ML-BASED MULTI-FLOOR IPS: A COMPARISON

MR.BURIN INTACHUEN
MR.MHADHANAGUL CHAROENPHON
MR.TANAKORN MANKHETWIT

A PROJECT REPORT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE BACHELOR OF ENGINEERING IN COMPUTER ENGINEERING

FACULTY OF ENGINEERING & INTERNATIONAL COLLEGE
MAHIDOL UNIVERSITY

2024

ML-BASED MULTI-FLOOR IPS: A COMPARISON

MR.BURIN INTACHUEN
MR.MHADHANAGUL CHAROENPHON
MR.TANAKORN MANKHETWIT

A PROJECT REPORT SUBMITTED IN PARTIAL
FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE
BACHELOR OF ENGINEERING IN COMPUTER ENGINEERING

FACULTY OF ENGINEERING & INTERNATIONAL COLLEGE
MAHIDOL UNIVERSITY
2024

Computer Engineering Project
entitled
ML-BASED MULTI-FLOOR IPS: A COMPARISON

Mr.Burin Intachuen
Researcher

Mr.Mhadhanagul Charoenphon
Researcher

Mr.Tanakorn Mankhetwit
Researcher

Dr. Sumeth Yuenyong, Ph.D. Signal Processing
Advisor

Thesis
entitled
ML-BASED MULTI-FLOOR IPS: A COMPARISON

was submitted to the Faculty of Engineering & International College,
Mahidiol University
for the degree of Bachelor of Engineering (Computer Engineering)
on
December 15, 2024

____

Dr. Sumeth Yuenyong, Ph.D. Signal Processing
Chair Committee

____

Dr. Charnon Pattiyanon, Ph.D. Information Science
Committee

____

Dr. Konlakorn Wongpatikaseree, Ph.D. Artificial Intelligence of Things
Committee

BIOGRAPHY

NAME	Mr.Burin Intachuen
DATE OF BIRTH	20 November 2002
BIRTHPLACE	Bangkok, Thailand
EDUCATION BACKGROUND	Mahidol International College
MOBILE PHONE	+6695-546-1591
E-MAIL	burin.int@gmail.com

NAME	Mr.Mhadhanagul Charoenphon
DATE OF BIRTH	16 Febuary 2003
BIRTHPLACE	Bangkok, Thailand
EDUCATION BACKGROUND	Mahidol International College
MOBILE PHONE	+6662-497-9193
E-MAIL	mhadhanagulc@gmail.com

BIOGRAPHY

NAME	Mr.Tanakorn Mankhetwit
DATE OF BIRTH	30 December 2002
BIRTHPLACE	Bangkok, Thailand
EDUCATION BACKGROUND	Mahidol International College
MOBILE PHONE	+6692-263-8884
E-MAIL	tanakornmankhetwit@gmail.com

ACKNOWLEDGEMENT

First and foremost, we would like to express our heartfelt gratitude to our advisor and co-advisors, Dr. Sumeth Yuenyong, Dr. Charnon Pattiyanon, and Dr. Konlakorn Wongpatikaseree, whose invaluable insight, advice, and expertise greatly assisted us in completing this project within the limited time frame.

In addition to our advisors, we would also like to extend our appreciation to our committee members, Dr. Sumeth Yuenyong, Dr. Charnon Pattiyanon, and Dr. Konlakorn Wongpatikaseree. Their insightful feedback, encouragement, and thought-provoking questions inspired us to reconsider and delve deeper into areas we had initially overlooked. This project could not have reached completion without their guidance and support.

Finally, we would like to express our gratitude to Mahidol University International College and the Faculty of Engineering, Mahidol University, for providing us with extensive resources and the financial support essential to conducting research for this project. Additionally, we thank CMKL University for their invaluable support through Dr. Charnon Pattiyanon's involvement in this senior project.

Mr.Burin Intachuen
Mr.Mhadhanagul Charoenphon
Mr.Tanakorn Mankhetwit

ชื่อโครงงาน	การเปรียบเทียบระบบกำหนดตำแหน่งหลายชั้นด้วย ML
ผู้วิจัย	นายบุรินทร์ อินทชื่น ICCI
	นายมทนกุล เจริญพร ICCI
	นายธนกร มั่นเขตวิทย์ ICCI
	ปริญญา วิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์
อาจารย์ที่ปรึกษา	ดร. สุเมธ ยืนยง
สำเร็จการศึกษา	15 ธันวาคม 2567

บทคัดย่อ

ระบบระบุตำแหน่งในอาคาร (Indoor Positioning Systems หรือ IPS) แบบดั้งเดิมมักอาศัยบีคอน Bluetooth Low Energy (BLE) และเทคนิคการทำลายนิ้วมือ (fingerprinting) เพื่อวัดค่าตัวชี้วัดความแรงของสัญญาณที่ได้รับ (RSSI) สำหรับคำนวณตำแหน่งผู้ใช้ผ่านการสามเหลี่ยม (trilateration) วิธีการแบบดั้งเดิมเหล่านี้มักต้องการฮาร์ดแวร์จำนวนมากและการวางแผนที่ละเอียดถี่ถ้วน งานวิจัยของเราแนะนำถึงความเป็นไปได้ของการใช้ระบบ IPS ที่ใช้จุดเชื่อมต่อ WiFi (AP) ที่มีอยู่เดิม และเทคนิคการเรียนรู้ของเครื่องขั้นสูง (ML) โดยไม่จำเป็นต้องใช้พิกัดอ้างอิง <X,Y> ซึ่งแตกต่างจากงานวิจัยก่อนหน้า ที่จำกัดอยู่ในห้องหรือชั้นเดียว ในงานวิจัยนี้ได้มีการพัฒนาเครื่องมือเพื่อเก็บข้อมูลสำหรับ IPS ที่สามารถทำงานได้ในพื้นที่บริเวณกว้าง และต่างชั้นกันได้ ด้วยวิธี ML และ Fingerprinting แนวทางของเรามีประสิทธิภาพสูงในสภาพแวดล้อมที่มีหลายชั้น โดยได้ทำการทดสอบในอาคารห้าชั้น แต่ละชั้นแบ่งออกเป็นแปดส่วน ขนาด 16.75 เมตร x 15 เมตร ระบบของเราใช้ความสามารถในการบอกค่าความแรงของสัญญาน wifi (RSSI) โทรศัพท์มือถือ แสดงให้เห็นถึงความเป็นไปได้ในการใช้ระบบ IPS ในพื้นที่สาธารณะ และศึกษาเปรียบเทียบแบบจำลอง ML รปแบบต่างๆ เพื่อกำหนดตำแหน่ง และพบว่า แบบจำลองโมเดล XGBoost Classifier ที่ผ่านการฝึกของเราสามารถทำนายตำแหน่งของผู้ใช้ในอาคารหลายชั้นได้ด้วยความแม่นยำถึง 82% ระบบ IPS ของเราเหมาะสำหรับการประยุกต์ใช้ที่หลากหลาย รวมถึงการนำทางในอาคาร การจัดการอาคาร การรวมเข้ากับอาคารอัจฉริยะ การจัดการฝูงชน และการโฆษณาตามตำแหน่งที่ตั้ง

คำสำคัญ : Indoor Positioning, Indoor Positioning System, XGBoost, Wifi, RSSI

33 หน้า

ML-BASED MULTI-FLOOR IPS: A COMPARISON

STUDENTS Mr. Burin Intachuen ICCI

Mr. Mhadhanagul Charoenphon ICCI

Mr. Tanakorn Mankhetwit ICCI

DEGREE Bachelor of Engineering (Computer Engineering)

