Northumbria University

Computing and Digital Technologies Project

Enhancing User Positioning: A Fusion of Wi-Fi and IMU Sensors with Support Vector Regression

Author:
Elvis Tony
W22062155
(10500 words)

Supervisor:
Dr. Saber Farag

A Dissertation submitted in fulfilment of the requirements for the degree of MSc.

in

Artificial Intelligence Technology

January 2024

Northumbria University
NEWCASTLE

Declaration of Authorship

I, Elvis Tony Raphael KOLLANNUR, declare that this thesis titled, 'Enhancing User Positioning: A Fusion of Wi-Fi and IMU Sensors with Support Vector Regression' and the work presented in it is my own. I confirm that this work submitted for assessment is my own and is expressed in my own words. Any uses made within it of the works of other authors in any form (e.g., ideas, equations, figures, text, tables, programs) are properly acknowledged at any point of their use. A list of the references employed is included.

Signed: **ELVIS TONY**

Date: **15th January, 2024**

Abstract

When GPS signals are missing or unreliable, like in buildings and subways, the combination of Wi-Fi positioning technology and Inertial Measurement Unit (IMU) sensors offers a viable substitute for precise indoor navigation. An effective solution for improved indoor localization is the combination of Wi-Fi positioning technology and Inertial Measurement Unit (IMU) sensors in scenarios where GPS signals are unavailable or unreliable, especially in indoor spaces like buildings and subways. Wifi localization makes use of signals from neighbouring access points to pinpoint the user's location, while IMU sensors record motion and rotational data. Combining these two complementary technologies yields significant benefits in confined and complex interior environments. It allows users to move around buildings with ease and supports real-time tracking. People may travel and explore indoor places more efficiently by combining IMU and Wi-Fi data, which successfully solves the problems that standard GPS-based navigation systems in such demanding situations provide.

The goal of this research project is to utilise IMU and Wi-Fi data for reliable location estimation by applying machine learning, the Support Vector Regression (SVR) algorithm. Based on the collected sensor data, the SVR algorithm—which is renowned for its capacity to model intricate relationships—is used to forecast changes in GPS locations. The effectiveness and benefits of using SVR for location estimation are discussed through a comparison of the machine learning method and traditional mathematical trajectory computations.

This study aims to improve outdoor and indoor navigation systems' accuracy and dependability by integrating SVR, especially in situations where GPS signals are erratic or nonexistent. In order to develop user positioning algorithms, the study attempts to show how effective the machine learning approach is at offering accurate and timely navigation solutions in difficult indoor conditions.

Acknowledgements

I extend my deepest gratitude to the Almighty, without whose grace and guidance this research would not have been possible. His infinite wisdom has been my constant source of strength and inspiration.

I would like to express my sincere appreciation to my Project Supervisor, Dr. Saber Farag, whose invaluable guidance, expertise, and unwavering support have been instrumental in shaping this research. His dedication to academic excellence has been a guiding light throughout this journey.

My heartfelt thanks go to my family for their unwavering support and understanding. Their encouragement and love have been my pillars of strength, providing me with the motivation to overcome challenges and pursue this academic endeavor.

I express my gratitude to my colleagues and well-wishers for their collaborative spirit and constructive feedback, which enriched the quality of this work. The camaraderie and shared enthusiasm made this research an enriching and rewarding experience.

Special thanks are extended to Dr. Rose Fong for her exceptional guidance throughout the dissertation period. Her expertise and insightful concepts have been invaluable in refining the quality of this research.

I am also grateful to Dr. Abdul Salih, the Program Leader, for his visionary leadership and encouragement.

To everyone who has played a role, big or small, in this academic journey, I am truly thankful. This achievement is a collective effort, and I am blessed to have such a supportive network.

Contents

List of Figures

Abbreviations

AI	**A**rtificial **I**ntelligence
AP	**A**ccess **P**oint
AR	**A**ugmented **R**eality
DL	**D**eep **L**earning
DNN	**D**eep **N**eural **N**etwork
GNSS	**G**lobal **N**avigation **S**atellite **S**ystem
GPS	**G**lobal **P**ositioning **S**ystem
IMU	**I**nertial **M**easurement **U**nit
KNN	**K**-**N**earest **N**eighbors
LAN	**L**ocal **A**rea **N**etwork
ML	**M**achine **L**earning
RSS	**R**eceived **S**ignal **S**trength
SVM	**S**upport **V**ector **M**achine
SVR	**S**upport **V**ector **R**egression
WiFi	**Wi**reless **Fi**delity
WLAN	**W**ireless **LAN**
UI	**U**ser **I**nterface
UK	**U**nited **K**ingdom
CSV	**C**omma-**S**eparated **V**alues
MPV	**M**inimum **P**roduct **V**iable
IEEE	**I**nstitute of **E**lectrical and **E**lectronics **E**ngineers
VR	**V**irtual **R**eality
RPE	**R**elative **P**osition **E**rror
ATE	**A**utomatic **T**est **E**quipment
UAV	**U**nmanned **A**erial **V**ehicle
BSSID	**B**asic **S**ervice **S**et **ID**entifier
MAC	**M**edia **A**ccess **C**ontrol
RMSE	**R**oot **M**ean **S**quare **E**rror

Chapter 1

Introduction

London is an international metropolis that is well-known for having a wide variety of transport options to meet the demands of its approximately 9 million inhabitants that live there. One of the most extensive and sophisticated urban transport networks in the world is made up of the city's complex network of buses, trains, trams, and the well-known London Underground (Wright, 2022). Meeting the needs of a high-density population that primarily depends on effective and connected modes of transportation requires a variety of transportation options.

With more than 270 stations spanning a vast network of lines, the London Underground, also known as the Tube, is the hub of the city's public transport system (Wright, 2022). In addition, thousands of bus lines cover the entire city, linking even the most isolated neighbourhoods (Wright, 2022). In addition, the Overground and Docklands Light Railway enhance suburban accessibility, guaranteeing a well-connected Greater London area.

1.1 Background

With more than 9 million residents, dependable and efficient mobility is essential to daily living and not just a convenience. The dense population of London highlights the necessity of an efficient public transport infrastructure in order to alleviate traffic jams, lessen the negative effects on the environment, and improve overall urban mobility (Statistics, 2022).

However, mobility is more than just efficiency and speed. It involves tying the several neighbourhoods of the city together to create a smooth, accessible network. TfL is aware of this, as seen by its commitment to providing accessible solutions like the well-known double-decker buses and the Oyster card system. They understand that diversity is necessary to keep London vibrant.

Naturally, no system is flawless. The transport tapestry of London has unique difficulties. Even seasoned commuters might be challenged by congestion jams, and figuring out the Tube's complex routes can sometimes feel like cracking a riddle. Nevertheless, a spirit of constant progress is evident despite these setbacks (Ferrari et al., 2014).

Digital devices can also be a distraction for commuters. Reports show that 15% of travelers face this issue. In the past year, passengers missed their stops a staggering 29 million times, with tourists, the elderly, and children finding city commuting particularly challenging. Buses, trains, trams, and the London Underground are the most common modes of travel in the UK, and missing stops is a common issue, often blamed on commuters nodding off during their journey (Innes, 2013).

1.2 Problem Domain

One of the main factors to this confusion lies in the complexity of London's transport system. Additionally, due to frequent maintenance projects, regular changes affect the routes and schedules which can be perplexing for those travellers who are unfamiliar with the modes of transportation available. The high traffic & gridlock on the roadways in London are another factor contributing to buses and public road transportation being delayed. Due to these delays, buses in London regularly miss their planned schedules, which can make the public transport unreliable in the eye of the working class.

Smart Navigation apps have become an integral part of an average commuter's toolkit. Google Maps, Apple Maps, Waze and Citymapper are some of the most popular tools that offer navigation services (Crozet and Coldefy, 2021, Tavmen, 2020). These apps provide personalized journey planning that helps users navigate London's complex transportation network effortlessly.

There are also special assistance programs in place to assist tourists, children and the elderly. Dedicated staff members are stationed at strategic transportation hubs to offer assistance, and educational resources are provided to make travelling easier for individuals who are unfamiliar with the city's public transit system.

In addition, the transportation authorities have worked with academic institutions to create interesting and instructional resources regarding the public transit network. The purpose of these materials is to help children and the elderly learn how to use the different types of transportation that are accessible in London. These resources are sent to schools and community centres in efforts to educate the population.

Another major factor contributing to this issue is the improper signage and faulty electronic alert systems which impact the passenger's confidence to know when to get off some bus and train routes. Inaccessibility on some buses and railways might result non passengers missing stops and being lost. It can be challenging for passengers with disabilities or those who are unfamiliar with the route to know when to get off some buses since they lack auditory announcements or visual displays of next stops.

Digital Devices can also be responsible as a distraction for the commuters. Innes (2013) highlights a statistic that suggests fifteen percent of travellers face this issue. Passengers have missed their stops 29 million times over the past year, according to their report. Commuting across the city can be challenging especially for tourists, the elderly and children.

Buses, Trains, Trams, and Tubes comprises the most common modes of travel in the United Kingdom. Most often, travellers are seen missing their stop and regretting their quick nap during their commute.

1.2.1 Proposed Concept

This research suggests a solution that uses smartphone sensors paired with machine learning algorithm to estimate a user's location, regardless they are above or below ground. In the current era, the spread of wireless networks across urban areas also encourage the use of leveraging Wireless networks to estimate the location of the user. With the fusion of both the Inertial and Wireless Sensor data, the location can be realised on a graphical map. For green and smart futuristic cities, location awareness is essential since it makes it possible to comprehend high-interest areas and how people interact with their surroundings. This research aims to encourage the use of IMU sensors paired with wireless networks to estimate device locations.

Chapter 2

Literature Review

The literature review chapter evaluates previous studies and scholarly publications in the topic critically. It combines key concepts, theories, and procedures to lay the groundwork for the current project. This chapter identifies knowledge gaps and establishes the theoretical framework that will guide the investigation.

Public transportation is a crucial aspect of urban living. Transportation allows the movement of large number of people efficiently and sustainably. In the recent years, many new additions have been made to the transport system to improve the speed and ease of travel. Although they enhance ease of access, it also poses as a complexity to senior citizens, children, and tourists. Innes (2013) also mentions in their article regarding how digital distractions such as smartphones and devices cause travellers to miss their stops.

2.1 GNSS

Over the years, multiple location estimation techniques have been developed to improve accuracy. One of the most commonly used strategy in vehicle tracking is using GPS based solutions which are also known as Global Navigation Satellite System (GNSS). GNSS provides location and time information anywhere on the planet. Zhuang et al. (2023) highlights in their paper titled "Multi-sensor integrated navigation/positioning systems using data fusion: From analytics-based to learning-based approaches" about the limitations of a GNSS only solution and instead proposes a hybrid approach using multiple sensors. Since most GPS satellites are note geo-stationary satellites, the range can vary significantly across different locations.

