最優秀賞

賞金 20 万円

多地点同時・多種気象データ観測器「P=System」

香川高等専門学校高松キャンパス

Entry No.
4

2018年10月26日

第7回 気象文化大賞
「高校・高専『気象観測機器コンテスト』」
コンテスト2次審査 実証実験報告書

香川高等専門学校

P-System

メンバー

電気情報工学科	5年	大西	哲
電気情報工学科	5年	笠松	雅史
電気情報工学科	5年	片山	大貴
電気情報工学科	5年	新谷	悠真
電気情報工学科	5年	吉川	悠斗

指導教員

電気情報工学科　村上　幸一

目次

1 製作背景と目的

2 ミッション

2.1 プロジェクトの概要 3
2.2 最終目標 3

3 使用機器

4 システムの概要

4.1 センサノードの概要 (屋外用センサノード) 15
4.1.1 機器の構成方法 15
4.1.2 気象状況の測定と計算方法 16
4.1.3 パケットの作成と送信 17
4.2 センサノードの概要 (屋内用センサノード) 19
4.2.1 空気清浄度の測定 19
4.2.2 パケットの送信 20
4.3 ゲートウェイの概要 21
4.3.1 センサノードとゲートウェイの接続 21
4.3.2 センサノードからのパケットの受信 22
4.3.3 Web サーバへの送信 23
4.3.4 AI サーバへの送信 23
4.4 サーバの概要 24
4.4.1 サーバの構成方法 24
4.4.2 アドバイス文の生成・天気の予想 25
4.4.3 ブログ記事へのアドバイス文の投稿 25
4.4.4 LINE アプリへのアドバイス文の投稿 26
4.4.5 ゲートウェイや AI サーバから Web サーバへのデータ送信 27

4.5 開発環境と動作要件 28
4.5.1 開発環境 28
4.5.2 動作要件 31

5 Neural Network の概要

5.1 AI の概要 33
5.1.1 Neural Network とは 33
5.1.2 深層学習の仕組み 36
5.1.3 Neural Network Console 39
5.2 室外アドバイス文章の生成 40
5.2.1 生成の方法 40
5.2.2 学習方法 42
5.2.3 学習結果 45
5.3 室内アドバイス文章の生成 46
5.3.1 生成の方法 46
5.3.2 学習方法 48
5.3.3 学習結果 51
5.4 気温の予測 53
5.4.1 予測の方法 53
5.4.2 LSTM について 53
5.4.3 学習方法 59
5.4.4 学習結果 62

6 フロントエンドの紹介

6.1 センサ情報 (屋内) 63
6.2 センサ情報 (屋外) 65
6.3 MAP 67
6.4 グラフ 68
6.5 気温予測 69
6.6 AR(拡張現実) 70

7 実験方法と結果

7.1 実験方法 72

7.2 実験結果 73

8 考察

8.1 気温の予測の精度 80
8.2 気温の予測の改善案 84
8.2.1 データセットの形式の見直し 84
8.2.2 ネットワークモデルの見直し 84

9 評価

9.1 ミッションの達成度と作品の有用性 86
9.2 今後の展望 87

10 感想

11 参考資料

要旨

近年の異常気象を鑑みると，正確な気象情報を迅速に収集・解析するシステムの構築は急務である．しかし，気象情報の収集・解析・提供を一元的・自動的に行なってくれるシステムは，一般には普及していない．P-System は，個人で所有することのできる，「気象情報の収集・解析・提供を一元的・自動的に行なってくれるシステム」である．

このシステムの実現のために，データの収集にワイヤレスセンサネットワーク，データの解析に人工知能 (AI)，データの提供のために Web アプリケーションを利用した．

そして，本システムを実際に運用・体験することで，システムの有用性・利便性を確認することができた．本レポートでは，システムの概要・運用結果・機器の有用性について述べる．

1 製作背景と目的

P-System は，天気に関する各種情報を収集し，人工知能による処理を経てから，ユーザに対して Web サイトや SNS を利用して情報を提供するシステムである．

このシステムの制作背景には，以下のようなものがある．

(1) 現状の天気予報サービスでは，広範囲 (県や市町村単位) の天気予報は提供されているが，自宅付近など，局所的な天気予報サービスは提供されていない．
(2) 天気情報は把握できても，その情報を元にどのような行動を取るべきなのかが分かり難い．
(3) 屋内の環境情報 (気温・湿度・空気室など) と天気情報を統合して情報提供やアドバイスを提供するサービスは少ない．

そこで私達は，「みんなのデータでつくる，自分のための天気サービス」をコンセプトに，P-System を制作した．

P-System の主な目的は，以下のものがある．

(1) 屋内外の気象データを測定する．単に気温や湿度だけでなく，空気室や PM2.5 のような，健康に直接影響を与えるものも測定する．
(2) 測定されたデータを元に，将来の気温予想を行ったり，その日の行動や生活面でのアドバイス文章も提供する．
(3) データの提供には Web サイトや SNS，その他のマルチメディア技術を利用して，ユーザにとって分かりやすく，楽しいものにする．

2 ミッション

2.1 プロジェクトの概要

1 章に示しているように，普段私たちが利用している天気予報は大多数の人々に向けたものであり，必ずしも自分にとって有用なものになるとは限らない．そこで，私たちは「超局所的 (自宅，自宅周辺) な天気予報」，「気候に合わせた行動アドバイス」，「SNS 等の身近なコンテンツでの予報の閲覧」の 3 つの要素を組み合わせた「**自分のための天気予報**」をコンセプトにシステムの開発を目標とした．

今回のシステムは，以下のような動作をする．

1. マイコン制御されたセンサを用いて，データを測定する．
2. 測定したデータをゲートウェイを介して，サーバに転送する．
3. 測定した気象データを用いて解析し，行動アドバイスを生成する．
4. 測定した気象データ・生成した行動アドバイスを Web ページ・SNS 上で閲覧できるように表示する．

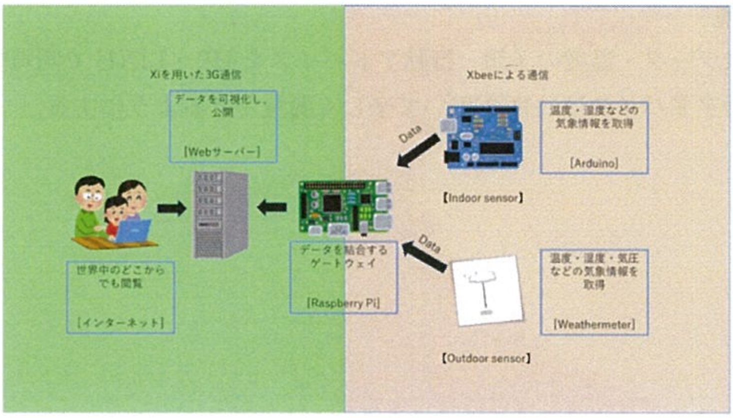


図 **2.1**　プロジェクトの概要

2.2 最終目標

私たちは **2.1** 節で示した「自分のための天気予報」を実現するために，「自宅内，自宅周辺の気象データを回収すること」，「回収した気象データを解析し，気象予測を行うこと」，「気象予測に基づくアドバイスを行うこと」を達成することを最終目標とした．

上記目標を達成するために，具体的な 5 つのシステム別ミッションを設定した．

■ 自宅内・自宅周辺の気象データの測定・取得

温湿度センサ，気圧センサなどのセンサを用いて基本的な気象データを測定する．また，今年度話題となった熱中症問題の対策のためにも，得られた気象データから体感温度，熱中症温度などの算出も行う．

また，気象データを取得する機器，気体の状態を取得する機器と，それらのデータを一元的に管理・転送する機器 (ゲートウェイ) による WSN(Wireless Sensor Network：ワイヤレスセンサネットワーク) を構築する．

ゲートウェイに送られた各種データは，CSV ファイルや JSON ファイルなどのデータ編集が容易なファイルに変換する．

■ 大気汚染物質・空気室の測定

ダストセンサを用いて自宅周辺の大気汚染物質の一種である微小粒子状物質 (PM2.5) の分布密度を測定する．

また，BME680 を用いて空気室 (IAQ) を測定できる．IAQ とは，一般に建物内等の空気中のガス成分量を指す．IAQ の算出には，ガス抵抗と温度の値を利用すること．

■ 深層学習を用いたデータ解析

ゲートウェイに送られてきた気象データを深層学習を用いて解析してからサーバーに送信し，行動アドバイスの生成・温度予測を行う．

■ データの閲覧

取得した気象データ・温度の予想・行動アドバイスを HP・LINE で閲覧できるようにする．また，行動アドバイスは AR を用いて楽しく閲覧できるようにする．

3 使用機器

使用する機器を以下に示す．

■ Arduino

Arduino は，イタリアで開発された教育用マイコン機器である．温湿度センサや Dust-sensor，Wethermeter 等を用いて気象を測定する屋外用 SensorNode の制御マイコンとして利用した．

Arduino を採用した理由として，サンプルコードが豊富なため，開発が容易であること，ダストセンサのような利用者の少ないセンサについても，リファレンスやライブラリが豊富に用意されていること，mbed などのマイコンに比べて，安価であることなどが挙げられる．

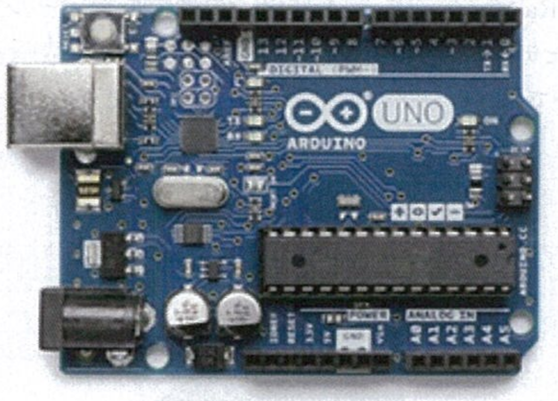


図 **3.1**　Arduino UNO Rev3

■ Raspberry Pi

Raspberry Pi はラズベリーパイ財団の開発したシングルボードコンピュータである．この機器を用いて，Arduino から送られたデータをインターネット回線を用いてサーバにアップロードするためのゲートウェイとして利用した．

その他，屋外用センサである BME680 を制御する用途にも利用する．

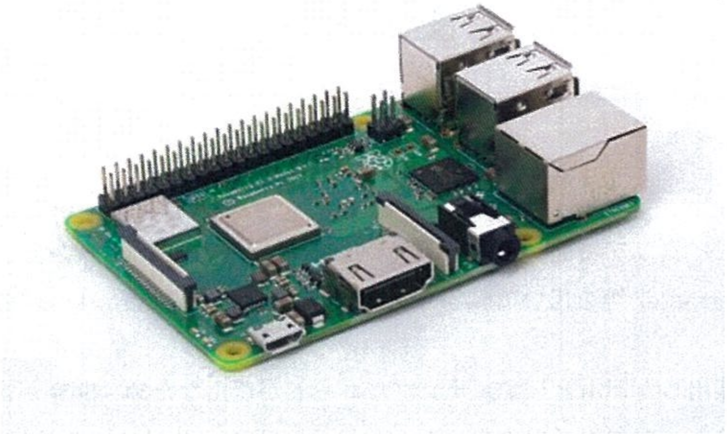


図 **3.2**　Raspberry Pi

■ XBee

図 **3.3** の機器は，Raspberry Pi と Arduino の通信モジュールである．これを用いて Arduino から取得したデータを Raspberry Pi に送信する．

また Raspberry Pi では，シリアル通信を用いて，XBee からデータを受信する．

図 **3.3**　XBee

■ XBee シールド

図 **3.4** の機器は Arduino に搭載する XBee 用のシールドである．この機器を用いて Arduino と XBee を接続する．

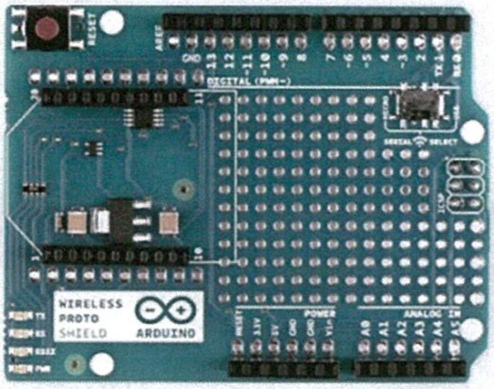


図 **3.4**　Arduino 用シールド

■ BME280

図 **3.5** のセンサは温度・湿度・気圧を測定することができる BME280 センサである．通信形式は I^2C であり，Arduino によって制御される slave デバイスである．また BME280 は，温度・湿度・気圧を同時に測定することができ，また精度が他のセンサに比べて良いので採用する．本システムでは屋外用に使用した．

表 **3.1**　BME280 の特性表

	レンジ	誤差
温度	−40 〜 85 °C	±1 °C
湿度	−	±3 %
気圧	300 〜 1100 hPa	±1 hPa

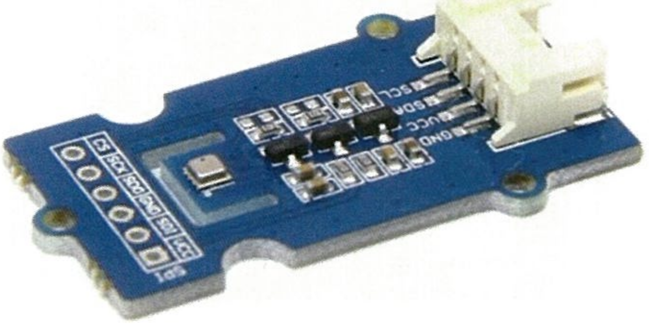


図 3.5　Grove BME280

■ **Grove Dust Sensor**

図 **3.6** のセンサは，ダストの濃度を測定することにより環境中の空気の質を表すセンサである．空気中の粒子状物質レベル (PM レベル) は，所与の時間単位での低パルス占有時間 (LPO 時間) をカウントすることによって 1 m^3 に何個の粒子が存在するかを測定することができる．この LPO 時間は PM 濃度に比例する．本システムでは屋外用で使用する．

また，DustSensor は Arduino によって制御される Slave デバイスである．

図 3.6　Grove DustSensor

■ **Grove Base Shield**

図 **3.7** の機器はアナログ，ディジタル，I^2C 通信に対応しており，数多くのセンサを接続できる．今回は屋外用の Arduino に対して 1 台取り付けて，BME280 と Weathermeter を接続するために使用する．

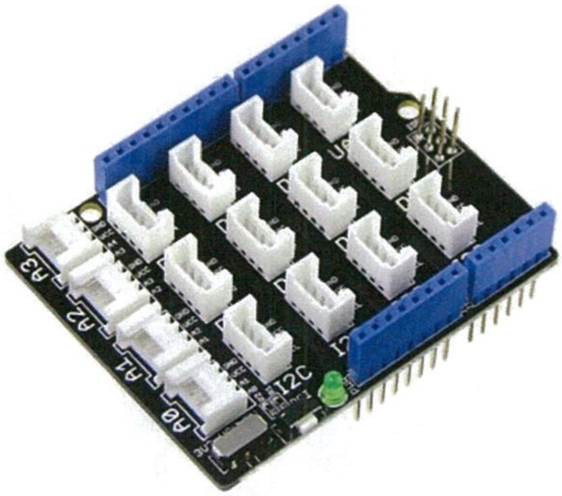


図 **3.7**　Grove Base shield

■ **Weathermeter**

図 **3.8** の機器及び Weatherduino で風向，風速，雨量，照度の 4 つを測定することができる．本システムでは屋外用で使用する．

また，Weathermeter は Weatherduino によって制御される Slave デバイスである．

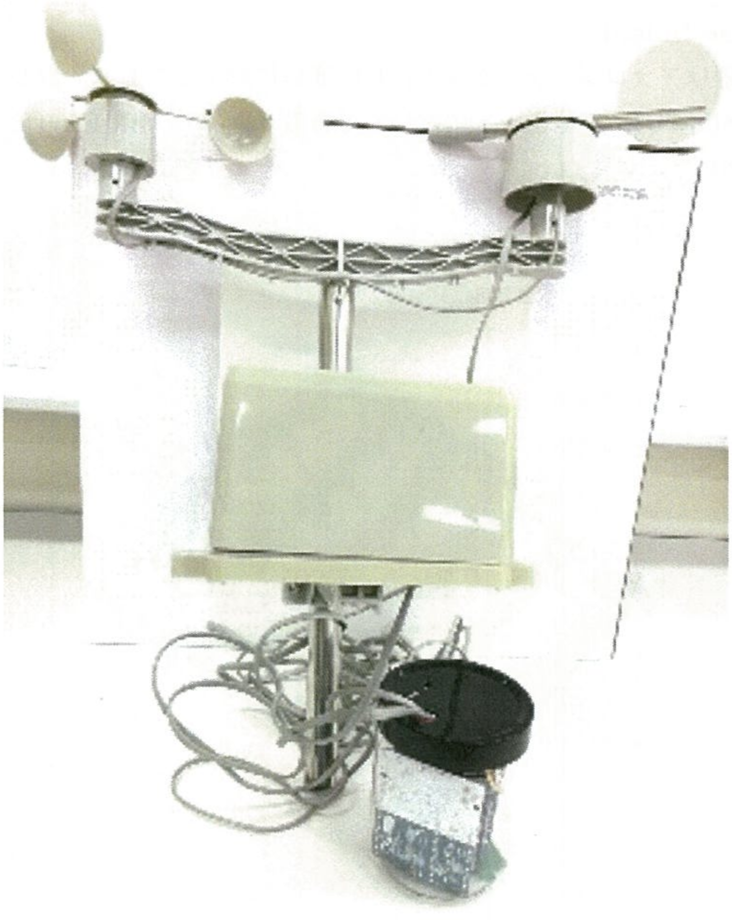


図 **3.8**　Weathermeter

■ BME680

図 **3.9** のセンサは温度・湿度・気圧・IAQ を測定することができるセンサである．通信形式は I^2C である．温度・湿度・気圧を同時に測定することができ，精度が他のセンサに比べて良いので採用する．本システムでは屋内用に使用する予定だ．

表 **3.2**　BME680 の特性表

	レンジ	誤差
温度	−40 ～ 85 °C	±1 °C
湿度	−	±3 %
気圧	300 ～ 1100 hPa	±0.6 hPa

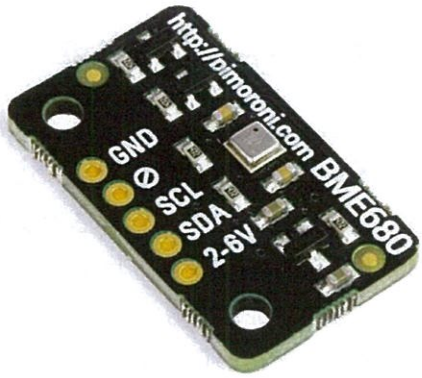


図 **3.9**　BME680

4 システムの概要

本章では，システムの構成方法やシステムの動作原理について説明する．

本システムを機能で大別すると，気象状況を測定するセンサノード (屋内用・屋外用)，測定したデータを整形し，サーバに送信するゲートウェイ，データを保存し，システムの利用者にデータを提供するサーバから構成される．

各機能と部品の対応と，データの流れについて示した図を，図 **4.1** に示す．また，サブシステム・コンポーネント間の関係を示した図 (Function and Physical Architecture) を，図 **4.2** に示す．

まず，屋内の気象状況を屋内用センサノード，屋外の気象状況を屋外用センサノードを用いて測定する．

屋内用センサノードの測定データについては，ゲートウェイである Raspberry Pi に有線で送信される．屋外用センサノードの測定データについては，一度 Arduino で処理され，Arduino から Raspberry Pi に無線送信される．無線送信には，XBee を用いている．

Raspberry Pi に送信されたデータは，サーバに送信される．ゲートウェイは，サーバとセンサノードの中継役を担っている．このとき，サーバにとって処理しやすいデータのフォーマットに変換されてから送信される．

サーバは，データを保存したり閲覧するための Web サーバと，気温の予想や天気アドバイス文章生成のための AI サーバの 2 つがある．

AI サーバでは，データを元に，将来の気温予想を行ったり，その日の天気の傾向や服装のポイント・傘の要・不要についてのアドバイス文を生成し，Web サーバにアップロードする．

Web サーバでは，ゲートウェイから送信される気象データや，AI サーバから送信される予想・アドバイス文章をデータベースに格納したり，ブログ記事に投稿する．その後，Web サイト上で，これらのデータを表・グラフ等，ユーザにわかりやすい形で提供する．

なお．センサノードやゲートウェイという単語は，「機能」によってシステムを分類した呼称であり，「部品」として分類したものではない．つまり，「Raspberry Pi = ゲートウェイ」，「Arduino = センサノード」などではなく，「データを中継するもの = ゲートウェイ」，「データを測定するもの = センサノード」である．

なので，1 つの部品が 2 つの機能を保つ場合は，2 つの呼称を持っている．例えば，ラズベリーパイはデータの中継と屋内の気象観測を行うため，ゲートウェイとセンサノードの両方の役割を担っている．

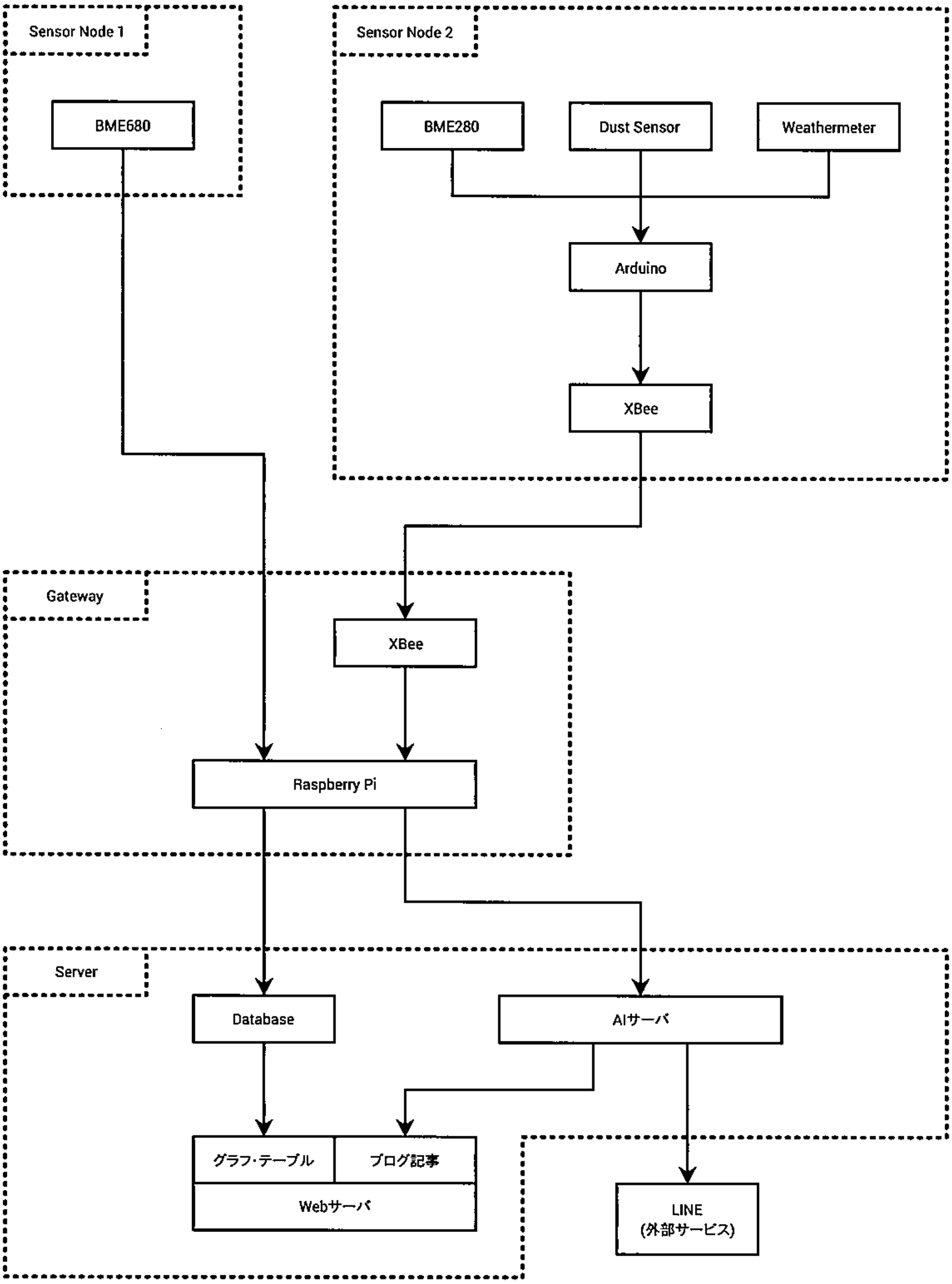

Sensor Node 1
BME680
Sensor Node 2
BME280
Dust Sensor
Weathermeter
Arduino
XBee
Gateway
XBee
Raspberry Pi
Server
Database
AIサーバ
グラフ·テーブル
ブログ記事
Webサーバ
LINE
(外部サービス)