PROJECT ADVISOR Dr. Sumeth Yuenyong, Ph.D. Signal Processing

DATE OF GRADUATION December 15, 2024

ABSTRACT

Indoor Positioning Systems (IPS) have traditionally relied on Bluetooth Low Energy (BLE) beacons and fingerprinting techniques to measure Received Signal Strength Indicator (RSSI) for user location calculation via trilateration. These conventional methods often require extensive hardware and meticulous planning. Our study introduces the feasibility of an IPS solution that utilizes existing WiFi Access Points (APs) and advanced Machine Learning (ML) techniques, eliminating the need for reference <X,Y> coordinates. Unlike previous research, often limited to single-room or single-floor environments, our approach functions in multi-floor settings. Tested in a five-floor building with eight segments per floor in a 16.75m x 15m grid, our system uses mobile phones' WiFi RSSI measurement capabilities, demonstrating IPS viability in public spaces. Our paper evaluates ML models to determine the best performer, finding that our trained XGBoost Classifier achieves 82% accuracy in predicting user locations across floors. While the results highlight the potential of our framework in multi-floor settings, further research and optimization are needed to improve scalability. Our IPS framework is well-suited for diverse applications, including Indoor Navigation, Building Management, Smart Building Integration, Crowd Management, and Location-Based Advertising, and beyond.

KEYWORDS : Indoor Positioning, Indoor Positioning System, XGBoost, Wifi, RSSI

33 Pages

CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 Background

People often rely on the Global Navigation Satellite System (GNSS) for navigation, the most well-known one is the Global Positioning System (GPS). There are other systems such as GLONASS, Galileo, Beidou and more. Regardless of what system is being used, satellites continuously transmit radio signals towards ground-based receivers on Earth. The receiver then measures the time of travel of the signal, calculates it with the distances of multiple satellites, and uses trilateration to pinpoint its position. However, its accuracy diminishes indoors due to various factors like low power signal that struggle to penetrate through dense materials, like concrete or metals. This creates issues like signal attenuation, scattering, shadowing, and blind spots, reducing model accuracy[13]. To address this, Indoor Positioning Systems (IPS) have been developed to track people and objects in indoor environments, which can be crucial for both emergency and non-emergency situations.

One common IPS method is trilateration, often implemented with Bluetooth Low Energy (BLE) devices. BLE is widely used due to its low power consumption and ease of deployment. In trilateration, Received Signal Strength Indicator (RSSI) fingerprinting measures signal strength in relation to distance from the transmitter [5]. The fingerprinting technique conducts IPS in two phases, offline and online phases. During the offline phase, RSSI data is collected to create radio maps, and in the online phase, end devices can calculate their position based on the assigned coordinates of each RSSI-emitting device, such as BLE beacons. This method works by determining distances between the phone and each BLE device and triangulating the position based on this data. Each location indoors has a mostly distinct RSSI signature, but RSSI is vulnerable to environmental noise, leading to frequent errors in location estimation[2]. To implement this method, often buildings are to be built to support the implementation and precision is required to assign each device their coordinates.

Many have speculated that by using Machine Learning (ML), it can learn to overcome random noise introduced within these settings. For example, H.T. Gidey et al. [8] use online heterogeneous transfer learning to improve accuracy by combining data from different domains. Other approaches include algorithms like k-Nearest Neighbor (kNN), Support Vector Machines (SVM), Decision Trees, Random Forests, and Neural Networks (NN). This implementation addresses the challenges of traditional IPS systems by making it a classification problem. In doing so, making the precision of positioning to be within acceptable range. While it may not provide pinpoint accuracy, it is reliable for determining the user's zone.

In our work, we will focus on a comparative analysis of different ML approaches applied to IPS. We aim to identify which methods provide the highest accuracy and reliability in estimating indoor positions. As well as being able to differentiate which floor the user is on. Through this comparison, we seek to provide insights into the strengths and limitations of each ML approach, helping to guide the selection of the most effective algorithms for various IPS applications.

Lastly, we will develop a mobile application to benchmark various ML algorithms, providing a comparison of their accuracy and performance in indoor localization systems using our experimental setup.

1.2 Objective

1. To demonstrate feasibility of implementing muti-floor Indoor Positioning System (IPS) using existing Machine Learning (ML) models based on data collected on campus, without having prior knowledge to Access Point Location.
2. To compare the accuracy of various ML algorithms (e.g., kNN, Random Forest, XGBoost, SVM) in the context of Multi-floor IPS.
3. To develop a mobile application that facilitates data collection and performance benchmarking of ML algorithms for IPS.

1.3 Scope

1. This project collects RSSI and location data in hallways across five floors of Mahidol Engineering building 3.
2. Modifying the environment (e.g., adding or removing Wi-Fi Access Points) is strictly forbidden.
3. Eight ML algorithm will be tested to localize users, including KNN, Random Forest, GaussianNB, XGBoost Classifier, XGBoost Random Forest, MLP, SVM, and Indoor Graph Neural Network (from prior work that inspired this project).
4. Mobile Application Platform: Android, on Android version 12 and 14.
5. The number of BSSIDs used is determined by how many are present during the recording.
6. Data is collected while standing still, holding the device at chest level, the rotation is random to keep data diverse, with both collection and positioning done via the mobile application.
7. Data collection and testing are conducted across 40 grids distributed over five floors. We aim to collect more than 1,000 data points, with data from each grid contributing to this total.
8. Movement of the furniture during the experiment is prohibited.
9. The grid size after segmentation is 16.75m x 15m.

1.4 Expected Results

1. The classifier-based approach will achieve an accuracy of 70% or more in grid-based indoor positioning across multiple floors.
2. The system will demonstrate usability of implementing IPS, under the assumption of no building knowledge. (e.g., un-uniformly distributed furniture).
3. The system will function effectively in a normal environment, defined as an area with sufficient public Wi-Fi Access Points.

1.5 Timeline

Table 1.1 Project Timeline

Plan	Timeline											
	2024											
	Jan	**Feb**	**Mar**	**Apr**	**May**	**Jun**	**Jul**	**Aug**	**Sep**	**Oct**	**Nov**	**Dec**
1: Study Sources and Do Research												
2: Design Methodology and Boundaries												
3: Machine Learning Implementation												
4: Develop Application												
5: Bug Fixing and Testing												
6: Thesis Write-Up												
7: Finalise Documents												

CHAPTER 2
LITERATURE REVIEW

This chapter presents a review of previous related work across three essential components: front-end, back-end, and data processing of indoor positioning systems (IPS). The front-end section clarifies the reasoning behind the choice of Flutter and the Android platform, including a description of the primary packages used for data collection. The back-end section surveys the requirements and limitations of existing IPS frameworks, discussing how other research works have addressed these challenges. Lastly, the data processing section outlines the established methodology encompassing feature selection for machine learning techniques, data cleaning methodologies, and significant packages required for model training in our mobile application.

2.1 Front-end

The front-end framework of the mobile application shares similarities with the work of Jo L. [11], particularly in its capability to create customizable grids and maps. However, this framework offers additional advantages. Unlike Jo L.'s framework, this design is specifically tailored to facilitate data recording, enabling seamless integration with the back-end service to support machine learning model training.

