

# 論理センサクラウド

## - クラウドとIoTの統合システム管理 -

串田 高幸<sup>1</sup> 永島 寛子<sup>2</sup> 加藤 由花<sup>3</sup>

概要：クラウドとIoTは、実用システムとして多くのところで使われている。しかし、IoTやクラウドの研究分野において、クラウドとIoTの統合システム管理の研究は、新規技術のイノベーションが少なく、まだ十分に行われていないことがわかっている。そのため、IoTの基幹となるセンサノードでのデバイス状況やデータの質をクラウドと同じレベルで管理して、プロダクション利用することが課題である。この論文では、クラウドとIoTの統合システム管理のための新プラットフォームである「論理センサクラウド」を提案する。論理センサクラウドでは、論理センサが、物理センサと同等の機能をもつだけでなく、高可用性とセンサデータの質を担保するためのセンサデータの補完を行う。また、統合システム管理では、センサノードから論理センサまでの統合的なライフサイクル管理と監視を行うことで、実時間データを使ったIoTアプリケーションをプロダクションで利用できるようにする。さらに、この論文では、論理センサクラウドの評価を行って、約100ms程度の転送処理遅延でIoTアプリケーションがセンサデータを利用できることを検証した。

### 1. はじめに

現在、クラウドとIoTは、実用システムとして多くのところで使われている。クラウドの本質的な特徴として、オンデマンドのセルフサービスや計測可能なサービスが上げられている。また、クラウドには、計算資源のリソースプールがある。このときのクラウドのリソースとは、コンピュータでは一般的なCPU、メモリ、ストレージ、ネットワークである。

一方、IoTでは、他の装置と通信可能なセンサノードが、相互接続可能なIoTデバイスとして提供されている。IoTデバイスが複数あって、それらが相互接続されているグループをIoTということが出来る[1]。また、IoT Cloudプラットフォームに関しては、数多くの関連研究が報告されている[2]。しかし、これらの研究においてクラウドとIoTの統合システム管理に関する研究は、まだ十分におこなわれていないのが現状である。また、この研究分野は、新規イノベーションが少ないこともわかっている。

ここでは、例として、IoTデバイス（センサノード）の温度センサから温度データを収集して、解析して表示するアプリケーションを使ったソリューションを説明する。物理量を測定するセンサノードでは、物理センサである温度セ

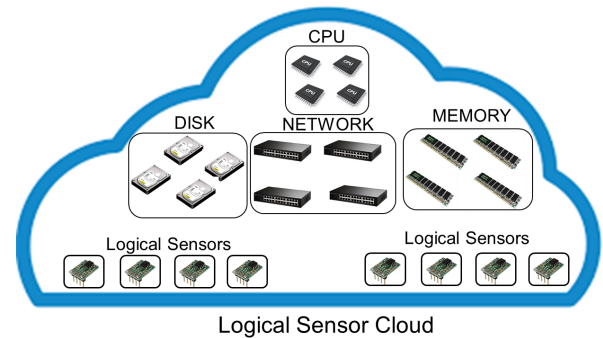


図1 論理センサクラウドの概念図

ンサによって、温度データをデジタル化して取り込むことができる。センサノードでは、温度データをネットワークを経由してサーバに送る。温度データを受け取ったサーバのアプリケーションは、収集した温度データをストレージに蓄積して、そのデータを解析して、ユーザに結果を提供する。この例では、センサノードとアプリケーションが垂直統合されている。つまり、センサノードは、特定の一つのアプリケーションによって排他的に使用されていて、他のアプリケーションからは容易に使うことができない状態になっている。このようにアプリケーションが、排他的に使用しているため、他のユーザが自らの用途やアプリケーションに合わせて、センサノードを自由に使うことができない。また、センサノードに取りつけられているセンサデバイスは、共通の提供方法や共有方法が確立されて

<sup>1</sup> 東京工科大学コンピュータサイエンス学部

<sup>2</sup> 東京女子大学大学院理学研究科

<sup>3</sup> 東京女子大学現代教養学部

いないために、パブリックでオープンな利用がうまくできていないことが課題である。

この例では、温度センサデバイスの特性や精度に関する管理データは、アプリケーションが直接利用しないため、通常はクラウドのアプリケーションに送られることはない。例えば、温度センサからの温度データが 13bit か 16bit でエンコードされているのか、また温度の精度は、何パーセントなのかというような特性の情報は、アプリケーションでは、プログラムを使って取得することができない。現在の仕組みの垂直統合では、温度センサのデバイスの特性や精度を、アプリケーションがプログラムによって、直接取得することは困難である。そのため、センサデバイスの管理として、センサデバイスの特性、状態やレジスタの設定に関する情報を取り扱う方法を確立する必要がある。

さらに、使用しているセンサデバイスが正常かどうか、API がいないためにアプリケーションではわからない。このように個別のセンサデバイスが正常かどうか、センサデータが正常かどうかを管理するための仕組みやアクセス方法も確立する必要がある。

一方、センサデータの転送時において、遅延、タイムアウト、データ損失が発生する可能性が常にある。その結果、取得したデータには欠損値や外れ値が含まれることになる。また、データ分析において異なるセンサデバイスのデータを結合するときに、センサデバイスによって計測間隔や単位の違いがあるために、データ値を補正することが必要になる。これらの補正を行わずに取得データをそのまま分析モデルに入力すると、正確な解析結果が得られないことがわかっている [3]。

著者らは、これまで IoT データの欠損値や外れ値の補完を行うための手法として APREP-S (Autonomous PRE-Processing for Sensor data) [4] の研究を進めてきた。APREP-S では、PBE (Programming by Example) アプローチ [5] によって、人の知識と機械学習を統合することでデータ補完を実現する手法を提案している。このような背景から、センサデータに対して、アプリケーションに入力する前に、収集したセンサデータの事前処理を行なってセンサデータの質を担保することが必要になる。

以上をまとめると、下記の課題がある。

- センサデバイスは、特定のアプリケーションに占有されている。オープンでパブリックな利用方法が確立されていない。
- センサデバイスの特徴、ステータスや設定情報が、アプリケーションに提供されていない。
- センサノードとクラウドの共有資源を統一的に扱うライフサイクル管理の手法が確立されていない。
- センサデータは、遅延、タイムアウト、データ損失が起こる可能性があるが、柔軟にセンサデータの質を担保するための機能が確立されていない。