図 4.1　データの流れ

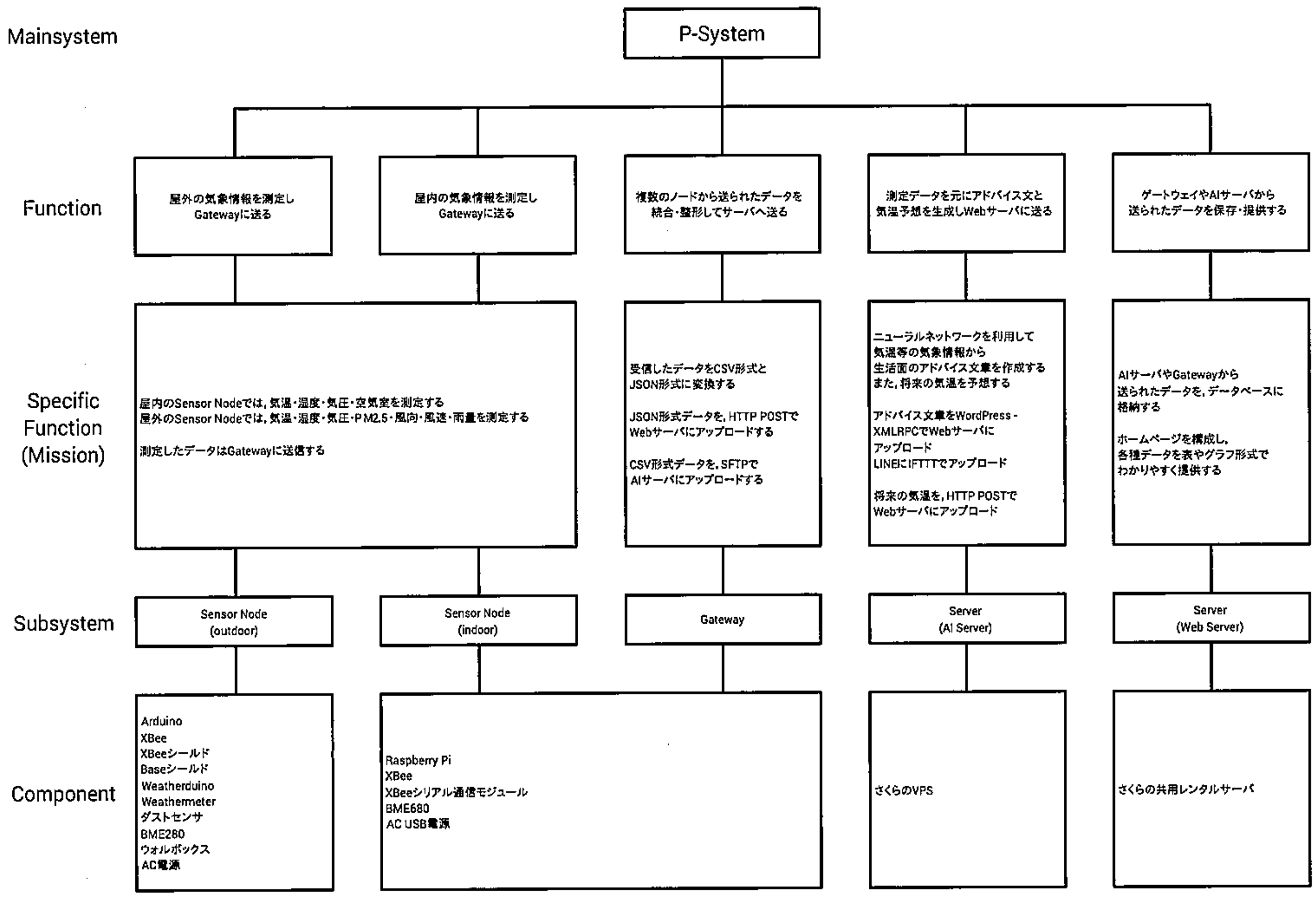


図 4.2　サブシステムと構成機器の対応図

4.1 センサノードの概要 (屋外用センサノード)

屋外用のセンサノードは，BME280 センサ・ダストセンサ・Weathermeter センサを搭載している．各センサは Arduino マイコンによって制御され，5 分に 1 回測定を行う．

また，測定した値から熱中症温度と体感温度を計算し，ゲートウェイへの送信パケットを作成する．

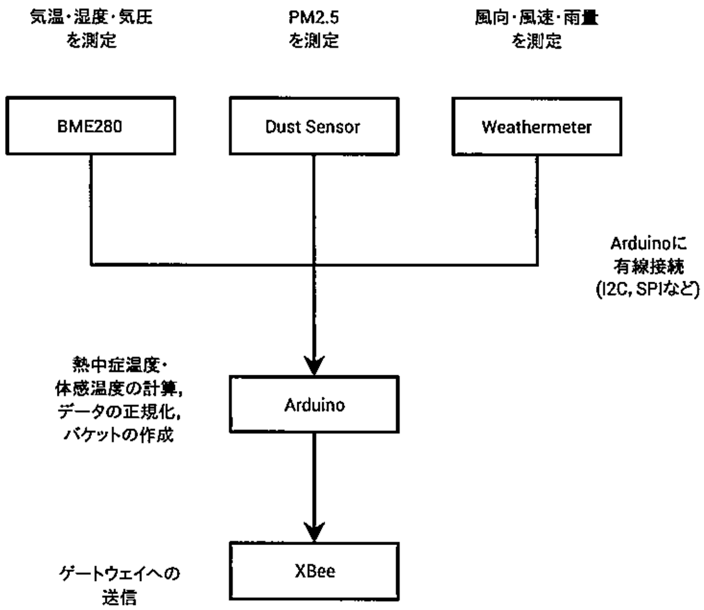


図 **4.3**　屋外用センサノードの構成と役割

4.1.1 機器の構成方法

屋外用センサノードは，Arduino に Weatherduino シールド，XBee シールド，Sensor シールドを接続して構成している．XBee シールドには ZigBee XBee S2C，Weathermeter シールドには Weathermeter，Sensor シールドには BME280 センサとダストセンサを取り付けている．

構成されたセンサノードは屋外に設置するため，風雨によってマイコン部分 (Arduino) が破損しないようにウォルボックスに収納し，センサ部分を外部に露出させている．

また，機器の電源には USB モバイルバッテリー電源 (5V 2A) を利用している．

(a) ウォルボックス内部　　　(b) ウォルボックス外部

図 4.4　機器の構成写真

図 **4.4(a)** において，ウォルボックス左側面に取り付けられているのがダストセンサである．また，図 **4.4(b)** において，ウォルボックス右側面に取り付けられているのが BME280 である．

4.1.2　気象状況の測定と計算方法

屋外用センサノードに搭載された BME280 センサは気温・気圧・湿度を，ダストセンサは空気中の PM2.5 の濃度を，Weathermeter センサは風向・風速・雨量を測定する．

風向のデータについては，風向計の向いている角度 (0°〜360°) が取得できるため，この値を 8 方位の風向情報 (N : 北，NE : 北東，E : 東，SE : 南東，S : 南，SW : 南西，W : 西，NW : 北西) に Arduino で変換してからパケットを生成している．

また，これらの気象データから，熱中症温度・体感温度の算出を Arduino で行う．測定された気温を T [°C]，湿度 (相対湿度) を H [%]，風速を v_w [m/s] とすると，熱中症温度 T_{wbgt} 及び体感温度 T_m は，以下の式で算出できる．

$$T_{wbgt} = 0.567T + 0.393e + 3.94 \tag{4.1}$$

$$T_m = 37 - \frac{37 - T}{0.68 - 0.0014H + \frac{1}{1.76 + 1.4v_w^{0.75}}} - 0.29T\left(1 - \frac{H}{100}\right) \tag{4.2}$$

ただし，式 (4.1) で用いられる e [hPa] は水蒸気圧を表し，式 (4.3) で求められる．

$$e = \frac{H}{100} \times 6.105 \times \exp\left\{17.27 \times \frac{T}{T + 237.7}\right\} \tag{4.3}$$

4.1.3 パケットの作成と送信

測定したデータを元に，熱中症温度・体感温度の算出を行う．また，ゲートウェイにデータを送信するため，　パケットを作成する．

熱中症温度・体感温度の算出は，**4.1.2** 項に示した方法で算出する．測定したデータと算出したデータを元に，パケットを作成し，XBee で Raspberry Pi に送信する．XBee は，ZigBee 規格に対応した小型の無線機であり，1 対 1 の通信においては，送信側が Router デバイス，受信側が Coordinator デバイスとなることが多い．

パケットのフォーマットは，API フレームパケットと呼ばれる方式である．XBee には，通信方式として API モードと AT モードが存在し，今回は API モードを採用している．API モードは設定が複雑であるが，データ損失・破損に関する信頼性や，データの衝突には強い通信形式である．

また，複数の XBee 同士でのメッシュ型通信にも対応できるため，センサノードの数を増やすことも可能である．

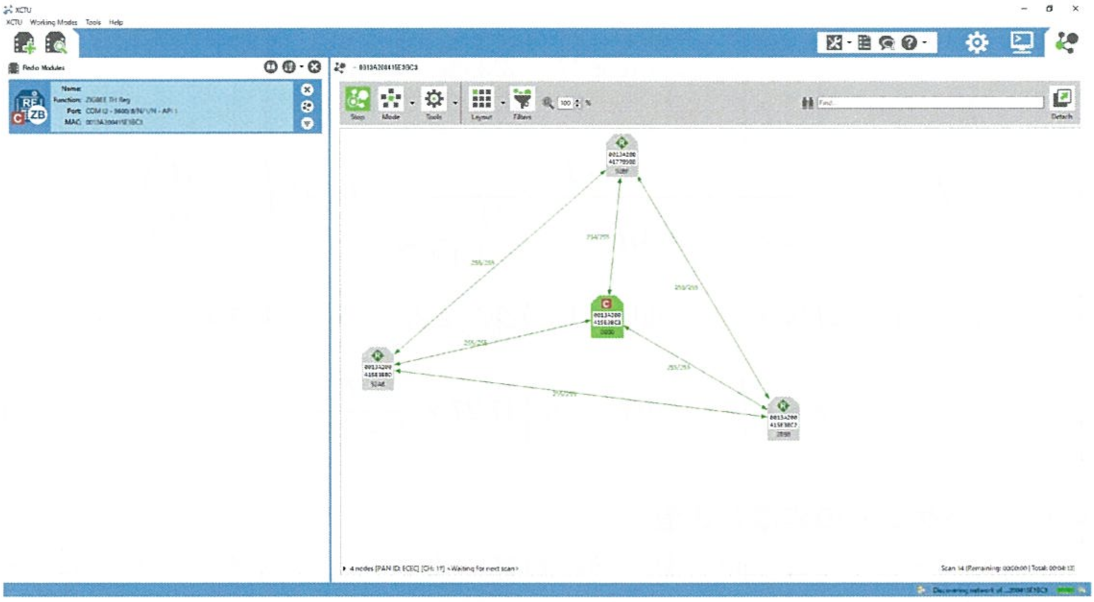


図 **4.5**　XBee の接続

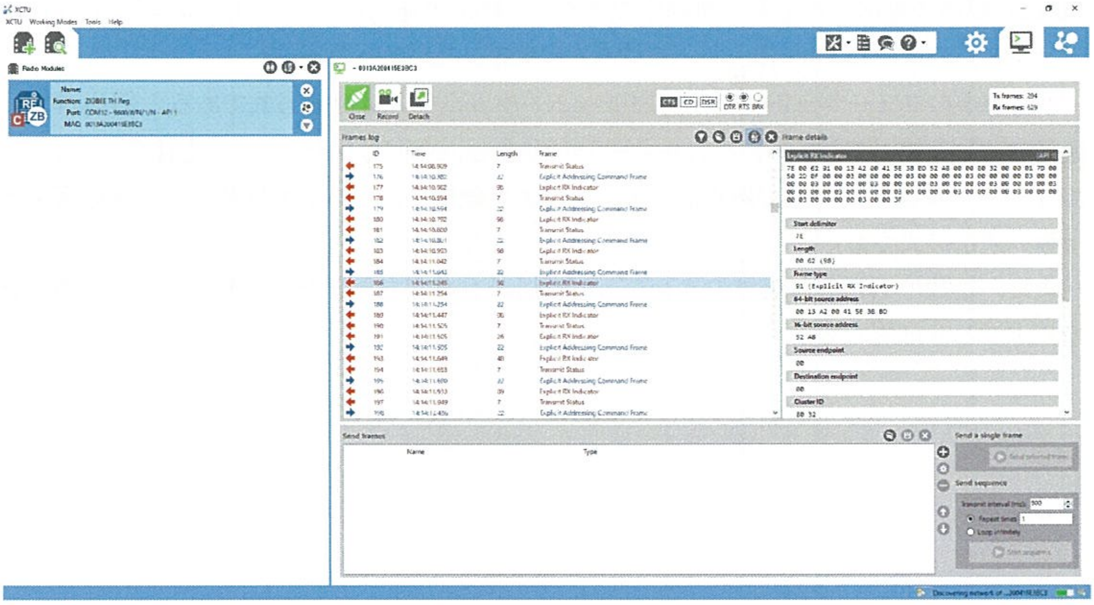


図 **4.6**　XBee でやり取りされるパケット

図 **4.6** で示したのが，XBee で送受信されるパケットの例である．API モードでは，パケットは 16 進数のフォーマットでやり取りされる．パケットの作成は，Arduino で行われる．

測定されるデータは整数値や実数値であるが，コンピュータ内部ではビット列として保存されているので，このビット列を 16 進数の文字列 (バイト列) に変換 (エンコード) して

送信している．

受信側となる Raspberry Pi(ゲートウェイ) では，このバイト列を数値情報に復元 (デコード) してから処理を行う．

4.2 センサノードの概要 (屋内用センサノード)

屋内用センサノードには，BME680 センサを利用している．BME680 センサは，Raspberry Pi を用いて制御する．BME680 センサは，気温・気圧・湿度・ガス抵抗値を測定することができ，湿度とガス抵抗の値から空気清浄度を計算する．

なお，屋内用センサノードとゲートウェイを構成する Raspberry Pi は同一であるため，Raspberry Pi は 1 台で屋内用センサノードとゲートウェイの両方の役割を果たしている点に注意が必要である．

4.2.1 空気清浄度の測定

空気清浄度を I_{aq}，湿度を H，ガス抵抗値を R_{gas}，湿度の基準値を H_b，ガス抵抗の基準値を R_{gb}，空気清浄度におけるガス抵抗と湿度の重みの比を $R_w : H_w$ (ただし $R_w + H_w = 1, R_w > 0 \cap H_w > 0$) とする．

湿度の基準値 H_b 及びガス抵抗の基準値 R_{gb} は，センサを起動してから暫くの間の湿度やガス抵抗の値の平均値とした．また，$R_w = 0.75, H_w = 0.25$ として計算した．

まずは，ガス抵抗と湿度の値から，空気清浄度の値の湿度分 S_H とガス抵抗分 S_R を求める．

$$S_H = \begin{cases} \dfrac{100 - H}{100 - H_b} \cdot (100H_w) & (H - H_b > 0) \\ \dfrac{H}{H_b} \cdot (100H_w) & (other) \end{cases} \tag{4.4}$$

$$S_R = \begin{cases} \dfrac{R_{gas}}{R_{gb}} \cdot (100R_w) & (R_{gb} - R_{gas} > 0) \\ 100 - \{(1 - R_w) \cdot 100\} & (other) \end{cases} \tag{4.5}$$

S_H と S_R より，空気清浄度が求まる．

$$I_{aq} = \{100 - (S_R + S_H)\} \times 5 \tag{4.6}$$

式 (4.4)，式 (4.5)，式 (4.6)，より，I_{aq} は 0 から 500 までの数値である．一般に，ガス抵抗の値が高いほど空気の質が良いため，I_{aq} に換算して考えると，I_{aq} の値が低いほうが空気の質は優れている．(空気清浄度が高い，表 **4.1** を参照)

表 4.1　I_{aq} の値の参考値

I_{aq} の値	空気の質
0～50	良好
51～100	普通
101～150	やや危険
151～200	危険
201～300	非常に危険
301～500	生活できない

4.2.2　パケットの送信

測定されたデータを元にして，パケットを作成する．本センサノードは，ゲートウェイと一体であるため，パケットの作成方法は **4.1.3** 項で示した方法とは異なる．

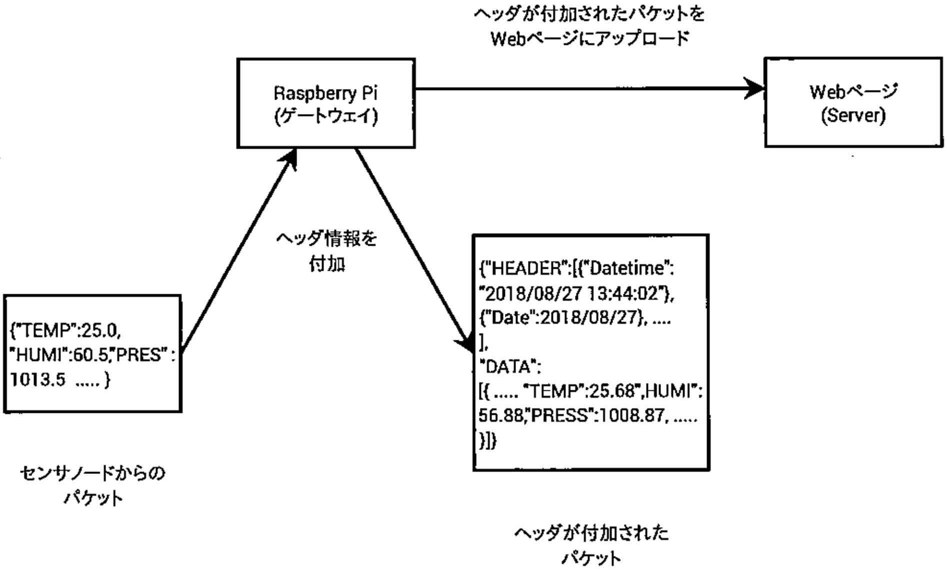


図 **4.7**　屋内用センサノードのパケット作成と送信

図 **4.7** に示したのが，屋内用センサノードで測定された値をサーバへの送信用フォーマットに変換する様子である．

屋内用センサノードで測定・算出された値は，JSON と呼ばれるフォーマットに変換された後，AI サーバや Web サーバに送信される．これらの機能についてはゲートウェイ側の機能に該当するため，**4.3** 節で詳しく説明する．

4.3 ゲートウェイの概要

ゲートウェイは，データの中継地点である．ゲートウェイの役割は，主に 3 つある．1 つ目の仕事は，センサノードからのデータを受信し，データを受信することである．2 つ目の仕事は，受信したデータを，サーバが処理しやすいフォーマットに整形することである．3 つ目の仕事は，整形したデータをサーバに送信することである．

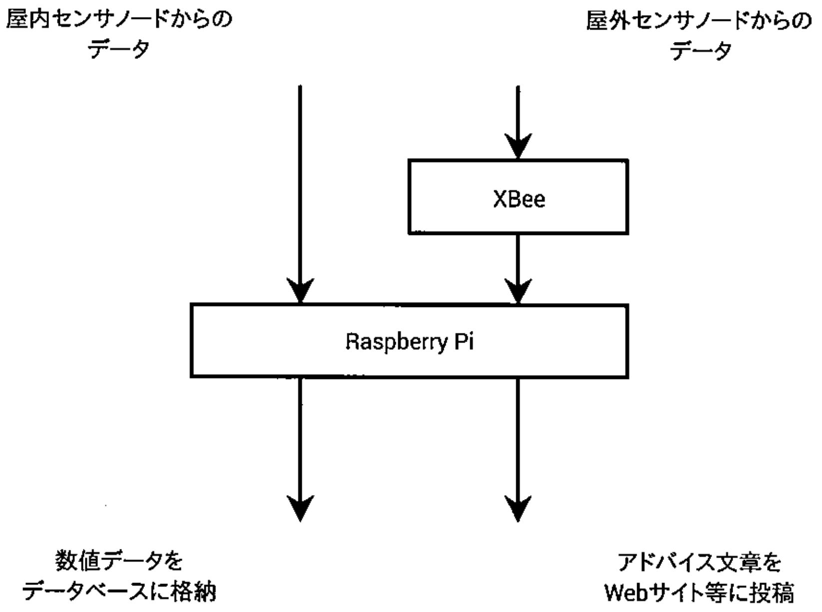


図 4.8　ゲートウェイにおけるデータの流れ

なお，サーバについては Web サーバと AI サーバに分かれているため，整形データの送信方法についてもそれぞれ説明する．

4.3.1　センサノードとゲートウェイの接続

屋外用センサノードとゲートウェイの接続は，**4.1.3** 項で述べたように，XBee を利用している．Raspberry Pi で XBee を利用する場合，Raspberry Pi の GPIO ピンに接続する方法と USB ポートを利用する方法があるが，このシステムでは後者を採用した．USB で接続すると，Raspberry Pi でシリアル通信を行うことで，センサノードからデータを受信できる．この場合，XBee のデバイス名を固定する必要があるため，Raspberry Pi の`/etc/udev/rules.d/99-local.rules` ファイルを編集してデバイス名を固定した．

屋内用センサノードとゲートウェイの接続は，Raspberry Pi の GPIO ピンを利用して直接接続されているため，パケットをそのまま受信できる．

4.3.2 センサノードからのパケットの受信

センサノードから送信されるデータは，直接 Web サーバや AI サーバに送信するにはフォーマットが不適切であるため，データを整形する必要がある．

屋内用センサノードから送信されたデータ (パケット) は，測定された数値情報が直接示されているため，測定時刻やセンサノードの名称などの情報を付加するだけでよい．

屋外用センサノードから送信されたデータは，バイト列であるため，数値情報に復元してから，屋内用センサノードと同様，測定時刻やセンサノードの名称などの情報を付加する．

```
indoor
{
    "TEMP":25.68,
    "HUMI":56.88,
    "PRESS":1008.87,
    "CO2":106611.15,
    "Additional_Sensor_1":96.00,
    "Additional_Sensor_2":"null",
    "Additional_Sensor_3":"null"
}
```

List 4.1　センサノードから送信されるパケット例

```
{
    "HEADER":
    [
        {"Datetime" : "2018/08/27 13:44:02"},
        {"Date" : "2018/08/27"},
        {"Time" : "13:44:02"},
        {"SN_Name":"indoor"},
        {"SN_id" : 2}
    ],

    "DATA":
    [
        {
            "Datetime" : "2018/08/27 13:44:02",
            "Date" : "2018/08/27",
            "Time" : "13:44:02",
            "TEMP":25.68,
            "HUMI":56.88,
            "PRESS":1008.87,
            "CO2":106611.15,
            "Additional_Sensor_1":96.00,
            "Additional_Sensor_2":"null",
            "Additional_Sensor_3":"null"
```

```
        }
    ]
}
```

List 4.2　ゲートウェイで情報が付加されたパケット例

List 4.1 に示したのが，屋内用センサノードから送信されているパケット例である．**List 4.2** に示したのが，測定時刻やセンサノードの名称などの情報が付加されたパケット例である．

4.3.3　Web サーバへの送信

気象データを長期的に保存するデータベースは，Web サーバ上に存在し，ゲートウェイ (Raspberry Pi) から直接データベースにアクセスすることはできないため (セキュリティ上の問題)，Web サーバ上のパケット受信ページに送信し，パケット受信ページからデータベースにアクセスすることで，データベースに保存される．パケット受信ページへのアクセスは，HTTP POST を利用している．

HTTP POST とは，HTTP を利用してクライアントからサーバにリクエストを送る手法の 1 つである．Web サイトのログイン画面で，ユーザ名やパスワードをシステム側に送信するときなどに利用されている．

HTTP を利用してクライアントからサーバにリクエストを送るその他の手法に HTTP GET がある．Google などの検索サイトにおいて，検索したい単語をシステム側に送信するときなどに利用されている．

HTTP GET は，データをシステム側に送信するとき，URL に情報を付記することによって送信している．そのため，送信内容は履歴として保存され，また送信内容が外部に公開される可能性があるためセキュリティが低い．また，HTTP GET は一度のリクエストで送信できる情報の量が少ないため，HTTP POST を採用した．

4.3.4　AI サーバへの送信

4.3.3 項の方法で，センサノードから受信したデータは，天気の予想やアドバイス文生成を行う AI サーバで利用するため，AI サーバにも送信される．

Web サーバへの送信には HTTP POST を利用したが，AI サーバへの送信には，SFTP を利用している．SFTP とは，SSH(Secure Shell) の技術を利用して，ファイルを転送するプロトコルである．

気温予想やアドバイス生成に関するプログラムは 1 時間に 1 回実行されるため，AI サーバへのデータ送信も 1 時間に 1 回行う．

AI サーバへのデータ送信フォーマットは，CSV 形式である．まず，センサノードから受信したデータを元にゲートウェイで CSV ファイルを生成し，1 時間に 1 回，AI サーバに

ファイルを転送する.