When compared to the mobile-based indoor navigation system proposed by Satan, A. [14], this approach demonstrates distinct advantages. Satan, A.'s system utilizes an Android application employing a direction-based navigation method, which includes a visual display of the user’s position, an arrow indicating the next direction, and image-based navigation showcasing the target location. While effective in certain scenarios, this approach encounters limitations in environments with multifloor models. For instance, reliance on image-based navigation may cause user confusion when floors have similar layouts or features, such as identical stairwell doors. Additionally, the arrow-based guidance system is redundant in this context, as the focus of this project is on accurately classifying a user’s location rather than providing directional navigation to a specific destination.

2.2 Back-end

The back-end framework of the mobile application addresses several requirements and limitations, such as handling high-volume data streams. Building upon Dmitris C. [4] the project depends on a web server built on the flask framework, which contributes to API responses for localization, path determination, and destination search. The usage of REST APIs also provides scalability, flexibility and independence [1]. However; as the project aims to present a unique approach to back-end design by focusing solely on data collection and employing TensorFlow Lite for model deployment on edge devices, which is something that, to the best of our knowledge, has never been a methodology that has been explicitly mentioned before. From initial findings, it presents potential applications in environments such as convention centers, shopping malls, and other infrastructures where GPS is insufficient for accurate positioning. Additionally, to build our API service, we will utilize the Go programming language instead of Python, prioritizing future-proofing.

The project aims to use a simple and easy to understand REST API, as the simplification makes it easier for users to use for further development in further studies.

2.3 Machine Learning

In the study titled "An Overview of Indoor Localization System for Human Activity Recognition (HAR) in Healthcare" [3], IPS without ML techniques often implement trilateration to determine a user's <X,Y> coordinates. Other methods may include triangulation, pseudo-range multilateration, and fingerprinting. Despite these efforts, IPS faces the inherent challenge of signal strength fluctuations in varying environments, contributing to variability in RSSI data. ML techniques have been introduced to mitigate this challenge [15]. With sufficient RSSI data, k-Nearest Neighbor (kNN) can perform well in classifying current positions. The study also proposes IndoorGNN as a viable option for classification. The research concludes that while kNN performs well with a high training ratio, IndoorGNN can outperform kNN, Multi-Layer Perceptron (MLP), Support Vector Machines (SVM), and Graph Neural Networks (GNN) when trained with low data.

2.4 Non ML IPS Techniques

1. Satan, A.'s work, reviewed earlier, employs a methodology that utilizes Bluetooth beacons in conjunction with Dijkstra's algorithm to calculate the optimal path for a user by navigating from node to node. While this approach offers precision in determining routes, it has notable drawbacks. One significant limitation is the requirement for regular maintenance of Bluetooth beacons, which may not be desirable due to resource constraints. Additionally, managing and connecting the nodes can be challenging, as the omission of even a single node can result in incorrect path calculations, potentially leading to significant deviations from the intended route.
2. Jo L.'s work [11] employs a straightforward methodology that utilizes the GPS functionality of a smartphone and overlays the location data onto a custom-built map. This approach is a standard and basic technique, offering the advantages of being easy to implement and construct, as it primarily relies on the phone's built-in GPS capabilities and open-source mapping software. One notable benefit of this methodology is its ability to function without requiring a Wi-Fi connection, as it uses the phone's GPS to determine the user's location on the map. However, the approach has significant limitations. The GPS module is not well-suited for multi-floor indoor positioning, as it can be disrupted by thick walls and other structural obstacles. Additionally, the map must be highly accurate to avoid causing confusion for users attempting to navigate the environment.

2.5 ML IPS Techniques

In another work, kNN can be used to determine the Euclidean distance between the immediate measurements of received signal strength (RSS) and the average RSS value associated with each fingerprint stored in a database. It subsequently finds one or more fingerprints nearest to the real-time sample signal RSS and calculates the averages or weighted averages of the positions of every fingerprint to estimate the target's position [16]. An indoor positioning system employing SVM is explored in the work of Khatab et al. [6], using a modified version of standard SVM called fuzzy Least Squares SVM (LS-SVM). This system operates in two phases: offline, where RSS values are collected and stored, and online, where the target position is estimated using fuzzy membership functions.

In another approach, the generation of a decision tree is based on a weak classifier utilizing the C4.5 algorithm. The adaptive boosting algorithm (AdaBoost) [7] constructs an ensemble model based on the earlier base models to estimate location, consisting of multiple weighted decision trees. During the offline calibration stage, an RSS fingerprinting database is established using collected RSS data, and a Random Forest classification model is trained with these input datasets. The trained model, which yields optimized classification results, can then be directly employed in the online positioning stage to estimate real-time tracking points [12].

Neural networks offer the capability to learn the correlation between 2D image coordinates and 3D world coordinates, allowing user location estimation via a camera without requiring complex mathematical algorithms [9]. The study referenced in [17] employs a three-layer feed-forward neural network to infer world coordinates from image coordinates captured by two cameras; however, this can also be achieved using just one camera. Using neural networks, object locations can be detected through camera input [10].

Despite the impressive findings from these studies, particularly regarding IndoorGNN, it is important to note that most tests were conducted in single-room environments, which may introduce biases due to the specific datasets used, such as the MNAV and UJIIndoorLoc datasets. Therefore, conducting a comparative study on ML techniques for IPS is warranted. The existing studies do not adequately address performance in expansive, multi-floor public areas, and the effectiveness of IndoorGNN in these contexts remains to be established. Thus, our study aims to fill this gap by exploring the feasibility of ML techniques for multi-floor IPS in real-world scenarios. While other implementations are noteworthy, we will closely resemble IndoorGNN in our approach, as our goal is to minimize hardware requirements, section the experimental area into grids, and simplify the implementation and testing process as much as possible.

CHAPTER 3
METHODOLOGY

This chapter is dedicated to explaining the study approach used for our comparative study of ML models in multi-floor IPS. It is structured as follows:

3.1 An Overview of the Comparative Study

The goal of this study is to evaluate the performance of various machine learning (ML) models in a multi-floor indoor positioning system (IPS) using Wi-Fi signal strength (RSSI) data. To achieve this, we designed a controlled experimental framework with well-defined variables to ensure a fair and meaningful comparison of model performance.

3.1.1 Independent Variables

The independent variables in this study represent the factors we intentionally manipulate or vary to observe their effects on the IPS model performance. These include:

1. **Machine Learning Models:** A diverse set of ML models, including Multi-Layer Perceptron (MLP), k-Nearest Neighbors (k-NN), Support Vector Machines (SVM), Random Forest (RF), XGBoost and IndoorGNN, are evaluated to understand their suitability for indoor positioning tasks.
2. **Grid Configuration and Size:** Each floor of the experimental area is divided into grids of 16.75 x 15 meters, representing distinct zones for classification. This segmentation approach allows us to assess how well each model can predict the correct grid based on input data.
3. **Data Collection Points:** Measurements are taken at five predefined points within each grid (center, top left, top right, bottom left, and bottom right). These points introduce variability in the training and testing datasets, reflecting real-world scenarios where users may move within a grid.

4. **Wi-Fi Signal Characteristics:** RSSI values from multiple Wi-Fi access points are used as input features. Variability in signal strength, caused by environmental factors, provides a robust test of the models' ability to generalize under noisy conditions.

3.1.2 Dependent Variables

The dependent variables are the measurable outcomes used to evaluate the effectiveness of the independent variables:

1. **Prediction Accuracy:** The primary metric for comparison is the accuracy with which each ML model can correctly classify a given data point into its corresponding grid.
2. **Accuracy and Precision:** These metrics provide additional insights into how well the models perform in terms of correctly identifying grids while minimizing false positives and false negatives.
3. **Generalization Performance:** The ability of the models to maintain accuracy across different floors or unseen datasets is evaluated to assess their robustness.