## Logical Sensor Cloud

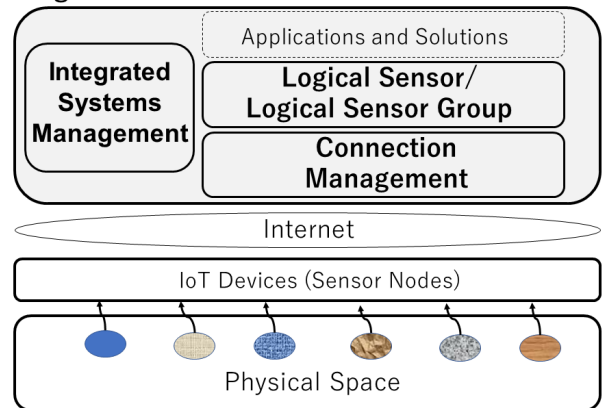


図 2 論理センサクラウドのアーキテクチャ

上記の課題を解決するため、この論文では、クラウドと IoT の統合システム管理のための新プラットフォーム「論理センサクラウド」を提案する。図 1 は、論理センサクラウド (Logical Sensor Cloud) の概念図である。図の論理センサクラウドでは、CPU、メモリ (MEMORY)、ディスク (DISK)、ネットワーク (NETWORK) の資源に加えて、論理センサ (Logical Sensor) が IaaS (Infrastructure as a Service) として提供されている。ここでの論理センサは、複数の仮想センサによって高可用性を提供することと、データ補完の機能によってセンサデータの質を担保する。また、論理センサクラウドは、DICOMO2019 にて報告した「クラウドとセンサの統合システム管理方式」を、拡張したアーキテクチャになる [6]。

この論文の第 2 章では、論理センサクラウドに関連する研究について述べる。第 3 章は、アーキテクチャとデザインについて述べる。第 4 章は、実際にどのように実装されるか実装について述べる。第 5 章は、論理センサクラウドに関する評価について述べる。第 6 章は、必要とされる機能や方式に関する議論について述べる。第 7 章は、この論文での貢献をサマリーしてまとめる。

## 2. 関連研究

この章では、論理センサとデータ補完機能の関連研究について述べる。

### 2.1 論理センサと統合システム管理

論理センサクラウドの提案の関連研究について述べる。センサクラウドのサーベイの結果が報告されている [7]。この論文のサーベイ報告では、Sensor-Cloud プラットフォームの関連研究が報告されている。このサーベイの最初に著者らの以前研究であるセンサクラウドを説明している。このセンサクラウドは、センサを仮想化した仮想センサによって、センサノード固有の制約である排他的な利用やユーザビリティを改善している [8]。

グリーン・センサ・クラウドでは, MMDD (Multi-Method Data Delivery) を提案している [9]. MMDD では, センサからセンサユーザまでのデータの配送について, ルールにもとづいておこなう方法を提案している.

一方で IoT デバイスに近くおくためのクラウドの拡張として, Fog computing という新しいコンピューティングパラダイムが, 提案されている [10][11]. Fog Computing の提案は, IoT デバイスに近いところに, クラウドのコンピューティング, ストレージ, ネットワーク機器を拡張することである. このようにクラウドとセンサデバイスの中間層として設置するものを Fog Computing と定義する. Fog Computing の論文では, Fog Computing のチャレンジを, 構造的な課題, サービスオリエンティッド, セキュリティに整理して報告している.

また, 中央に集めたコンピュータシステムやクラウドコンピューティングからのパラダイムのシフトとして Edge Computing が提案されている [12]. Edge Computing では, クラウドコンピューティングに集約されている機能や資源を Edge Computing に置く. この結果, 機能と資源が Edge Computing に分散されるため, ユーザには, よりよいサービスを提供することが可能になる.

また, IoT では, 26 個の IoT Cloud プラットフォームについて調査がなされている. 調査された IoT Cloud プラットフォームについて, アプリケーション開発, デバイス管理, システム管理, 多種多様な管理, データ管理, 解析, デプロイメント, モニタリング, 仮想化, のように研究のドメインに整理して, それぞれの機能を報告している [2]. 前記の報告では, 現在の IoT Cloud プラットフォームの問題点として, 標準化, 多様性, コンテンツ・アウェアネス, ミドルウェア, IoT ノードの識別, エネルギー管理, フォールトトレラントに分類して報告している.

著者らは, 以前の研究において, センサクラウドの提案と, それに関連した提案をしてきている [8] [13] [14] [15]. この論文で提案している論理センサクラウドの論理センサは, センサクラウドの仮想センサを拡張することによって実現される. また, 論理センサは, 高いパフォーマンスと高信頼性のセンサを API 経由でアプリケーションに提供することができる. これらの以前のセンサクラウドの研究は, 論理センサクラウドの以前アーキテクチャとして位置づけられる.

## 2.2 データ補完

論理センサで用いるデータ補完法は, 補完ターゲットの特徴に応じて, 予め定義しておいた複数の補完手法から 1 つを自動で選択することにより補完を実現するものである. 選択対象となる補完手法としては, これまで提案されてきた既存の補完手法を用いる. 例えば, スプライン補間 (複数の点を結ぶ滑らかな曲線を描くことによる補完) のよう

な古典的な手法, 時系列データを予測するための回帰モデル (季節性やトレンドを組み入れた時系列モデル) をベースとした手法 [16] や, 再帰的ニューラルネットワークを用いて系列データの予測を行う LSTM (Long Short-Term Memory) [17] である. 本研究では, 各補完ターゲットの特徴に応じて, これらの中から適した手法を選択することにより, 単一の手法のみを用いる場合よりも補完の精度を向上することを目指している.

補完ターゲットの特徴と, それに適した手法をどのように関連づけるかについては, PBE アプローチを用いる. PBE はプログラミングの自動化を目指した一連の研究であり, プログラムへの入出力の組を例示することにより, そこから自動でプログラムを生成する. 文生成 (自然言語の生成) の枠組みをプログラムの生成に応用した手法 [5] が良く知られている. ここでは, 機械学習により生成時間の大幅な高速化を実現している. 他分野への応用例もいくつか見られ, 例えば機械学習のための訓練データを生成するために PBE アプローチを用いた研究 [18] がある. 本研究では, 補完ターゲットの特徴を入力, 補完手法を出力として, 入力から出力を得るためのモデルを PBE により学習する. このとき, 入出力の組をシステムの外部から与えることになるが, この過程で人の知識をモデルに組み込む.

## 3. アーキテクチャとデザイン

この章では, 論理センサクラウドのアーキテクチャとデザインについて述べる. 図 2 は, 論理センサクラウドのアーキテクチャである. 論理センサクラウドは, 統合システム管理 (Integrated Systems Management), 論理センサ・論理センサグループ (Logical Sensor/Logical Sensor Group), 接続管理 (Connection Management) で構成されていて, 論理センサクラウドに接続されるインターネット (Internet), センサノード (Sensor Node) と物理空間 (Physical Space) がある.