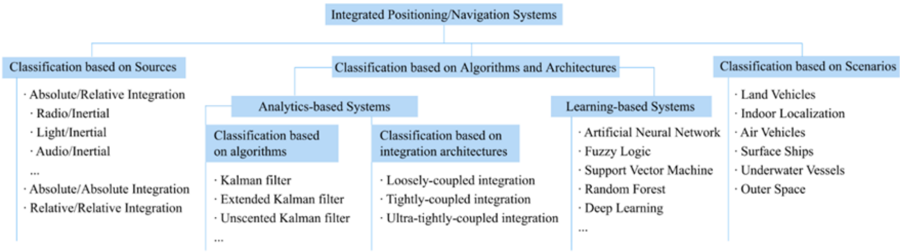


Figure 2.1: Navigating Methods

Figure 2.1 shows the various navigation systems implemented based on their applications. Although, GNSS provides an industry accepted accuracy, its versatility is questionable. Urban metropolitans often have subways and underground transport which could make GNSS based solutions hard to implement. Toy et al. (2022) supports this premise with the scenario of location tracking indoors (where GNSS signal is weak or non-existent) IMU sensors are employed to estimate the location.

Hussain et al. (2021) in their paper highlights the potential vulnerabilities of satellite-based positioning systems in urban canyons where signals are prone to scattering and reflection. The paper mentions the significant impact of Multi path (MP)/ Non Line Of Sight (NLOS) on the quality of the signal which leads to regular loss of signal and estimation errors greater than 10 meters.

2.2 WiFi based Localization

WiFi is a standard technology used for setting up a Wireless Local Area Networks (WLANs) in both home and office settings and officially termed as IEEE 802.11. The two primary bands in which WiFi networks propagate are the 2.4 GHz and 5 GHz bands (Faragher and Harle, 2015, Kotaru et al., 2015).

Kim et al. (2018) discusses the use of Wi-Fi signals for indoor localization in environments where GPS signals cannot reach. This is accomplished by gathering a database of location fingerprints and measuring received signal strengths (RSSs) from wireless network equipment. The closest match between a device's RSS value and the fingerprints of known spots in the database can be utilised for estimating a device's position.

Modern Wi-Fi fingerprinting systems take a hierarchical approach and estimate a device's building, floor at a single instance. Due to less parameter adjustment and improved scalability, machine learning techniques, particularly deep neural networks (DNNs), offers intriguing solutions to Wi-Fi fingerprinting.

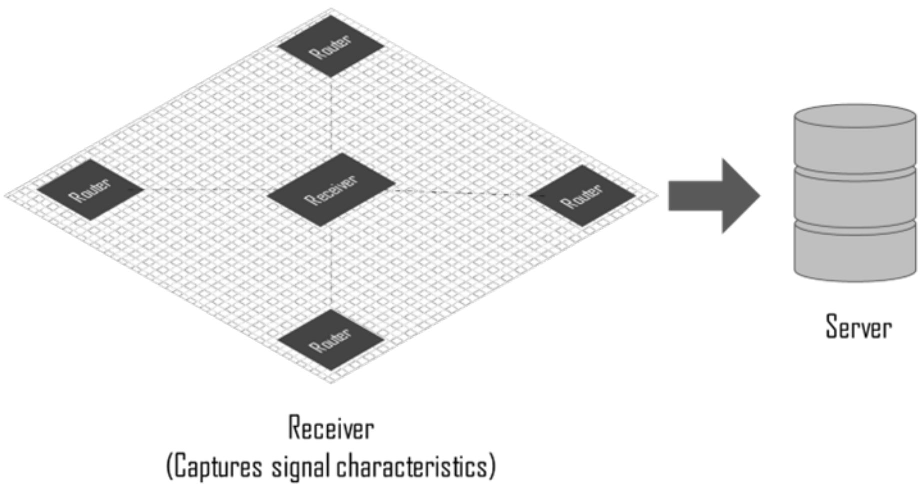


Figure 2.2: Wireless Device in Wireless network zones

Kim et al. (2018) also explains a feasibility analysis of the author's Campus Visitor Service System's use of DNNs for hierarchical building and floor classification and floor-level location estimation. The researchers used realtime RSS data collected at one of the buildings on the campus to develop a prototype indoor localization system for floor-level location estimate. Location awareness is a critical technology for green and smart futuristic cities as it enables the understanding of high interest regions and how people interact with the environment.

Mok and Retscher (2007) also agrees that indoor environments provide challenging scenarios and GNSS based solutions are often unreliable or even unavailable. With the increase of skyscrapers and high-rise buildings, the possibility of using a GNSS based location system has reduced due to the layers of concrete and cement. The authors propose a system that integrates GNSS with Wireless networks and compares two approaches that are Wi-Fi trilateration and Wi-Fi Fingerprinting. Using this approach, a positional accuracy of up to 3 metres was achieved.

2.3 Inertial Measurement Unit (IMU)

IMU oriented approaches involve the use of multiple sensors that includes Accelerometer, Gyroscope and Magnetometer. In many scenarios, GNSS based solutions may not be effective and may be more power intensive.

Toy et al. (2022) highlights in their paper about the importance of accurate positioning for autonomous vehicles and the use of IMU dead reckoning as an alternative to GPS-based systems. There are various methods that employ sensor fusion techniques for estimating position, where most methods making use of a Kalman filter. The proposed method involves creating two odometry values using angle and velocity data determined with the help of the IMU sensor data speed information from the vehicle Controller Area Network (CAN).

The paper demonstrates the effectiveness of the method proposed that works by comparing the odometer data (speed) with the actual expected value (ground truth) using RMSE measure, ATE (Absolute Trajectory Error) and RPE (Relative Pose Error). The paper also presents information about the performance with Noisy IMU sensor data which need to be filtered. Furthermore, the paper suggests that IMU dead reckoning can be a viable alternative to GPS-based systems in situations where GPS data is inaccessible or weak, and the proposed method can provide accurate positioning for autonomous vehicles. Dead reckoning is carried out with the aid of IMU sensors which extends the application to both indoors and outdoors.

According to a research conducted by Prikhodko et al. (2018), using a tactical grade gyroscope in an IMU module and the speedometer the researchers were able to estimate the attitude and velocity. The study concludes with a minimal error of about 30 meters which is comparable to the GPS based system with an accuracy of 15 meters.

A research conducted by Schmutz et al. (2020) explains how the use of IMU sensors can be used as an effective alternative to GNSS solutions in determining the speed of each stride of horses. The study judges two approaches: signal based approach and ML based approach on the basis of accuracy. The ML based approach showed an error less than 0.6 meters per second without the use of GPS signals to estimate the speed. The model was developed using one IMU unit to enhance the usability of the tool for indoor and outdoor applications.

The model was implemented using the SVM algorithm and can be transformed to support other motion based estimation. This paper shows a feasible approach in estimating a device's path and speed with the help of IMU sensors.

2.4 Similar Research

Li and Zhang (2022) presents a novel solution to the problem of Unmanned Aerial Vehicles (UAVs) placement in inside corridor environments, where traditional techniques based on GPS signals have limitations. To improve accuracy, the suggested approach makes use of WiFi and Inertial Measurement Unit (IMU) data.

A zone partition-based Weighted K Nearest Neighbours (WKNN) algorithm for improving WiFi positioning accuracy is one of the main elements of the suggested approach.

In order to optimise UAV placement, this algorithm divides the corridor into zones and gives WiFi access points (APs) different weights depending on where they are. Their paper offers a viable approach to UAV location in interior corridor environments. The suggested approach works better than existing methods and is distinguished by its accuracy, dependability, and flexibility to different indoor situations.

Jin et al. (2014) presents a technique for real-time WiFi fingerprinting placement using inertial measurement unit (IMU) readings. To tackle the RSS variation problem, the proposed IMU-assisted nearest neighbour selection algorithm selects out unnecessary reference points based on position prediction with IMU data. In a real-world indoor context, the suggested technique was assessed and compared to the standard K-nearest neighbours (KNN) selection and the IMU-based dead-reckoning location. The suggested technique had an average positioning error of 2.41 meters, whereas the KNN-based fingerprinting algorithm and the IMU-based dead-reckoning positioning had errors of 3.57 meters and 15.27 meters, respectively (Jin et al., 2014).

Since the introduction of various location-based services, indoor positioning has been growing quickly (Brida et al., 2021). A lot of study has gone into utilising WiFi networks to determine the whereabouts of users because of how affordable and widely available it is. It is common practice to locate fingerprints based on received signal strength (RSS) pattern matching.

2.5 Contribution

The proposed solution attempts to re-implement this similar approach in a different environment. The benefit of this implementation would promote the use of low-power sensors to work in synchrony to generate trajectory when GPS positioning is unavailable or too expensive. Using GPS sensors in devices can be power depleting each time it contacts the positioning satellite for information. Macealois (2023) mentions in their article, explaining how the frequency of GPS pings can have significant impact to the longevity of the device's battery.

The proposed solution makes use of lower range sensors that consume lesser battery and yields results not far from the GPS observations. Making use of sensors to improve local awareness, we can reproduce similar observations without the need for pinging satellites. Underground tunnels, Subways, bridges, buildings etc. have a major inaccessibility to GPS positioning, therefore making it viable candidates to implement this concept.

2.5.1 Selecting Sensor Sources

Inertial measurement sensors are found in many devices such as Smartphones, Smartwatches, Fitness bands, VR headsets, delivery robots etc. This research revolves around the use of smartphone sensors as the source of IMU readings along with GPS readings (whenever possible). Popular alternatives to gathering IMU data is using IMU based development boards such as Arduino UNO paired with an IMU extension. According to the research around various IMU sensor sources as a candidate for Binaural Head-Tracking by Franček et al. (2023), results indicated that the majority of smartphones have acceptable data output with negligible latency which make it suitable for capturing data in a efficient manner without the use of development boards.

Chapter 3

Concept & Methodology

The Concept & Methodology chapter defines the conceptual framework and describes the approaches chosen to tackle the problem discussed. It delves into the theoretical foundations, the reasoning behind methodological decisions, and the overall approach to data gathering, analysis, and interpretation.

3.1 Proposed Concept

By tracking a user's movement and changes in orientation within a building, IMU (Inertial Measurement Unit) sensors are able to record their path.