3.1.3 Control Variables

To ensure the validity of the experiment, several control variables are maintained throughout the study:

1. **Consistent Data Collection Process:** Data is collected under similar environmental conditions, with the same number of data points and measurement points per grid.
2. **Preprocessing Pipeline:** All models are trained and tested on the same preprocessed dataset to eliminate bias introduced by inconsistent data preparation.
3. **Controlled Environment:** The study is conducted in a multi-floor setting with minimal external interferences to provide a realistic but controlled comparison.

This structure ensures that the experiment systematically evaluates the suitability of different ML models for indoor positioning tasks, focusing on accuracy, robustness, and computational efficiency. The following sections will elaborate on the data collection, processing, model selection, and implementation processes integral to this study.

3.2 The Data Collection Process

The data collection process was meticulously designed to ensure the creation of a robust and diverse dataset for training and evaluating the indoor positioning system (IPS). The goal was to gather Wi-Fi Received Signal Strength Indicator (RSSI) data across multiple grids within a multi-floor environment, with careful consideration given to both central and boundary areas of each grid.

3.2.1 Initial Data Collection Strategy

Data collection began at the center of each grid, focusing on points in close proximity to ensure a dense and representative dataset in the initial stages. By collecting data points near the center of each grid, we established a reliable core for each zone, which is critical for training machine learning models with high accuracy in areas of high confidence.

3.2.2 Progressive Expansion of the Collection Area

After establishing a dense dataset near the center of each grid, the data collection area was gradually expanded outward. This progressive approach allowed for the inclusion of boundary areas, which are typically more challenging to classify due to signal overlap and environmental noise. The expansion was performed systematically to ensure an even distribution of data points across the grid while avoiding significant gaps in coverage.

3.2.3 Treatment of Boundary Areas

Areas near grid boundaries pose a unique challenge for indoor positioning systems, as signal strength from Wi-Fi access points may not distinctly differentiate between adjacent grids. To address this, boundary regions where data collection was not performed were designated as *grey zones*. These zones are regions where classification into one grid or the other is less critical, as they are expected to exhibit overlap in signal characteristics. This treatment allows for a more realistic and practical representation of the indoor environment, reducing the impact of ambiguous areas on overall model performance.

3.2.4 Considerations for Environmental Variability

The data collection process was conducted under relatively consistent environmental conditions to minimize the influence of transient factors such as human movement or temporary obstructions. This ensures that the collected dataset reflects the typical signal strength distributions in a controlled environment, enabling fair comparisons between machine learning models.

3.2.5 Summary of Data Collection Protocol

- Data points were collected at five predefined locations within each grid (center, top left, top right, bottom left, and bottom right) to establish a structured dataset.
- The collection process began at the center of each grid and expanded outward systematically.
- Boundary areas were intentionally left sparse, with uncollected zones designated as *grey zones*.
- Consistent environmental conditions were maintained throughout the data collection process.

This strategy ensures the dataset includes a balance of high-confidence central data points and more variable boundary data, providing a comprehensive foundation for training and evaluating machine learning models for indoor positioning.

3.3 The Data Processing Process

Data collected from back-end services is stored in a PostgreSQL database. Below is the structure of the relational data involved in data collection. Lines connecting tables represent foreign keys, and highlights represent super keys.

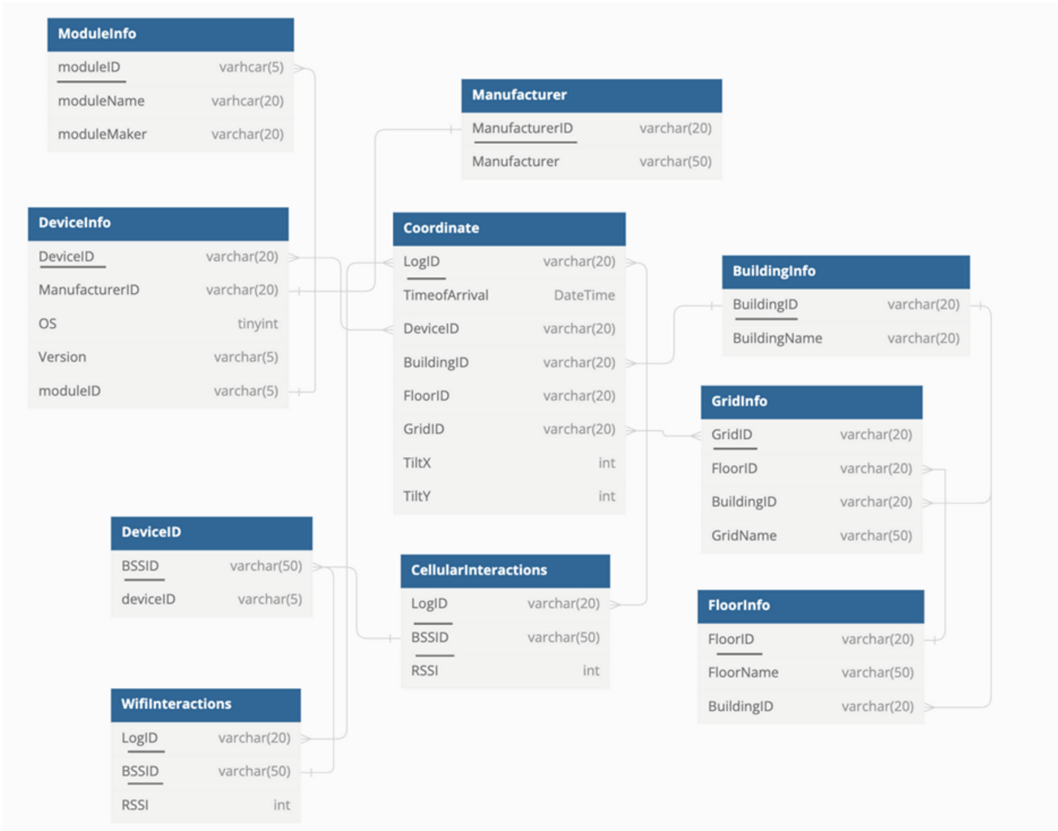

Figure 3.1 Relational Database

Conceptually, these data points are linked in a relational context. However, our database tables do not implement these constraints to mitigate performance loss during data writing.

NoSQL databases were considered but were found less suitable for handling the structured spatial data and complex queries required for IPS.

LogID	RSSI#1	RSSI#2	RSSI#3	...	RSSI#9	RSSI#10	Classification
S0001	-40db	-20db	-10db	...	-5db	0db	1
S0002				...			
S0003				...			
S0004				...			
S0005				...			
S0006				...			

Figure 3.2 Mock-Up Labeled Data

Figure 3.2 represents each data point, its features, and classification. The LogID represents each unique data point, and RSSI values are the features. If specific data does not contain the RSSI of certain Wi-Fi APs, those are automatically assigned -100dBm, the lowest value a smartphone can capture. Classification represents the labeling of collected data. For example, data collected at Grid ID 1 is labeled with 1 as its grid ID.

3.4 The Selection of ML Models for Comparison

The selection of machine learning (ML) models for this study was guided by the need to evaluate a diverse range of algorithms commonly used in classification tasks, particularly in indoor positioning systems (IPS). Each model was chosen for its relevance and proven effectiveness in similar tasks, as well as its ability to handle the characteristics of our collected dataset. To ensure fair comparisons, each model underwent parameter optimization through systematic looping, aiming to achieve the best test accuracy.