図 3 は, 論理センサクラウドの基本構成である. 図の物理空間にあるオブジェクトに対して, センサノードは, 物理空間のオブジェクトから得られる物理量をセンサによってデジタル化する. センサノードは, デジタル化したデータをネットワークを経由して論理センサクラウドの接続管理に送る. このときセンサデータとともに, センサデバイスのステータスデータや管理データも, 同時にセンサノードから接続管理に転送される. 接続管理は, これらのデータをそれぞれの論理センサに転送する. アプリケーションやソリューションは, API を使って論理センサにアクセスすることによって, 物理センサと同等の情報を入手できる. また, センサノードを管理するために, 統合システム管理とセンサノードの間で管理情報のやりとりをおこなう. 論理センサクラウドでは, 統合システム管理は, クラウド資源, 論理センサ・論理センサグループ及び IoT デ

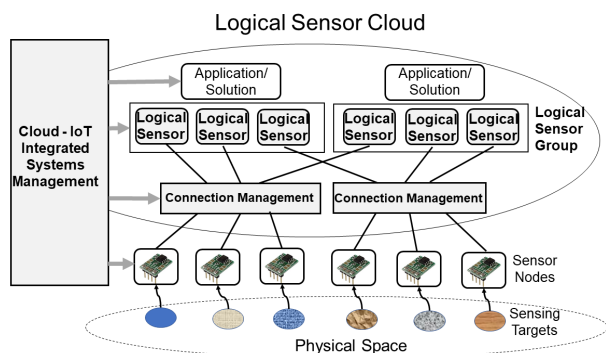


図3 基本構成図

パイスに対するシステム管理をおこなう。

### 3.1 論理センサと論理センサグループ

#### 3.1.1 論理センサ

論理センサは、クラウド上で物理センサと同等の機能を持つソフトウェアコンポーネントである。また、論理センサグループは、論理センサの特徴や特性をもとに複数の論理センサをグループ化して、一つの単位として管理するための方法である。

図6は、論理センサの構成である。論理センサは、図6のように複数の仮想センサ (Virtual Sensor)、データ前処理 (Data Pre-Process) によるデータ補完と API から構成される。また、論理センサは、実際の物理センサのセンサデータや特性と同じ情報を提供するコンポーネントである。また、論理センサは、ソフトウェアコンポーネントとして構成されて提供されるため、同じセンサデータや特性をもった論理センサを複数生成して、複数のユーザが独立して同時に使うことができる。

論理センサは、物理センサと同等の機能を提供することに加えて、センサデータの補完によってセンサデータの質を担保することや、複数の仮想センサによる高可用性や処理能力を確保することを可能にする。以下が論理センサの特徴である。

- 物理センサと同じ情報をクラウド上でアプリケーションに API 経由で提供する。
- 複数の仮想センサによって高可用性を提供することによって、信頼性を高める。
- データ補完の機能によって、提供するセンサデータの質を担保する。

#### 3.1.2 論理センサグループ

論理センサクラウドでは、通常のクラウドに加えて論理センサグループを、クラウド上でユーザのリクエストに応じて提供することができる。ユーザは、論理センサクラウドに対してクラウドのリソースの一つとして論理センサグループをリクエストする。この論理センサグループは、複数の論理センサで構成される。論理センサグループを構成する個々の論理センサは、通常の物理センサに加えて、可

用性の向上やセンサデータの質の向上のためのデータ補完機能を備えている。この論理センサは、センサデータだけでなく、物理センサの特性や精度、物理センサの状態に関する情報についても API 経由で提供する。アプリケーションは、このような情報を得ることによって、論理センサ API を使ってあたかも物理センサと同じかそれ以上の機能を使うことができる。

図4は、論理センサグループである。論理センサグループは、1つあるいは複数の論理センサをグループ化したものである。論理センサグループは、それぞれの論理センサが、コンピューティングリソースとしては、小さい単位であることと、論理センサが数多くになるために、複数の論理センサをグループとして、1つの単位として扱うための方法である。また、論理的なグループであるため、どのような論理センサの組み合わせであってもグループ化して、論理センサグループとして取り扱うことが可能である。図4は、論理センサグループであって、 $LS_1$  から  $LS_6$  までの合計6つ論理センサがあって、これらが1つの論理センサグループになる。この論理センサグループは、統合システム管理によって登録されて管理されている。

例えば、論理センサを  $LS$  とすると、複数の論理センサは、 $LS_i$  として表わされる。このとき  $i$  は、 $1 \dots n$  になる。また、論理センサが複数集まった論理センサグループ  $LSG_j$  は、 $LSG_j = \{LS_i, LS_{i+1}, \dots, LS_{n-1}, LS_n\}$  になる。このとき、 $LSG_j$  において  $j = 1 \dots n$  になる。また、論理センサは、同時に複数の論理センサグループに属することができるため、論理センサグループの種類は、論理センサのあらゆる組み合わせになる。

論理センサクラウドにおいて、論理センサをサービスとして提供する場合、個別の論理センサごとにライフサイクルを管理するのではなく、論理センサグループごとのライフサイクル管理を行う。

例えば、東京都八王子市の温度センサ群という論理センサグループを作る。このグループには、東京都八王子市に設置された論理温度センサが50個入っているとすると、また、この論理センサグループは、複数の論理センサを自在に組み合わせることで作ることができる。下記のリストが、論理センサグループの一例である。

- 同じ組織にある複数の論理センサ。例えば、東京工科大学の複数の論理センサのグループ。
- 同じ物理センサ（デバイス）をもった論理センサ。例えば、PM2.5 検出の論理センサのグループ。
- 同じ地域あるいは、同じ場所にある論理センサ。例えば、東京都八王子市の論理センサのグループ。
- 同じ目的で利用できる異なった種類の論理センサ。例えば、天気状況（温度、湿度、気圧）の論理センサ。

また、論理センサクラウドでは、ユーザからのアクセスや監視対象も、この論理センサグループ単位になる。この



ように論理センサグループは、ソフトウェアのオブジェクトである論理センサが複数集まって構成される。また、それぞれの論理センサグループは、独立して個別に扱うことができる。そのため、論理センサグループの論理センサの追加と削除は、それぞれ個別に行うことになる。