AI サーバへの送信と Web サーバへの送信で送信方法が異なる理由について述べる. Web サーバへ送信されたデータは, データベースに格納される. データベースへのアクセスに用いられるプログラムには PHP と呼ばれる言語が主に利用されており, この言語は Web アプリケーションを構成することに向いている. そこで, Web アプリケーションで広く利用されている JSON フォーマットや HTTP POST を利用している.

一方 AI サーバに送信されたデータは, 人工知能 (ニューラルネットワーク) によって利用される. 人工知能では Python と呼ばれる言語が利用されており, 数値計算などに利用されることが多い. そこで, データの読み書きを容易に行える CSV 形式を利用した. また, AI サーバには, ある程度のデータを纏めて送信するため, ファイル単位で送信するほうが都合が良い. そこで, ファイル転送プロコトルである SFTP を利用した.

4.4 サーバの概要

サーバは, Web サーバと AI サーバに別れている.

AI サーバは, ゲートウェイから送信されるデータを元に, アドバイス文章生成や気温の予測を行い, その結果を Web サーバに送る.

Web サーバは, ゲートウェイや AI サーバから送信されたデータをデータベースに格納したり, ブログ記事として投稿する役割を果たしている. また, データベースに格納されたデータを, HP 上でシステムの利用者にわかりやすく表示する機能を果たしている.

ゲートウェイや AI サーバから Web サーバに送信されるデータは,「その日の天気の傾向や服装のポイント, 傘の要・不要についてのアドバイス文章」,「将来の天気 (気温等) の予測値」,「現在の気温や湿度, 気圧等の気象状況」の 3 つである.

「その日の天気の傾向や服装のポイント, 傘の要・不要についてのアドバイス文章」については, サーバ上のブログに記事としてアップロードされる.

「将来の天気 (気温等) の予測値」,「現在の気温や湿度, 気圧等の気象状況」については, 表にまとめ, ユーザにわかりやすいように表示している. さらに「現在の気温や湿度, 気圧等の気象状況」については, グラフ機能を用いてグラフを作成している.

その他, 表の内容を CSV ファイルでダウンロードしたり, マップ上に観測機器の設置箇所を表示させる等, 高度利用者向けのサービスも提供している.

4.4.1 サーバの構成方法

AI サーバの構成には, さくらインターネット株式会社の VPS(さくらの VPS) を利用している. サーバ上の OS には, Linux ディストリビューションの 1 つである Ubuntu OS を利用している.

Webサーバの構成には，さくらインターネット株式会社の共用レンタルサーバ (さくらのレンタルサーバ) を利用している．

Webサーバ上には，データを保存するデータベースと，利用者にデータを提供するWebページがある．データベースの管理システムにはMySQLを利用し，Webページの構成にはWordPressを使用した．

Wordpressとは2003年にオープンソースのブログ作成システムとして公開されたもので，現在はブログの他にも企業サイト，公共機関サイトなど多方面にわたるウェブサイトで利用されている．

本システムのWebページは屋内の気象データを表示する「センサ情報 (屋内)」，屋外の気象データを表示する「センサ情報 (屋外)」，本システムを導入しているほかの地域の気象データを一度に閲覧できる「Map」，温湿度の変化をリアルタイムでグラフ化してくれる「グラフ」，1時間周期で24時間後までの温度変化の予測結果を表示する「天気予報」で構成されている．

4.4.2 アドバイス文の生成・天気の予想

4.3.4 項で示した，ゲートウェイからAIサーバにSFTPで送信されたデータを元に，アドバイス文章を生成したり，将来の天気を予想したりする．

SFTPはHTTP POSTと異なりファイル転送プロコトルなので，ゲートウェイから送信される時刻に合わせて，文章生成や天気予想の実行を行う必要はない．AIサーバでは，CRONを利用して，文章生成や天気予想のプログラムを1時間に1回実行し，その結果をWebサーバに送信する．

ここで，アドバイス文の投稿にはpython wordpress xmlrpcを利用している．(**4.4.3** 項参照)

気温の予想値の送信には，ゲートウェイからWebサーバに気象データを送信したとき同様，HTTP POSTを利用している．(**4.3.3** 項及び **4.4.5** 項)

具体的な文章生成や天気予想の手法については，**5** 章に示してある．

4.4.3 ブログ記事へのアドバイス文の投稿

4.4.2 項で生成されたアドバイス文章は，サーバ上のブログ記事として，アップロードされる．

記事の投稿には，python wordpress xmlrpcを利用した．python wordpress xmlrpcは，WordPressにおいて，外部システムからリモートで記事投稿や画像のアップロードを行う際に利用されるプロトコルであるxmlrpcをPythonで利用できるようにしたモジュールである．

テキストファイルとして保存されたアドバイス文章を，python wordpress xmlrpcをインポートしたPythonプログラム上でリードし，その内容をブログ記事としてアップロー

ドしている．

4.4.4 LINE アプリへのアドバイス文の投稿

LINE へのアドバイス文の投稿は IFTTT(イフト：IF This Then That) というサービスを利用することで実現している．IFTTT は，ある Web サービスと別の Web サービスを連携させて，新しいサービスにすることができるシステムである．例えば，Gmail のメールアドレスに特定のアドレスからのメールが届いたならば，SMS に通知するといったことが簡単に実現できる．

処理の流れとしては IFTTT の設定で，「トリガー」という発動条件を設定し，その条件を満たすことで「アクション (実行)」に移行させるというものである．システムの構成においてプログラミングは不要であり，すでに IFTTT で提供されている既存の連携サービスとユーザ・コミュニティが構成したサービス (IFTTT では「レシピ」と呼ぶ) を利用することにより，アクセス許可やフィルターの条件などを記述するだけで済む．つまり，IFTTT の名前の由来である「もし (IF)『トリガー (This)』ならば (Then)『アクション (That)』する」の「This」と「That」を対応サービスから選択すれば，新しいサービスの作成が可能となる．

アドバイス文を投稿するという機能を IFTTT で実現するためには，「AI サーバから Web リクエストが届いたならば，LINE へメッセージを送信する」という風にしてやればよい．今回は以下の手順で準備を行った．

(1) IFTTT の LINE 連携ページで通知先の LINE アカウントを登録
(2) Web リクエストの送り先である Webhooks の URL の取得
(3) トリガー (This) とアクション (That) を設定

まず，(1) の「IFTTT の LINE 連携ページで通知先の LINE アカウントを登録」である．これを行うことで通知先の LINE アカウントに，LINE の公式アカウント"LINE Notify"が友達追加される．LINE Notify とは，LINE が提供する外部サービスとの通知連携サービスであり，LINE 以外のアプリや Web サービスと API で連携し，各サービスにきた通知を LINE のメッセージで送信することができる．IFTTT 以外にも，GitHub や Mackerel というサービスと連携が可能である．アドバイス文は，LINE Notify から通知され，個人だけでなく LINE でグループ作成することによって複数人への通知も可能となる．

次に，(2) の「Web リクエストの送り先である Webhooks の URL の取得」である．Webhooks は指定の URL にリクエスト送信する事をトリガーに，また指定の URL に POST 送信などができる IFTTT のツールである．Webhooks により，Raspberry Pi, IFTTT，LINE の連携が可能となった．

最後に，(3) の「トリガー (This) とアクション (That) を設定」である．トリガーとなる部分には Webhooks の「Receive a webrequest」を選択，アクションとなる部分には LINE

の「Send message」を選択することによって，指定の URL に Web リクエストが届いたならば，LINE へメッセージを送信するという設定となった．

運用の際は Webhooks の URL に Web リクエストに送信するシェルスクリプトを実行コマンドとともに引数を入力することによって，Web リクエストが Webhooks の指定 URL に送信され，LINE へ引数が送信される．また，今回は URL で示される場所からデータを様々なプロトコルを用いて送受信することができるコマンドである curl を用いて，Webhooks の指定 URL に Web リクエストを送信している．

4.4.5 ゲートウェイや AI サーバから Web サーバへのデータ送信

Web サーバ上のデータベースには，屋内用センサノードの値を格納するテーブル，屋外用センサノードの値を格納するテーブル，天気の予測値を格納するテーブル，データの受信記録を格納するテーブルを用意しておく．

ゲートウェイや AI サーバは，これらのテーブルにアクセスして気象データを格納するのが望ましいが，レンタルサーバの仕様上，直接データベースにアクセスすることができない．

そこで，WordPress の Web ページ内に，データベースにアクセスするための PHP コードが動作する固定 PHP ページを作成しておく．WordPress は，ブログ記事のページとは別に，固定ページを作成することが可能で，この固定ページ内では，任意の PHP コードが実行可能である．

ゲートウェイや AI サーバはこの PHP ページに対して HTTP POST メソッドを利用して JSON 形式のパケット (**List 4.2** 参照) を送信する (図 **4.9** の緑色矢印)．パケットを受信した PHP ページは，受け取ったパケットを連想配列に格納し，連想配列の要素をデータベースに格納する (青色矢印)．

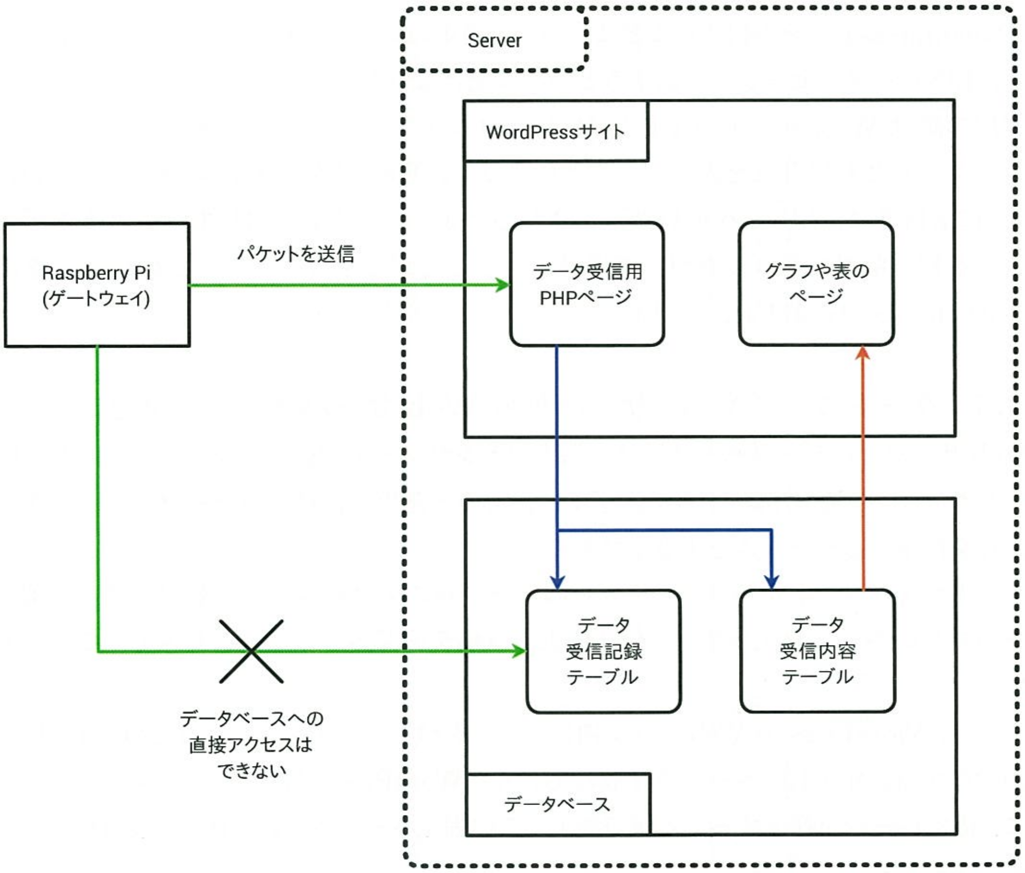


図 **4.9**　ゲートウェイからサーバへのデータ送信

図 **4.9** にはテーブルが 2 つしかないが，実際には本項の冒頭で述べた 4 つのテーブルが存在する．また，PHP の受信ページも複数存在し，ゲートウェイはパケットの種類 (屋内データのパケット，屋外データのパケット，予測値のパケット) に応じて送信先のページを選択する．

4.5　開発環境と動作要件

4.5.1　開発環境

■ 屋外用センサノードの開発環境

屋外用センサノードの開発環境を，表 **4.2** に示す．

表 4.2　屋外用センサノードの開発環境

使用言語	Arduino 言語
開発環境	Arduino IDE 1.8.5
使用ライブラリ	Grove-BME280-master PString3 Streaming G7Weather3

■ 屋外用センサノード・ゲートウェイの開発環境

屋内用センサノード及びゲートウェイの開発環境を，表 **4.3** に示す.

表 4.3　屋内用センサノードとゲートウェイの開発環境

使用言語	Python 3.5.2
開発環境	Visual Studio Code Notepad++
使用 OS	Raspbian Stretch with desktop, June 2018
使用ライブラリ	bme680 urllib paramiko

■ **AI** サーバの開発環境

AI サーバの開発環境を，表 **4.4** に示す.

表 4.4　AI サーバの開発環境

使用言語	Python 3.5.2
開発環境	Visual Studio Code Notepad++ Sony Neural Network Console
使用サービス	さくらの VPS 2G (さくらインターネット株式会社) IFTTT
使用 OS	Ubuntu 16.04.5 amd64
使用ライブラリ	Wordpress xmlrpc paramiko urllib curl nnabla

■ **Web サーバの開発環境**

Web サーバの開発環境を，表 **4.5** に示す．

表 4.5　Web サーバの開発環境

使用言語	PHP 7.2 HTML JavaScript
使用サービス	さくらのレンタルサーバ スタンダード (さくらインターネット株式会社)
データベース	MySQL 5.7
ライブラリ・API	jQuery audio.js A-Frame AR.js Google Chart API Google Map API
使用ソフトウェア	WordPress Apache 2.4.33
使用 OS	FreeBSD 9.1-RELEASE-p24 amd64
SSL サーバ構成	nginx + OpenSSL
SSL サーバ証明書認証局	Comodo

4.5.2　動作要件

■ 屋外用センサノードの動作要件

屋外用センサノードが動作するためには，以下の要件を満たす必要がある．

(1) 常に 5 V から 9 V，1.5 A 以上の電源に接続できること (AC アダプタが接続できること)
(2) ゲートウェイとの距離が 20 m 以下であること
(3) 直射日光が当たらないこと
(4) 風通しの良い場所であること．屋根がないこと
(5) 台風等の強風・豪雨では使用しないこと

■ 屋内用センサノード及びゲートウェイの動作要件

屋内用センサノード及びゲートウェイが動作するためには，以下の要件を満たす必要がある．

(1) 常に 5 V，2 A 以上の電源に接続できること (AC アダプタが接続できること)

(2) センサノードとの距離が 20 m 以下であること
(3) 窓や空調機の近く等，気体の環境の変化が大きい場所の近くには置かないこと
(4) 日光が当たらないこと
(5) Wi-Fi の電波がよく届く場所であること

■ **AI サーバの動作要件**

AI サーバが動作するためには，以下の要件を満たす必要がある．

(1) Python 3.4 以上の実行が可能であること
(2) 使用したライブラリがインストール可能であること
(3) CRON を利用してプログラムの提示実行が可能であること
(4) インターネットとの接続状況が安定していること

■ **Web サーバの動作要件**

Web サーバが動作するためには，以下の要件を満たす必要がある．

(1) PHP，JavaScript を利用したホームページが構成できること
(2) WordPress がインストール可能であること
(3) データベースが利用可能であること
(4) 使用したライブラリ・API が全て動作可能であること
(5) インターネットとの接続状況が安定していること

5 Neural Network の概要

5.1 AI の概要

ここでは，気象データの予測やアドバイス文章の生成に利用した AI(Artificial Inteligence) について説明する．

AI で課題を解決するにはさまざまな手法があるが，本システムでは Deep Learning を利用している．Deep Learning とは，多層の Neural Network による Machine Learning の手法の 1 つである．

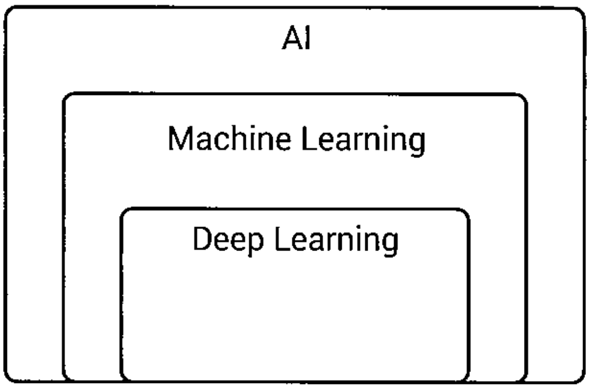


図 5.1　AI の枠組み

5.1.1 Neural Network とは

ここでは，Neural Network について説明する．Neural Network とは，脳機能に見られるいくつかの特性を計算機上のシミュレーションによって表現することを目指した数学モデルである．

一般的な Neural Network のモデル図を図 5.2 に示す．

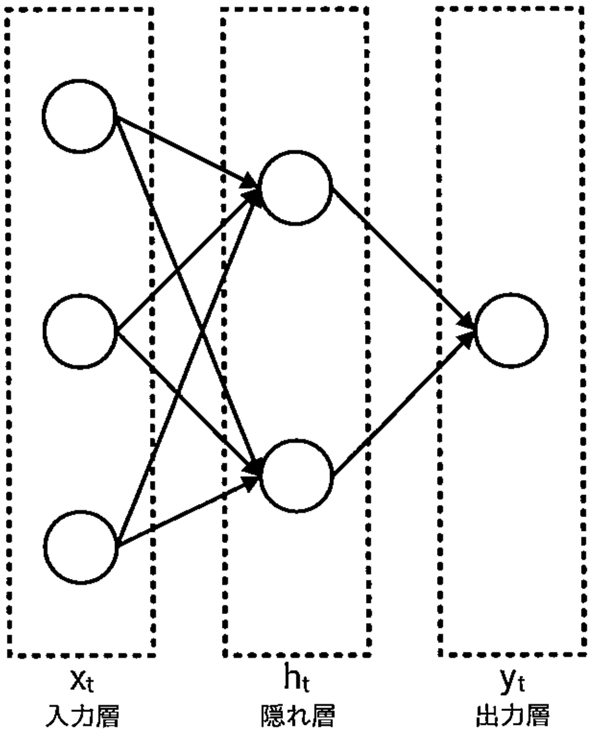


図 **5.2**　Neural Network のモデル

左から順に入力層，隠れ層，出力層となっている．入力層から入力されたデータは，定められた計算が行われながら，隠れ層を経由して出力層に順伝播する．入力データは −1 〜+1 の間に収まるように正規化されている必要がある．

そして 1 つ 1 つの丸いものはニューロン (ノード) と呼ばれている．各ニューロンは，1 つ以上の入力を受け取り，1 つの出力 (シナプス) を生成する関数とも考えられる．各ニューロンには，重み・バイアスのパラメータがあり，活性化関数を持っている．

重み：ニューロン同士の結合強度ともいう．入力ごとに重みが乗算される．

バイアス：ニューロンの発火のしやすさに関わります．重みが 1 の入力とも考えらる．

活性化関数：出力を計算する関数です．同じ層のニューロンの活性化関数は共通である．

1 つのニューロンの動作は図 **5.3** のようになり，数式で記述すると次のようになる．

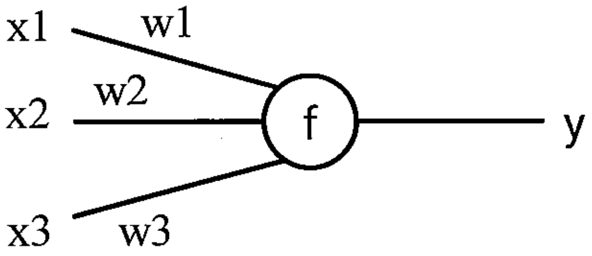


図 **5.3**　1 つのニューロンの動作

- 入力：$x1, x2, x3$
- 重み：$w1, w2, w3$
- バイアス:b
- 活性化関数：$f(u)$
- 出力:y

とすると

$$u = x1 \times w1 + x2 \times w2 + x3 \times w3 + b$$
$$y = f(u)$$

よって

$$y = f(x1 \times w1 + x2 \times w2 + x3 \times w3 + b)$$

となる．

1つ1つの動作は単純であるが，ニューロンの数を増やすほど，層を深くするほど複雑な関数を表現することができる．

活性化関数を $f(u)$ としたが，実際には，図 **5.4** に示すようにランプ関数 (ReLU) や図 **5.5** に示すようにシグモイド関数や図 **5.6** に示すように双曲線関数 (tanh) などがあります．

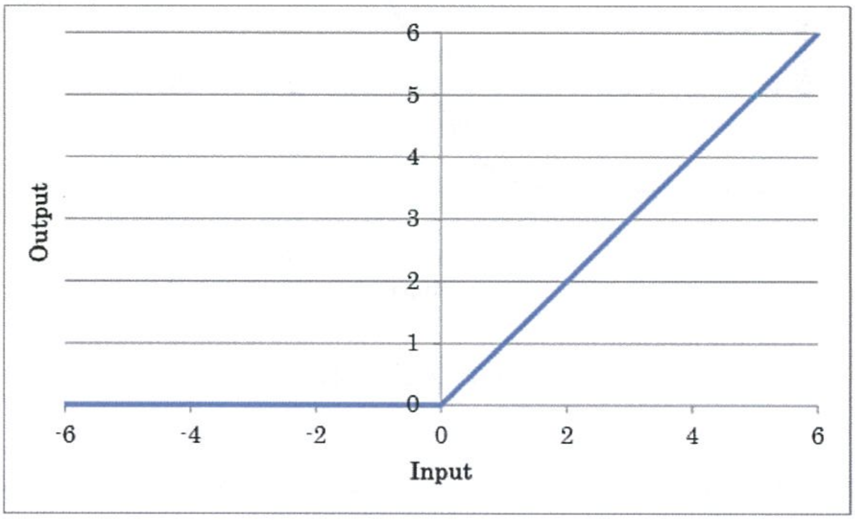


図 **5.4**　ReLU 関数

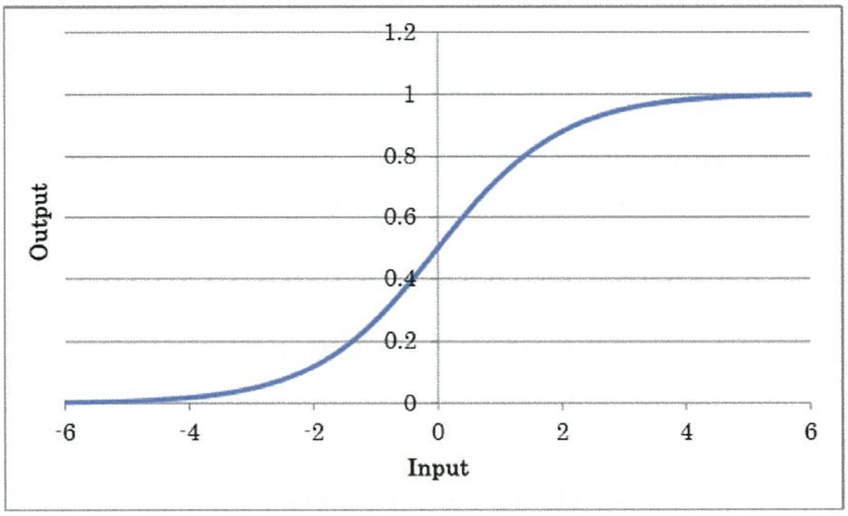


図 5.5　シグモイド関数

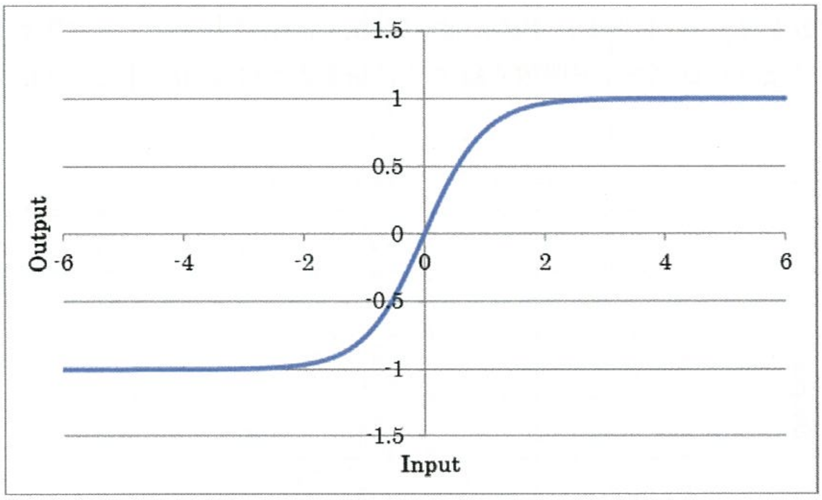


図 5.6　双曲線関数

また，Neural Network にも様々なモデルがあり，DNN(Deep Neural Network)・CNN(Convolution Neural Network)・RNN(Recurrent Neural Network) などがある．