3.4.1 Parameter Optimization Process

The parameter optimization process was implemented as an iterative search, where key hyperparameters for each model were varied within predefined ranges. The goal was to identify the configuration that produced the highest test accuracy for each algorithm. Below is a summary of the models and their respective parameter optimization strategies:

- **k-Nearest Neighbors (k-NN):** The primary hyperparameter for k-NN is the number of neighbors (*n_neighbors*). We looped through values from 1 to 100, systematically evaluating the model's performance for each value. The optimal *n_neighbors* was selected based on the highest test accuracy observed during this process.
- **Random Forest:** For Random Forest, the key hyperparameter is the number of estimators (*n_estimators*), which determines the number of decision trees in the ensemble. We performed a loop through 100 different values, ranging from 1 to 100. The optimal number of estimators was chosen based on the configuration that maximized test accuracy.
- **XGBoost Random Forest and XGBoost Classifier:** Similar to Random Forest, the XGBoost Random Forest and XGBoost Classifier models were tuned by

looping through 100 values for *n_estimators*. The best-performing configuration was identified based on the same accuracy criterion.

- **Multi-Layer Perceptron (MLP):** For MLP, the number and configuration of hidden layers were critical hyperparameters. Various hidden layer architectures were explored, including different numbers of layers and neurons per layer. Each configuration was evaluated, and the architecture that achieved the highest test accuracy was selected as the optimal model.
- **Support Vector Machine (SVM):** The key hyperparameter for SVM is the regularization parameter (C), which controls the trade-off between achieving a low error on the training data and minimizing model complexity. We performed a loop over 100 values for C, systematically evaluating the effect of each value on test accuracy. The optimal C value was selected based on the best performance observed.
- **IndoorGNN:** For the IndoorGNN model, we followed the parameter optimization strategy outlined in the original paper. The same range of hyperparameters and tuning methodology were employed, ensuring consistency with the model's established benchmarks. We looped through all relevant parameters as specified in the original study, selecting the configuration that produced the best results on our dataset.

3.4.2 Summary of Parameter Tuning Process

Each model's performance was systematically evaluated across multiple hyperparameter configurations, with test accuracy serving as the primary metric for selection. By employing this rigorous looping process, we ensured that all models were compared under their best-performing configurations, providing a fair and comprehensive evaluation of their suitability for indoor positioning tasks.

3.5 The Implementation of the Data Collection and IPS Mobile Application

3.5.1 Experimental Setup

Data was collected from five floors, each divided into grids. Measurements were taken from five points per grid ("Center," "Top Left," "Top Right," "Bottom Left," "Bottom Right").

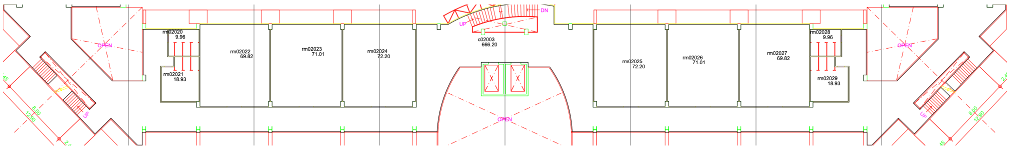

Figure 3.3 Measurement Area from One of the Five Floors

This setup balances grid size with practical accuracy, enabling effective classification for real-world applications.

3.5.2 Measurement Error and Tolerance Levels

Grid-based systems inherently involve some error due to environmental interferences. However, this method balances acceptable error margins with functionality for practical applications.

3.5.3 System Architecture

The system architecture is composed of four primary segments, as illustrated in Figure 3.4: Front-end, Back-end, Database, and Machine Learning (ML). This modular design meets specific requirements and addresses limitations in IPS applications. Each segment has unique roles, interactions, and constraints critical to the system's overall functionality.

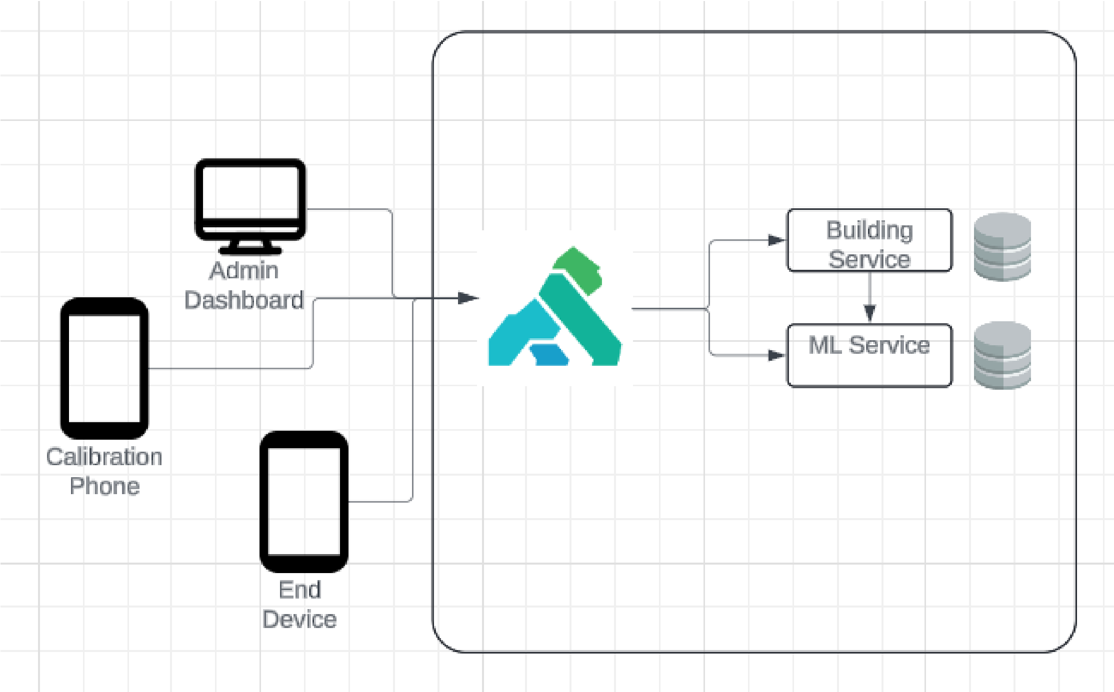

Figure 3.4 Overall System Architecture Diagram

3.5.4 Front-end

The front-end application is Flutter-based, supporting cross-platform compatibility and leveraging packages like wifi_scan to fetch Wi-Fi access point information for ML model use. The UI/UX is optimized for minimal lag in data visualization.

Figure 3.5 illustrates a mock-up of grid segmentation used to track user movement across zones.