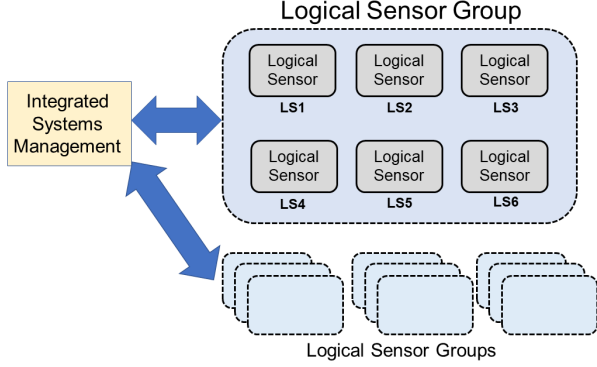


図 4 論理センサグループ

### 3.2 データ補完

#### 3.2.1 手法の概要

センサデータの質を担保するために、論理センサのデータ前処理では、APREP-S [4] によるデータ補完手法を用いる。APREP-S は、欠損値・外れ値の検出、およびデータ補完には既存手法を用いるが、外部入力として分析者の補完に関する知識をモデルに組み込むことで、補完の精度を向上させる点に特徴がある。補完手法は、補完ターゲットの特徴に応じて適切な手法が選択されるが、この関連付けを行うための推論モデルを、外部入力を利用したベイズフィルタにより逐次的に更新していくことで、人の知識の組み込みを実現する。

ここでは、複数の補完手法（例えば ?? 章に示した手法など）を予め定義しておいた上で、以下の手順に従い推論モデルの更新を行う。

- (1) 補完ターゲットを検出し、その特徴を抽出
- (2) 推論モデルを用いて、補完手法ごとの適合度を算出
- (3) 分析者は、適合度を参考にしながら補完手法を選択
- (4) ベイズフィルタにより推論モデルを更新
- (5) 以上を繰り返す

論理センサの前処理では、推論モデルの学習はバックエンドで行う。センサデータの前処理は、論理センサに組み込まれた処理エンジン（学習済推論モデル）によって行い、これを非同期で更新済のものと入れ替えることによって、最新の推論モデルの利用を可能にする。以下、推論モデルの詳細を述べる。

#### 3.2.2 モデルの定義

観測データとして  $N$  個の入力値  $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$  および出力値  $\mathbf{Y} = \{\mathbf{y}_1, \dots, \mathbf{y}_N\}$  があるものとする。入力値

#### Algorithm 1 学習フェーズ

INPUT:  $\{\mathbf{X}, \mathbf{Y}\}$

OUTPUT:  $\alpha, \beta$

```

1:  $\alpha \leftarrow \mathcal{N}(\mu_\alpha, \sigma_\alpha), \beta \leftarrow \mathcal{N}(\mu_\beta, \Sigma_\beta)$ 
2: for  $n \leftarrow 1$  to  $N$  do
3:   for  $k \leftarrow 1$  to  $K$  do
4:      $\mathbf{g}_k = \alpha_k + \beta_k \mathbf{x}_n \leftarrow \alpha_k, \beta_k, \mathbf{x}_n$ 
5:      $\mathbf{f}_k = \exp(\mathbf{g}_k) / \sum_{k'=1}^K \exp(\mathbf{g}_{k'}) \leftarrow \mathbf{g}_k$ 
6:      $\text{Cat}(\mathbf{y}_n | \mathbf{f}_k) \leftarrow \mathbf{y}_n, \mathbf{f}_k$ 
7:      $\alpha_k, \beta_k \leftarrow \text{sampling with Cat}(\mathbf{y}_n | \mathbf{f}_k)$ 
8:   end for
9: end for
10: return  $\alpha, \beta$ 
```

$\mathbf{x}_i \in \mathbb{R}^M$  は  $i$  番目の補完ターゲットの特徴を表現するベクトルであり、これを  $K$  個（予め定義しておいた補完手法の数）あるクラスの 1 つに分類する問題として推論モデルを構築する。具体的には、多次元ベクトル  $\mathbf{y}_i \in \{0, 1\}^K$  が  $\sum_{k=1}^K y_{i,k} = 1$  を満たすとし、これが入力値  $\mathbf{x}_i$  に対して、次のようなカテゴリカル分布にしたがって出力されると仮定する。

$$p(\mathbf{Y} | \mathbf{X}, \mathbf{W}) = \prod_{i=1}^N p(\mathbf{y}_i | \mathbf{x}_i, \mathbf{W}) = \prod_{i=1}^N \text{Cat}(\mathbf{y}_i | \mathbf{f}(\mathbf{x}_i, \mathbf{W})). \quad (1)$$

ここで行列  $\mathbf{W} \in \mathbb{R}^{M \times K}$  はこのモデルのパラメータである。式 (1) における非線形関数  $f$  については、 $K$  次元のソフトマックス関数を使うことにし、ソフトマックス関数の入力値には、ガウス事前分布から生成される 2 つのパラメータ  $\alpha, \beta$  を持つ一次関数  $\alpha + \beta \mathbf{x}_i$  を仮定する。各次元  $k$  に関して関数を具体的に記述すると、以下のようになる。

$$f_k(\mathbf{x}_i, \mathbf{W}) = \frac{\exp(\alpha_k + \beta_k \mathbf{x}_i)}{\sum_{k'=1}^K \exp(\alpha_{k'} + \beta_{k'} \mathbf{x}_i)}. \quad (2)$$

式 (2) を式 (1) に代入したものが、ここでの推論モデルとなる。

#### 3.2.3 学習フェーズ

学習フェーズでは、入力値と出力値の組  $\{\mathbf{X}, \mathbf{Y}\}$  を訓練データセットとして与えることにより、式 (1) における  $\mathbf{W}$  の事後分布（具体的には  $\alpha$  と  $\beta$  の分布）を学習する。事後分布は以下のように与えられる。

$$p(\mathbf{W} | \mathbf{Y}, \mathbf{X}) \propto p(\mathbf{Y} | \mathbf{X}, \mathbf{W}) p(\mathbf{W}). \quad (3)$$

APREP-S では、MCMC 法（Markov Chain Monte Carlo methods）を用いて大量のサンプルを取得することにより、このパラメータ学習を行う。モデル学習のアルゴリズムを Algorithm 1 に示す。

#### 3.2.4 推論フェーズ

学習フェーズで得られた推論モデルを用いると、新たな補完ターゲットの特徴を表現するベクトル  $\mathbf{x}_*$  が与えられ

たときに、 $\mathbf{y}_*$  の予測分布を以下のように推論することができる。

$$p(\mathbf{y}_*|\mathbf{Y}, \mathbf{x}_*, \mathbf{X}) = \int p(\mathbf{y}_*|\mathbf{x}_*, \mathbf{W})p(\mathbf{W}|\mathbf{Y}, \mathbf{X})d\mathbf{W}. \quad (4)$$