5.1.2　深層学習の仕組み

深層学習とは，Neural Network を利用した機械学習の一種である．
図 **5.7** に 1 層の Neural Network の処理の流れを示す．

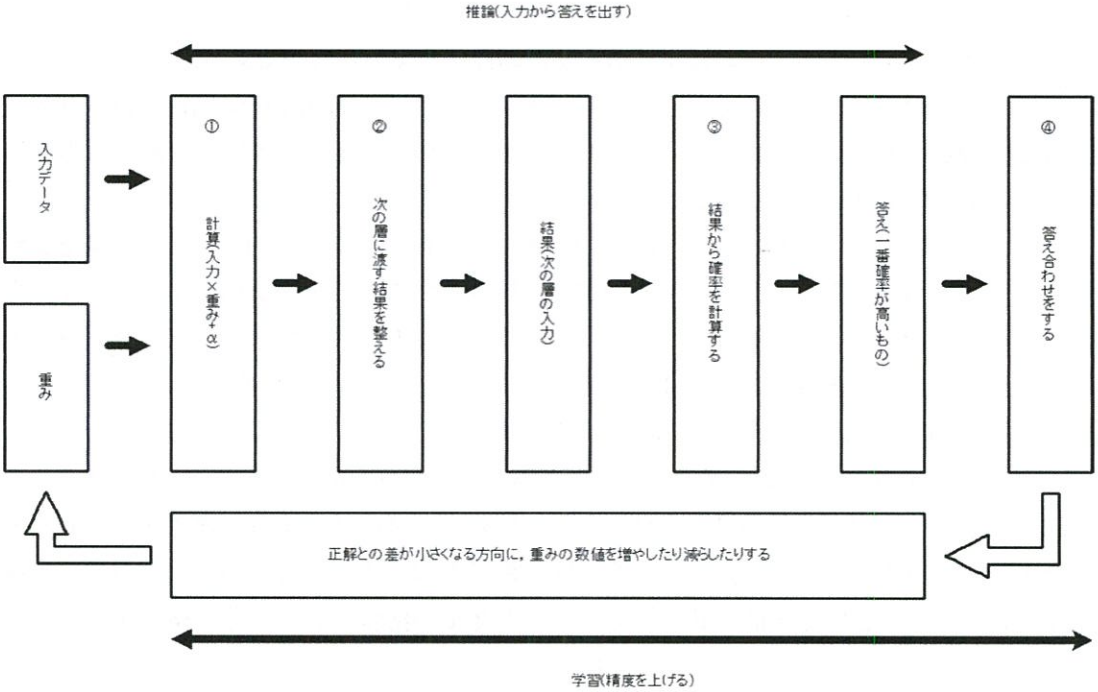


図 **5.7**　Neural Network の処理の流れ

①〜③の答えを出すところまでで止めるのが推論, ①〜④まで行い，重みパラメータを更新していくのが学習である．

Neural Network の出力と目的関数の出力 (正解データ) との誤差を小さくすることを，学習を進めることと捉える．つまり，誤差 (誤差関数) が小さくなるようなパラメータ (重み・バイアス) を求めることを学習という．

例えば，誤差関数が図 **5.8** のようなものだとする．

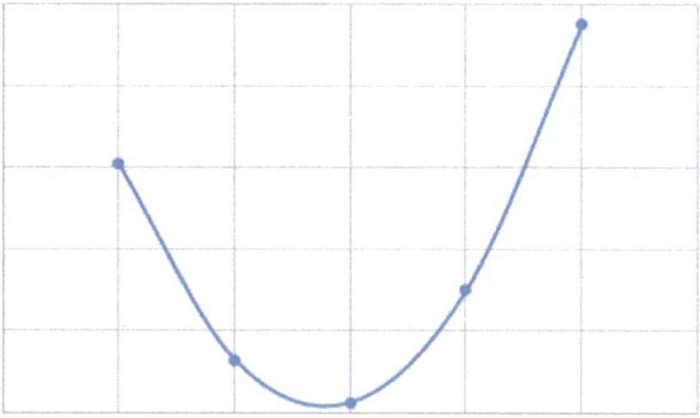


図 **5.8**　誤差関数の例

この場合，誤差関数の値を最小にするには次の方法が考えられる．誤差関数を E として，$E'=0$ となる w を考える．これはよくある関数の極値を求める方法である．

しかし，この方法は現実的ではない．その理由は，実際の Neural Network のパラメータは膨大な数であるからである．パラメータの数だけ偏微分をして多次元の連立方程式を解くのは実時間では不可能である．

そこで勾配降下法を利用する．これは，「勾配」が「降下」する方向にパラメータを更新しようという方法である．これを学習回数分，繰り返すことで誤差を小さくしていく．当然「勾配」を計算するには微分を行うが，勾配降下法は方程式を解かないため計算量が少なくなる．また，勾配降下法にも種類があり，最急降下法 (バッチ勾配降下法)・確率的勾配降下法 (SGD)・ミニバッチ勾配降下法などがある．

しかし，Neural Network の構造が複雑になると誤差関数を微分するのは困難である．この問題を解決したのが，誤差逆伝播法である．誤差逆伝播法は，微分の連鎖律（合成関数の微分）の性質を利用する．誤差逆伝播法は，図 **5.9** に示す通りである．出力の誤差を微分して，1 つ前の層に逆伝播させて微分して，また 1 つ前のという手順で微分を行う．簡単な微分を繰り返すことで計算量を減らしている．

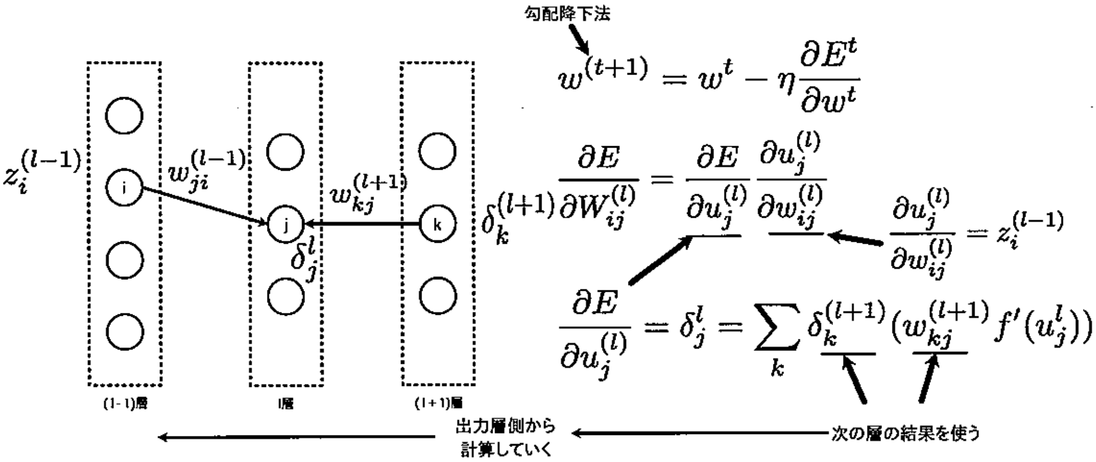


図 **5.9**　誤差逆伝播法

誤差逆伝播法は，誤差を逆伝播させて微分を行うものでしたが，実際に微分されるのは活性化関数である．

実際にシグモイド関数の微分を行ってみる．シグモイド関数を $f(x)$ とすると，

$$f(x) = \frac{1}{1+e^{-x}}$$

であるから，シグモイド関数の微分は

$$f'(x) = (1 - f(x))f(x)$$

このように簡単に微分を計算することができる．そして，同じ層のニューロンが共通の活性化関数を持つ理由はそうしておくことで計算が楽になるためである．

微分 (勾配) の計算ができたため，パラメータを更新することができる．実際に，パラメータをどういう計算で更新するかにも様々な方法がある．

- Momentum・Nesterov の加速勾配降下法
- Adagrad
- Adadelta
- RMSprop
- Adam

現在のところ，Adam が最も評価されている手法のようである．

Adam	$m_t = \beta_1 m_{t-1} + (1-\beta_1)g_t$ $v_t = \beta_2 m_{t-1} + (1-\beta_2)g_t^2$ $\hat{m}_t = \dfrac{m_t}{1-\beta_1^t} \quad \hat{v}_t = \dfrac{m_t}{1-\beta_2^t}$ $\theta_{t+1} = \theta_t - \alpha \dfrac{\hat{m}_t}{\sqrt{\hat{v}_t}+\varepsilon}$

図 5.10　パラメータの更新法 (Adam)

ここまで Neural Network の学習の仕組みを説明してきた．

まとめると，深層学習とは Neural Network の最適パラメータを求めることである．パラメータは勾配降下法で少しずつ更新するが，パラメータの更新には勾配を求める (誤差を微分する) 必要があり，それは誤差逆伝播法により簡単に勾配を求めることができるから，その勾配からパラメータを更新するというものである．

5.1.3 Neural Network Console

ニューラルネットワークの設計手法を紹介する．

ネットワークモデルは Neural Network Console(以下 NNC) というソフトウェアを用いて設計した．

NNC とは 2017 年 8 月 17 日から Sony より無償提供されているディープラーニング (深層学習) のプログラムを生成できる統合開発環境である．NNC のコアライブラリには，同じく Sony よりオープンソースで公開されている nnabla という機械学習ライブラリを使用している．本来，ディープラーニングのプログラム開発は，入力に応じて最適な Neural Network を構築した上で，性能の向上に向けた試行錯誤を繰り返して Neural Network を最適化した後，製品やサービスへの搭載に至ることができる．また，一般的なディープラー

ニングのプログラム生成では，Neural Network の構造を，プログラムのコードを記述して，その関数ブロックを組み合わせることで構築する．しかし，NNC を利用することで，関数ブロックの概念はそのままに，簡便な形で GUI 上に表現されており，画面上には，コンポーネントの形でレイヤー (関数ブロック) が用意されている．それらを自由に配置するという直感的な操作で Neural Network の設計，学習，評価などを行いながらディープラーニングのプログラムを効率よく開発することができる．

5.2 室外アドバイス文章の生成

ここでは，アドバイス文章の生成方法について説明する．

5.2.1 生成の方法

はじめに，文章生成方法の大まかな流れについて説明する．

1. ゲートウェイから AI が動作している VPS に CSV データが送られてくる．
2. CSV データを NNC で作成した AI に入力し推論結果を得る．
3. 推論結果をもとにアドバイス文章を生成する．

推論によって得られる結果は，体感温度の情報を 10 段階に，WBGT の情報を 5 段階に，雨量の情報を 6 段階に，風速の情報を 13 段階に分類したものである．したがって，生成されるアドバイス文章の種類は 10×5×6×13 の 3900 種類である．アドバイス文章は表 **5.1** ～表 **5.4** に対応して決められる．

表 **5.1** 体感温度の段階

体感温度の値 [℃]	アドバイス文
～−12.2	凍るくらいに寒いです．
−12.1～-6.7	すごく寒く感じるでしょう．
-6.6～−1.1	寒く感じるでしょう．
−1.0～4.4	すこし寒く感じるでしょう．
4.5～10.0	涼しく感じるでしょう．
10.1～15.6	心地よく感じるでしょう．
15.7～21.1	暖かく感じるでしょう．
21.2～27.2	暑く感じるでしょう．
27.3～32.2	すごく暑く感じるでしょう．
32.3～37.8	溶けるくらいに暑いです．

表 5.2　WBGT の段階 (室外用アドバイス文章)

WBGT の値 [℃]	アドバイス文
〜21 .0	熱中症の危険はほぼありません.
21.1〜25.0	運動の際は熱中症に気を付けましょう.
25.1〜28.0	炎天下はできるだけ避けたほうがいいです.
28.1〜31.0	激しい運動は控えたほうがいいです.
31.1〜	運動は控えましょう.

表 5.3　雨量の段階

雨量 [mm]	アドバイス文
0	降っていないです.
0.1〜10.0	やや強い雨です.
10.1〜20.0	強い雨です.
20.1〜30.0	激しい雨です.
30.1〜50.0	非常に激しい雨です.
50.1〜	猛烈な雨です.

表 5.4　風速の段階

風速 [m/s]	アドバイス文
0〜0.2	平穏です.
0.3〜1.5	至軽風です.
1.6〜3.3	軽風です.
3.4〜5.4	軟風です.
5.5〜7.9	和風です.
8.0〜10.7	疾風です.
10.8〜13.8	雄風です.
13.9〜17.1	強風です.
17.2〜20.7	疾強風です.
20.8〜24.4	大強風です.
24.5〜28.4	全強風です.
28.5〜32.7	暴風です.
32.8〜	颱風です.

5.2.2　学習方法

はじめに，学習に用いたデータセットについて説明する.

データは，気象庁の HP から高松の気象データを 2013 年〜2018 年の 5 年分をダウンロードした. 気象データの内訳は，1 時間ごとの日時・気温・雨量・風速・湿度である. 以上のデータより，学習に用いるデータセットを作成した.

データセットの形は，入力データ 1 行 7 列と正解データ 1 行 4 列の 1 行 11 列である. 入力データは，−1〜+1 に収まるように正規化がされている. 出力データは，分類問題として学習させるため，あらかじめクラス分けが施されている.

データセットの持っている情報について説明する. 入力データは，日付・気温・雨量・風速・湿度・体感温度・WBGT（熱中症指数）の情報を持っている. 正解データは，体感温度・WBGT（熱中症指数）・雨量・風速の情報を持っている.

作成したデータセットは，43800 件である. このうち，80 %の 35040 件を学習用データとし，残りの 20 %の 8760 件を検証用データセットとした.

データセットの例を図 **5.11** に示す.

	A	B	C	D	E	F	G	H	I	J	K
1	x__1:date	x__2:temp	x__3:rain	x__4:wind	x__5:humi	x__6:sensi	x__7:wbgt	sensible	wbgt	rain	wind_speed
2	-0.99	0.009	0	0.008	0.61	-0.01743	0.060138	2	0	0	1
3	-0.98977	0.002	0	0.017	0.66	-0.0643	0.056614	2	0	0	2
4	-0.98955	-0.001	0	0.015	0.69	-0.06291	0.055281	2	0	0	2
5	-0.98932	0.006	0	0.013	0.7	-0.0486	0.060359	2	0	0	1
6	-0.9891	0.016	0	0.014	0.7	-0.04171	0.067342	2	0	0	1
7	-0.98887	0.016	0	0.013	0.76	-0.04167	0.068959	2	0	0	1
8	-0.98864	0.011	0	0.013	0.81	-0.05037	0.0667	2	0	0	1
9	-0.98842	0.007	0	0.001	0.79	0.029422	0.063327	3	0	0	0
10	-0.98819	0.025	0	0.009	0.73	-0.01306	0.074562	2	0	0	1
11	-0.98797	0.041	0	0.016	0.77	-0.02499	0.087439	2	0	0	2
12	-0.98774	0.024	0	0.02	0.89	-0.06411	0.078413	2	0	0	2
13	-0.98751	0.038	0	0.019	0.82	-0.04004	0.086796	2	0	0	2
14	-0.98729	0.056	0	0.02	0.7	-0.01546	0.096181	2	0	0	2
15	-0.98706	0.071	0	0.028	0.61	-0.01022	0.103847	3	0	0	2
16	-0.98684	0.078	0	0.018	0.56	0.018802	0.106923	3	0	0	2
17	-0.98661	0.072	0	0.022	0.55	0.004211	0.102185	3	0	0	2
18	-0.98638	0.071	0	0.016	0.61	0.014759	0.103847	3	0	0	2
19	-0.98616	0.064	0	0.017	0.7	0.001065	0.102142	3	0	0	2

図 **5.11**　データセットの例 (室外用アドバイス文章)

次に，作成したネットワーク図を図 **5.12** に示す．

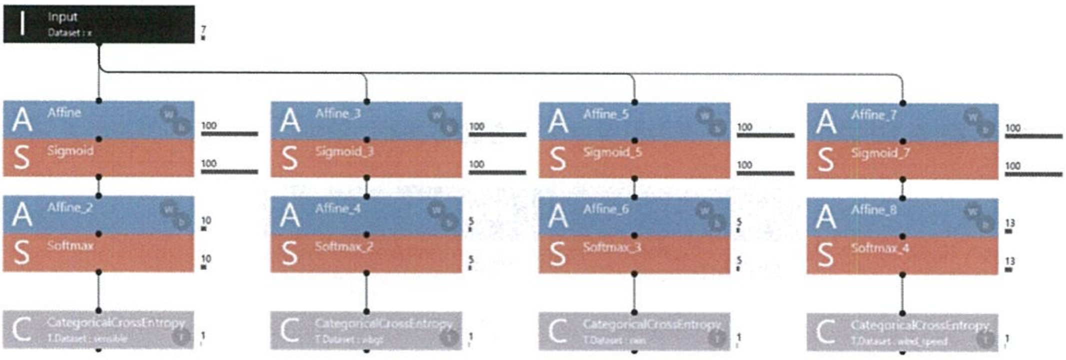


図 **5.12**　室外用文章生成の Neural Network 図

まず，上記のネットワーク図について説明する．これは 4 つの DNN(Deep Neural Network) としてとらえることができる．それぞれのネットワークは同じ構造（入力層，中間層 1，中間層 2，出力層）をとっている．各層の説明を表 **5.5** に示す．

表 5.5　ネットワーク構造 (室外用アドバイス文章)

層	説明
入力層	1 行 7 列のデータを入力としている
中間層 1	ノード数は 100，活性化関数は Sigmoid を設定している
中間層 2	ノード数は分類数によるため，左から順に 10，5，6，13 となっている 活性化関数は Softmax を設定する
出力層	誤差関数には CategoricalCrossEntropy を設定している

次に，図 **5.12** の各レイヤーについて説明する．

Affine　全ての入力値から，指定する全ての出力ニューロンへの結合を持つ全結合層である．

Tanh　入力値の tanh による処理結果が出力される．tanh の特性により出力は -1.0〜$+1.0$ の範囲で出力される．

Sigmoid　入力値の Sigmoid による処理結果が出力される．Sigmoid の特性により出力は 0.0〜1.0 の範囲で出力される．そのため，確率を得たいときに使用される．

CategoricalCrossEntropy　カテゴリ Index で与えられるデータセットの変数との相互情報量を最小化する Neural Network の出力層である．

次に，図 **5.12** のネットワークの計算コストを図 **5.13** に示す．

Type	Value
Output	877
CostParameter	6,533
CostAdd	866
CostMultiply	0
CostMultiplyAdd	6,100
CostDivision	433
CostExp	433
CostIf	0

図 **5.13**　ネットワークの計算コスト (室外用アドバイス文章)

次に，学習に使用した勾配法について説明する．ミニバッチ学習（確率的勾配法）で学習し，勾配の更新には Adam(Adaptive moment estimation) を利用した．

最後に，学習におけるパラメータを表 **5.6** に示す．

表 **5.6**　パラメータ (室外用アドバイス文章)

Evaluate network "MainRuntime" using "Validation" dataset	
Batch size	128
solver:Adam	Learning rate(Alpha):0.001 Beta1:0.9 Beta2:0.999 Epsilon:1e-08
Weight decay is not applied	

5.2.3　学習結果

学習の様子を表した図を図 **5.14** に示す．

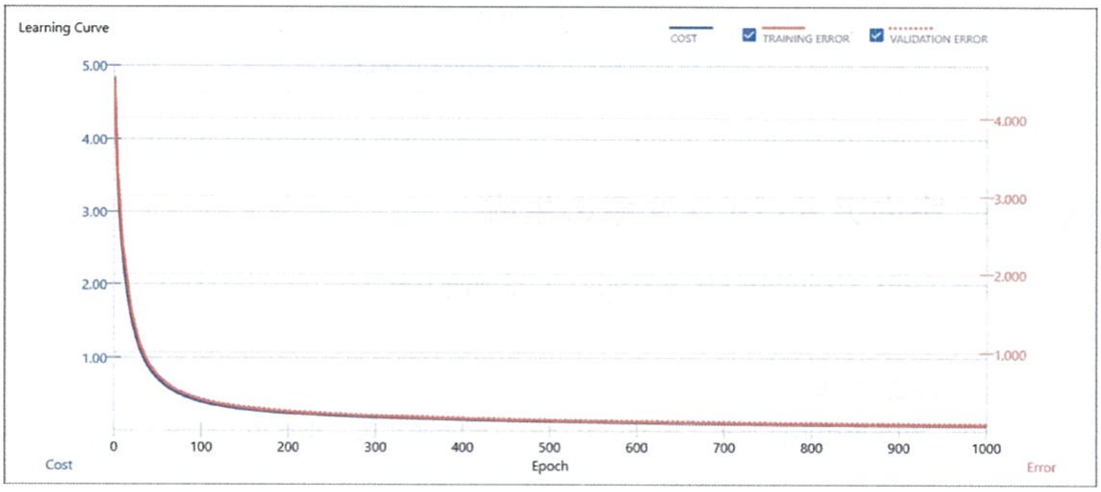


図 **5.14**　学習曲線 (室外用アドバイス文章)

横軸は学習世代数，左縦軸は学習データにおけるコスト関数の値，右縦軸は学習データおよび評価データにおける誤差関数の値を示している．

右縦軸の学習データおよび評価データにおける誤差関数が世代数が増えるごとに減少していることから，正しく学習が進んでいることがわかる．

次に，学習結果の精度を表 **5.7** に示す．

表 5.7　学習結果 (室外用アドバイス文章)

Evaluate network "MainRuntime" using "Validation" dataset	
Variable:rain	Accuracy:0.9996 Avg.Precision:0.7539 Avg.Recall:0.8326 Avg.F-Measures:0.7868
Variable:sensible	Accuracy:0.9937 Avg.Precision:0.9902 Avg.Recall:0.9204 Avg.F-Measures:0.9367
Variable:wbgt	Accuracy:0.9956 Avg.Precision:0.9930 Avg.Recall:0.9917 Avg.F-Measures:0.9923
Variable:wind_speed	Accuracy:0.9995 Avg.Precision:0.9994 Avg.Recall:0.9877 Avg.F-Measures:0.9934

5.3 室内アドバイス文章の生成

ここでは，アドバイス文章の生成方法について説明する．

5.3.1 生成の方法

はじめに，文章生成方法の大まかな流れについて説明する．生成の方法は，室外アドバイス文章の生成法とほぼ同じである．

1. ゲートウェイから AI が動作している VPS に CSV データが送られてくる．
2. CSV データを NNC で作成した AI に入力し推論結果を得る．
3. IAQ（Indoor Air Quality）の情報は閾値を設けて 6 段階に分類している．
4. 推論結果をもとにアドバイス文章を生成する．