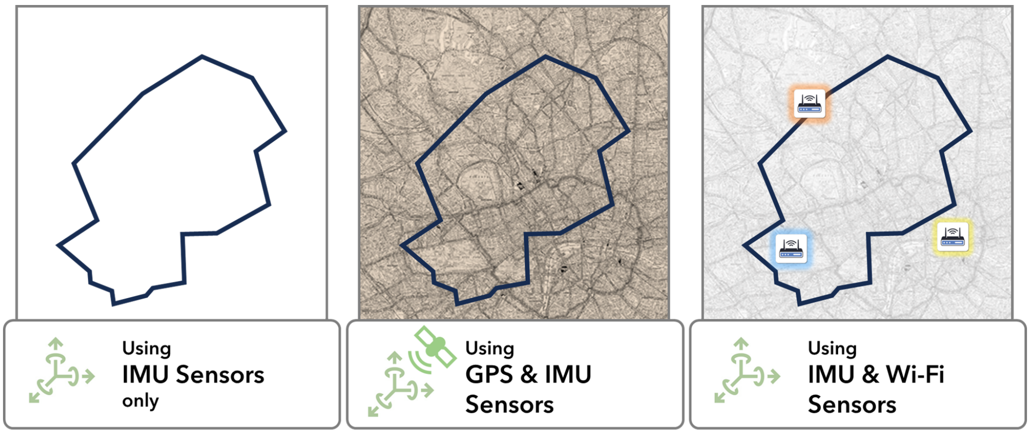


FIGURE 3.1: Combination of Sensors vs Proposed Concept

While pairing an IMU with a GPS unit is a popular method for outdoor navigation, it is not appropriate for subterranean or indoor locations. Because GPS depends on satellite signals, which are frequently obstructed or attenuated by buildings, it is unreliable for tracking users inside of them as well as in subterranean subways and train systems. For the purpose of providing precise location information in such circumstances, the integration of IMU with alternative positioning technologies, such as WiFi fingerprinting or other indoor positioning techniques, becomes essential. IMU sensors are prone to drift and can collect inaccuracies over time, so while this provides a trajectory or path, it does not provide exact location information (Leitch et al., 2023).

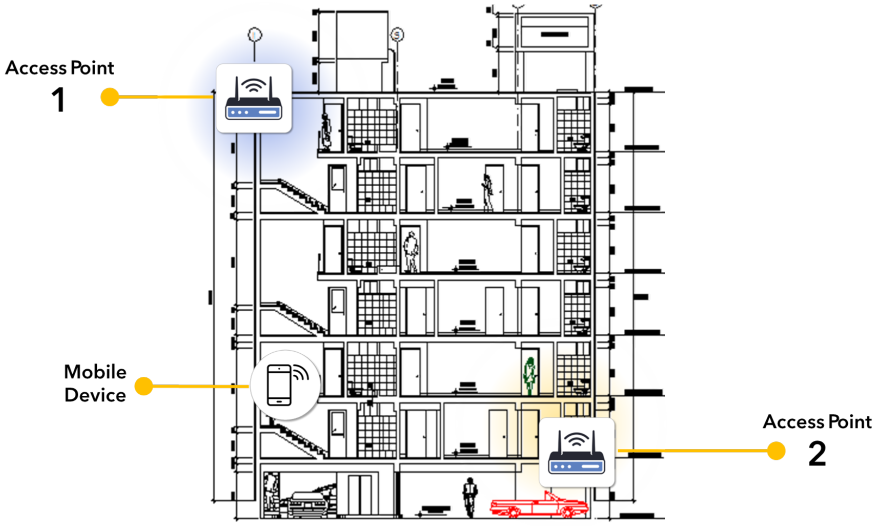


FIGURE 3.2: Use Case 1: Tracking User movement in commercial buildings

This problem can be fixed by pairing this track with known WiFi fingerprinted locations. Mapping the signal intensities of neighbouring WiFi access points at particular places within the building is known as WiFi fingerprinting (as seen in figure 3.2). The system determines the user's current position by comparing the path (recorded by the IMU) with the known WiFi fingerprints. Accurate indoor positioning and improved user comprehension of one's location on a building map are made possible by the combination of WiFi fingerprinting and IMU data. Li and Zhang (2022) explains in their paper about leveraging this concept to navigate drones in confined indoor spaces.

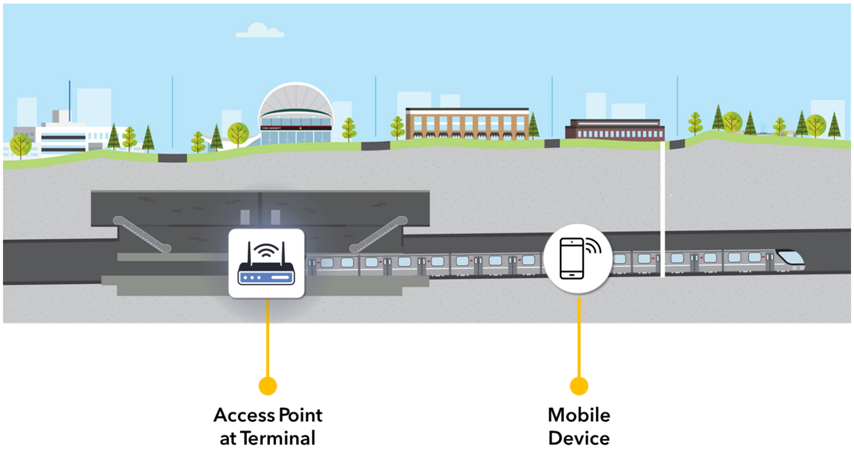


FIGURE 3.3: Use Case 2: Tracking User in underground tunnels

A similar approach can also implemented in subways and tunnels. A wireless network map can be constructed with the help of an initial probing of the route of the underground rails, which can then be used as a look up table for estimating the start and end points (as seen in Figure 3.3).

3.2 Data Sources

The data collected for this research includes readings from Inertial measurement unit (IMU), GPS Coordinates (for training the model and hold-out validation) and Wireless Networks (fingerprinting).

3.2.1 Inertial Measurement Unit

- **Accelerometers** are used to measure linear motion, including direction and speed variations. The accelerometer tracks changes in the user's position and provides information on their step count, speed, and direction.
- **Gyroscope** measures orientation and rotational motion. It assists in identifying if a user is spinning in position, turning, or changing direction.
- A **Magnetometer** is a device that measures variations in the Earth's magnetic field and is used to find out where the user is in relation to the magnetic north of the planet.

3.2.2 Wireless Networks

Wi-Fi, Bluetooth, and other wireless technologies, as well as wireless networks, are now essential parts of contemporary communication systems. These technologies allow devices to interact without physical wires by offering flexible and convenient communication. While Bluetooth is frequently used for close-quarters device communication, Wi-Fi is specifically utilised for internet access.

3.2.3 WiFi Networks

Wi-Fi runs in the 2.4 GHz and 5 GHz frequency ranges and is based on IEEE 802.11 specifications. It makes it possible for devices to wirelessly join a Local Area Network (LAN), giving them access to the internet and easing communication amongst other devices on the network. Access points (APs) that broadcast signals and client devices that connect to these access points make up Wi-Fi networks.

WiFi Fingerprinting

Wi-Fi Fingerprinting is the process of identifying host devices based on their device artifacts that includes output signals, channels, hardware IDs or signatures. Wi-Fi Fingerprinting can be performed with the help of two stages, Offline and Online stages. Wireless networks

A Wi-Fi fingerprint can include the following parameters:

- **AP Name**: The name used to broadcast the signal. It is often called as Wi-Fi Name.
- **AP Signal Strength**: The signal power that is being received to the client's device.
- **AP MAC Address**: The physical address of the access point that is a 2 digit hexadecimal value. This is also known as BSSID for some routers and access points.
- **AP Security Type**: Each Access Point can support multiple Security Standards. Popular standards include WPA2, WPA2-PSK and WEP.

Wi-Fi fingerprints helps identify unique networks that may have similar Access Point names. Although Access Point passwords can also be used to uniquely identify networks, this research concept aims to leverage these networks without attempting to connect to host.

3.3 Methodology

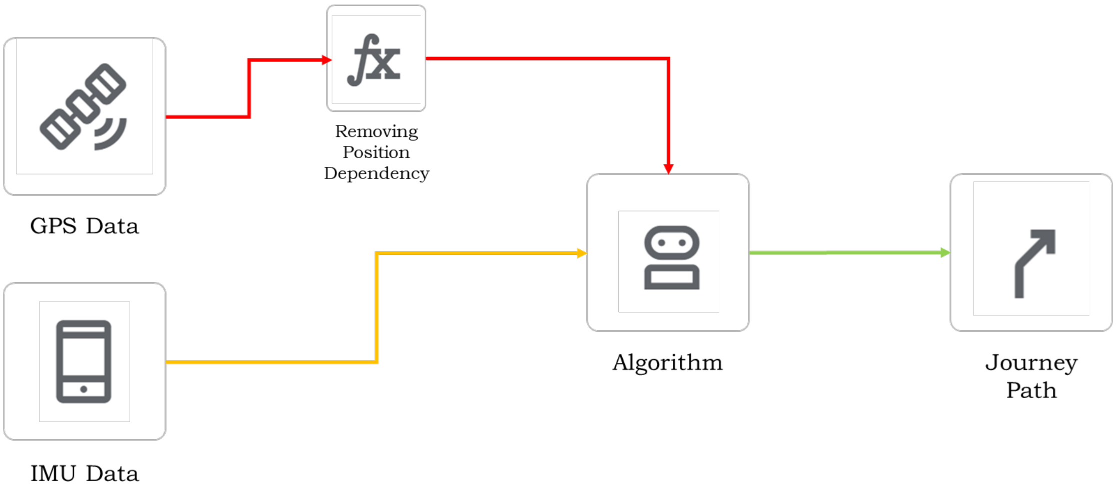


FIGURE 3.4: Proposed Methodology

Figure 3.4 shows how the Data from the IMU & GPS sensors are fed into the algorithm to generate a journey path. Given the research involves gathering data manually, the research process was broken down into stages to clearly define each process and its inner workings. The Stages are as follows:

1. Device selection
2. Data collection
3. Data pre-processing
4. Algorithm

3.3.1 Device Selection

In the previous chapter, it was shown that Arduino boards offer a compelling choice for gathering IMU sensor data. This was due to its developer friendly design and ability to upgrade based on the needs of the research.

Arduino UNO with IMU Board

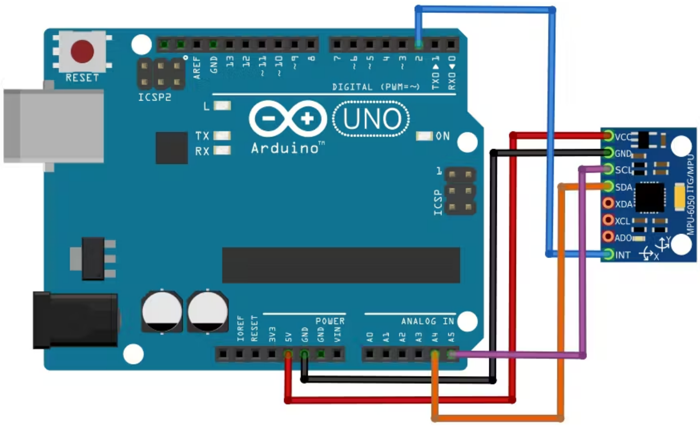


FIGURE 3.5: Arduino UNO with MPU 6050

Arduino UNO boards by default does not come equipped with IMU sensors, to allow the board to gather the sensor readings, an IMU extension board called MPU 6050 is connected via the serial pins (as seen in Figure 3.5).