APREP-S では、具体的なパラメータの実現値を  $\mathbf{W}^{(1)}, \dots, \mathbf{W}^{(L)}$  のように  $L$  個サンプリングすることで、 $\mathbf{y}_*$  の期待値を以下のように計算する。

$$\langle \mathbf{y}_* \rangle \approx \frac{1}{L} \sum_{l=1}^L f(\mathbf{x}_*, \mathbf{W}^{(l)}). \quad (5)$$

この値は、ある特徴量  $\mathbf{x}_*$  に対する、補完手法ごとの適合度を表す。例えば、 $K=3$  であり、 $\langle \mathbf{y}_* \rangle = (0.23 \ 0.06 \ 0.71)^\top$  であったとする。この特徴量に対しては、手法1の適合度（手法1が適しているであろう確率）が0.23、手法2の適合度が0.06、手法3の適合度が0.71であると考えられる。この適合度と補完ターゲットの特徴を参考にしながら、分析者は、実際に利用する補完手法を選択することになる。その結果  $\mathbf{Y}$  が更新され、APREP-S はこの更新された  $\mathbf{Y}$  を用いて新たな学習を行う。例えば、上記の例で手法1を選択した場合、 $\mathbf{y}_* = (1 \ 0 \ 0)^\top$  と更新される。

### 3.3 統合システム管理

論理センサクラウドでは、クラウドとIoTの統合システム管理として、クラウドのIT資源とIoTの管理資源を統合的に管理する手法を使う。論理センサクラウドの場合、統合システム管理は、クラウドのIT資源、論理センサ、センサノードを統合して管理するためのソフトウェアである。この統合管理システムには、プロビジョニング、ログイン、モニタリング、アカウント管理、登録の機能がある。まとめると、クラウドとIoTの統合システム管理として、次の項目がある。

- ユーザからのリクエストによって論理センサと論理センサグループをプロビジョニング、デプロビジョニングをするためのライフサイクル管理。
- センサノード、接続管理、論理センサの機能を監視するためのロギング及びモニタリング機能。
- ユーザごとの論理センサ、論理センサグループ及びIT資源の使用量を記録するためのアカウント管理機能。
- 論理センサグループ、論理センサとセンサノードを論理センサクラウドに登録、更新、削除をするための登録機能。

統合システム管理では、ライフサイクル管理を行なう。クラウドのITリソースのライフサイクル管理の方法は、確立されている。そのため、ここでは、論理センサグループのライフサイクル管理について説明する。論理センサグループは、そのグループを生成する前に、事前にカタログとして登録しておく。このカタログには、複数の論理センサグ

ループが登録されることになる。また、論理センサグループの定義や、論理センサの登録もカタログ登録前に行う。論理センサの登録は、複数のセンサノードを仮想化した仮想センサと、その仮想センサから出てくるデータのためのデータ補完を構成して登録することになる。

例えば、温度センサのためのセンサノードであるセンサノード  $sn_1$  とセンサノード  $sn_2$  があったとする。このとき、それらが仮想化された仮想化センサ  $vs_1$  と仮想化センサ  $vs_2$  が、それぞれ統合システム管理に登録される。論理センサである  $ls_a$  は、これらの  $vs_1$  と  $vs_2$  と、データ補完の  $dp_a$  によって構成されていて登録される。さらに、他の論理センサ  $ls_b, ls_c, ls_d$  も同様に登録される。この  $\{ls_a, ls_b, ls_c, ls_d\}$  に対して、論理センサグループは、 $lsg_\alpha$  として登録する。この論理センサグループも複数登録することができるため、例えば、 $\{lsg_\alpha, lsg_\beta, lsg_\gamma\}$  として登録する。このように統合システム管理では、センサノード、仮想センサ、データ補完、論理センサ、論理センサグループをそれぞれ登録して管理して、これらの特性情報とともに構成情報として記録しておく。

## 4. 実装

この章では、論理センサクラウドの実装について説明する。センサノード、論理センサ、接続管理、統合システム管理は、コンテナ上に実装する。また、それぞれのコンポーネントは、マイクロサービスとして実装する。これらのマイクロサービス間のデータ交換は、REST API によっておこなう。

### 4.1 センサノード

この研究の検証に用いられるセンサノードのハードウェアは、Espressif 社の ESP32 であって、Xtensa デュアルコア 32 ビット LX6 マイクロプロセッサ、4MiB のフラッシュメモリ、インターフェイスとして GPIO と WiFi がある。またソフトウェアは、ESP32 用の MicroPython(v1.12) を使用する。センサのハードウェアは、温度、湿度、気圧を取得することができる Bosch 社の BME280 を使用する。ESP32 と BME280 は、I<sup>2</sup>C によって接続されて、センサデータ以外に、特性データとレジスタ状態を取得している。ESP32 は、WiFi を使ってインターネットにセンサに関連した情報を送ることができる。図9のセンサノードが、ESP32、BME280、MicroPython を示している。センサノードのソフトウェアは、大きくわけて次の機能をもっている。

- (1) センサからデータを取得して、接続管理に送信する機能。
- (2) センサデバイスのステータスやレジスタ情報を送信する機能。
- (3) センサノードのハードウェア固有の情報（ESP32、

BME280)を送信する機能。

- (4) センサノードのソフトウェアのステータス情報を送信する機能。
- (5) 接続管理からセンサノードに対する問い合わせに返答する機能。
- (6) 接続管理にセンサノードを登録する機能（一連の登録動作のために必要な機能）。

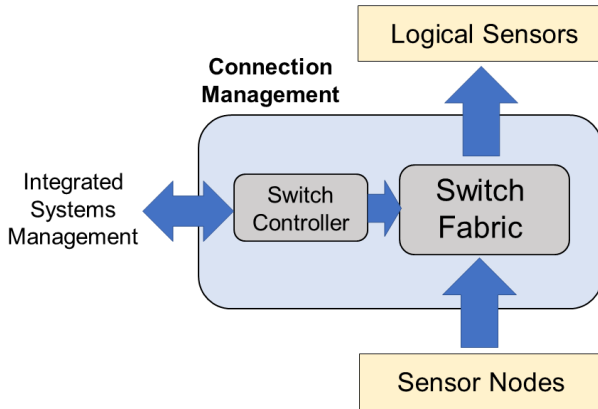


図 5 接続管理

#### 4.2 接続管理

図5は、接続管理である。接続管理は、センサノードからのセンサデータ、固有情報、ステータスを、プロビジョニングされた論理センサに転送する機能をもっている。1つの接続管理は、複数のセンサノードと複数の論理センサ（ソフトウェアコンポーネントであるため、センサノードよりも数が増える。）との間の接続管理を行っている。そのため、センサノードと論理センサ間の転送遅延を最小限することが必要になる。接続するセンサノードと論理センサが多くなってくると、スケーラビリティに対する考慮が必要である。また、接続管理は、統合システム管理とのコマンドやデータのやりとりを行なうことによって、センサノードと論理センサの適切な接続を行なう。以下が、接続管理の機能である。