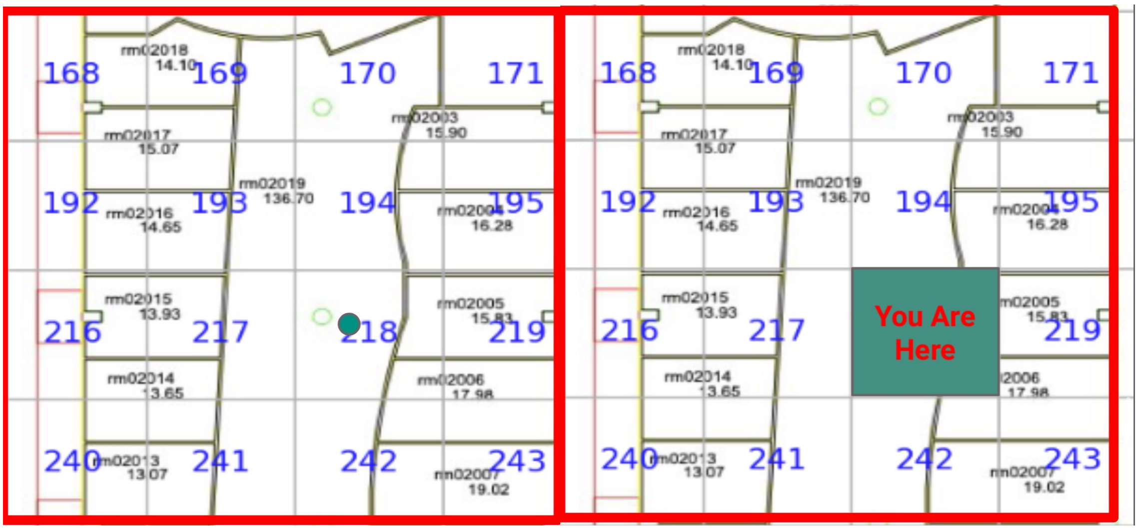

Figure 3.5 Mock-Up Grid Segmentation

3.5.5 Back-end

The back-end handles HTTP requests, supports scalable data access, and includes a RESTful API developed with Go and Python. PostgreSQL manages structured spatial data, while Docker containerization ensures deployment consistency.

Figure 3.1 depicts the relational database structure.

Deploying the Service

A self-hosted backend server using Coolify and Cloudflared zero trust ensures secure deployment. Coolify simplifies environment setup and log management, while Cloudflared masks the backend and database, enhancing security.

CHAPTER 4
RESULT

4.1 Data Collection Results

The dataset used in this study consisted of 5,442 data points collected across multiple runs, with 1,510 unique BSSID columns. These data points were manually gathered by the students involved in the project. A key decision during preprocessing was to retain all WiFi BSSIDs, even if some appeared as noise. The rationale behind this approach was to preserve the full spectrum of WiFi signals in the environment, as dropping potentially noisy data could have excluded valuable signals, which might influence the model's accuracy.

The distribution of the data included varying signal strengths (RSSI values) from different access points across multiple floors, ensuring a diverse set of conditions for model evaluation. The team was cautious about using data cleaning techniques, such as Lasso reduction, as these might have removed important signal patterns, thus affecting the representation and analysis of the results.

4.2 IPS Accuracy Results

In the following section, we evaluate the accuracy of various machine learning models used for location prediction based on the collected RSSI data, first without and then with the inclusion of latitude, longitude, and altitude data. It is important to note that the geographical coordinates were recorded during data collection for benchmarking purposes and were not originally considered as part of the independent variables for model prediction.

4.2.1 Model Performance without Longitude and Latitude

In this experiment, we assessed the performance of different models based only on RSSI values from the WiFi access points, excluding any geographical data. The results are summarized as follows:

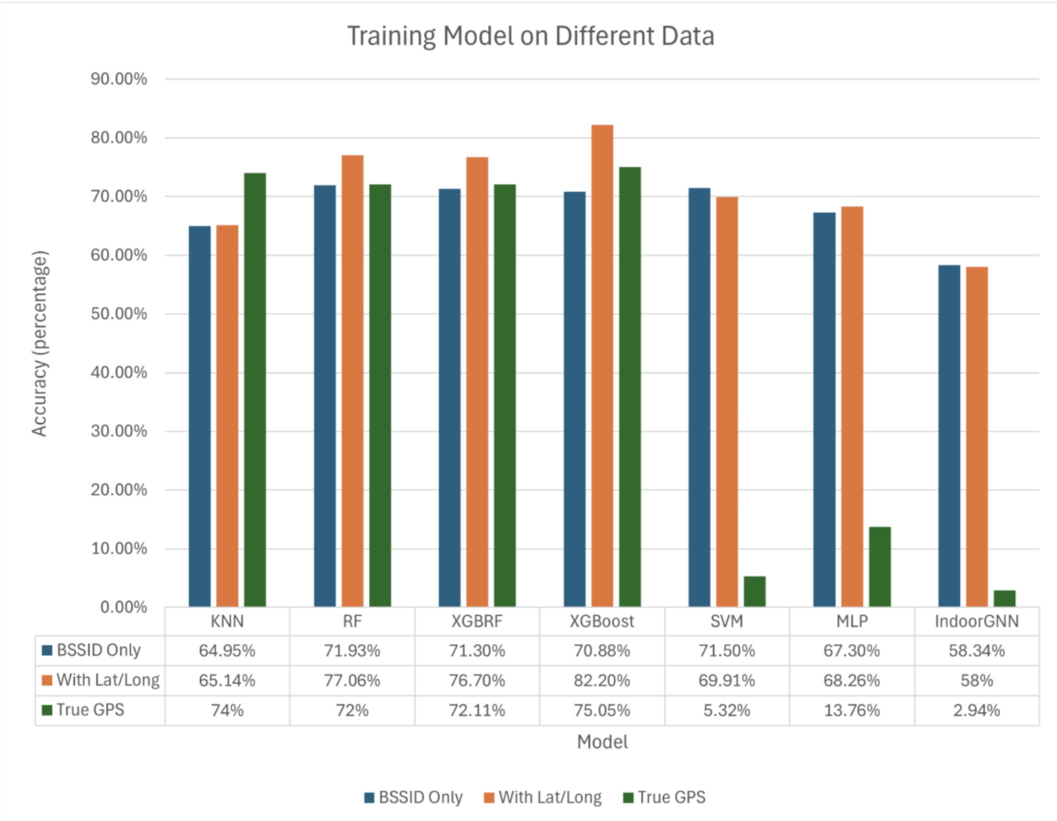

Figure 4.1 ML Model performance without geographical data

- **K-Nearest Neighbors (KNN, K = 22):** Achieved 64.95% accuracy, showing limited success in a multi-floor environment.
- **Random Forest Classifier (n = 96):** Improved performance to 71.93%, demonstrating the effectiveness of ensemble methods.
- **XGBoost RF Classifier (n = 15):** Achieved 71.30% accuracy, showing that the random forest variant of XGBoost performed similarly to the Random Forest model.
- **XGBoost Classifier (n = 5):** Achieved 70.80% accuracy, slightly lower than the ensemble methods but still competitive.
- **Multilayer Perceptron (MLP):** Reached 67.30% accuracy, showing moderate performance, likely due to sensitivity to data scaling.
- **Support Vector Machine (SVM, C = 1.5):** Performed similarly to the Random Forest and XGBoost models, with 71.50% accuracy.
- **Indoor Graph Neural Network (GNN):** Underperformed with 58.34% accu-

racy, likely due to the complexity of modeling the graph structure in a multi-floor environment.

4.2.2 Model Performance with Longitude, Latitude, and Altitude

The inclusion of latitude, longitude, and altitude data, though not initially intended as features for prediction, was tested to explore if geographical coordinates might improve model performance. The results were as follows:

- **K-Nearest Neighbors (KNN, K = 22):** A marginal improvement to 65.14% accuracy, suggesting a slight benefit from including geographical data.
- **Random Forest Classifier (n = 81):** Performance increased substantially to 77.06%, demonstrating the advantage of incorporating geographical data alongside RSSI.
- **XGBoost RF Classifier (n = 93):** Achieved 76.70% accuracy, benefiting similarly from the additional data.
- **XGBoost Classifier (n = 51):** The standalone XGBoost model reached 82.20% accuracy, the highest accuracy achieved across all models tested, suggesting that geographical data can be a useful complement to RSSI-based predictions.
- **Multilayer Perceptron (MLP):** Achieved a modest improvement to 68.26% accuracy.
- **Support Vector Machine (SVM, C = 1.5):** Experienced a slight drop in performance with geographical data, achieving 69.91% accuracy.
- **Indoor Graph Neural Network (GNN):** Continued to underperform with 58.00% accuracy, showing no significant improvement from adding geographical features.