Smartphone Sensors

Smartphones on the other hand are equipped with these sensors to enhance the user experience. Popular breakthroughs such as Augmented Reality, Motion tracking etc. rely on these sensors and are now part of almost all smartphones released to date.

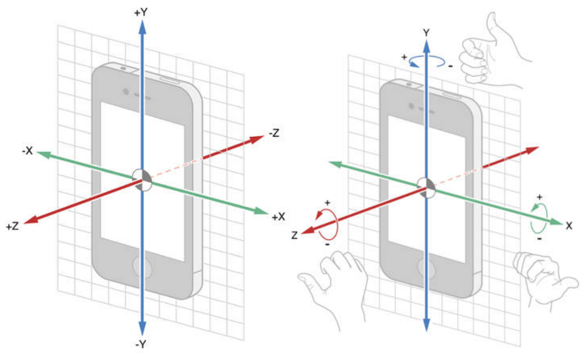


FIGURE 3.6: Smartphone Sensors — Coordinates

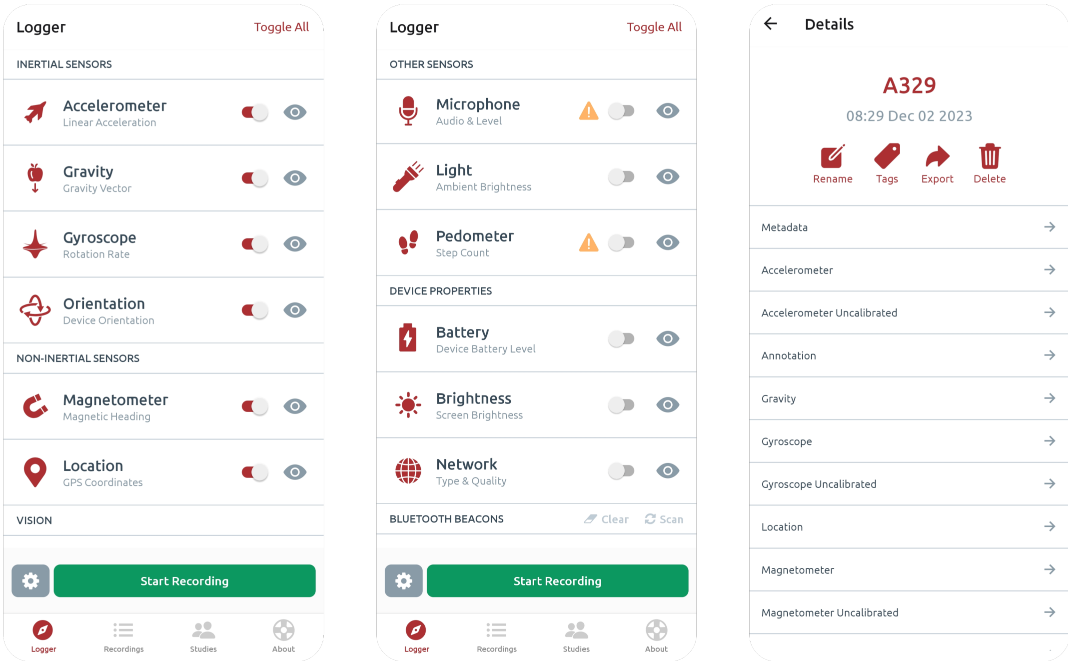
Using the IMU sensors on the smartphone can be cost effective since, an Arduino UNO board would indicate a single time purchase along with its essential accessories and would also require expertise handling the code. Figure 3.6 shows the position and movement coordinates that can be extracted from the on-board sensors.

3.3.2 Data Collection

The data required for the research was collected manually rather than relying on pre-existing datasets to understand the real world problems and issues that would entail when deploying a viable concept. The data points that need to be captured were explained in the previous section. To facilitate the data collection process from the smartphone, an interface must be used to communicate between the individual sensors and the device.

Smartphone Sensor Collection

To aid in the data collection capture, Choi (2022) a data scientist in the UK developed an android application that captures internal sensor readings with a user-friendly design. Figure 3.7 shows the intuitive UI that was used to record sensor data.



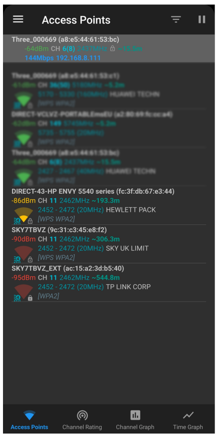
Available Sensors (page 1) Available Sensors (page 2) View Recordings & Data

FIGURE 3.7: Sensor Logger App

Using the application, the following sensors are recorded:

1. Accelerometer
2. Gyroscope
3. Magnetometer
4. Location

Along with the Inertial sensors, the Wireless networks are also scanned in frequent intervals to create a database of WiFi fingerprints and their locations. Figure 3.8 shows how the App gathers data and how it was parsed into a table using Python and pandas.



WIFI Analyser App

```
Time Stamp|SSID|BSSID|Strength|Primary Channel|Primary Frequency|Center
Channel|Center Frequency|Width (Range)|Distance|Timestamp|802.11mc|Security
2023/12/08-07:44:13|SWR WiFi|00:ee:ab:38:a9:10|-92dBm|1|2412MHz|1|2412MHz|20MHz
(2402 - 2422)|~393.7m|1186726761386|false|[ESS]
2023/12/08-07:44:13|SWR WiFi|00:ee:ab:a5:10:d0|-99dBm|6|2437MHz|6|2437MHz|20MHz
(2427 - 2447)|~872.3m|1186726761386|false|[ESS]
2023/12/08-07:44:13|#FreeStationWiFi|38:45:3b:18:bc:f8|-
72dBm|2|2417MHz|2|2417MHz|20MHz (2407 - 2427)|~39.3m|1186726761386|false|[ESS]
```

Exported Data
(Pipe separated values)

	Time Stamp	SSID	BSSID	Strength	Primary Channel	Primary Frequency	Center Channel	Center Frequency	Width (Range)	Distance	Timestamp	802.11mc	Security	Station
0	2023/05/05-18:20:16	TfL_Wi-Fi	00:09:90:01:0f:c5	-39dBm	1	2412MHz	1	2412MHz	20MHz (2402 - 2422)	~0.9m	17704440487	False	[ESS]	Bond street
1	2023/05/05-18:20:16	1234	b8:11:4b:fb:a3:c1	-64dBm	6	2437MHz	6	2437MHz	20MHz (2427 - 2447)	~15.5m	17696406985	False	[WPA2-EAP/SHA1+?-CCMP][RSN-EAP/SHA1+?-CCMP][ESS]	Bond street
2	2023/05/05-18:20:16	Ra1In3tP0l	b8:11:4b:fb:a3:c3	-61dBm	6	2437MHz	6	2437MHz	20MHz (2427 - 2447)	~11.0m	17696406989	False	[WPA2-PSK-CCMP][RSN-PSK-CCMP][ESS]	Bond street
3	2023/05/05-18:20:16		b8:11:4b:fb:a3:c4	-56dBm	36	5180MHz	38	5190MHz	40MHz (5170 - 5210)	~2.9m	17696337100	False	[WPA2-PSK-CCMP][RSN-PSK-CCMP][ESS]	Bond street
4	2023/05/05-18:20:16	MTR_Crossrail	b8:11:4b:fb:a3:c5	-56dBm	36	5180MHz	38	5190MHz	40MHz (5170 - 5210)	~2.9m	17696406982	False	[WPA2-PSK-CCMP][RSN-PSK-CCMP][ESS]	Bond street
5	2023/05/05-18:20:16	VodafoneWiFi	b8:11:4b:fb:a3:c7	-56dBm	36	5180MHz	38	5190MHz	40MHz (5170 - 5210)	~2.9m	17696406980	False	[WPA2-EAP/SHA1-CCMP][RSN-EAP/SHA1-CCMP][ESS]	Bond street

Parsed with Python & Pandas

FIGURE 3.8: WiFi Analyser App

3.3.3 Data Pre-processing

Data preparation is an important phase in our project, which is concerned with calculating vehicle trajectories using sensor data. Various obstacles must be overcome in order for the data to be ready for analysis and modelling.

To begin, missing values in sensor data, which might occur as a result of signal dropouts or sensor malfunctions, are handled using techniques such as interpolation or discarding null values to ensure data continuity. Second, data integration was carried out to aggregate data from multiple sensors, align timestamps, and create a consistent dataset.

Normalisation and scaling are used to handle differences in scales between sensors, ensuring that all features contribute equally. Outliers are discovered and handled to prevent skewing of analysis or model training, and redundant information was deleted to reduce dimensionality and computing complexity.

Time (frequency) synchronization was critical for ensuring that timestamps from various sensors match to the same point in time. Feature engineering is another standard preprocessing step that involves adding or manipulating features to improve the model's capacity to capture data patterns. Feature engineering in detail will be revisited later in the report.

3.3.3.1 Data Extraction & Merging

During the data preprocessing stage, the gathered information—which was originally saved in compressed zip files—was extracted and placed into various folders. The next step was to combine data collected from multiple sensors into a single file using Pandas, a data processing module in Python. After the combination, one notable finding was that GPS sensors updated less frequently than Inertial Measurement Unit (IMU) sensors, which resulted in null values that needed to be handled later.

3.3.3.2 Journey Start Assumption

Every trip that was documented had a trajectory by nature, which made it possible to plot latitude and longitude graphically. To ensure uniformity, a presumption approach was taken in light of the unknown start and finish points of these journeys where the starting point for all journeys was set at coordinates (0,0).

3.3.3.3 Position Dependency

GPS coordinates contain an initial position from which the journey begins. Training a model with this initial position would prove to be futile unless the initial position is also provided to the model. To simplify the learning algorithm and remove the position dependency from the GPS Position coordinates, consecutive differences between pairs of rows were performed.

$$\delta x = x(t) - x(t_{n-1})$$

$$\delta y = y(t) - y(t_{n-1)}$$

Where, $x(t)$ is current reading, $x(t_{n-1})$ is the previous reading. and dx is the change in position. Here, two new columns X_change and Y_change were created. These columns were then used in the subsequent stages below.

3.3.3.4 Handling Duplicate Values

Duplicate values, especially those originating from the IMU sensors, were collectively removed from the recorded file in order to further enhance the uniformity of the dataset. Managing null values in the GPS information (formed due to the different sampling frequency) was a complex task. Errors occurred in the first efforts using the forward fill (FFILL) and backward fill (BFILL) techniques. Interpolation then became a more sensible solution, gradually approximating missing values by using neighbouring data points. Remaining null values were either set to 0 or eliminated since they could not be interpolated.

Every observation trial underwent this thorough preparation method once, which ensured consistency and homogeneity throughout the dataset. By the time this procedure was finished, all of the sensor data that had been gathered had been combined into a single file, with each observation / recording file also intact.