- (1) センサノードとのデータの送受信の機能。
- (2) 論理センサとのデータの送受信の機能。
- (3) 複数のセンサノードのデータを複数の適切な論理センサに転送。
- (4) 統合システム管理とのデータの送受信の機能。

この場合、一つのセンサノードから複数の論理センサを作ることができる。このとき接続管理では、同じデータを複数の論理センサに転送して、複数の同じ論理センサを使うことができるようにする。

#### 4.3 論理センサ

図6は、論理センサの構成である。論理センサは、1つ

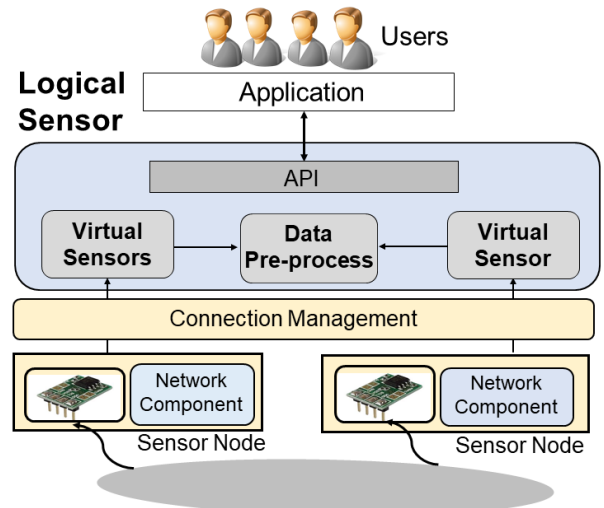


図 6 論理センサ

または、複数の仮想センサと、センサデータのデータ補完から構成されている。この論理センサは、センサデータ、プロパティ、ステータスをアプリケーションに提供するためにREST APIを用意している。アプリケーションは、このREST APIを利用して、センサデータ、プロパティ、ステータスを取得してその用途に使用する。上記のうち、論理センサは、次のコンポーネントから構成される。

- 前処理によってセンサデータを担保する機能
- 物理センサと同等の機能を持った複数の仮想センサ
- 論理センサを高可用性を持たせるための機能
- 接続管理コンポーネントと接続するための機能
- ユーザからのアクセスを受け付けるAPIと認証機能
- 統合システム管理との送受信の機能

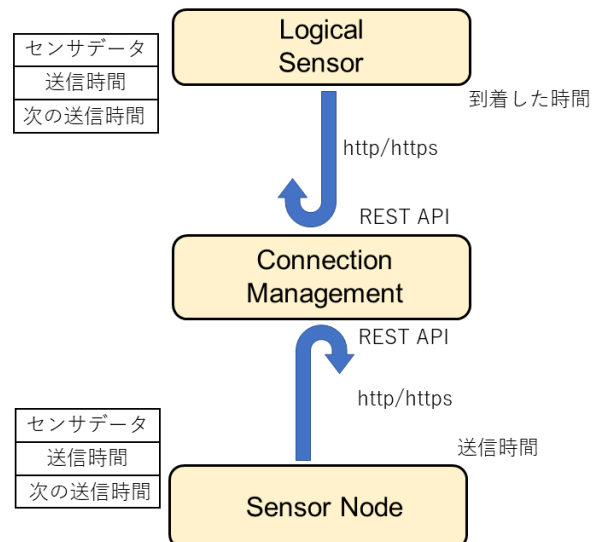


図 7 同期処理

#### 4.4 同期処理

センサノードからセンサデータを接続管理を経由して論

理センサに大きな遅延なく転送することが必要になる。通常、センサデータは、一定時間の間隔でセンサデバイスから取得されて、送り出される。例えば、温度センサであれば、30 秒ごとに 1 回、センサデバイスから温度データが取得されて送り出される。このとき、センサデータをセンサノードから次に送出する予定時間を、接続管理にセンサデータとともに送る。このことによって、接続管理に接続される論理センサは、センサデータとともに、次に送出予定時間を受け取ることができる。この時間もとにして、データを取得するための接続管理への次のアクセスの時間を求めることができる。これは、センサデータが接続管理に届いた時間に、論理センサが接続管理にアクセスして最新のデータを入手することができる。この論文では、このように次の時間を送ることによって同期させて、最短遅延時間でデータを受け取る方法を使う。

図 7 は、この同期方法の実装である。センサノードと論理センサは、いずれも REST API のクライアントであり、接続管理には http/https によってアクセスして、REST API でデータのやりとりができる。図 7 のうちセンサの送信時間を  $t_s$  と、センサの次の送信時間を  $t_{s+1}$  とする。このデータを、センサデータとともに送り出す。論理センサにセンサデータが到着して、アプリケーションがデータを使うことができる時間を  $t_r$  となる。論理センサに届いたデータの中に、次の送信時間  $t_{s+1}$  がある。論理センサは、 $t_{s+1}$  に、あわせて REST API にアクセスして次のセンサデータを取得する。また、論理センサで取得されたセンサデータは、REST API 経由でユーザに提供される。

#### 4.5 統合システム管理

図 8 が、統合システム管理の構成である。統合システム管理には、プロビジョニング (Provisioning)、ロギング (Logging)、モニタリング (Monitoring)、アカウンティング (Accounting)、登録 (Registration)、グラフィカル・ユーザインターフェイス (GUI) がある。これらのコンポーネントは、論理センサ・論理センサグループ、接続管理、センサノードに対してコマンド出す機能と、その結果とステータスを得るための機能を持っている。

### 5. 評価

#### 処理遅延の評価

この章では、実際のソフトウェアを使った処理遅延の測定を行う。この遅延評価は、センサデータを取得したところから、論理センサクラウドを使って、実際にアプリケーションがデータを取得したところまでの遅延時間として評価を行なう。この遅延時間が、論理センサクラウドに必要な最小限の時間となる。