推論されて得られる結果は，気象データから，温度の情報を 5 段階に，湿度の情報を 3 段階に，WBGT の情報を 5 段階に分類している．

この分類した結果をもとにアドバイス文章を生成している．したがって，生成されるア

ドバイス文章の種類は 5×3×5×6 の 450 種類である．アドバイス文章は表 **5.8**～表 **5.11** に対応して決められる．

表 **5.8**　室内温度の段階

気温 [℃]				アドバイス文章
春	夏	秋	冬	
～22.0	～23.0	～19.0	～17.0	室温を上げたほうがいいです．
22.1～25.0	23.1～26.0	19.1～22.0	17.1～20.0	室温を少し上げたほうがいいです．
25.1～26.0	26.1～28.0	22.1～25.0	20.1～23.0	快適に過ごせる室温です．
26.1～29.0	28.1～31.0	25.1～28.0	23.1～26.0	室温を少し下げたほうがいいです．
29.1～	31.1～	28.1～	26.1～	室温を上げたほうがいいです．

表 **5.9**　湿度の段階

湿度 [%]	アドバイス文章
～40.0	加湿器を使ったほうがいいです．
40.1～60.0	快適に過ごせる湿度です．
60.1～	除湿器を使ったほうがいいです．

表 **5.10**　WBGT の段階 (室内用アドバイス文章)

WBGT の値 [℃]	アドバイス文
～21.0	熱中症の危険はほぼありません．
21.1～25.0	運動の際は熱中症に気を付けましょう．
25.1～28.0	炎天下はできるだけ避けたほうがいいです．
28.1～31.0	室温の上昇に注意しましょう．
31.1～	外出は控えたほうがいいです．

表 5.11　IAQ の段階

IAQ の値	アドバイス文章
0〜50	空気清浄度は非常にいいです．
51〜100	空気清浄度に問題はありません．
101〜150	空気が少し汚れています．
151〜200	換気をしたほうがいいです．
201〜300	換気をしてください．
301〜500	生活できないくらいに空気が汚いです．

5.3.2　学習方法

はじめに，学習に用いたデータセットについて説明する．

データは，気象庁の HP から高松の気象データを 2013 年〜2018 年の 5 年分をダウンロードした．気象データの内訳は，1 時間ごとの日時・気温・湿度である．以上のデータより，学習に用いるデータセットを作成した．

データセットの形は，入力データ 1 行 4 列と正解データ 1 行 3 列の 1 行 7 列である．入力データは，−1〜+1 に収まるように正規化がされている．出力データは，分類問題として学習させるため，あらかじめクラス分けが施されている．

データセットの持っている情報について説明する．入力データは，日付・気温・湿度・WBGT(熱中症指数) の情報を持っている．正解データは，気温・湿度・WBGT(熱中症指数) の情報を持っている．

作成したデータセットは，43800 件である．このうち，80 %の 35040 件を学習用データとし，残りの 20 %の 8760 件を検証用データセットとした．

データセットの例を図 **5.15** に示す．

	A	B	C	D	E	F	G
1	x__1:date	x__2:temp	x__3:humi	x__4:wbgt	temp	humi	wbgt
2	-0.99	0.009	0.61	0.060138	0	2	0
3	-0.98977	0.002	0.66	0.056614	0	2	0
4	-0.98955	-0.001	0.69	0.055281	0	2	0
5	-0.98932	0.006	0.7	0.060359	0	2	0
6	-0.9891	0.016	0.7	0.067342	0	2	0
7	-0.98887	0.016	0.76	0.068959	0	3	0
8	-0.98864	0.011	0.81	0.0667	0	3	0
9	-0.98842	0.007	0.79	0.063327	0	3	0
10	-0.98819	0.025	0.73	0.074562	0	2	0
11	-0.98797	0.041	0.77	0.087439	0	3	0
12	-0.98774	0.024	0.89	0.078413	0	3	0
13	-0.98751	0.038	0.82	0.086796	0	3	0
14	-0.98729	0.056	0.7	0.096181	0	2	0
15	-0.98706	0.071	0.61	0.103847	0	2	0
16	-0.98684	0.078	0.56	0.106923	0	1	0
17	-0.98661	0.072	0.55	0.102185	0	1	0
18	-0.98638	0.071	0.61	0.103847	0	2	0
19	-0.98616	0.064	0.7	0.102142	0	2	0
20	-0.98593	0.051	0.79	0.095598	0	3	0

図 5.15　データセットの例 (室内用アドバイス文章)

次に，作成したネットワーク図を図 **5.16** に示す．

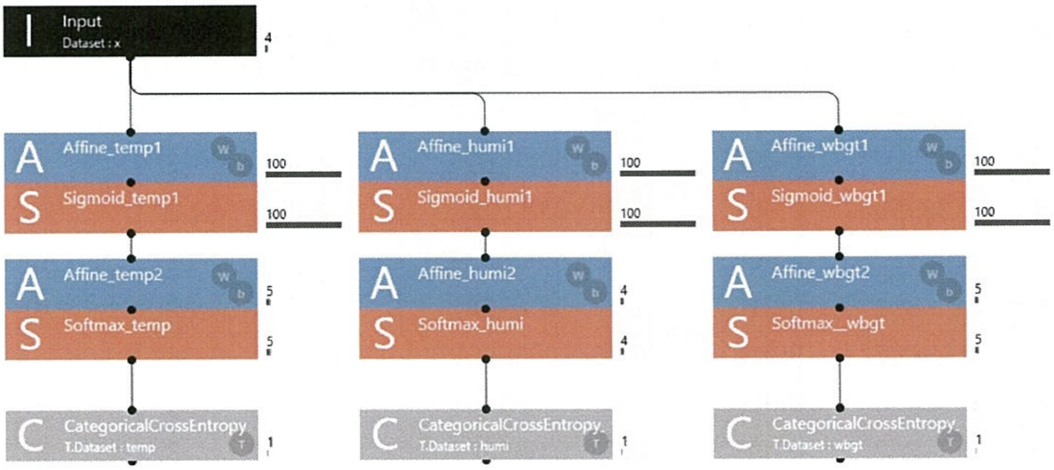


図 5.16　室内用文章生成のニューラルネットワーク図

まず，上記のネットワーク図について説明する．これは 3 つの DNN(Deep Neural Network) としてとらえることができる．それぞれのネットワークは同じ構造（入力層，中間層 1，中間層 2，出力層）をとっている．各層の説明を表 **5.12** に示す．

表 5.12　ネットワーク構造 (室内用アドバイス文章)

層	説明
入力層	1 行 4 列のデータを入力としている
中間層 1	ノード数は 100，活性化関数は Sigmoid を設定している
中間層 2	ノード数は分類数によるため，左から順に 5，4，5 となっている 活性化関数は Softmax を設定する
出力層	誤差関数には CategoricalCrossEntropy を設定している

各レイヤーについては，ネットワークが室外アドバイス文章の生成と同様であるため，省略する．

次に，図 **5.16** のネットワークの計算コストを図 **5.17** に示す．

Type	Value
Output	635
CostParameter	2,914
CostAdd	628
CostMultiply	0
CostMultiplyAdd	2,600
CostDivision	314
CostExp	314
CostIf	0

図 **5.17**　ネットワークの計算コスト (室内用アドバイス文章)

次に，学習に使用した勾配法について説明する．ミニバッチ学習 (確率的勾配法) で学習し，勾配の更新には Adam(Adaptive moment estimation) を利用した．

最後に，学習におけるパラメータを表 **5.13** に示す．

表 **5.13**　パラメータ (室内用アドバイス文章)

Evaluate network ”MainRuntime” using ”Validation” dataset	
Batch size	128
solver:Adam	Learning rate(Alpha):0.001 Beta1:0.9 Beta2:0.999 Epsilon:1e-08
Weight decay is not applied	

5.3.3　学習結果

学習の様子を表した図を図 **5.18** に示す．

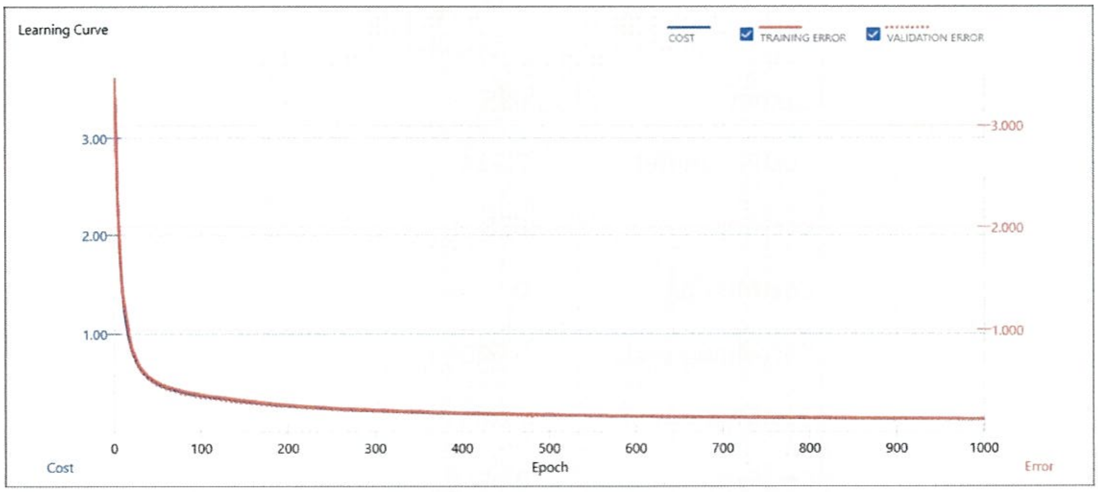


図 5.18　学習曲線 (室内用アドバイス文章)

横軸は学習世代数，左縦軸は学習データにおけるコスト関数の値，右縦軸は学習データおよび評価データにおける誤差関数の値を示している．

右縦軸の学習データおよび評価データにおける誤差関数が世代数が増えるごとに減少していることから，正しく学習が進んでいることがわかる．

次に，学習結果の精度を表 **5.14** に示す．

表 5.14　学習結果 (室内用アドバイス文章)

Evaluate network "MainRuntime" using "Validation" dataset	
Variable:humi	Accuracy:1 Avg.Precision:1 Avg.Recall:1 Avg.F-Measures:1
Variable:temp	Accuracy:0.9715 Avg.Precision:0.9422 Avg.Recall:0.9416 Avg.F-Measures:0.9418
Variable:wbgt	Accuracy:0.9976 Avg.Precision:0.9960 Avg.Recall:0.9948 Avg.F-Measures:0.9954

5.4 気温の予測

ここでは，気温の予測について説明する.

5.4.1 予測の方法

ここでは，気温の予測方法について説明する.

まず，ゲートウェイから AI が動作している VPS(AI サーバ) に CSV データが送られてくる. 次に，その CSV データを NNC で作成した AI に入力し推論結果を得る.

1. ゲートウェイから AI が動作している VPS に CSV データが送られてくる.
2. CSV データを NNC で作成した AI に入力し推論結果を得る.
3. CSV データに推論結果を反映させる.
4. 2. と 3. を 24 回繰り返す.

入力データは，過去 24 時間分の日付・気温・湿度を正規化してある. そして，推論によって得られる結果は，1 時間後の気温と湿度の予測値である. この出力された気温・湿度を入力データに加え，それを 24 回繰り返すことにより 24 時間後までの気温・湿度を予測している.

5.4.2 LSTM について

過去のデータから未来のデータを予測するように，時系列 (連続) データを扱う Neural Network のことを RNN(Recurrent Neural Network) という. RNN は，図 **5.19** のように隠れ層の値を再び隠れ層に入力するという構造をしているのが特徴であり，未来のデータを予測させるために使われることが多い.

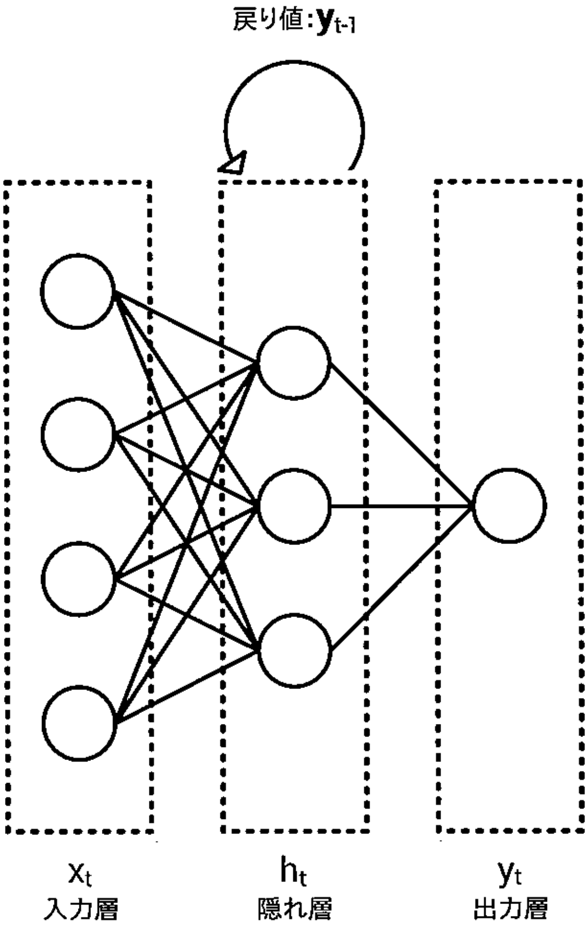


図 5.19　RNN のモデル

隠れ層の値を再び隠れ層に入力するということから通常の Neural Network と違いが大きいものと思われるが，RNN を展開すると図 **5.20** のようになる．RNN は，同じネットワークの複数のコピーであり，それぞれが後続の Network に値を渡していると考えることができるため，セル (図の点線枠内) が鎖状に並んでいるようにみえる．これは時系列データを扱うために，配列やリストと密接な関係があることを示している．また，セル内の構造としては単一の tanh 層という，非常に単純な構造をしている．しかし，通常の RNN では長時間前のデータを利用しようとすると，勾配が消滅するという問題があり，短時間のデータしか処理できないのが欠点だった．

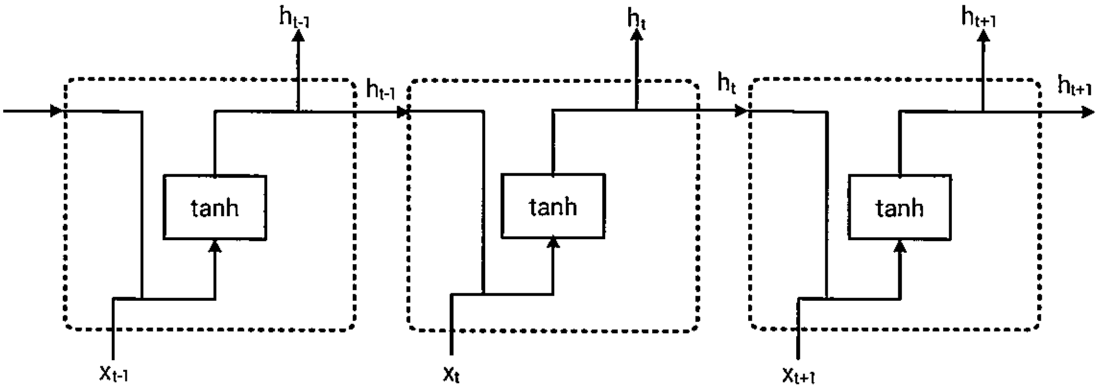


図 5.20　RNN の展開図

そのため，今回予測を行う AI の Neural Network には，RNN の短時間のデータしか処理できないという欠点を改善した LSTMN(Long Short Term Memory Network) を使用した．構成は図 **5.21** のように RNN の隠れ層を LSTM Block と呼ばれるメモリと 3 つのゲートを持つ Block に置き換える形なっている．

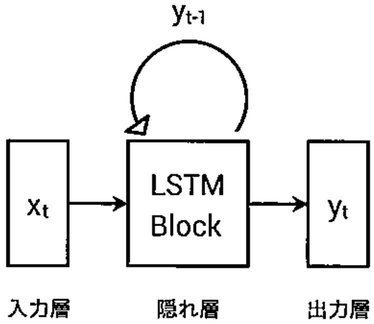


図 5.21　LSTMN

LSTMN を展開すると図 **5.22** のようになり，RNN と同じようにセル (LSTM Block) が鎖状に並んでいる．これは時系列データを扱うために，配列やリストと密接な関係があることを示している．図 **5.22** のそれぞれの線は，ベクトル全体を，一つのノードの出力から他のノードの入力に運ぶ．円は，ベクトルの加算のような，一点の操作を，四角は学習される Neural Network の層を表している．

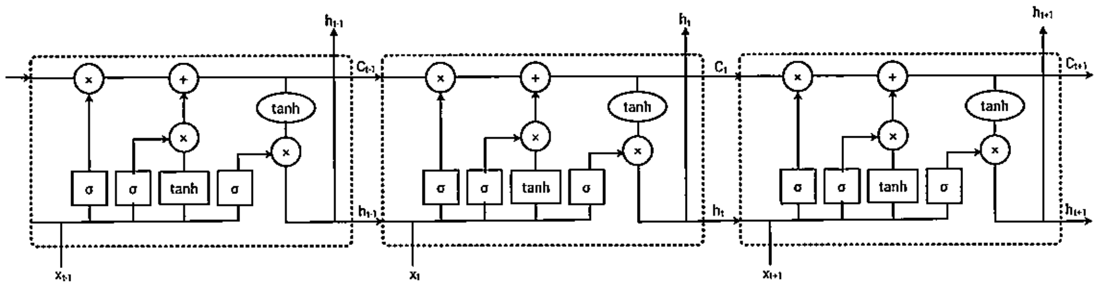


図 **5.22**　LSTMN の展開図

セルの状態はベルトコンベアに似ており，線形相互作用のみを伴い，鎖全体を流れる．LSTM Block は，セルの状態に対し情報を削除したり，追加したりする機能を持っており，この操作はゲートとよばれる構造により制御されている．

ゲートは選択的に情報を通す方法であり，図 **5.23** のように Sigmoid 層とある一点の乗算により構成されている．Sigmoid 層は 0 から 1 までの数値を出力し，この出力は対応する情報が通過する割合を表す実数であり，例えば，0 は「情報を何も通さない」を，1 は「情報をすべて通す」を意味する．LSTM Block は，これらのゲートを通じてセルの状態の情報の保護と制御を実現しており，入力ゲート・出力ゲート・忘却ゲートという 3 つのゲートを持っている．

図 **5.23**　ゲートの構造

LSTM Block の最初のステップは，Sigmoid 層 (忘却ゲート) によって，セルの状態から捨てる情報の判定をすることである．これは,$h_t - _1$ とx_tにより決まり，式 (5.1) によって表すことができる．

$$f_t = \sigma(W_f \times [h_t{-}_1, x_t] + b_f) \tag{5.1}$$

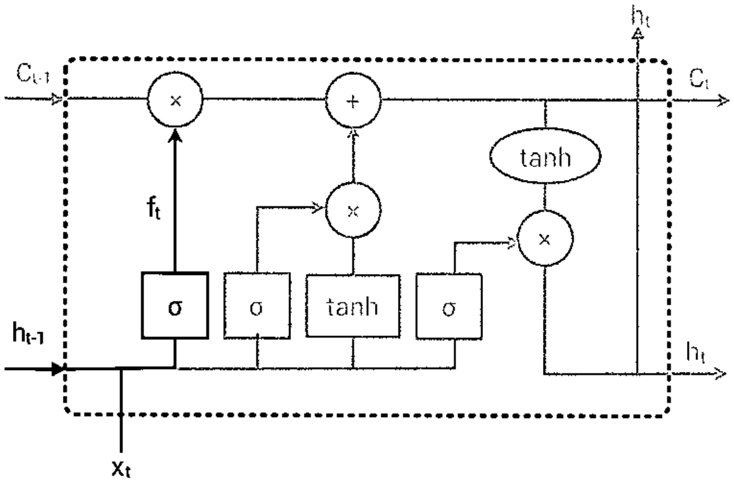


図 5.24　忘却ゲート

次のステップは，セルの状態に保存する新たな情報を判定することである．これには 2 つの部分があり，まず，Sigmoid 層 (入力ゲート) によって，どの情報を更新するかを判定する．次に，tanh 層は，セルの状態に新たに加える情報の候補値のベクトル $\bar{C_t}$ を作成する．これらは，それぞれ式 (5.2) と式 (5.3) によって表すことができる．

$$i_t = \sigma(W_i \times [h_t-_1, x_t] + b_i) \tag{5.2}$$

$$\bar{C_t} = tanh(W_c \times [h_t-_1, x_t] + b_C) \tag{5.3}$$

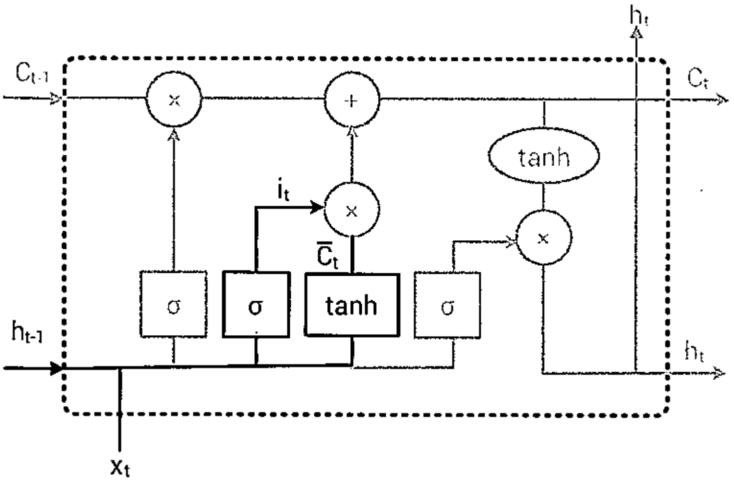


図 5.25　入力ゲート

そして次のステップでは，古いセルの状態 C_{t-1} から新しいセルの状態 C_t から新しいセルの状態 C_t に更新することである．忘却ゲートの出力である f_c と古いセルの状態で乗算

を行い，忘却ゲートで忘れると判定したものを忘れさせる．次に，状態値を更新すると決定した割合でスケーリングされた，新たな候補値である i_t と $\bar{C}_t$ の積とセルの状態で加算を行うことで新しいセルの状態に更新される．これは，式 (5.4) によって表すことができる．

$$C_t = f_t \times C_t{-_1} + i_t \times \bar{C}_t \tag{5.4}$$

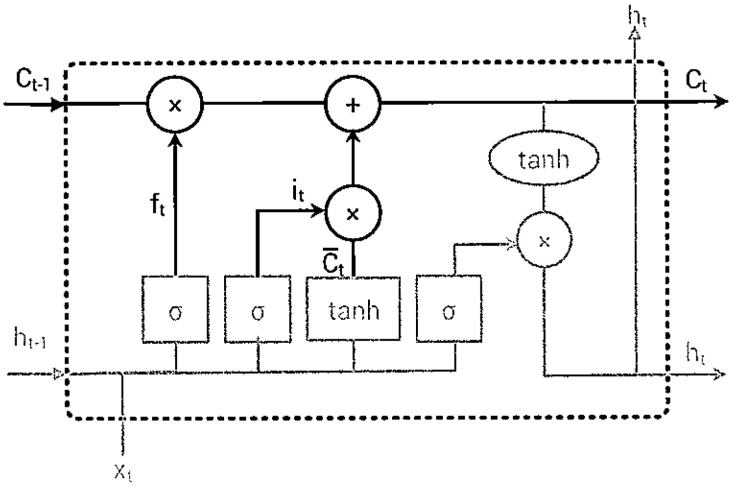


図 **5.26**　セルの状態の更新

最後のステップは，出力するものを判定することである．この出力は，セルの状態に基づいて行われるが，Sigmoid 層 (出力ゲート) によってフィルタリングされているものである．まず，Sigmoid 層ではセルの状態のどの部分を出力するかを判定する．その後，判定された部分のみ出力するため，セルの状態を −1〜+1 の範囲に圧縮するために tanh を適用し，Sigmoid 層の出力である o_t と乗算を行うことで，フィルタリングされた情報を出力する．これは，式 (5.5) と式 (5.6) によって表すことができる．

$$o_t = \sigma(W_o \times [h_t{-_1}, x_t] + b_o) \tag{5.5}$$

$$h_t = o_t * tanh(C_t) \tag{5.6}$$

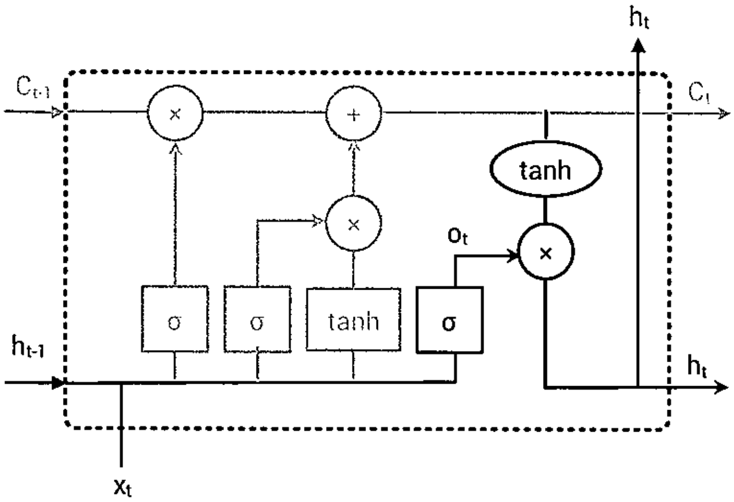


図 5.27　出力ゲート

5.4.3　学習方法

はじめに，学習に用いたデータセットについて説明する．

データは，アドバイス文章の生成と同様に，気象庁の HP から高松の気象データを 2013 年～2018 年の 5 年分をダウンロードした．気象データの内訳は，1 時間ごとの日時・気温・湿度である．以上のデータより，学習に用いるデータセットを作成した．