3.3.4 Algorithm

To calculate the trajectory, there were two approaches that seemed to fit the criteria.

- Mathematical Approach
- Machine Learning Approach

3.3.4.1 Mathematical Approach

The mathematical approach that was adopted involved calculating the angular velocity at each instance and multiplying the scalar acceleration to previous position to produce the new position.

$$p(t) = \int \int a(t)\,dt\,dt$$

In discrete form, considering small time intervals Δt, the formula becomes:

$$p(t_n) = p(t_{n-1}) + v(t_{n-1})\Delta t + \frac{1}{2}a(t_{n-1})(\Delta t)^2$$

Here:

- $p(t_n)$ represents the position of the device at time t_n.
- $v(t_{n-1})$ is the velocity of the device at the previous time step t_{n-1}.
- $a(t_{n-1})$ is the acceleration of the device at the previous time step t_{n-1}.
- Δt is the time period between consecutive measurements.

3.3.4.2 Machine Learning Approach

To enhance the robustness of the system, various kinds of models were employed, that includes the Linear Regression, Polynomial Regression, Support Vector Regression (SVR), and lastly, Multi-output Regressor.

The Support Vector Regression Algorithm was adopted due to its ability to calculate inter-dependencies between target variables. Because of its capacity to efficiently handle problems involving multiple target variables, the Multi-output Regressor, a variant of the SVR algorithm, stands out as an excellent choice in the machine learning framework. The Multi-output Regressor was used to predict the position changes based on the IMU readings.

Cumulative Summation of Positions Changes

The model outputs were position changes with respect to the IMU readings. To produce a graphical representation, the start point of the journey was assumed to be at (0,0) and cumulative summation was performed on both columns. This would yield a graph that starts from (0,0) and mimic the shape of the trajectory of the vehicle.

3.3.5 WiFi Location Estimation

Following the generation of the graphical representation of the projected path, the next step was to estimate the geographical origin of the travel using local WiFi fingerprints. This smart use of WiFi fingerprints capitalises on the distinct signal patterns present in different locales, allowing the model to geographically locate the starting point of the vehicle's journey. This additional phase improves the model's practical usefulness by providing insights not only into the path prediction but also into the spatial environment of the journey's start.

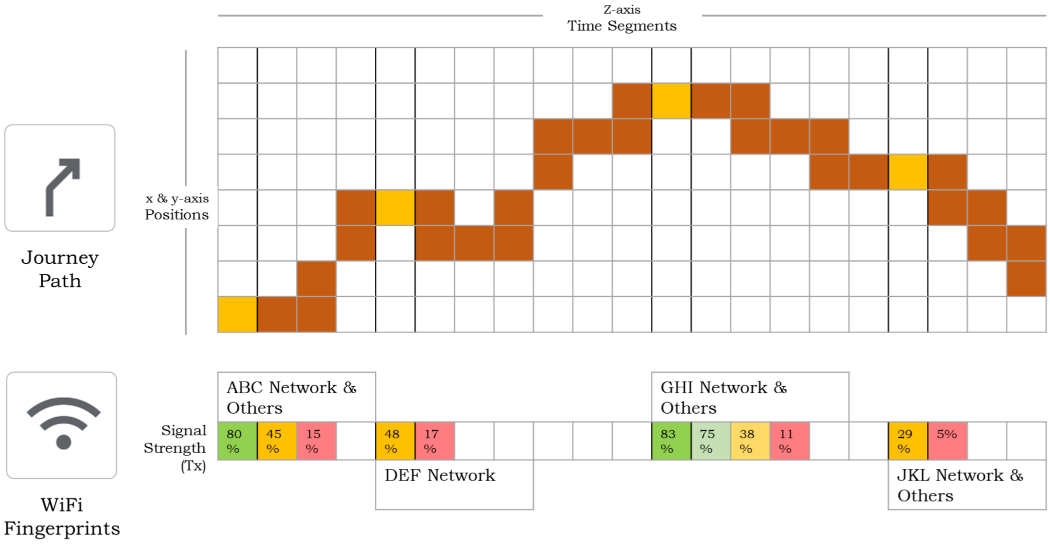


FIGURE 3.9: Location Estimation with Known Fingerprints

Figure 3.9 shows the concept where the Signal strengths of the Wireless Networks encountered is recorded and later compared to the list of known locations to estimate the route traveled.

Chapter 4

Implementation & Testing

This chapter 'Implementation & Testing', describes in detail how the proposed concepts and approaches were put into practice. It goes over the use of algorithms, the integration of technologies, and the execution of trials or tests. The emphasis is on technological application, demonstrating how theoretical notions may be translated into real-world solutions.

4.1 Code Implementation

The code explained below was developed to calculate a vehicle's trajectory based on GPS coordinates and data from an Inertial Measurement Unit (IMU) sensor. The code has been divided into segments based on the stage of processing.

- Importing necessary libraries
- Dataset Selection
- Model Training
- Model Saving (Pickling)
- Model Testing
- Model Validation

Figure 4.1 shows the data flow diagram for the implementation where the sensor data is split into *train* and *test* data. This is then pushed to the Algorithm for training and testing. The learned model is then exported with the process of *Pickling* and then executed on a Hold-out validation test set.

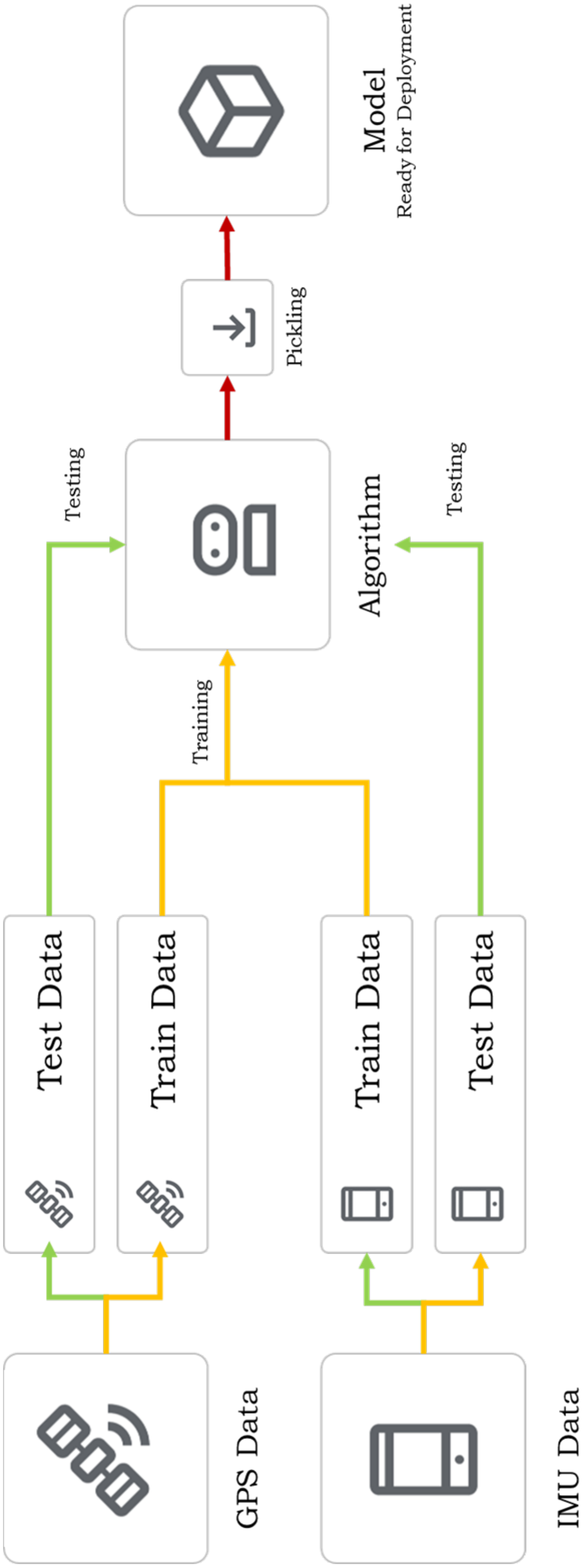


FIGURE 4.1: Proposed Concept - Detailed

4.1.1 Importing necessary libraries

```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

[138]

```python
from sklearn.model_selection import train_test_split
from sklearn.svm import SVR
from sklearn.metrics import mean_squared_error
from sklearn.preprocessing import StandardScaler
from sklearn.multioutput import MultiOutputRegressor
```

[150]

FIGURE 4.2: Library imports

Essential libraries are imported in the first section of the code. For numerical calculations and data manipulation, numpy and pandas are used, and matplotlib is used to visualise GPS track. File operations are handled with the help of the os library (Figure 4.2).

4.1.2 Dataset Selection

Selecting Training Set

```python
_start = 12
_end = 14
_source = sources[_start:_end+1]
_source
```

[141]

```
... array(['Forbury_Road_(stop_EL)-2023-11-28_14-02-13',
          'Reading_College-2023-12-02_15-45-53',
          'Station_Road-2023-12-08_07-53-14'], dtype=object)
```

```python
df_selected = df[df['source'].isin(_source)]
```

[142]

Select Testing Set

```python
_start_t = 2
_end_t = 2
_test = sources[_start_t:_end_t+1]
_test
```

[172]

```
... array(['2023-04-21_17-03-28'], dtype=object)
```

```python
df_test = df[df['source'].isin(_source)]
```

[144]

FIGURE 4.3: Dataset selection for Train & Test

The location to the dataset folder and the CSV file containing the global observations are specified in this section. Ensuring well-structured access to the dataset is crucial for the subsequent analysis of the data. To make sure the model is assessed on data that hasn't been seen yet, the code then chooses a different subset to test the model's performance (Figure 4.3).

This step comprises choosing pertinent columns for both GPS and IMU data, as well as removing extraneous data to free up memory. It guarantees memory efficiency and concentrates on important data columns needed for further investigation.

4.1.3 Model Training

The required libraries for the algorithm such as SVR from the scikit −learn package are imported.

4.1.3.1 Input & Output Features

```
cols_X = ['x_acc', 'y_acc', 'z_acc', 'x_gyro', 'y_gyro', 'z_gyro','x_mag', 'y_mag', 'z_mag']
cols_Y = ['latitude_delta','longitude_delta']
```
[151]

```
X = df_imu[cols_X].values
y = df_gps[cols_Y].values
```
[152]

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```
[153]

FIGURE 4.4: Splitting the dataset into X,Y

In order to train the machine learning model, the significant columns, such as latitude and longitude changes for the output features and accelerometer, gyroscope, and magnetometer values were chosen as the algorithm's input features (Figure 4.4). To evaluate the model, this phase separates the data into training and testing sets.