従来からの垂直アプリケーションでは、データの転送時間だけがセンサからアプリケーションまでの遅延時間で

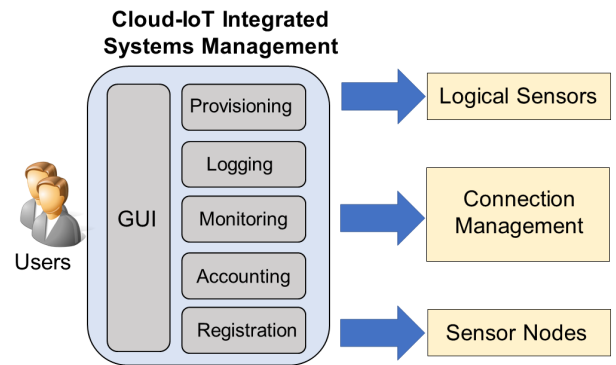


図 8 統合システム管理

あって、それ以外の遅延時間は、同じホスト内でのコピーであって、転送時間に比べて小さいために無視することができると思定される。

図 9 のように論理センサクラウドでは、センサノードから接続管理を経由して、論理センサからアプリケーションにデータが送られる。そのため、それぞれのコンポーネントでの転送処理があり、それらを立ち合わせた時間が、トータルの遅延時間になる。

また、センサデバイスの特徴として、センサデバイスでのデータの送信間隔は、コンピュータの転送時間に対して、一定間隔であって、十分に時間間隔が大きいものであると定義する。つまり、データ転送は、一定間隔の時間で行われていて、それらの間隔が十分に空いているということである。例えば、温度センサであれば、固定長 (バイト) の数 10 バイトの温度データを 1 分間に 1 回送るだけで、毎回の 1 分という送信時間の間隔は、常に一定であるということである。また、転送時間が数十 msec だとすると、送信時間の間隔の 1 分は、転送時間よりも十分に大きいということになる。図 10 は、処理遅延の測定方法である。センサノードから送られたセンサデータは、途中、接続管理を経由して、論理センサに届いて、アプリケーションに提供される。センサノードから送られた時間を  $t_1$  として、論理センサからユーザに提供される時間を  $t_2$  とすると、遅延時間  $t_d$  は、 $t_d = t_2 - t_1$  になる。

図 12 は、処理遅延の分布である。縦軸が、遅延時間 (ミリ秒) であり、横軸はシーケンス番号である。図をみると、100 ミリ秒以下での遅延処理時間になっているが、少数であるが大きく遅延しているデータになっている。また、図 12 は、図 12 のデータを頻度分布にしたものである。これをみると 50-100 ミリ秒にピークがあり、そこから頻度が減少していることがわかる。このように遅延時間の分布が、50-100 ミリ秒にピークがあるということは、センサノードから接続管理を経由して論理センサの間のセンサデータの同期している。センサデータが 50-100 ミリ秒の同期遅延で、論理センサから取り出せるということになる。ネットワークの遅延を入れたデータではないが、また、ネット



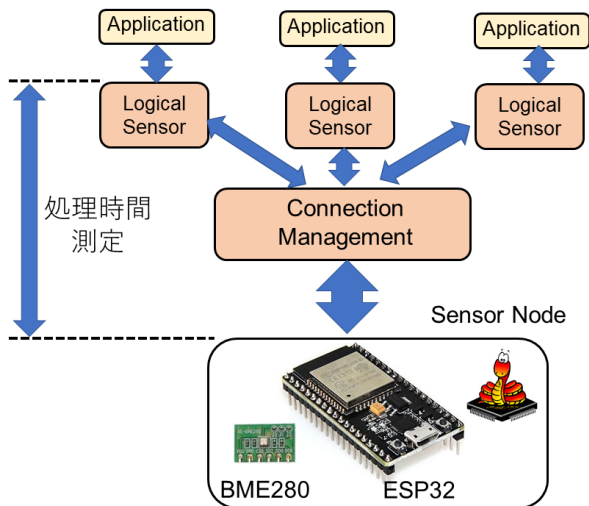


図 9 実験の構成と遅延測定

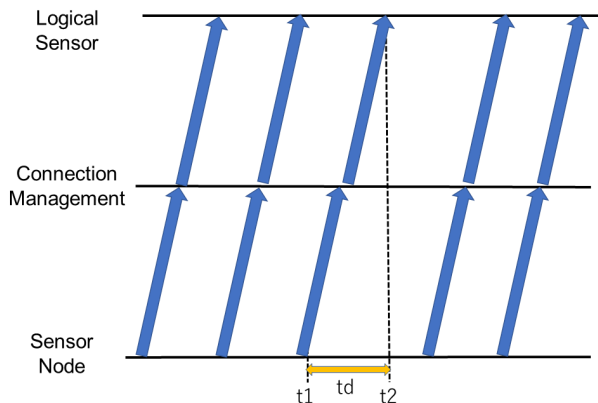


図 10 処理遅延の測定

ワークのデータ転送遅延を最大 50 ミリ秒とすると、大半のセンサデータは、最大でも約 150 ミリ秒程度の遅延で、論理センサからデータを取り出すことができる。このように論理センサクラウドは、使用するセンサデータが環境センサからのデータのように時間にセンシティブでない場合には、十分に実用範囲にあるといえることができる。

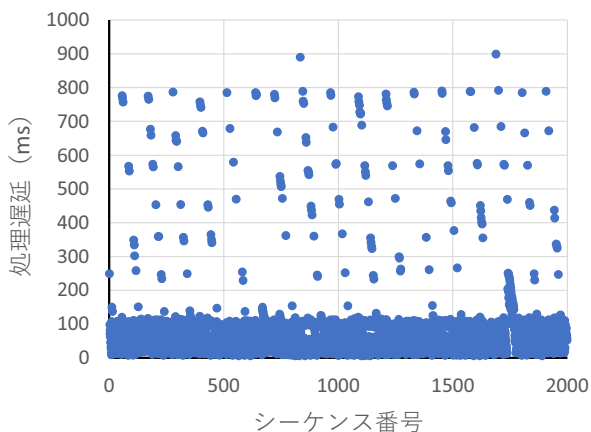


図 11 処理遅延の分布

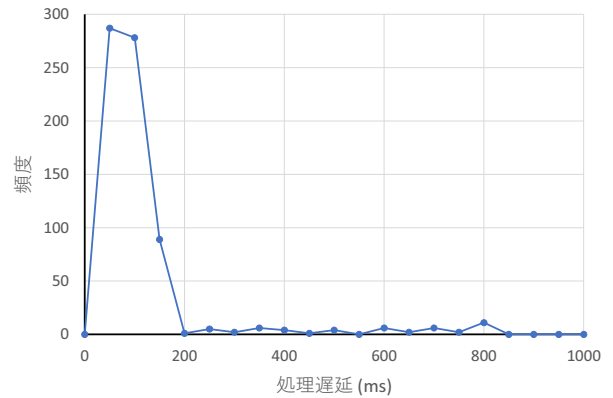


図 12 処理遅延の頻度分布

## 6. 議論

論理センサクラウドの接続管理では、REST API の Web サーバと疑似的な同期機能を使ってメッセージを転送している。これは、MQ Telemetry Transport(MQTT) や Advanced Message Queuing Protocol (AMQP) のようなメッセージキューとは異なって、サーバやクライアントでメッセージ専用のプロトコルを必要とせず、また、メッセージキューのパフォーマンスやスケーラビリティとは関係ない [19]。また、キューの一時的な記録とは異なるため、実装や遅延時間にとって有利である。さらに通常の Web サーバのスケーラビリティの手法やソフトウェアを使うことができるため、比較的簡単に大規模サーバを構築することができる。今後、メッセージ転送に関して、メッセージキューとの定量的な比較を行って評価をすることが必要である。