データセットの形は，入力データ 24 行 3 列と正解データ 24 行 2 列である．入力データは，−1～+1 に収まるように正規化がされている．

データセットの持っている情報について説明する．入力データは，日付・気温・湿度の情報を持っている．正解データは，気温・湿度の情報を持っている．

作成したデータセットは，43800 件である．このうち，80 %の 35040 件を学習用データとし，残りの 20 %の 8760 件を検証用データセットとした．

データセットの例を図 5.28 に示す．

	A	B	C	D	E	F	G	H	I	J
1	-0.98977	0.002	0.66							
2	-0.98955	-0.001	0.69							
3	-0.98932	0.006	0.7							
4	-0.9891	0.016	0.7							
5	-0.98887	0.016	0.76							
6	-0.98864	0.011	0.81							
7	-0.98842	0.007	0.79							
8	-0.98819	0.025	0.73							
9	-0.98797	0.041	0.77							
10	-0.98774	0.024	0.89							
11	-0.98751	0.038	0.82							
12	-0.98729	0.056	0.7							
13	-0.98706	0.071	0.61							
14	-0.98684	0.078	0.56							
15	-0.98661	0.072	0.55							
16	-0.98638	0.071	0.61							
17	-0.98616	0.064	0.7							

	A	B	C
1	0.002	0.66	
2	-0.001	0.69	
3	0.006	0.7	
4	0.016	0.7	
5	0.016	0.76	
6	0.011	0.81	
7	0.007	0.79	
8	0.025	0.73	
9	0.041	0.77	
10	0.024	0.89	
11	0.038	0.82	
12	0.056	0.7	
13	0.071	0.61	
14	0.078	0.56	
15	0.072	0.55	
16	0.071	0.61	
17	0.064	0.7	

図 **5.28**　データセットの例 (気温の予測)

次に，作成したネットワーク図を図 **5.29** に示す．

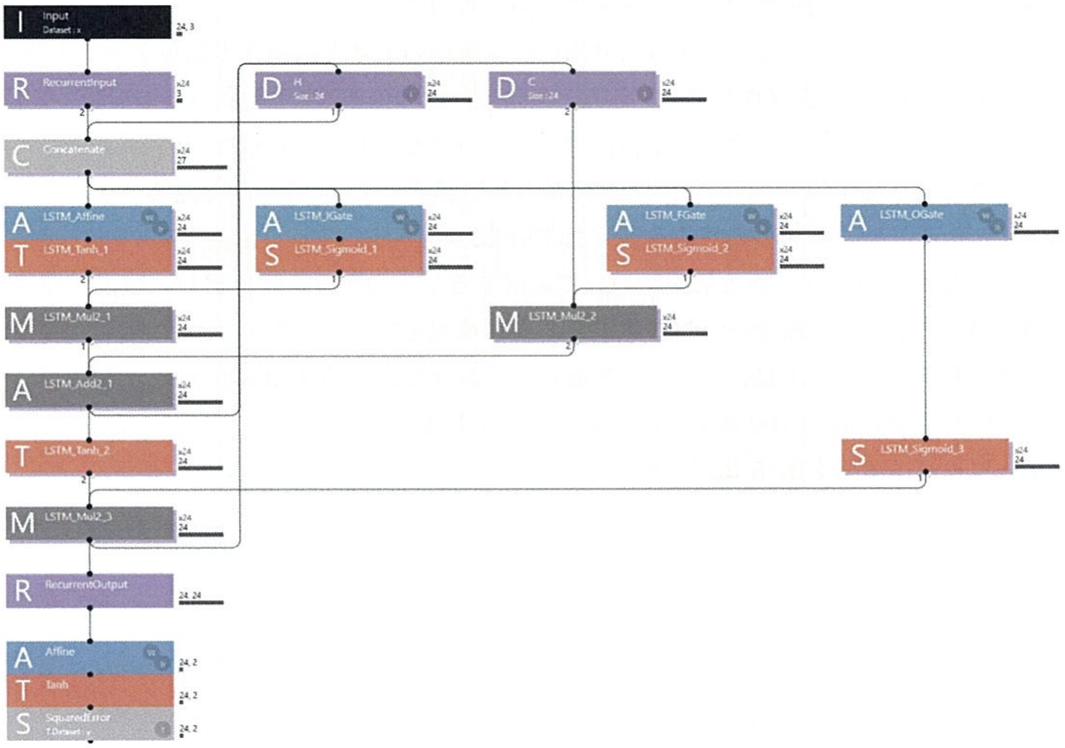


図 **5.29**　気温予測のニューラルネットワーク図

次に，図 **5.29** の各レイヤーについて説明する．

Affine(LSTM_Affine)　全ての入力値から，指定する全ての出力ニューロンへの結合を持つ

全結合層である.

Tanh(LSTM_Tanh)　入力値の tanh による処理結果が出力される. tanh の特性により出力は -1.0〜$+1.0$ の範囲で出力される.

Sigmoid(LSTM_Sigmoid)　入力値の Sigmoid による処理結果が出力される. Sigmoid の特性により出力は 0.0〜1.0 の範囲で出力される. そのため, 確率を得たいときに使用される.

SquaredError　データセットの変数との二乗誤差を最小化する Neural Network の出力層である. Neural Network により回帰問題を解く際 (連続値を出力する Neural Network を最適化する際) に使用される.

次に, 図 **5.29** のネットワークの計算コストを図 **5.30** に示す.

Type	Value
Output	10,152
CostParameter	92,208
CostAdd	7,056
CostMultiply	1,728
CostMultiplyAdd	89,856
CostDivision	2,928
CostExp	2,928
CostIf	0

図 **5.30**　ネットワークの計算コスト (気温の予測)

次に, 学習に使用した勾配法について説明する. ミニバッチ学習 (確率的勾配法) で学習し, 勾配の更新には Adam(Adaptive moment estimation) を利用した.

最後に, 学習におけるパラメータを表 **5.15** に示す.

表 **5.15**　パラメータ (気温の予測)

Evaluate network "MainRuntime" using "Validation" dataset	
Batch size	64
solver:Adam	Learning rate(Alpha):0.001 Beta1:0.9 Beta2:0.999 Epsilon:1e-08
Weight decay is not applied	

5.4.4　学習結果

学習の様子を表した図を図 **5.31** に示す．

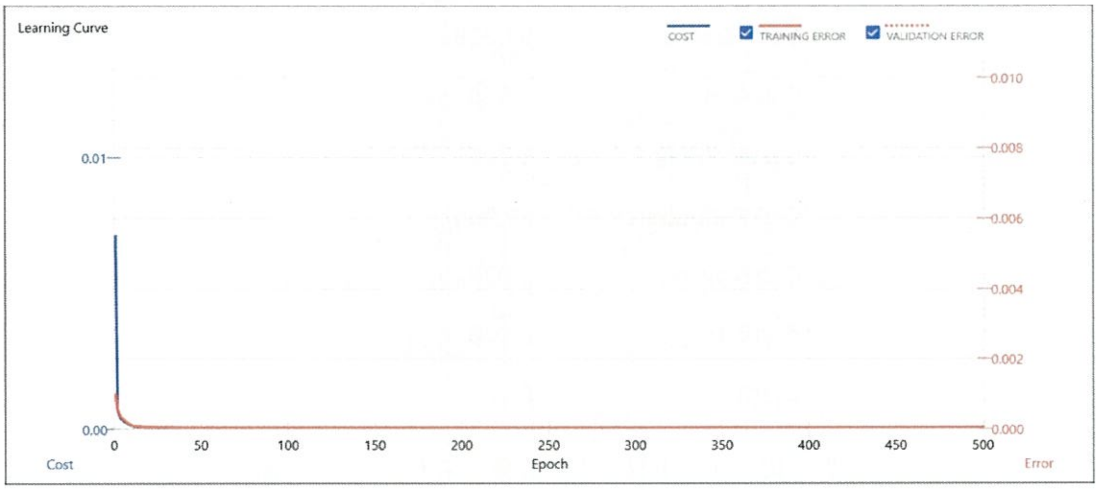


図 **5.31**　学習曲線 (気温の予測)

横軸は学習世代数，左縦軸は学習データにおけるコスト関数の値，右縦軸は学習データおよび評価データにおける誤差関数の値を示している．

右縦軸の学習データおよび評価データにおける誤差関数が世代数が増えるごとに減少していることから，正しく学習が進んでいることがわかる．

6 フロントエンドの紹介

本システムは，測定結果を利用者が直接閲覧できるのが特徴である．そこで，利用者がデータを閲覧する Web ページの部分の主要機能を紹介する．

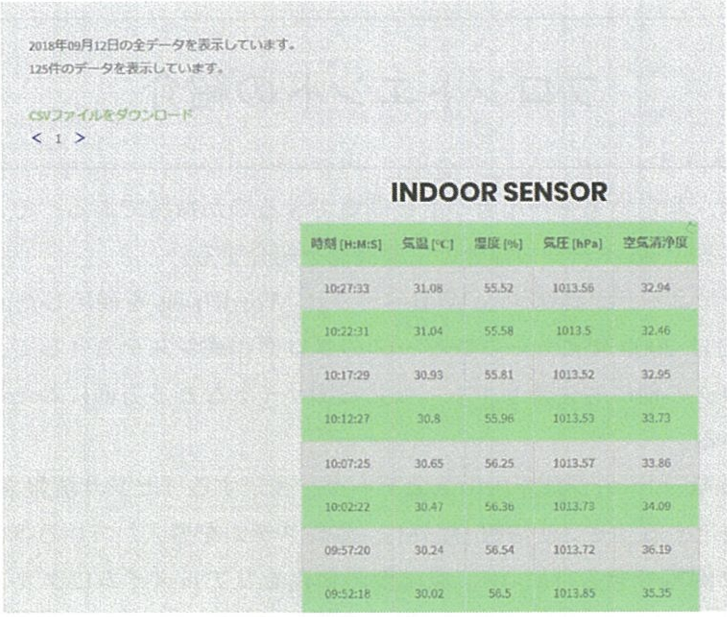
本システムではユーザが閲覧する Web ページに WordPress を使用した．

Wordpress とは 2003 年にオープンソースのブログ作成システムとして公開されたもので，現在はブログの他にも企業サイト，公共機関サイトなど多方面にわたるウェブサイトで利用されている．

本システムの Web ページは屋内の気象データを表示する「センサ情報 (屋内)」，屋外の気象データを表示する「センサ情報 (屋外)」，本システムを導入しているほかの地域の気象データを一度に閲覧できる「Map」，温湿度の変化をリアルタイムでグラフ化してくれる「グラフ」1 時間周期で 24 時間後までの温度変化の予測結果を表示する「気温予測」で構成している．

6.1 センサ情報 (屋内)

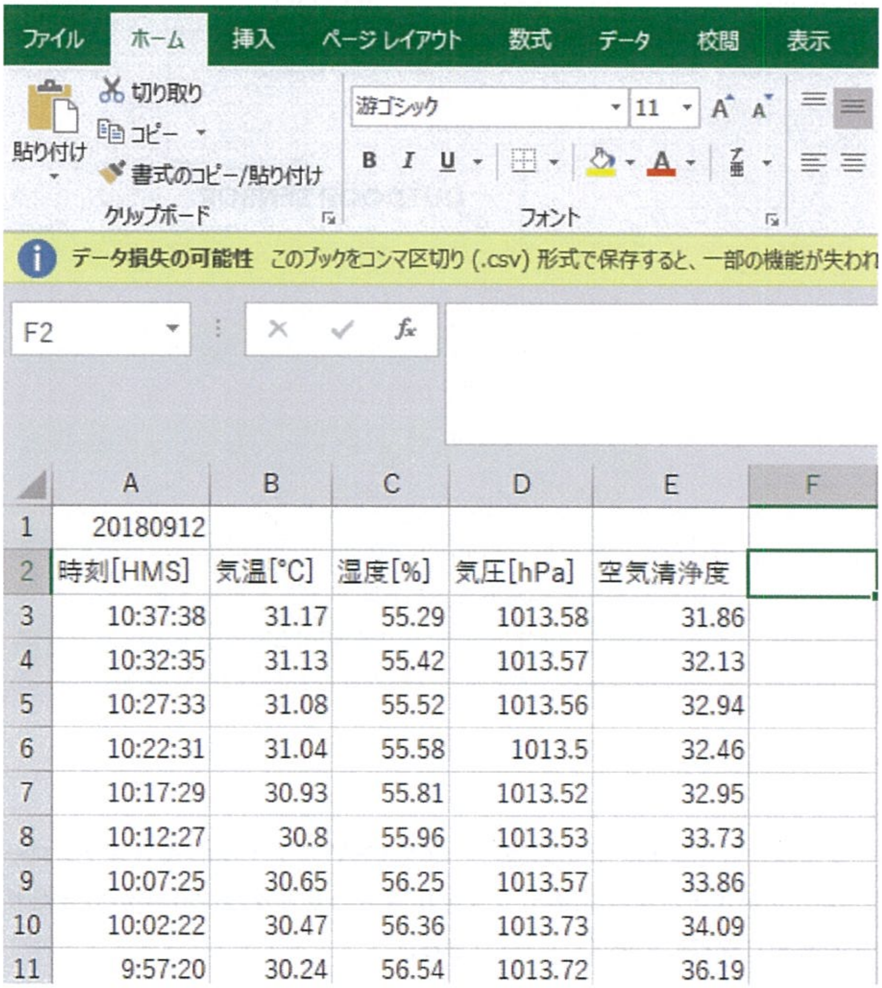
屋内の気象測定には **3** 章で示した BME680 を使用した．BME680 で測定した気象データは **3** 章で示したデータベースに格納され PHP により，表形式で Web ページに表示させた．また，測定した全データを表示させるのではなく，ある 1 日の測定データだけに限定して表示させることで Web ページの動作を軽くさせている．さらに，スクロールさせる手間を省かせるために JavaScript でページ送りの機能を実装した．



時刻 [H:M:S]	気温 [℃]	湿度 [%]	気圧 [hPa]	空気清浄度
10:27:33	31.08	55.52	1013.56	32.94
10:22:31	31.04	55.58	1013.5	32.46
10:17:29	30.93	55.81	1013.52	32.95
10:12:27	30.8	55.96	1013.53	33.73
10:07:25	30.65	56.25	1013.57	33.86
10:02:22	30.47	56.36	1013.73	34.09
09:57:20	30.24	56.54	1013.72	36.19
09:52:18	30.02	56.5	1013.85	35.35

図 6.1　センサ情報 (屋内) ページ

このページでは，表示されている気象データを CSV ファイル形式でダウンロードすることもでき，ユーザがさまざまな気象データを組み合わせてグラフ作成ができたり，履歴としてユーザの端末に保存しておくこともできる．

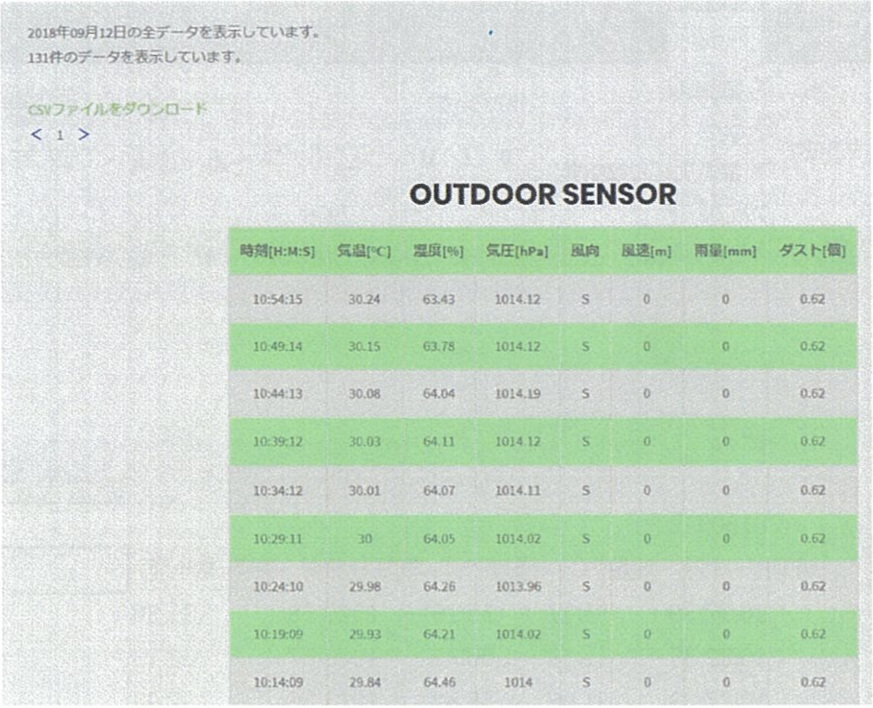


	A	B	C	D	E	F
1	20180912					
2	時刻[HMS]	気温[°C]	湿度[%]	気圧[hPa]	空気清浄度	
3	10:37:38	31.17	55.29	1013.58	31.86	
4	10:32:35	31.13	55.42	1013.57	32.13	
5	10:27:33	31.08	55.52	1013.56	32.94	
6	10:22:31	31.04	55.58	1013.5	32.46	
7	10:17:29	30.93	55.81	1013.52	32.95	
8	10:12:27	30.8	55.96	1013.53	33.73	
9	10:07:25	30.65	56.25	1013.57	33.86	
10	10:02:22	30.47	56.36	1013.73	34.09	
11	9:57:20	30.24	56.54	1013.72	36.19	

図 **6.2**　CSV ファイル出力結果

6.2　センサ情報 (屋外)

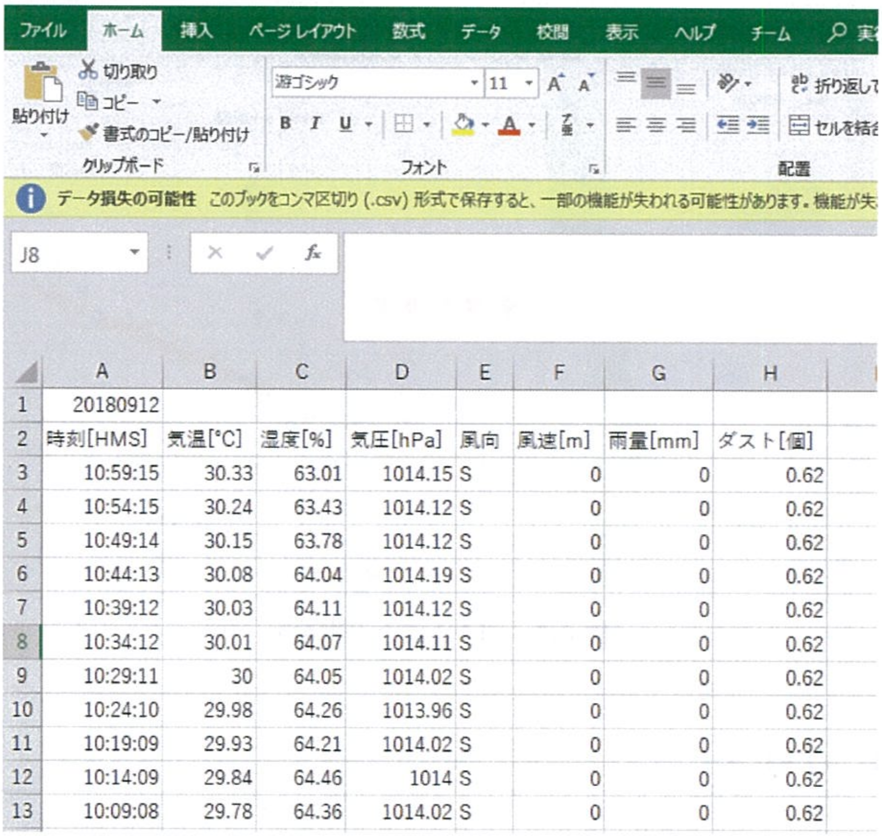
屋外の気象測定には **3** 章で示した Weathermeter，BME280，ダストセンサ (以下屋外センサ) を使用した．屋外センサで測定した気象データは **3** 章で示したデータベースに格納され PHP により，表形式で Web ページに表示させた．また，測定した全データを表示させるのではなく，ある 1 日の測定データだけに限定して表示させることで Web ページの動作を軽くさせている．さらに，スクロールさせる手間を省かせるために JavaScript でページ送りの機能を実装した．



時刻[H:M:S]	気温[℃]	湿度[%]	気圧[hPa]	風向	風速[m]	雨量[mm]	ダスト[個]
10:54:15	30.24	63.43	1014.12	S	0	0	0.62
10:49:14	30.15	63.78	1014.12	S	0	0	0.62
10:44:13	30.08	64.04	1014.19	S	0	0	0.62
10:39:12	30.03	64.11	1014.12	S	0	0	0.62
10:34:12	30.01	64.07	1014.11	S	0	0	0.62
10:29:11	30	64.05	1014.02	S	0	0	0.62
10:24:10	29.98	64.26	1013.96	S	0	0	0.62
10:19:09	29.93	64.21	1014.02	S	0	0	0.62
10:14:09	29.84	64.46	1014	S	0	0	0.62

図 **6.3**　センサ情報 (屋外) ページ

このページでは，表示されている気象データを CSV ファイル形式でダウンロードすることもでき，ユーザがさまざまな気象データを組み合わせてグラフ作成ができたり，履歴としてユーザの端末に保存しておくこともできる．



	A	B	C	D	E	F	G	H
1	20180912							
2	時刻[HMS]	気温[°C]	湿度[%]	気圧[hPa]	風向	風速[m]	雨量[mm]	ダスト[個]
3	10:59:15	30.33	63.01	1014.15	S	0	0	0.62
4	10:54:15	30.24	63.43	1014.12	S	0	0	0.62
5	10:49:14	30.15	63.78	1014.12	S	0	0	0.62
6	10:44:13	30.08	64.04	1014.19	S	0	0	0.62
7	10:39:12	30.03	64.11	1014.12	S	0	0	0.62
8	10:34:12	30.01	64.07	1014.11	S	0	0	0.62
9	10:29:11	30	64.05	1014.02	S	0	0	0.62
10	10:24:10	29.98	64.26	1013.96	S	0	0	0.62
11	10:19:09	29.93	64.21	1014.02	S	0	0	0.62
12	10:14:09	29.84	64.46	1014	S	0	0	0.62
13	10:09:08	29.78	64.36	1014.02	S	0	0	0.62

図 **6.4**　CSV ファイル出力結果

6.3 MAP

Map 機能は JavaScript で Google map 上でシステムを導入している場所にマーカーを配置して，そのマーカーをクリックすると Web ページへのリンクが表示されるようになっている．リンクをクリックするとシステムを導入している場所のセンサ情報の Web ページが表示され，ユーザがほかの地域の気象データと比較，検討することもできる．