```
scaler_X = StandardScaler()
scaler_y = StandardScaler()
X_train_scaled = scaler_X.fit_transform(X_train)
y_train_scaled = scaler_y.fit_transform(y_train)
```
[154]

FIGURE 4.5: Dataset Feature Scaling

```
svr_model = SVR(kernel='rbf', C=100, epsilon=0.1)
multioutput_regressor = MultiOutputRegressor(svr_model)
multioutput_regressor.fit(X_train_scaled, y_train_scaled)
```

[155]

FIGURE 4.6: Model Training

StandardScaler() is used to scale data in order to ensure that the model is trained on uniform data (Figure 4.5). In order to predict many output variables at once, the multioutput regressor is used by the algorithm to initialise and train the SVR model (Figure 4.6).

```
def gpsgraph(df_gps):
    longitude_gps = df_gps['longitude'].values
    latitude_gps = df_gps['latitude'].values

    latitude_gps = latitude_gps - latitude_gps[0]
    longitude_gps = longitude_gps  - longitude_gps[0]

    plt.figure()
    plt.scatter(longitude_gps, latitude_gps, label='GPS Path', marker='o')
    plt.xlabel('Longitude')
    plt.ylabel('Latitude')
    plt.title('GPS Path Visualization')
    plt.legend()
    plt.grid(True)
    plt.show()
```

[139]

FIGURE 4.7: GPS Graph - Function

```
gpsgraph(df_gps)
```

[149]

...

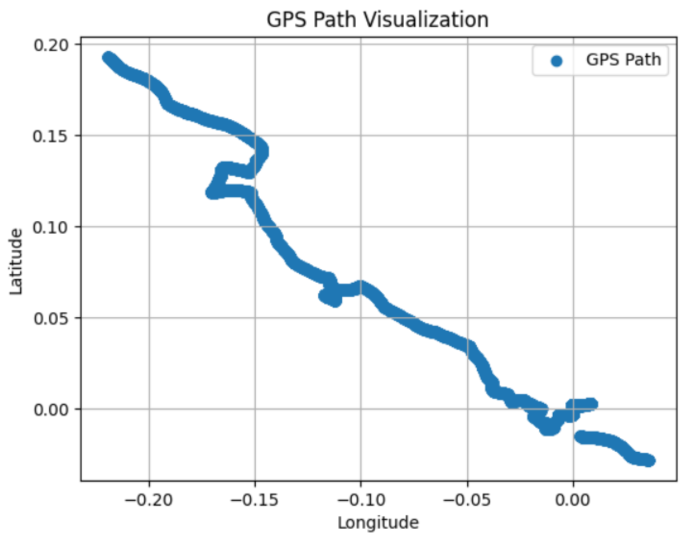


FIGURE 4.8: GPS Graph - Graphical Representation

To visualise the ground truth GPS path, a function called gpsgraph() is defined (Figure 4.7). The function was used to represent the ground truth to later compare with the predicted results (Figure 4.8).

4.1.4 Model Saving (Pickling)

Model saving, often known as pickling, is the process of serialising a trained machine learning model to a file for subsequent use. It is critical because it permits the model's learned parameters and structure to be preserved.

Saving a model allows it to be easily loaded and reused without the requirement for retraining, which saves computational resources and time. This is especially significant when it comes to deploying models in real-world applications, where efficient storage and quick access to pre-trained models are required for smooth integration into diverse systems.

The trained model is saved as a file for easy deployment. This is made possible with the library called pickle. Pickling is the process of saving a variable or object as a bytes file which can be imported and restored with the library. Figure 4.9 shows the process of saving the model as svr_model.model.

```
[169]
import pickle
```

```
[171]
with open('models/svr_multi.model','wb') as m:
    pickle.dump(multioutput_regressor,m)
```

FIGURE 4.9: Saving the model for future use - Pickling

4.1.5 Model Testing

Model testing is the process of evaluating the performance of a machine learning model using a dataset that was not encountered during training. It is imperative to evaluate the model's robustness towards new data, confirming its reliability and effectiveness in making accurate predictions outside of the training set.

The dataset that is used during the Testing phase (Hold out) is selected using the code seen in Figure (4.10).

```
df_test = df[df['source'].isin(_test)]
```

[247]

```
df_gps_test = df_test[['time','longitude','latitude']]
df_imu_test = df_test[cols_X]
```

[248]

Figure 4.10: Selecting Test Dataset (Hold Out Validation)

```
X_input_scaled = scaler_X.transform(X_input)
```

[253]

```
preds = multioutput_regressor.predict(X_input_scaled)
```

[254]

```
df_pred = pd.DataFrame()
df_pred['longitude_change'] = preds[:,0]
df_pred['latitude_change'] = preds[:,1]