一方で、論理センサクラウドのスケーラビリティについて考慮する必要がある。接続管理は、REST API の Web サーバとして構成しているため、スケーラブルな構成で大規模なシステムにすることが可能である。また、センサノードや論理センサの数が増大するような構成にするときには、複数の接続管理によるスケーラブルな構成にする必要がある。しかし、スケーラブル構成にしても、センサノードから、接続管理を経由して、論理センサからアプリケーションへのデータ転送を最低限の遅延時間に保つ必要がある。そのため、センサデータの一時記憶や接続管理の階層化構造による遅延の増加はできる限り避ける必要がある。スケーラブルな構成についても今後考慮すべき課題である。

## 7. おわりに

この論文では、IoT デバイスを信頼性が必要なプロダクションシステムとして利用できる基盤技術として論理センサクラウドを提案した。

論理センサクラウドは、IoT デバイスを利用するときに、

論理センサを使うことによって、特定のアプリケーションに占有されることなく、オープンでパブリックな利用ができる。また、論理センサには、データ前処理によるデータ補完機能があって、その機能を使うことによってデータの質を担保することができる。また、論理センサの API によって、IoT デバイスの特徴や精度を取得することができる。さらに、論理センサから IoT デバイスのステータスや設定の管理情報を API 経由で取得することができる。さらに、論理センサグループの定義によって、管理は個別の論理センサではなく、論理センサグループでできるようになる。

以上のことから論理センサクラウドは、センサクラウドを拡張したアーキテクチャとして、実用に供するプロダクションシステムに利用可能な基盤とすることができる。

謝辞 本研究の一部は、JSPS 科研費 20K11776 の助成を受けたものである。

## 参考文献

- [1] B. Dorsemaine, J. Gaulier, J. Wary, N. Kheir, and P. Urien, "Internet of things: A definition taxonomy," in *2015 9th International Conference on Next Generation Mobile Applications, Services and Technologies*, 2015, pp. 72–77.
- [2] "A survey of iot cloud platforms," *Future Computing and Informatics Journal*, vol. 1, no. 1, pp. 35–46, 2016.
- [3] Z. Qi, H. Wang, J. Li, and H. Gao, "Impacts of dirty data: and experimental evaluation," *arXiv:1803.06071*, 2018.
- [4] H. Nagashima and Y. Kato, "Recommendation of imputing value for sensor data based on programming by example," *Journal of Information Processing*, vol. 28, pp. 102–111, 2020.
- [5] A. K. Menon, O. Tamuz, S. Gulwani, B. Lampson, and A. T. Kalai, "A machine learning framework for programming by example," in *Proc. of the 30th International Conference on International Conference on Machine Learning (ICML 2013)*, vol. 28, 2013.
- [6] 串田高幸, "クラウドとセンサの統合システム管理方式," in *情報処理学会マルチメディア, 分散協調とモバイルシンポジウム 2019 論文集*, vol. 2019, June 2019, pp. 985–991.
- [7] A. Alamri, W. S. Ansari, M. M. Hassan, M. S. Hossain, A. Alelaiwi, and M. A. Hossain, "A survey on sensor-cloud: Architecture, applications, and approaches," *International Journal of Distributed Sensor Networks*, vol. 9, no. 2, p. 917923, 2013.
- [8] M. Yuriyama and T. Kushida, "Sensor-cloud infrastructure-physical sensor management with virtualized sensors on cloud computing," in *2010 13th International Conference on Network-Based Information Systems*. IEEE, 2010, pp. 1–8.
- [9] C. Zhu, V. C. M. Leung, K. Wang, L. T. Yang, and Y. Zhang, "Multi-method data delivery for green sensor-cloud," *IEEE Communications Magazine*, vol. 55, no. 5, pp. 176–182, 2017.
- [10] A. V. Dastjerdi and R. Buyya, "Fog computing: Helping the internet of things realize its potential," *Computer*, vol. 49, no. 8, pp. 112–116, 2016.
- [11] R. Mahmud, R. Kotagiri, and R. Buyya, *Fog Computing: A Taxonomy, Survey and Future Directions*. Singapore: Springer Singapore, 2018, pp. 103–130.
- [12] P. Garcia Lopez, A. Montresor, D. Epema, A. Datta, T. Higashino, A. Iamnitchi, M. Barcellos, P. Felber, and E. Riviere, "Edge-centric computing: Vision and challenges," *SIGCOMM Comput. Commun. Rev.*, vol. 45, no. 5, p. 37–42, Sep. 2015. [Online]. Available: <https://doi.org/10.1145/2831347.2831354>
- [13] M. Yuriyama and T. Kushida, "Integrated cloud computing environment with it resources and sensor devices," *International Journal of Space-Based and Situated Computing*, vol. 1, no. 2-3, pp. 163–173, 2011.
- [14] H. Takahashi and T. Kushida, "Innovative iaas management system for sensor devices and it resources," *International Journal of Distributed Sensor Networks*, vol. 10, no. 6, p. 931968, 2014.
- [15] 高橋ひとみ and 串田高幸, "センサークラウド: センサデバイスを it 資源とする拡張クラウド環境," *情報処理学会論文誌*, vol. 54, no. 2, pp. 672–687, Feb 2013.
- [16] S. J. Taylor and B. Letham, "Forecasting at scale," *The American Statistician*, vol. 72, no. 1, pp. 37–45, 2018.
- [17] S. Hochreiter and J. A. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [18] M. Narita and T. Igarashi, "Programming-by-example for data transformation to improve machine learning performance," in *Proc. of the 24th International Conference on Intelligent User Interfaces: Companion (IUI 2019)*, 2019, pp. 113–114.
- [19] D. Happ, N. Karowski, T. Menzel, V. Handziski, and A. Wolisz, "Meeting iot platform requirements with open pub/sub solutions," *Annals of Telecommunications*, vol. 72, no. 1-2, pp. 41–52, 2017.