システムを導入していない都道府県がある場合は，外部サイトより気象データを JSON 形式で取得し，マーカーをクリックするとリアルタイムの気象データを閲覧できるようになっている．

図 **6.5**　MAP 出力 (開発中)

6.4　グラフ

グラフ機能は JavaScript で，Google Charts ライブラリを使用して表示させている．グラフは屋内センサ及び屋外センサから取得した温湿度データを基に最新 30 件分の温湿度データをグラフ表示させている．グラフ左側の縦軸は温度ラベル，グラフ右側の縦軸は湿度ラベル，横軸は時間ラベルにしている．ユーザがよく使うであろう，温湿度のグラフ表示を Web ページ上で行うことで，CSV ファイルをダウンロードする手間を省くことができ，Web ページ上で温湿度の変化に対する，比較，検討を行うことができる．また，温湿度の変化に応じてグラフの縦軸も自動で変化するのでいつでも見やすく最良なグラフを見ることができる．

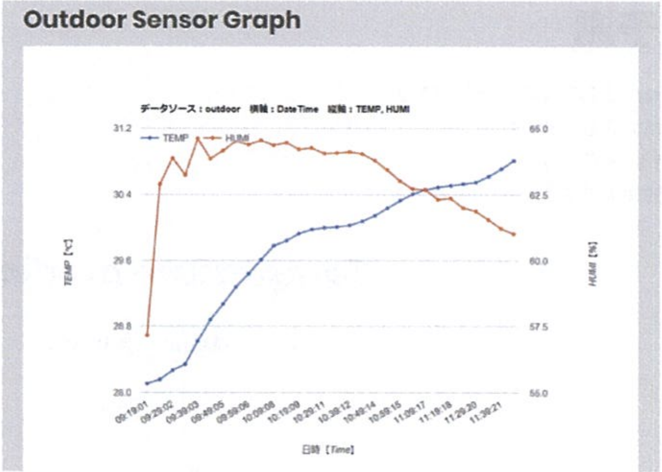


図 **6.6**　屋外センサグラフ

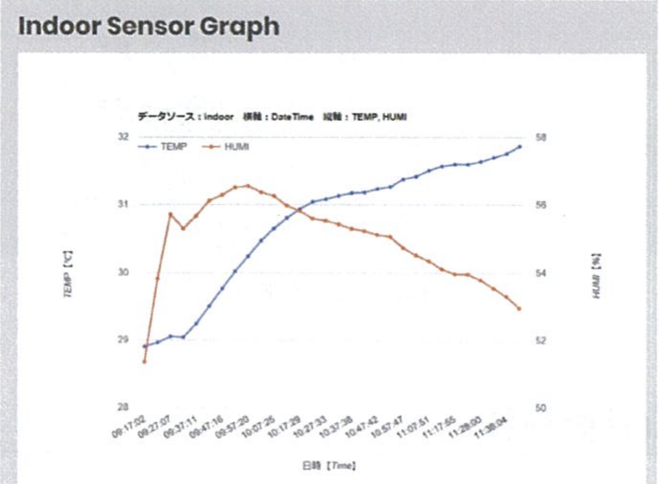


図 **6.7**　屋内センサグラフ

6.5　気温予測

気温予測機能は **5** 章で示した，NN(ニューラルネットワーク) により，気象庁よりインポートした過去 5 年分の気温データと現在の気温データを基にして 1 時間おきに 1 時間ごと 24 時間分の気温予測データを PHP で表形式で Web ページに表示させている．また，その予測に応じて NN がアドバイス文を自動で生成し，ユーザに「昼から気温が下がるので厚手のコートや雨に備え傘を持って出かけましょう」などの最適なアドバイスを行う．

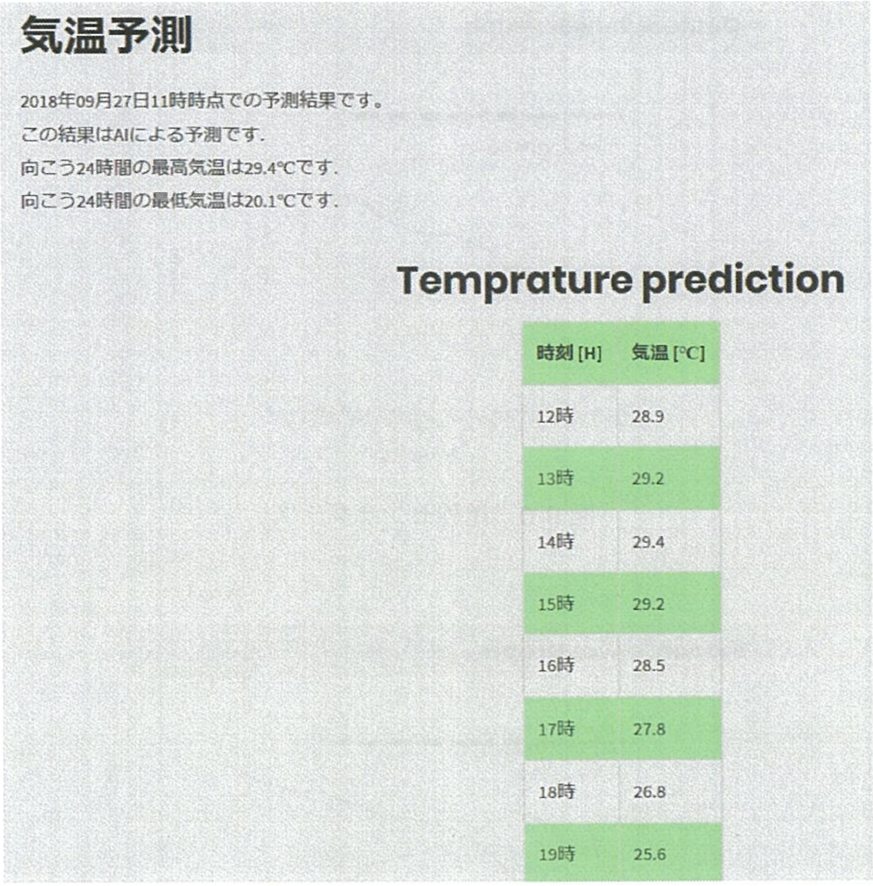


図 **6.8**　気温予測ページ

6.6　AR(拡張現実)

AR 機能では AR.js，aframe により 3D キャラクターを表示させ，audio.js と PHP により動的にアドバイス文を発話させている．また，発話させるだけではなく画面上にアドバイス文をテキストで表示させている．

使用方法としては，AR ページでカメラを起動させ，AR マーカーを読み取ることでキャラクターを表示させ，アドバイス文を発話させることができる．

図 6.9　AR ページ

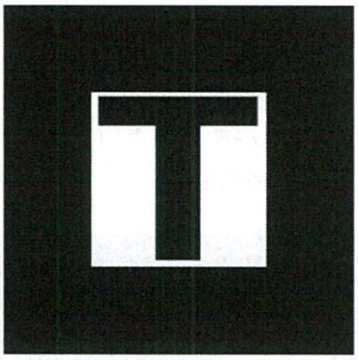


図 6.10　AR マーカー

7 実験方法と結果

7.1 実験方法

実際に作成した機器を利用して，8 月から 9 月にかけて運用を行った．ただし，開発と実験を交互に行い，機能の追加を図ったため，最終的なデータは，9 月 23 日以降に取得できている．

屋外用センサノードの運用を行った場所は，香川高等専門学校高松キャンパス　専攻科棟 5F　螺旋階段踊り場である．運用場所の環境を，表 **7.1** に示す．

表 **7.1**　実験場所

緯度	34.311
経度	134.010
高度	34 m

屋内用センサノードの運用を行った場所は，香川高等専門学校高松キャンパス　専攻科棟 5F　共同実験室及びゼミ・談話室である．

図 **7.1**　屋内用センサノードの外観図

図 7.2　屋外用センサノードの外観図

図 7.1 や図 7.2 に示しているのが，運用中のセンサノード及びゲートウェイである．

7.2 実験結果

本システムの仕様として，測定したデータは自動的に整形・保存されるため，直接的な測定値を閲覧する場合には，データベースにアクセスする必要がある．また，測定間隔も非常に短いので，データ量が多く，本レポートに生データを直接記述することもできない．(単純な情報の数を総計すると，1 週間の運用で数万件のデータになる)

6 章の冒頭でも述べたとおり，本システムは測定結果を利用者が直接閲覧できるのが特徴である．そのため，6 章で示したフロントエンド (システム全体のうち，利用者が直接閲覧・操作できる部分) が事実上の実験結果となる．

そこで，6 章のスクリーンショットを再掲し，実験結果の解説を行う．6 章では，システムの機能的な側面を紹介したが，本節では測定項目や精度・有用性に重みを置いた解説とする．

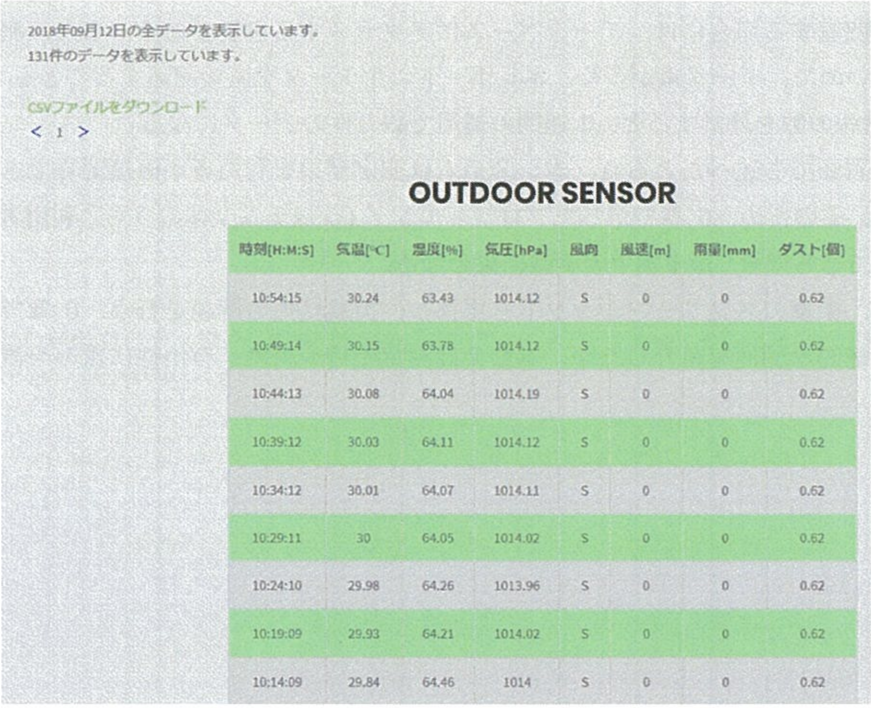
2018年09月12日の全データを表示しています。
125件のデータを表示しています。

CSVファイルをダウンロード
< 1 >

INDOOR SENSOR

時刻[H:M:S]	気温[℃]	湿度[%]	気圧[hPa]	空気清浄度
10:27:33	31.08	55.52	1013.56	32.94
10:22:31	31.04	55.58	1013.5	32.46
10:17:29	30.93	55.81	1013.52	32.95
10:12:27	30.8	55.96	1013.53	33.73
10:07:25	30.65	56.25	1013.57	33.86
10:02:22	30.47	56.36	1013.73	34.09
09:57:20	30.24	56.54	1013.72	36.19
09:52:18	30.02	56.5	1013.85	35.35

図 **7.3**　センサ情報 (屋内) ページ (再掲)


2018年09月12日の全データを表示しています。
131件のデータを表示しています。

CSVファイルをダウンロード
< 1 >

OUTDOOR SENSOR

時刻[H:M:S]	気温[℃]	湿度[%]	気圧[hPa]	風向	風速[m]	雨量[mm]	ダスト[個]
10:54:15	30.24	63.43	1014.12	S	0	0	0.62
10:49:14	30.15	63.78	1014.12	S	0	0	0.62
10:44:13	30.08	64.04	1014.19	S	0	0	0.62
10:39:12	30.03	64.11	1014.12	S	0	0	0.62
10:34:12	30.01	64.07	1014.11	S	0	0	0.62
10:29:11	30	64.05	1014.02	S	0	0	0.62
10:24:10	29.98	64.26	1013.96	S	0	0	0.62
10:19:09	29.93	64.21	1014.02	S	0	0	0.62
10:14:09	29.84	64.46	1014	S	0	0	0.62

図 **7.4**　センサ情報 (屋外) ページ (再掲)

図 **7.3** と図 **7.4** に示したのが，屋内・屋外のセンサ情報である．本システムで最も測定値を直接的な表現で掲示している部分である．

屋内用センサノードでは気圧・湿度・気温・空気清浄度・熱中症温度 (WBGT) を，屋外用センサノードでは気圧・湿度・気温・風向・風速・雨量・PM2.5・体感温度・熱中症温度 (WBGT) を閲覧できる．これらのデータに付いては日本で汎用的に用いられる単位への変換等は行っているが，測定値を直接掲載している．よって，これらのデータの精度は，センサの精度にのみ依存する．

また，WBGT・体感温度・空気清浄度は計算値であるが，WBGT については気温と湿度からのみ算出しているので，真値と比べて若干の誤差がある (本来，WBGT の算出には黒球温度も必要である)．

表による表示は，最も情報量が多い一方で，直感的にはそのデータの意味を解釈し難いという問題がある．気温や湿度については比較的容易にその意味を解釈できるが，「熱中症温度が○○度である」というデータは大半の利用者にとっては意味を理解し難い．このデータの意味を理解するためには，熱中症温度の知識と，熱中症の危険がある熱中症温度の目安を事前知識として有しておく必要があるからである．

そのため，気温や湿度以外のデータを直接的に表現しているのは，表のみである．表による表示は，システムや気象の知識に慣れている高度利用者向けとして開発した．

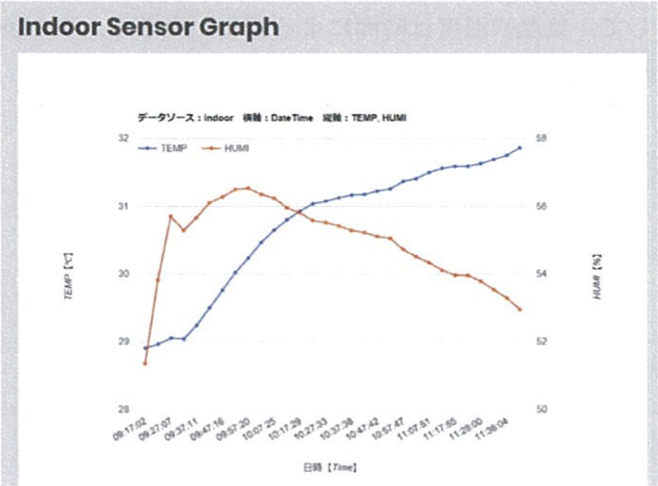


図 **7.5**　屋内センサグラフ (再掲)

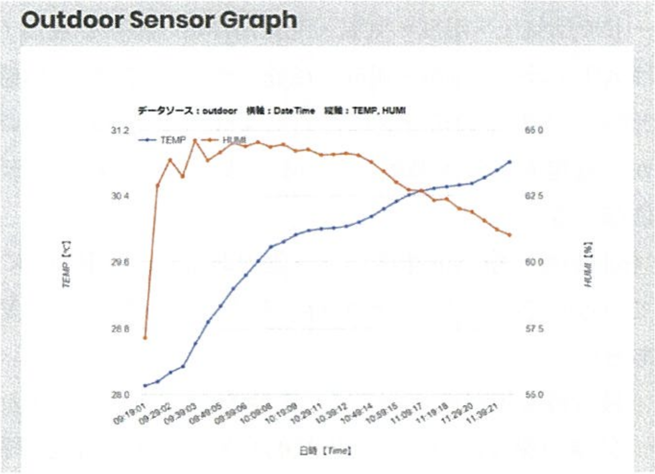


図 **7.6**　屋外センサグラフ (再掲)

図 **7.5** と図 **7.6** に示したのが，センサ情報を元にプロットしたグラフである．グラフは，測定値をそのままプロットしているので，グラフの精度についても，センサの精度にのみ依存する．グラフでプロットしているのは，気温と湿度の 2 項目である．気温と湿度を採用した理由として，気温や湿度は時間による変化が大きく，また多くの人が一般的に慣れ親しんでいるため，理解しやすい事が挙げられる．

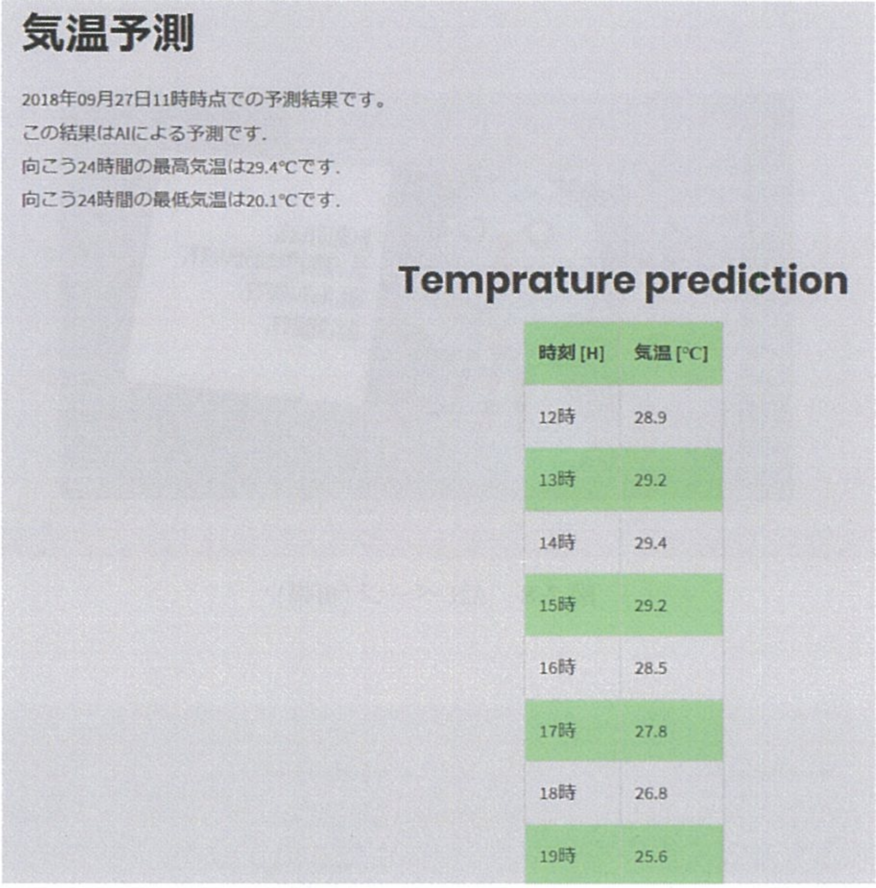


図 **7.7**　気温予測ページ (再掲)

図 **7.7** に示したのが，気温の予測である．気温の予測についても，数値を直接掲載しているが，これについても，気温は時間による変化が大きく，また多くの人が一般的に慣れ親しんでいるため，理解しやすいためである．

図 7.8　AR ページ (再掲)

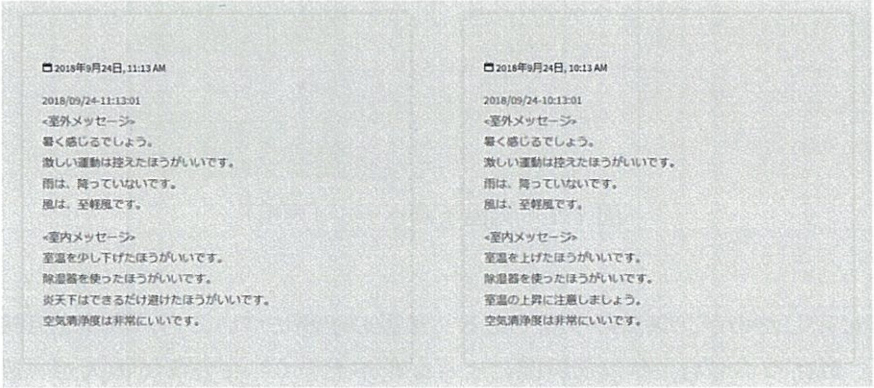


図 7.9　Web サイト上の記事

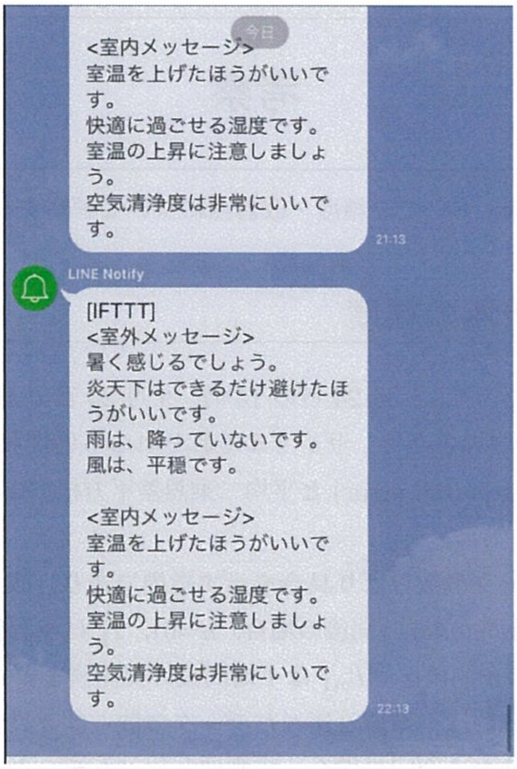


図 **7.10**　LINE での投稿

図 **7.8**・図 **7.9**・図 **7.10** に示したのが，アドバイス文章である．アドバイス文章は，**5** 章でも述べたように，熱中症温度・湿度・降雨など，その変化が生活に直接影響するものを元にして生成されている．

これらのアドバイス文章は，AR コンテンツ・Wordpress 上での記事・LINE への投稿によって確認できる．LINE は，予め登録しておいたユーザに配信される．AR コンテンツは，文章画像のほか，機械音声によってアドバイス文章を読み上げる機能も搭載している．

8 考察

本章では，気温の予測においての精度・改善点について考察する．

8.1 気温の予測の精度

精度を確かめるにあたって，9 月 24 日の 15 時，9 月 25 日の 10 時と 22 時，9 月 26 日の 15 時の予測値をサンプルとした．サンプルそれぞれの 6 時間後，12 時間後，24 時間後の平均二乗誤差 (Mean squared error) と平均二乗誤差平方根 (Root mean squared error) を算出した．

平均二乗誤差は，データの散らばり具合を示す指標であり，式 (8.1) で表すことができる．この時，n は経過時間 (ex.6 時間後の場合，n=6)，T_p は予測値，T_m は実測値を表す．また，T_{p1} は 1 時間後の予測値を，T_{m1} は 1 時間後の実測値を表す．

分散と似ているが，分散は平均値と比べたデータの散らばり具合を表すものであり，平均二乗誤差は平均値ではなく今回の場合，実測値と比べたデータの散らばり具合を表すものである．

$$MSE = \frac{1}{n}\sum_{k=1}^{n}(T_{pk} - T_{mk})^2 \tag{8.1}$$

平均二乗誤差平方根は，平均二乗誤差を平方根をとったものであるため，式 (8.2) で表すことができる．平均二乗誤差は，単位の次元が元の二乗となるため，平方根をとることで単位をあわせることができる．

$$RMSE = \sqrt{MSE} \tag{8.2}$$

これらの値は，小さければ小さいほど精度が良いといえる．

以下に，サンプルそれぞれの予測値と実測値のグラフを図に，平均二乗誤差と平均二乗誤差平方根を表に示す．また，グラフの縦軸は気温，横軸は経過時間 n(式 (8.1) の n) である．