df_pred['longitudes'] = np.cumsum(df_pred['longitude_change'].values)
df_pred['latitudes'] = np.cumsum(df_pred['latitude_change'].values)
```

[255]

Figure 4.11: Model Testing

The dataset is scaled using the StandardScaler() used previously. Scaling is crucial to ensure all the variables are uniform and within the given range. This reduces the presence of outliers. Figure 4.11 shows the code used to perform the Scaling process.

The model is then executed on the new input (hold-out) dataset. The model produces delta changes that need to be cumulatively summed to produce positions on the graph. The step that performs the cumulative summation is seen in Figure 4.11.

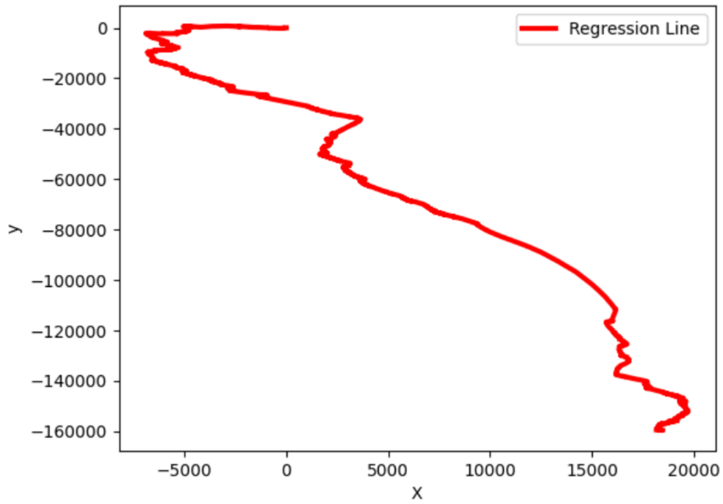


FIGURE 4.12: Journey Predicted (SVR Model)

Figure 4.12 shows the model's prediction after the cumulative summation process. The model is seen to have generated a highly correlated graph that contains major points of similarity to the original journey path (The original journey path can be seen in Figure 4.8).

4.2 Model Validation

Model validation is the process of evaluating the performance and robustness of a machine learning model on unseen data. To ensure reliable predictions on fresh data, the dataset is partitioned into training and testing sets, the model is trained on the training set, and its accuracy is evaluated on the testing set.

Holdout validation is a machine learning approach used to evaluate the performance of a model by separating the dataset into three independent sets: training, validation, and test. The training set is used to train the model, the test set is used to evaluate the model's performance on previously unknown data.

1. **Training set:** This subset of the dataset is used to train the machine learning model. To create predictions, the model learns patterns and relationships in the data.
2. **Validation Set:** Once the model has been trained, it must be fine-tuned to maximise its performance. For this purpose, the validation set is critical. The model is assessed on the validation set, and hyperparameters or the model architecture are tweaked to improve performance.

3. **Test Set:** After the model has been trained and fine-tuned, it is evaluated on a new test set. This last assessment provides an unbiased assessment of the model's capacity to generalise to new, previously unknown data. It assists in determining how effectively the model is predicted to function in real-world circumstances.

In this research, the Hold-Out Validation method is used to validate the performance of the model. Among the various datasets, there are multiple journeys recorded across route A. Additional information about the other routes can be found in Appendix A.

Sensor Readings Evaluation

Initial examination of sensor readings, including accelerometer, gyroscope, and magnetometer data, demonstrated that the mathematical model was capable of producing graphs that corresponds to the actual journey path. The observed considerable variances after particular periods, on the other hand, highlighted the need for further improvement in addressing incremental errors. This emphasised the significance of resolving the model's calculation complexity and adaptability over time.

4.2.1 Mathematical Approach

Using a mathematical method resulted in graphs that initially reflected the actual path. However, a noticeable variation in the expected trajectory developed after a precise point. This divergence indicated the presence of an unaccounted component having a significant influence on sensor data, necessitating a further in-depth investigation of contributory variables.

4.2.2 Machine Learning Approach

Simple Regression Approach

Linear Regression and Polynomial Regression were used for testing, with unfavourable results, especially the Linear Regression technique that produced a dense cloud of points focused in the centre of the graph, indicating its limits in capturing the complex patterns present in the actual route.

Support Vector Regression

To examine the efficacy of the Support Vector Regression (SVR) model, a hold-out evaluation technique was performed by removing two journeys from the training process. Subsequent testing with these previously unreported datasets demonstrated a significant connection to the real path, demonstrating the model's generalisation potential.

Although the SVR model's output graph lacked precise definition, the estimated forecasts had resilient properties that were consistent with the actual voyage path. The inclusion of curves and bends in the anticipated trajectory, however less prominent, demonstrates the model's ability to capture crucial elements.

4.3 Findings

This SVR model's findings show encouraging results in vehicle trajectory estimation using a fusion of Inertial Measurement Unit (IMU) sensor data with WiFi fingerprinting.

- The Support Vector Regression (SVR) model produced significant positive results, generating accurate and well-defined graphs that closely resembled actual GPS trajectories. Despite initial success, the mathematical model revealed the need for additional complexity and access to larger datasets.
- The detection of device-specific faults highlighted the significance of comprehensive sensor calibration and validation to ensure the accuracy and dependability of trajectory predictions.
- Another important finding relating to the reliability of the wireless fingerprinting was the vulnerability to masquerading, better security protocols would be advised.

Chapter 5

Evaluation & Discussion

This chapter 'Evaluation & Discussion', evaluates the consequences of adopted approaches critically. It presents findings, investigates project success or failure, and explores ramifications. This chapter strives to provide a more nuanced perspective of results by revealing strengths, weaknesses, and potential areas for development.

Reasoning

The Models were evaluated on the unseen journeys (ROUTE A & B). Figure 5.1 shows the performance of each approach. It can be understood that the mathematical approach and the SVR approach yielded the better results compared to the Simple Regression.

The Mathematical approach although produced a similar looking graph, the model seemed to have drifted away from the general heading. It can be speculated that the noise that was part of the dataset might have caused this distortion. The Support Vector Regression model has however adapted to this by learning around the noise thresholds of the device to minimise the drift from the original journey path.

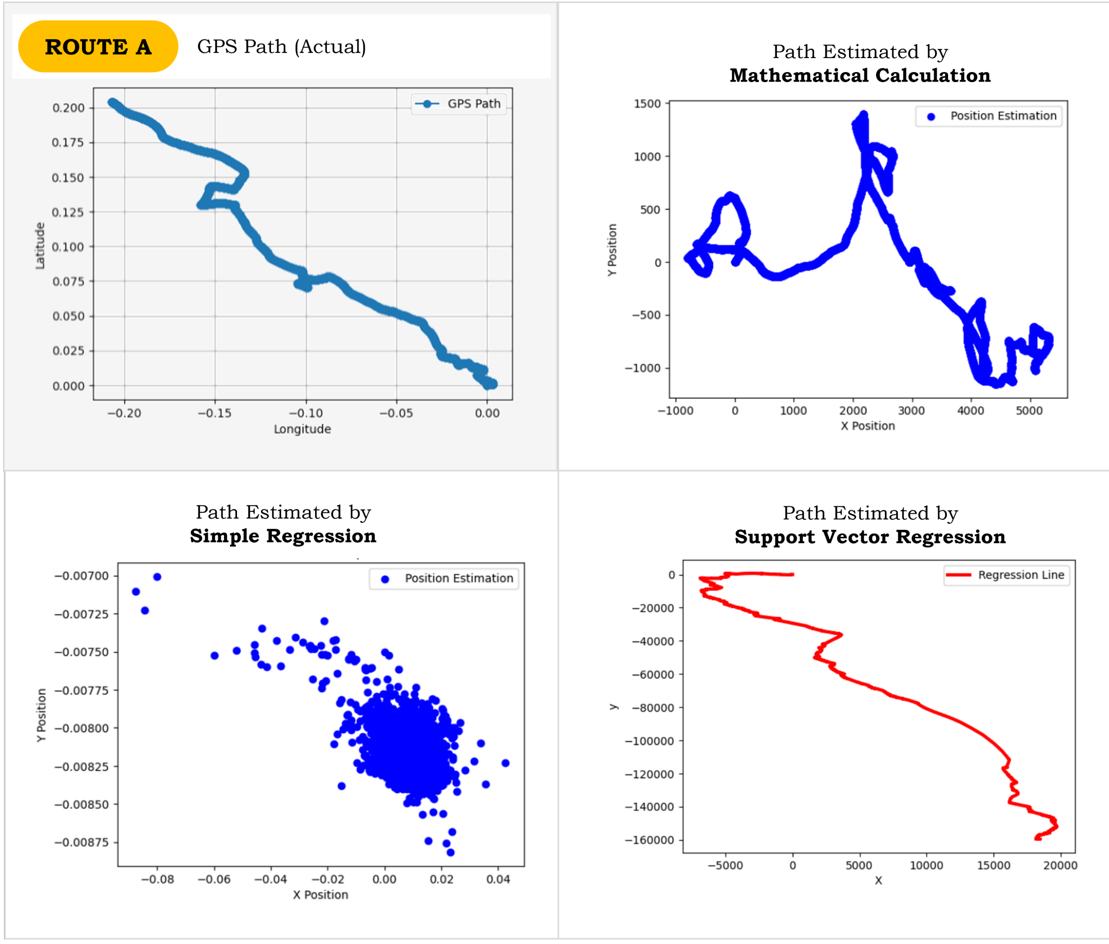


FIGURE 5.1: Route A

The SVR Algorithm was also tested on another path that was different from the majority of the dataset mode of transportation. Route B (as seen in Figure 5.2) was a journey made by Railways unlike Route A which was completed by a bus.

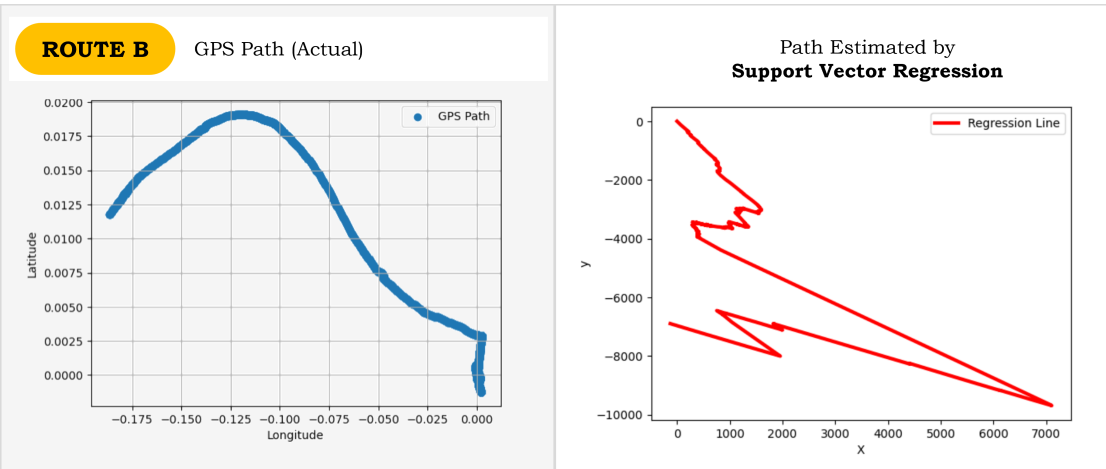


FIGURE 5.2: Route B

In the Figure 5.3, it can be seen that the patterns highlighted can be found in the generated graph. The major distortion is caused by the outlier point. The other curves (green and orange) can be corresponded to the SVR generated graph.

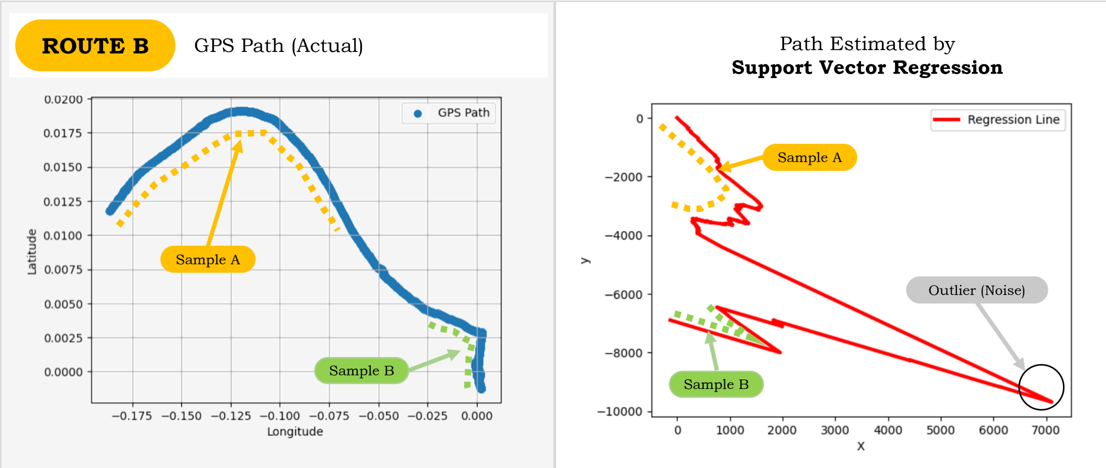


FIGURE 5.3: Route B (Points of Similarity)

5.1 Model Evaluation

The model testing and evaluation phase is crucial in determining the usefulness and dependability of the suggested models for predicting vehicle trajectories. Each model was rigorously tested to determine its strengths, flaws, and overall performance, including the mathematical method, Linear Regression, and Support Vector Regression (SVR).

5.1.1 Mathematical Model

The mathematical model originally produced a graph that resembled the journey path, but it showed significant variations at various intervals. This divergence indicated the model's inability in adjusting to incremental errors after reaching a certain stage, emphasising the significance of refining computation complexity for long-term accuracy.

5.1.2 Machine Learning Model

Simple Regression Model

Linear regression failed to capture the complex interactions between individual Inertial Measurement Unit (IMU) sensors. Linear regression's fundamental simplicity limited its ability to simulate the intricacy inherent in sensor data, emphasising the need for more sophisticated techniques to account for nuanced interactions.

Support Vector Regression Model

The Support Vector Regression (SVR) model produced well-defined graphs that closely resembled actual GPS trajectories. This achievement establishes SVR as a reliable alternative, especially when dealing with complex data interactions. However, it is critical to recognise that SVR performance can be further optimised with access to larger datasets, allowing for improved model construction and accuracy. Furthermore, increasing the dataset to include readings from various sensors can help to eliminate inherent biases and create a more generalizable model.

5.2 Discussion

Potential Issues & Considerations

5.2.1 Input Data Noise

The accuracy of the model depends greatly on the quality of the input data. Any noise, outliers, or inconsistencies in sensor measurements might have an effect on model performance.

Inaccuracies, noise, or outliers in the sensor measurements are possible causes for concern. Sensors are prone to several interferences that can add inaccuracies into the data, such as gyroscopes, accelerometers, and magnetometers. Inaccurate readings could result from, for example, sudden shocks or outside magnetic fields (Jalilian et al., 2017, Lee et al., 2019). In order to recognise and deal with outliers effectively, a thorough data preprocessing stage is essential. It is possible to improve the quality of the sensor data by using methods like smoothing filters or outlier detection algorithms.

Procedures for sensor calibration should also be taken into account. Systematic errors and biases in sensor readings can be decreased by calibration. To further improve data reliability, routine calibration checks should be made, particularly in dynamic situations.

5.2.2 Feature Engineering