4.2.3 Comparison with GPS-Based Models

Additionally, we compared the performance of models trained with GPS data, including latitude, longitude, and altitude, to provide a benchmark for the potential improvement in location accuracy. The results for GPS-based models were as follows:

- **K-Nearest Neighbors (KNN):** Achieved 74.00% accuracy.
- **Random Forest Classifier (RF):** Achieved 72.00% accuracy.

- **XGBoost RF Classifier (XGBRF):** Achieved 72.11% accuracy.
- **XGBoost Classifier (XGBoost):** Achieved 75.05% accuracy.
- **Support Vector Machine (SVM):** Achieved 5.32% accuracy, showing poor performance with GPS data.
- **Multilayer Perceptron (MLP):** Achieved 13.76% accuracy.
- **Indoor Graph Neural Network (IndoorGNN):** Achieved 2.94% accuracy, significantly underperforming even with GPS data.

These results were included in the same graph to compare the performance of WiFi-based models with the models trained using GPS data.

4.3 Summary of Results

The data collected in this study allowed for a thorough analysis of WiFi signal characteristics and their relationship with location prediction accuracy. By retaining all WiFi BSSIDs and avoiding data cleaning techniques that could eliminate potentially useful signals, the study maintained a robust and comprehensive dataset.

The results showed that including latitude, longitude, and altitude data from mobile phones (which were logged during data collection for benchmarking purposes) led to performance improvements in most models. The XGBoost Classifier, which achieved 82.20% accuracy, performed the best among all models tested. While Random Forest and XGBoost RF also showed strong performance, the Indoor GNN model faced challenges and underperformed in this multi-floor environment.

In comparison, the models trained with GPS data performed better than the WiFi-based models, with the XGBoost Classifier achieving 75.05% accuracy. However, GPS-based models such as SVM, MLP, and Indoor GNN did not perform well, suggesting that while GPS data can improve performance, it does not guarantee high accuracy in this scenario.

CHAPTER 5
DISCUSSIONS

5.1 Data Collection

For the initial data collection method, two devices were used. An ROG Ally 3 and an ROG Ally 6. They are both android phones, with the most notable difference being the Wi-Fi adapter. However; that difference is minimal and does not create a standard error. In conjunction with the devices inbuilt GPS module a Geolocator flutter package was used to collect the latitude, longitude and altitude values of the person's position. We weight the amount of data we collect each day into runs, with one run consisting of going from the first grid of a floor to the last grid of the floor (grid 1 to grid 8). A normal day of data collection consists of 8 runs per each of the 5 floors, on each device. This usually takes us 2 - 3 hours, the time taken can increase if we take . The datapoints we collect from this method lets us use 99% of the datapoints collected, with the 1% being datapoints collected via human error which are then deleted or tagged in the database as incorrect. On average, we collect 640 datapoints per day, 320 datapoints per phone.

5.2 Strength

The grid-based data collection method provides several strengths in our data gathering process. Primarily, it offers straightforward collection and organization. Dividing floors into grids simplifies labeling because the large area each grid covers makes it easy to identify and record locations, reducing the risk of mislabeling. Furthermore, the use of two ROG Ally devices ensures redundancy and consistency, adding reliability to the data by verifying measurements from two sources. This method's structure helps mitigate the potential for significant errors in location data due to the limitations of the devices, allowing us to focus on refining data quality rather than on error correction.

Furthermore, the grid-based data collection method also allow the creation of heatmaps. Heatmaps provide developers with convenient tools for analysis and enable a deeper understanding of data distribution by visually representing data density across grids. As demonstrated in Figure 5.1 below.

Another significant benefit of this method is that the implementation did not require extensive data cleaning. This streamlined approach ensures that the model could function effectively with minimal preprocessing. Chapter 4 further demonstrates that in similar settings, radio signal-only implementations can perform classification tasks as effectively as GPS-based implementations.

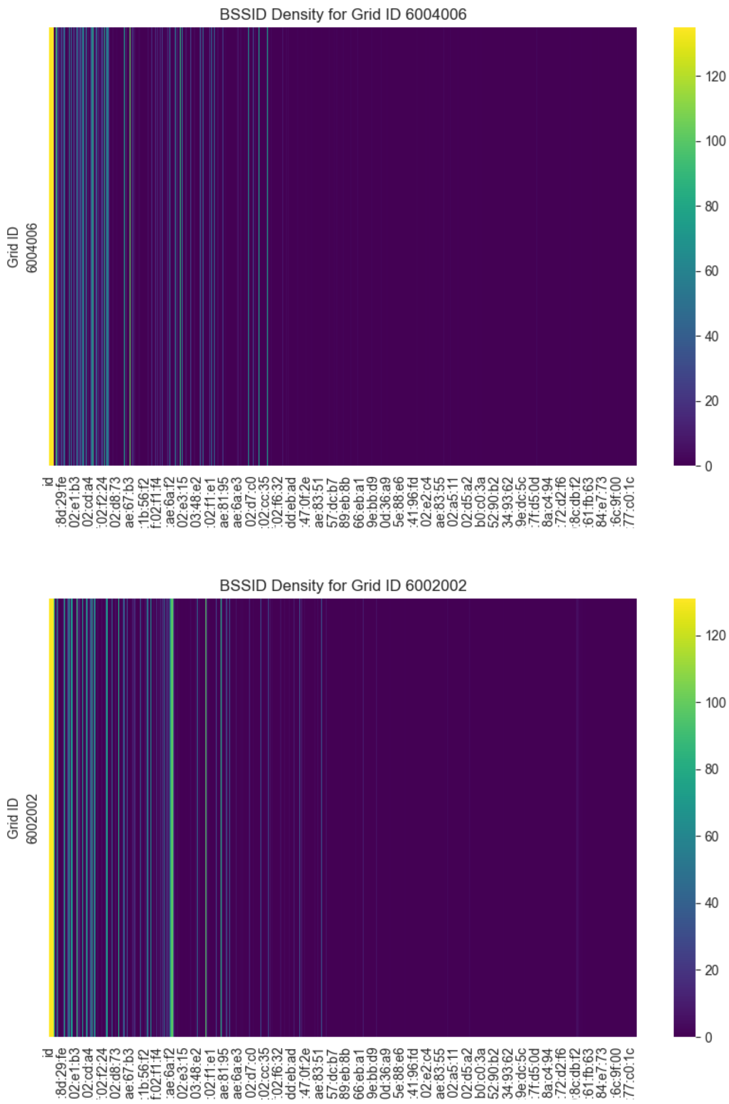

Figure 5.1 Heatmaps Generated for Grid IDs Ranging from 6001001 to 6005008

5.3 Weakness

However, this method has a few notable weaknesses. For instance, using grids means that data points represent larger areas rather than precise coordinates. This could lead to inaccuracies in applications that require high spatial precision since there is a trade-off between grid size and positional accuracy. Additionally, variations in device performance or environmental factors, such as interference with GPS signal quality on different floors, may introduce minor inconsistencies between data collected from different devices or runs. This limitation is partly addressed by tagging inaccurate data points, yet it still impacts the reliability of the overall dataset.

