

title

平本宗大 上村航平 大西真輝 佐野一樹 高屋友輔 平木晶 三好涼太 山下智也 吉原駿平

理工学研究科 数理情報プログラム

A-1. はじめに

A-2. 関連研究

A-3. 提案手法

A-4. 実験

A-5. 結果

A-6. まとめ

B-1. IoT デバイスと AI を用いた課題解決

1. 混雑度推定の有意性

2. Society5.0

B-2. システムの工夫

1. システムの概要

2. フロントエンドの工夫点

3. バックエンドの工夫点

続いて、バックエンドの工夫点について述べる。

(1) Flask の採用

バックエンド開発には、Python の Web フレームワークである Flask を採用した。Flask は、以下の点で本システムに適していると判断した。

- シンプルで使いやすい

シンプルで柔軟な設計により、少ないコードで API を構築できる。そのため、学習コストが低く、迅速なプロトタイプ作成や機能追加が可能となる。

- 軽量

必要最低限の機能のみ提供するため、動作が軽い。そのため、小規模なアプリケーション開発に適しており、リソースの限られた環境でも利用しやすい。

- フロントエンドとの連携が容易

フロントエンドで必要な API エンドポイントを迅速に構築でき、クライアントからのリクエストに柔軟に対応できる。また、JSON 形式のデータを簡単に送受信できるため、フロントエンドと異なるフレームワークを使用しても連携をスムーズに行える。

表 1: 予測結果を保存するテーブル

カラム名	データ型	制約	説明
id	INT	PRIMARY KEY, AUTO INCREMENT	識別子
timestamp	DATETIME	DEFAULT CURRENT_TIME STAMP	スキャン時刻
other_data	JSON	NOT NULL	位置情報, RSSI など

表 2: 作成した API

API 名	概要	レスポンスの例
get_prediction	最新の予測結果を取得	{"prediction": "Jan 2025 23:1
insert_scanned_data	スキャンデータをデータベースに保存	{"message": "}

(2) データベースの採用

本システムでは、モデルによる予測結果の保存方法として、データベース設計を採用した。これにより、拡張性や永続性を確保し、今後の機能追加やデータ増加に柔軟に対応できる設計を実現した。表 1 に示すように、予測結果は適切なスキーマ設計を通じて格納され、データの整合性を保ちながら、将来的な拡張にも対応可能な構成としている。

(3) 機能の API 化

本システムでは、バックエンド側で提供する機能を API として切り出し、フロントエンドとバックエンドの分離を図った。このアプローチにより、次のようなメリットが得られた。

- フロントエンドとバックエンドの独立性の向上

フロントエンドとバックエンドが API を通じて通信する構造にすることで、フロントエンド側の実装に依存せずにバックエンドの開発が可能になった。

- 再利用性の向上

各機能が API として分離されることで、その機能を他のシステムやサービスで再利用することが容易になった。

本システムで作成した API の概要について、表 2 に示す。

B-3. 開発体制

1. プロジェクト全体の体制

本節では、プロジェクト全体の組織体制について、1.1. 先行研究調査・BLE 計測機器の調査および作成、1.2. 混雑度推定アプリ【PASLTO AI】の作成の 2 つに分けて述べる。

1.1. 先行研究調査・BLE 計測機器の調査および作成

BLE 計測機器の作成に必要な知見を収集し、試作・検証を行う。

- 先行研究調査

BLE を用いて混雑度推定を行った研究事例、人数推定に必要なとなるデータを調査する。

- BLE 計測機器の調査および作成

先行研究調査で調査したデータを取得するよう計測機器を設計、試作する。計測機器側、サーバ側に役割分担をし、計測機器側からサーバに送信するプログラム、サーバ側で受信するプログラムを作成する。

1.2. 混雑度推定アプリ【PASLTO AI】の作成

BLE 計測機器との連携を実現するアプリを開発し、システムとしての統合を図る。バックエンド、フロントエンドに役割分担し作成する。

- フロントエンド

デザイン設計、UI 設計、開発を行う。

- バックエンド

API の構築、データベース設計を行う。

2. プロジェクト管理・コミュニケーション

本節では、我々がプロジェクトを円滑に進めるために使用していた管理ツールおよび進捗や問題の報告等を実現するための体制について述べる。

- プロジェクト管理ツール

今回のプロジェクトへ取り組むにあたり、IoT 機器による BLE 取得や混雑度を可視化させるシステムの処理内容を記述したソースコードの管理、並びにタスクの管理を実現するために以下の 2 つのツールを用いた。

1. GitHub

GitHub とは、Git[Git 23] を基盤とするリポジトリ（データベース）を用いたソースコード管理と開発者同士のコラボレーションを実現するプラットフォームのことである [Git 25]。

2. Notion

Notion とは、メモ・タスク管理・ドキュメント作成・データベース機能を統合した多機能な情報管理ツールのことである [Not 25]。

- コミュニケーション体制

進捗確認や課題の報告等を目的として対面の定例ミーティングを週 1 日で実施した。

3. スケジュールとマイルストーン

プロジェクトの開始から報告資料の作成までの一連のスケジュールおよび各過程ごとのマイルストーンを表 3 に示す。

参考文献

[Git 23] Git, <https://git-scm.com/> (2023)

[Git 25] GitHub, <https://github.co.jp/> (2025)

[Not 25] Notion, <https://www.notion.com/> (2025)

表 3: スケジュールとマイルストーン

期間	取り組み
2024/06	プロジェクト開始
2024/07 ~ 08	IoT 機器で BLE を取得するプログラムの作成
2024/09	学食で BLE の取得実験
2024/10 ~ 11	機械学習モデルの構築と性能評価
2024/12 ~ 2025/01	混雑度可視化のプロトタイプシステムの構築
2025/02	学食でシステムの試運転
2025/03	報告資料の作成