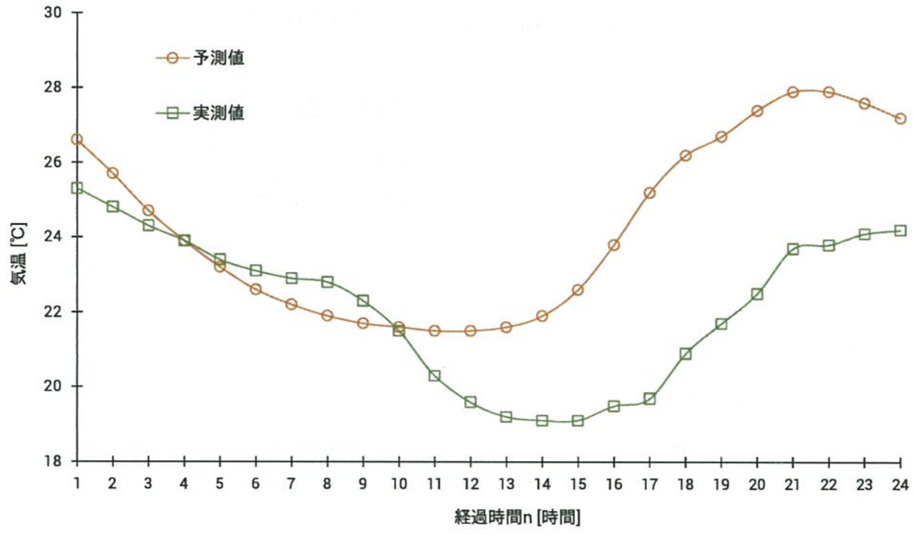


図 8.1　予測値と実測値 (24 日 15 時)

表 8.1　MSE と RMSE(24 日 15 時)

	6 時間	12 時間	24 時間
MSE	0.492	0.806	9.04
RMSE	0.701	0.898	3.00

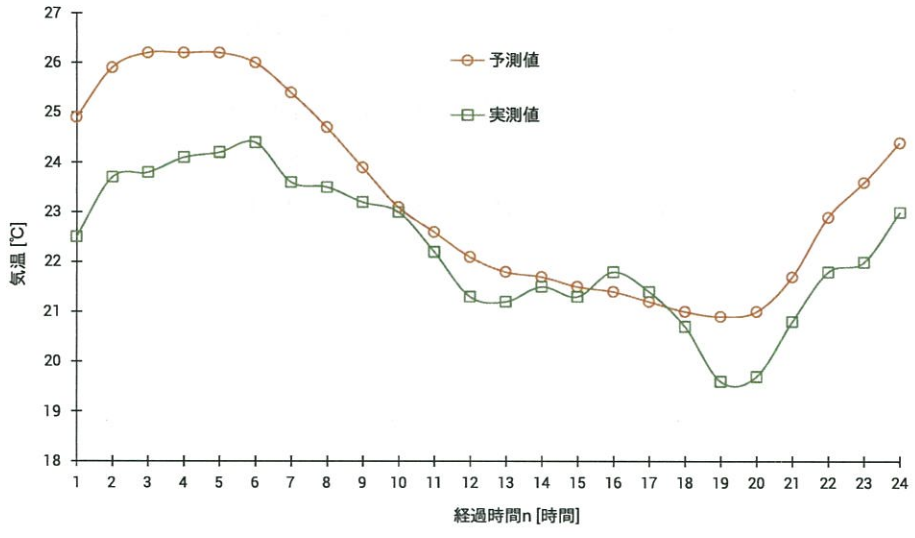


図 8.2　予測値と実測値 (25 日 10 時)

表 **8.2**　MSE と RMSE(25 日 10 時)

	6 時間	12 時間	24 時間
MSE	4.56	2.78	1.83
RMSE	2.13	1.67	1.35

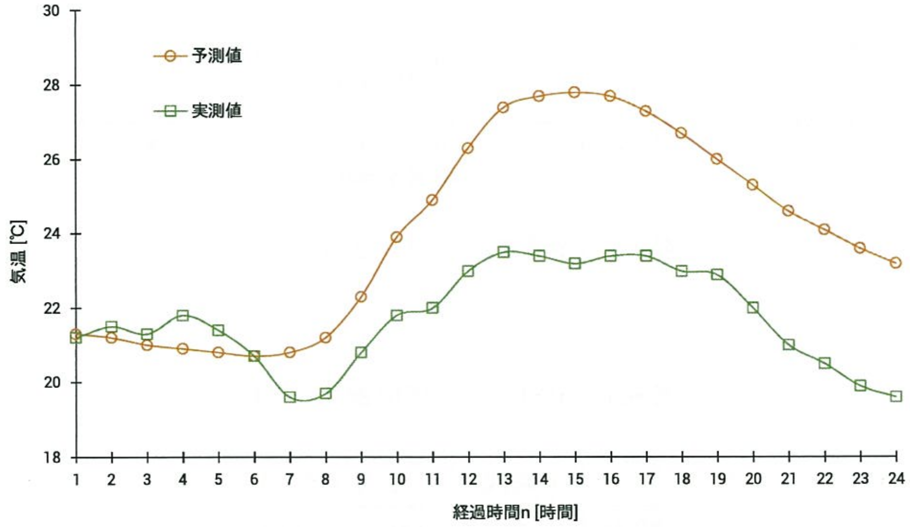


図 **8.3**　予測値と実測値 (25 日 22 時)

表 **8.3**　MSE と RMSE(25 日 22 時)

	6 時間	12 時間	24 時間
MSE	0.227	2.58	8.60
RMSE	0.476	1.67	2.93

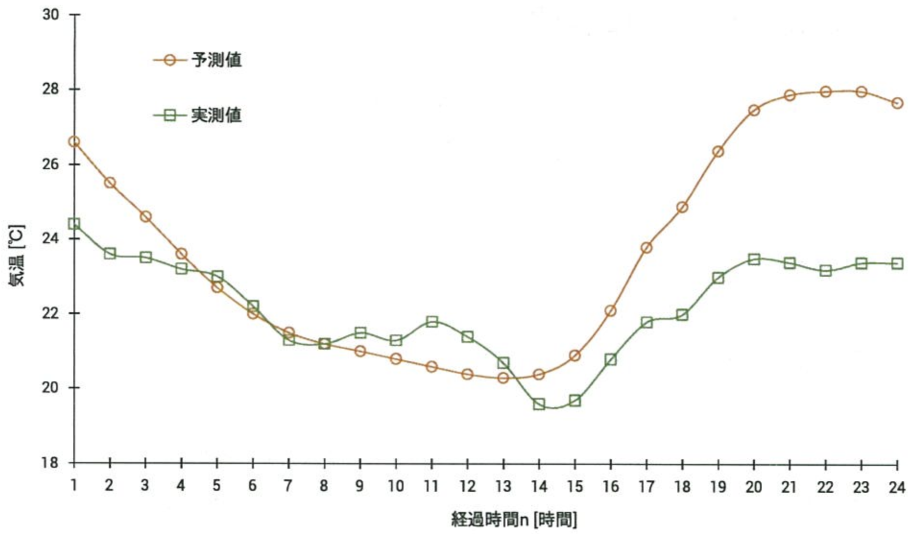


図 8.4　予測値と実測値 (26 日 15 時)

表 8.4　MSE と RMSE(26 日 15 時)

	6 時間	12 時間	24 時間
MSE	1.66	1.08	5.82
RMSE	1.29	1.04	2.41

これらの結果より，以下のようなことがわかる．

- 予測値は実測値より序盤は小さく，終盤は大きい値を出していることが多い．

例えば，24 日の 15 時のデータでは経過時間が 4 時間から 9 時間の間は予測値は実測値を下回り，経過時間が 10 時間以降は大きく上回っている．25 日の 10 時のデータが例外にも思えるが，このデータは最初から実測値よりも，2 ℃以上大きい予測値を出していたため，序盤に予測値が実測値を下回らなかったと考えられる．

- 時間が経てば経つほど，誤差は大きくなっていく．

表を見たらわかる通り，RMSE は経過時間が長くなるにつれて大きくなっていることから，時間が経てば経つほど，誤差は大きくなっていることがわかる．25 日の 10 時の予測データは例外となってしまうがそれ以外には当てはまる．

8.2 気温の予測の改善案

誤差が大きくなった理由やこうすればより精度は向上するなど
気温の予測の改善案として次のことが挙げられる.

- データセットの形式の見直し
- ネットワークモデルの見直し

8.2.1 データセットの形式の見直し

データセットの形式について考察する.

今回, 学習に用いたデータセットは, 入力データが過去 24 時間分の日時・気温・湿度であり, この入力に対する正解データが 1 時間後の気温・湿度である. このようなデータセットでは気温の予測を行うには十分ではなかったと考えられる.

気温の変化を支配する主な要因として日射量があげられる. 日射量は直達日射量・散乱日射量・全天日射量の主に 3 種類に分類されるが, ここでは特に区別はしない.

また, 気圧も気温の変化に関係している. 例えば, 低気圧なら天気が悪いと考えられる. 天気が悪いと雲量が多くなり日射量が少なくなってしまう. このように気圧が気温に与える影響は大きい.

この 2 つのデータ (日射量・気圧) をデータセットに組み込むことで, 気温の予測精度が向上すると考えられる.

以上のようにデータセット形式の改善案を挙げた. しかし, これを実践するには少し問題がある. それは, 気圧・日射量を測定するセンサを新たに導入する必要がある点である.

そして, 気温の変化は様々な要因が複雑に絡み合っているため, どのようなデータセット形式が適切であるかの検証も必要である.

8.2.2 ネットワークモデルの見直し

ネットワークモデルについて考察する.

今回, 深層学習で採用したネットワークモデルは LSTM である. この LSTM は, 時系列データを処理するのが得意で未来のデータを予測する際に利用される.

ネットワークモデルとして LSTM を過信しすぎてしまい, 学習の精度に大きな影響を与えるハイパーパラメータ (ノード数や活性化関数等) の決定を十分に検討することができなかった.

ハイパーパラメータを決定する方法としては, ランダムサーチやベイズ最適化などが挙げられる. このうち, ランダムサーチの機能は, 深層学習を行う際に使用したソフトウェアである NeuralNetworkConsole(NNC) に組み込まれている. NNC のランダムサーチはハイパーパラメータの探索を行うだけでなく, ネットワークモデル自体の探索も行ってく

れる．

このような機能を利用することで，適切なハイパーパラメータの設定やより優れたネットワークモデルを採用し，学習することで気温の予測精度が向上すると考えられる．また，今回は学習に充てられる時間の都合もあり学習回数は 1000 回としたが，さらに増やすことで精度が向上すると考えられる．

しかし，むやみに学習回数を増やしたり，ランダムサーチにより得たハイパーパラメータを過信しすぎると過学習に陥ってしまう可能性もある．それ防ぐために，学習させたモデルの精度を確かめる K-fold Cross-Validation といった手法で検証を行う必要があるだろう．

9 評価

本章では作成したシステムの，ミッションの達成度・作品の有用性・今後の展望について述べる．

9.1 ミッションの達成度と作品の有用性

2.2 節で述べたミッションの達成度について考察する．

■ 自宅内・自宅周辺の気象データの測定・取得について

このミッションは，すべて達成することができた．測定に成功したデータは，屋外が温湿度・気圧・風向・風速・雨量・PM2.5 濃度，屋内が温湿度・気圧・空気室である．

屋内用のデータについてはその正確性を確認することができないが，屋外用のセンサの測定した値と気象庁が公開しているデータを比較し，高精度なデータであることが確認できた．測定場所が異なるので，特に風向や風速は気象庁のデータとは異なったが，変化の間隔等については概ね一致していた．

また，熱中症温度や体感温度の算出にも成功した．これらの値は，近年問題になっている熱中症に直接関わる指数であり，熱中症対策 (屋外での運動の可否や空調機を動作させるタイミング等) の参考値として極めて有用であると考える．

■ 大気汚染物質・空気室の測定について

上記のミッション同様，大気汚染物質や空気室についても測定することに成功した．前年度までのシステムでは，大気汚染物質を測定するためのダストセンサを動作させることはできたが，その結果から大気汚染物質を算出することはできなかったが，その算出にも成功した．

空気室の測定については，初めての測定であったが，空気室の値と人体への影響の関連性まで調査することに成功した．

■ 深層学習を用いたデータ解析について

深層学習を用いて予想される気温データは，24 時間の平均誤差が 1 [°C] 程度，6 時間・12 時間の平均誤差が 0.5 [°C] 程度と，十分実用的な域に達していると考えられる．

これらの予想結果はシステムにより測定された値から導出されるため，センサノードを設置している場所ごとの予想値を得られることができる．

また，天気情報を元に生成されるアドバイス文も，熱中症温度などを絡めたアドバイス文となっており，ユーザにとって有用なものであると思う．

■ データの閲覧について

測定されたデータは，すべてホームページで閲覧できるようにした．過去の作品では測定されたデータはただ SD カード上に保存されるだけであったり，表示を行う場合も，JSON 形式での表示や CSV 形式での表示であったりと，ユーザにとって閲覧し難い方法であった．

JSON 形式や CSV は，一見するとただの文字列であり，またその数値から考察を行うことも難しい．そこでm Excel 等のソフトウェアを用いてグラフ化等の作業を手動で行う必要があった．

しかし本システムでは，ホームページを利用して表やグラフ上でデータを提供し，ユーザにとって使いやすく理解しやすいシステムとした．また，AR 機能を利用してキャラクターが天気案内をするなど，ユーザが楽しめるシステムを構成することができた．

一方で，データはすべてデータベースに保存したり，CSV でのダウンロード機能を搭載することで，上級者にも利用しやすいシステムにした．

9.2 今後の展望

本システムは，発案時に掲げた機能はすべて搭載することができ，その実用性が確認された．このシステムでは人工知能を利用し，局所的な天気予報を行うことに重きをおいている．

したがって，今後このシステムを多くの場所で導入することによって，広い範囲の天気予報を実施することや，天気予報の精度向上につながると考えられる．

また今回は，気象観測機器コンテストの大会日程上，測定されたデータから熱中症温度や体感温度等の夏に重要視される数値の算出・考察を多くおこなっていた．今後は空気が乾燥し，インフルエンザ等の感染症が流行する時期になるので，インフルエンザ等への対策アドバイス文の生成を行うなど，季節にあったシステムに更新していくなどの改良も考えられる．

このシステムのユーザが増えれば，気温予想やアドバイス文生成のための情報がより多く入手できることになり，より精度の向上が期待できる．その一方で，他のユーザのための気温予想やアドバイス文生成に自分が測定したデータが利用されることもあり，結果として利用者全体の間でシステムの精度向上の好循環が生まれることになると考える．

10 感想

■

これまで私達が制作してきたシステムは「データを測定するためのシステム」でした．つまり，気象観測機器を制作することには拘っても，そのデータをどう解析・考察し，どのように提供するのかということは二の次となっていました．

しかし，機器を作成し，データを収集することだけで満足していては，本当の意味での気象観測機器ではないのではないかと考えました．ユーザに使ってもらえる機器とは，正確な値が測定できるだけでなく，それらのデータをわかりやすく提供できるものだと考えたのです．

そこで今回のシステムでは，メンバー全員が「わかりやすいデータの提供」ということに重みをおいて開発を行いました．例えば，単に気温などの単純な情報ではなく，熱中症温度などの計算値を利用し，その値を直接提供するだけでなく，アドバイス文という形で提供しました．また，AR 機能などを利用して，子供にもわかりやすく，楽しめるような形で提供しました．

もちろん，これまでに培ってきたデータの測定技術やデータ処理技術についても，いくつか進歩がありました．Raspberry Pi を用いて直接 BME680 センサを制御し，コストダウンを行なったのもその 1 つです．

このように，今回の作品は，「データの収集」から「データの表示」までが一元的に行えるシステムになったのではないかと思います．

今回の気象観測機器コンテストは，5 回目の参加です．まさか高専という学生生活の中で，気象観測機器コンテスト皆勤賞となるとは思っていもいませんでした．それだけに，本コンテストは私に多くの知見と経験をもたらしてくれました．初めて出場した第 3 回の時と比べて，少しずつではありますが，作品の質を向上させることができたのではないかと思います．

2 次審査を通過した暁には，この作品をより楽しく便利に利用していただくために，これまでとは一風変わったプレゼンを予定しています．ぜひご期待ください．

電気情報工学科 5 年 大西哲

■

私は今回で気象観測機器コンテストへの出場が 3 回目となります．今回製作したシステムは今まで開発してきた中で 1 番の規模を誇るものとなりました．これほどのシステムを完成させることができ正直驚いています．

私は Web サーバと AI サーバの製作を担当しました．この 2 つの機能は，気象データを

有効利用するのに大きな役割を果たしていると考えます．

これまでは，所得したデータはcsvファイルに出力するかコンソールに表示するだけの味気ないものでありましたが，Webサーバでは表やグラフ形式で表示することによってデータの確認を容易なものにすることができました．

また，人工知能をAIサーバに実装することができました．出来には満足してはいませんが，近年注目されている技術を導入することができたという意味ではこれは大きな成果といえるでしょう．

私は5年生ということもあり，今回が最後の気象観測コンテストで残念です．しかし，最後のコンテストにふさわしいシステムを製作することができ，とても満足しています．

電気情報工学科5年 笠松雅史

■

私はセンサノードの製作を担当しました．私は今回の気象観測コンテストが人生で初めて参加するコンテストであり，右も左も分からない状況でした．しかし，何とかメンバーの力を借りて屋外の気象データを測定するセンサノードの製作に成功しました．

私が今回の作品製作過程で最も重要であると感じたことは「ほう・れん・そう」です．

私は今までグループで1つの物を協力して作るという経験がありませんでした．しかし，他のメンバーは今までコンテストを何度も経験している猛者ばかり．最初は分からないことが恥ずかしいと思い，問題を自分の中で抱え込んでばかりしていました．しかし，そんなことを気にしていても作業を遅らせるだけで何も有益なことは生まれません．私はそのことに気が付いた時にくだらないプライドは捨てました．

そして私からメンバーに分からないことを聞くとメンバーの皆はとても優しく，丁寧に様々なことを教えてくれました．そのおかげで新しい知識が身に付き，他のメンバーとの仲も以前より深まりました．

思い返してみても，とてもためになる半年間でした．私は来春から社会人になります．今回のコンテストで学んだことを社会人生活でも生かしていけるように，残り半年の学生生活を過ごしていきたいと思います．

電気情報工学科5年 片山大貴

■

私は，1年生から4年生まで成層圏気球による気象観測を行ってきました．今年は，成層圏気球による気象観測を後輩に託し新たなシステムの開発を行いました．それが今回の作品である「P-system」です．今回初めて，Webページの制作に挑戦しました．ユーザーが見やすく，扱いやすいページにするのに大変苦労しました．以前の気象観測機器コンテストの作品では，外部のWebサービスを使って，データをアップロードしていましたが，今回は1からサーバーを構築してWebページで気象データを見たり，加工したりできるよう

にしました.

私は，今回初めて PHP と Javascript というものを知りました．この短期間で少しだけですが使えるようになったのはまあまあの成長ではないのかなと思います.

今回の気象観測機器コンテストは，5 回目の参加となります．昨年の感想で最後になりますと記述しておきながら，また出場してしまいました．本コンテストで，プレゼンテーションとは何なのか，プレゼンテーションの要領などを学ぶことができました.

最後に，5 年間気象観測機器コンテストに出場してきて，3 度優秀賞を頂きました．しかし，未だトップを取ったことがありません．5 年間の集大成となる今回のコンテストで，あの大きいトロフィーが頂けるように，審査員，会場の目を引くプレゼンテーションをしたいと思います.

電気情報工学科 5 年 新谷悠真

■

私にとって，今回の気象観測機器コンテストは，第 4 回以来の参加となります．第 4 回の気象観測機器コンテストは技術や知識が未熟で，あまりチームに貢献することができなかったため，悔やしさの残るものでした.

しかし，「P-System」はメンバー全員が力を合わせることで，完成したシステムです．チームに貢献することできたこと，何より楽しく製作が行えたことを非常に嬉しく感じます.

また，製作を通して私自身の成長も感じます．私は，Gateway と AI サーバ部分の製作を担当しました．今まで使ったことのない機器やプログラミング言語を使用したり，数多くのトラブルシューティングを行ったりと，製作に携ったからこそ得られる経験や知識を身につけることができたと思います.

このように自分の成長にも繋がった機会をいただけたこと感謝しています．「P-System」は，私たちの高専生活の集大成となる作品であり，今まで作製してきた作品の中でも一番の規模と一番の質を誇るものだと自信を持って言えます．是非とも，コンテストの最終選考の場で，ポスターセッションやプレゼンテーションを行えればと思います.

電気情報工学科 5 年 吉川悠斗

11 参考資料

■ センサノードの参考資料

(1) 乾球温度・湿球温度・湿度の関係

https://www.jstage.jst.go.jp/article/sasj1971/9/2/9_2_39/_pdf

(2) BME680 センサ　参考ライブラリ・データシート等

https://www.switch-science.com/catalog/3570/

https://github.com/pimoroni/bme680-python

https://github.com/G6EJD/BME680-Example

https://ae-bst.resource.bosch.com/media/_tech/media/datasheets/BST-BME680-DS001-00.pdf

(3) WBGT の計算

http://www.chrononhotonthologos.com/misc/hygromtr.htm

(4) Arduino で XBee API 通信を行う方法

http://www.geocities.jp/zattouka/GarageHouse/micon/XBee/XBee4.htm

https://qiita.com/h-sakano/items/15e1b269d3ceb03cf499

(5) XBee ZB S2C のデータシート

https://www.digi.com/resources/documentation/digidocs/pdfs/90002002.pdf

■ ゲートウェイの参考資料

(1) Raspberry Pi で SFTP を利用する

https://librabuch.jp/blog/2014/03/python3_paramiko_sftp_scp/

(2) Python におけるデータ入出力（CSV，JSON）

https://qiita.com/Amtkxa/items/1bc945bf434ae910a85f

(3) WordPress に記事を投稿する

https://python-wordpress-xmlrpc.readthedocs.io/en/latest/

https://k99-tech.com/blog/archives/941

(4) USB のデバイス名を固定する

http://www.raspberrypirulo.net/entry/2017/05/04/USB%E3%81%AE%E3%83%87%E3%83%90%E3%82%A4%E3%82%B9%E5%90%8D%E3%82%92%E5%9B%BA%E5%AE%9A%E3%81%99%E3%82%8B%E6%96%B9%E6%B3%95

■ AI サーバの参考資料

(1) nnabla の導入
https://github.com/sony/nnabla

(2) nnabla の使い方
https://nnabla.readthedocs.io/en/latest/index.html
http://arakan-pgm-ai.hatenablog.com/archive/category/NNabla

(3) NeuralNetworkConsole の使い方
https://dl.sony.com/ja/
http://arakan-pgm-ai.hatenablog.com/archive/category/
%E3%83%8B%E3%83%A5%E3%83%BC%E3%83%A9%E3%83%AB
%E3%83%8D%E3%83%83%E3%83%88%E3%82%B3%E3%83%B3
%E3%82%BD%E3%83%BC%E3%83%AB

(4) Raspberry Pi から LINE へプッシュ通知を行う
https://mimimopu.com/raspberry-pi_line/

(5) プログラムの定時実行
http://ksknet.net/linux/post_2.html

■ Web サーバの参考資料

(1) PHP で JSON データを処理する方法
https://qiita.com/fantm21/items/603cbabf2e78cb08133e

(2) WordPress における PHP ショートコードの実行方法
http://fx.yoshiron.com/php-file-include-wordpress-posts/

(3) 気象データのテーブル表示方法
http://konayuki1026.blogspot.com/2011/09/php_23.html

(4) 折れ線グラフの作成方法
https://www.petitmonte.com/javascript/chart_linechart.html
https://developers.google.com/chart/interactive/docs/reference#DataTable
https://qiita.com/aion/items/8c5b536ce64e89d0af28
http://hbnist76.blog.fc2.com/blog-entry-401.html
https://www.xmisao.com/2014/01/29/google-charts-two-yaxis-line-chart.html

(5) Web ページで音声データを再生する方法
https://github.com/kolber/audiojs
http://kolber.github.io/audiojs/demos/test6.html

(6) Web ページで音声データを再生する方法
https://github.com/kolber/audiojs

(7) CSV ファイル形式でダウンロード
https://www.sejuku.net/blog/25648

http://codaholic.org/?p=1800

http://php.net/manual/ja/function.fputcsv.php

https://blog.supersonico.info/?p=869

https://php.programmer-reference.com/php-mysql-csv-download/

http://wp-school.info/plugin-downloadmanager

(8) 地図を表示する方法

http://www.nanchatte.com/map/showDifferentInfoWindowOnEachMarker.html

http://ponk.jp/jquery/basic/pagination

https://lab.syncer.jp/Web/API/Google_Maps/JavaScript/