Although heat maps provide a helpful visual representation of data distribution, they reveal certain limitations in our approach. Specifically, the data collection process includes significant noise. Many access points used in training data rarely appear consistently in specific grids. This meant that, while we collected data and tried to overcome poor accuracy by adding more datapoints, the collection time could have been cut had there been data filtering in regard to access points used for training.

5.4 Model Performance

Various machine learning models were evaluated for their suitability in this context. IndoorGNN, claims to perform well at generalizations based on limited data. However, it struggled in our blind testing setting. While IndoorGNN could potentially succeed if the BSSIDs were tightly controlled, this was not confirmed in this particular study. On the other hand, Random Forest consistently emerged as the best-performing model. Despite being outpaced by XGBoost in some scenarios, Random Forest showed consistent performance throughout our data collection cycles, maintaining reliability as more data was collected.

5.5 Application

The findings of this study can help developers and property owners implement Indoor Positioning Systems more effectively and accurately. For property owners, particularly those renting out spaces, a zone-based IPS offers a practical solution without requiring extensive data collecting. Developers can design applications that collect all available BSSIDs within a property and use this data to create an IPS, eliminating much of the filtering typically required. This approach benefits from continuous usage, as daily data collection can incrementally improve the system's accuracy.

In terms of accuracy, the system provides reliable results to indicate whether users are in specific zones or general areas. This makes it particularly valuable for applications such as evacuation systems, where precise positioning is less critical than knowing the number of occupants in a given area.

CHAPTER 6
CONCLUSION

6.1 Conclusion

To conclude, this report demonstrates the feasibility of a multi-floor Indoor Positioning System using existing Machine Learning models and the data collected on the campus' existing WiFi infrastructure. As well as a comparison on the accuracy of the various existing ML algorithms in indoor positioning applications. Our finding shows that adding latitude and longitude data significantly enhanced the accuracy of most models. With XGBoost Classifier (n = 51) performing the best at the accuracy of 82.20%, Random Forest (n = 81) and XGBoost RF Classifier (n = 93) were also competitive at 77.06% and 76.70% respectively. All the while the Indoor GNN model underperformed. Note that no data preprocessing was involved except for data normalization in MLP and indoor GNN.

The grid-based approach allowed for efficient and reliable coverage while minimizing human error. Which leads to the retention of approximately 99% of the data collected daily. However, the lack of clear physical boundaries for each grid does cause periodic misclassification. The altitude measurements of the built-in GPS modules were unsatisfactory leading to finding an alternative method to approximate altitude. With our result at hand, it is suggesting that the existing WiFi infrastructure can support reliable indoor positioning without high hardware costs or data-cleaning techniques.

6.2 Future Work

While our findings indicate the viability of using the campus's existing WiFi infrastructure for multi-floor indoor positioning, several areas could further enhance accuracy and reliability in future research. First, investigating data cleaning techniques could help refine the model training process and improve the system's overall performance by reducing noise and irrelevant data.

Additionally, reducing the grid size could be explored to see the effect on accuracy; a smaller grid could potentially lead to more precise location estimates, though at the cost of increased computational demand. Another promising area would be transitioning from classification to regression-based methods to allow for continuous position estimation, which may enable smoother location tracking across grid boundaries.

REFERENCES

[1] What is a rest api and how can it benefit your business? URL https://mailchimp.com/resources/rest-api/.

[2]

[3] L. Bibbò, R. Carotenuto, and F. Della Corte. An overview of indoor localization system for human activity recognition (har) in healthcare. *Sensors*, 22(21), 2022. ISSN 1424-8220. doi: 10.3390/s22218119. URL https://www.mdpi.com/1424-8220/22/21/8119.

[4] D. Christodoulou. Developing an indoor localisation and wayfinding app for a university library.

[5] D. Csík, □. Odry, and P. Sarcevic. Comparison of rssi-based fingerprinting methods for indoor localization. In *2022 IEEE 20th Jubilee International Symposium on Intelligent Systems and Informatics (SISY)*, pages 000273--000278, 2022. doi: 10.1109/SISY56759.2022.10036270.

[6] Z. Ezzati Khatab, V. Moghtadaiee, and S. A. Ghorashi. A fingerprint-based technique for indoor localization using fuzzy least squares support vector machine. In *2017 Iranian Conference on Electrical Engineering (ICEE)*, pages 1944--1949, 2017. doi: 10.1109/IranianCEE.2017.7985373.

[7] Y. Freund, R. Schapire, and N. Abe. A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence*, 14(771-780):1612, 1999.

[8] H. T. Gidey, X. Guo, K. Zhong, L. Li, and Y. Zhang. Ohettlal: An online transfer learning method for fingerprint-based indoor positioning. *Sensors*, 22:38, 2022. doi: https://doi.org/10.3390/s22239044.

[9] M. S. Ifthekhar, N. Saha, and Y. M. Jang. Neural network based indoor positioning technique in optical camera communication system. In *2014 International*

Conference on Indoor Positioning and Indoor Navigation (IPIN), pages 431--435, 2014. doi: 10.1109/IPIN.2014.7275513.

[10] J. Jun and C. Kim. Robust camera calibration using neural network. In *Proceedings of IEEE. IEEE Region 10 Conference. TENCON 99.'Multimedia Technology for Asia-Pacific Information Infrastructure'(Cat. No. 99CH37030)*, volume 1, pages 694--697. IEEE, 1999.

[11] J. Link, P. Smith, N. Viol, and K. Wehrle. Footpath: Accurate map-based indoor navigation using smartphones. pages 1--8, 09 2011. ISBN 978-1-4577-1805-2. doi: 10.1109/IPIN.2011.6071934.

[12] N. A. Maung Maung, B. Y. Lwi, and S. Thida. An enhanced rss fingerprinting-based wireless indoor positioning using random forest classifier. In *2020 International Conference on Advanced Information Technologies (ICAIT)*, pages 59--63, 2020. doi: 10.1109/ICAIT51105.2020.9261776.

[13] A. Nessa, B. Adhikari, F. Hussain, and X. N. Fernando. A survey of machine learning for indoor positioning. *IEEE Access*, 8:214945--214965, 2020. doi: 10.1109/ACCESS.2020.3039271.

[14] A. Satan. Bluetooth-based indoor navigation mobile system. pages 332--337, 05 2018. doi: 10.1109/CarpathianCC.2018.8399651.

[15] R. Vishwakarma, R. B. Joshi, and S. Mishra. Indoorgnn: A graph neural network based approach for indoor localization using wifi rssi. In V. Goyal, N. Kumar, S. S. Bhowmick, P. Goyal, N. Goyal, and D. Kumar, editors, *Big Data and Artificial Intelligence*, pages 150--165, Cham, 2023. Springer Nature Switzerland. ISBN 978-3-031-49601-1.

[16] L. Xuanmin, Q. Yang, Y. Wenle, and Y. Fan. An improved dynamic prediction fingerprint localization algorithm based on knn. In *2016 Sixth International Conference on Instrumentation & Measurement, Computer, Communication and Control (IMCCC)*, pages 289--292, 2016. doi: 10.1109/IMCCC.2016.107.

[17] Q. Zhao, F. Wang, and Z. Sun. Using neural network technique in vision-based robot curve tracking. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3817--3822. IEEE, 2006.