Additional Feature engineering maybe required. Currently in the study, the delta changes calculated can be considered as feature engineering. Depending on the properties of the data, further feature engineering may be required to improve model performance. Enhancing the performance of the model can be achieved by adjusting its features through experimentation and domain expertise.

5.2.3 Model Complexity

SVR can be considered to be a complex model which can also result in over-engineering. Exploring other suitable models can perhaps uncover algorithms that might perform better with lesser compute resources.

It is crucial to fine-tune hyperparameters such kernel parameters and the regularisation parameter (C). A complex model might have an extremely close fit to the training set, catching anomalies and noise, but it might not translate well to new data. An alternative would be to investigate more straightforward models with linear kernels, which would strike a compromise between identifying trends and avoiding unnecessary complexity.

5.2.4 Overfitting or Underfitting

When a model picks up noise from the training set, it is said to be overfitted, which results in poor generalisation on fresh data. Conversely, underfitting occurs when a model is overly simplistic to adequately represent the underlying patterns in the data. With the potential for complexity that SVR brings, overfitting issues must be properly addressed.

There is a risk of overfitting or underfitting, particularly if the dataset is limited or the model is too complex. Regularisation approaches and simpler models may be beneficial.

5.2.5 Dataset Size and Diversity

The possibility of improving SVR with larger datasets highlights the importance of data volume and diversity in training machine learning models, which contribute to their adaptability and generalisation.

5.2.6 Device-Specific Limitation & Bias

The mention of device-specific errors emphasises the importance of strict sensor calibration and validation to assure model accuracy, particularly in real-world applications where devices may display variances or defects.

The proposed approach's effectiveness has numerous possible limits that should be carefully considered. Device-specific errors can negatively influence the accuracy of the journey's forecasts. If the data collection device contains a malfunctioning sensor, the fusion of sensor signals from the accelerometer, gyroscope, and magnetometer may be disrupted. Faulty sensors can add flaws in the data, resulting in inaccurate predictions and lowering the model's overall reliability.

Gathering the sensor readings from the same device can incur device bias that can influence the model. The model may inherit the device's tolerance of noise and may not be robust. Gathering sensor readings from other devices (Smartphones, Arduino boards, Raspberry based development boards) can improve the model's ability to respond to generic data.

5.2.7 Wireless Fingerprint Vulnerabilities

Recognising wireless fingerprint vulnerabilities emphasises the significance of robust security protocols, which ensure that the model's predictions are not jeopardised by potential network attribute disguising or spoofing.

Furthermore, relying on wireless fingerprints exposes a set of vulnerabilities that must be addressed. While wireless fingerprints, which include unique identifiers like SSID, security protocols, and MAC addresses, provide a reliable method of differentiating across networks, they are not impervious to manipulation. Because many of the identifying elements in wireless fingerprints may be actively set and mimicked, one possible weakness is the ability to masquerade as another network. This creates the chance for hostile actors to trick the model by mimicking the features of a legal network, resulting in inaccurate location estimations.

For instance, an attacker may intentionally construct a network to mimic the properties of a legitimate network, prompting the model to link the device's location with the forged network. This type of spoofing poses a substantial danger to the accuracy of location estimate, particularly in contexts where security measures are not rigorously enforced.

5.2.8 Model Bias

Since the current model is trained on both Buses and Railway based transportation, the balance between both types of transportation must be maintained to ensure the model is not biased towards a certain mode. Exposing the algorithm to different types of transportation can also help the model establish better correlations between the sensor readings.

5.3 Ethical, Legal and Social Issues (ELSI)

Addressing Ethical, Legal, and Social Issues (ELSI) is critical to the development and implementation of any technology, including the proposed project. In this research, the following aspects were explored:

Ethical Issues

- **Privacy Concerns:**
  Collecting user's location information may raise justifiable privacy concerns. To reassure the users of this concept, the location information can be collected by accepting volunteers to feed into the machine learning model. Once a strong model is developed, the user's device will not have the need to share location data, instead the processing would take place in the user's device.

- **Transparency**
  Providing information without a cause can raise suspicions, therefore, data collected as part of the ongoing improvement of the model can be paired with a detailed use of the data and the methods that would be employed to make use of the data.

  To avoid any model bias, Regular comparisons with existing location technologies can be performed and evaluated to fine tune the model parameters.

Legal Issues

- **Data Protection Compliance (GDPR)**
  As this research involves the use of personal information such as the location of the individual (via the mobile device), the further research can supplemented by obtaining an informed consent from the users of the application.

- **Secure Data Handling**
  Since the model uses location information to improve the accuracy of the model, strict measures can be put in place to safeguard the data on reliable storage solutions.

- **Intellectual Property**
  This research and associated source code operates with open source algorithms and software and will continue to be open source with an (MIT License) to encourage other researchers and developers to fine tune the model freely.

Social Issues

- **Accessibility**
  This concept hopes to accommodate a diverse user base that includes those with disabilities. The project will however, require further datasets that were collected from certain demographics to ensure a reliable model performance.

Chapter 6

Conclusion & Recommendations

This research project served as a foundational exploration, particularly targeted at producing a proof of concept for calculating a vehicle's trajectory through the combination of Inertial Measurement Unit (IMU) sensor data with WiFi fingerprinting for location estimate. The models used, which included a mathematical approach, Linear Regression, and Support Vector Regression (SVR), produced notably positive results. However, significant limitations and areas for improvement were discovered during the course of the project.

The models, particularly the SVR, demonstrated significant promise in producing accurate predictions, closely resembling actual GPS trajectories. Nonetheless, further complexity and access to larger, less noisy datasets are identified as essential components for enhancing the models' predictive capabilities. The recognition of device-specific errors and vulnerabilities in wireless fingerprinting underscores the importance of continuous refinement and robustness in real-world applications.

It is critical to recognise that the visual representations created by these models were not without flaws; nonetheless, they have the potential to achieve outstanding accuracy with additional refinement via dataset training and diligent model tweaking. This round of testing acts as a proof of concept, demonstrating the model's capacity to anticipate a vehicle's path regardless of topographical conditions. The findings show the proposed approach's practicality and potential usefulness in the context of real-world applications.

Recommendations

- **Sensor Data Quality**
  Investigate and improve sensor data quality by experimenting with advanced noise reduction techniques and outlier detection approaches. Ensuring high-quality input is critical for the positioning system's precision.

- **Intermediate Machine Learning Model**
  Create an intermediate machine learning model to limit the impact of device vibrations and free mobility within the vehicle. This improvement tries to separate and record actual vehicle movement more precisely.

- **WiFi Fingerprints Crowdsourcing**
  Using crowdsourcing approaches to acquire a varied variety of WiFi fingerprinted locations can greatly contribute to a more comprehensive and accurate dataset. This method ensures a wider range of environmental variables and scenarios.

- **Movement Threshold Setting**
  Using dynamic thresholds based on various kinds of transportation helps improve the model's adaptability. For example, distinguishing between modes such as vehicles, buses, and trains based on their average degrees of turning can lead to more nuanced and accurate predictions.

Finally, the research presents a solid foundation for the incorporation of IMU sensors and WiFi fingerprinting in vehicle trajectory estimation. The identified areas for improvement pave the way for future efforts to develop and expand the approach's applicability in real-world circumstances. The possibility for accurate and dependable trajectory estimation grows as technology develops and datasets become more diversified.

Appendix A

Estimated Path Predictions

ROUTE A

- **START**: Reading Rail Station
- **END** : Berinsfield (Oxfordshire)
- **MODE** : Bus

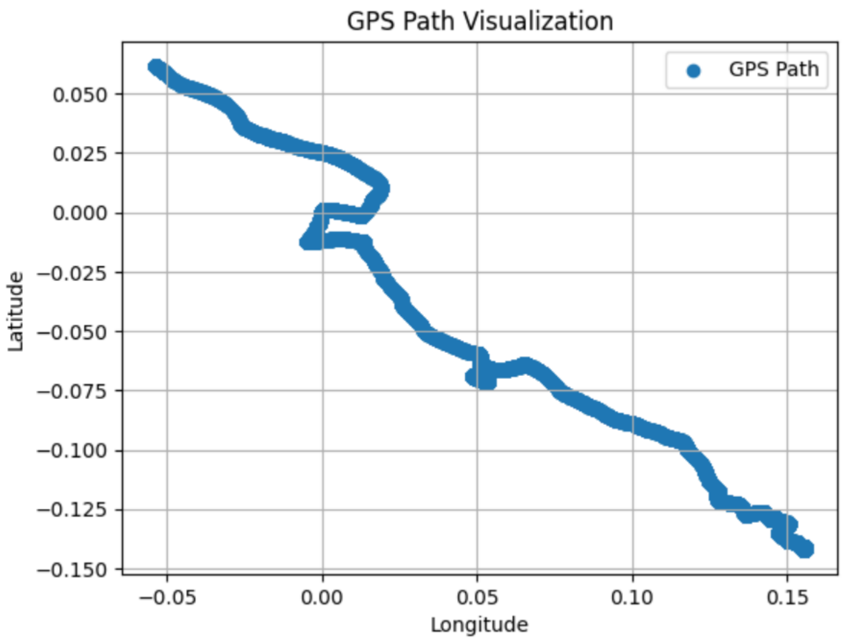


Figure A.1: Route A - Actual Path

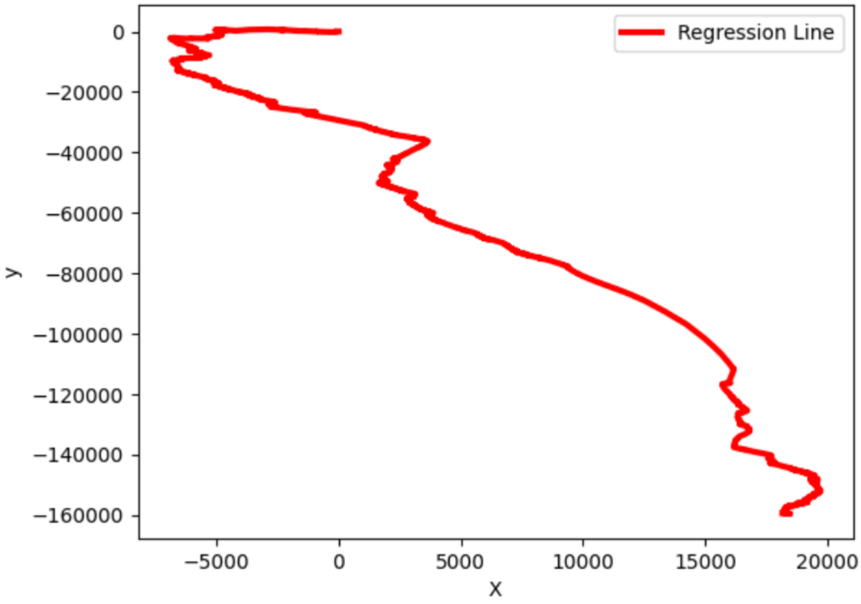


FIGURE A.2: Route A - ML Predicted Path

ROUTE B

- **START**: Earley Station
- **STOP**: Stop at Reading Rail Station
- **END** : Berinsfield (Oxfordshire)
- **MODE** : Bus + Rail

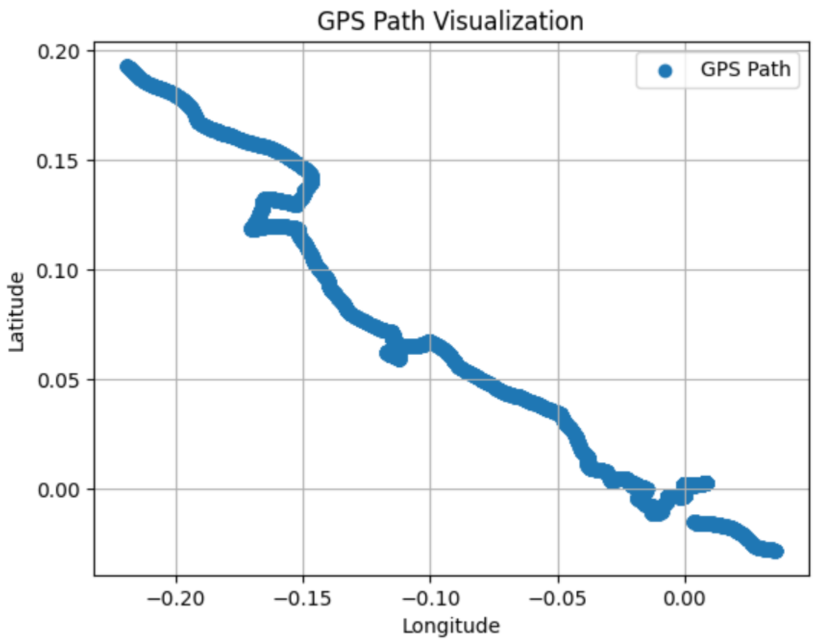


FIGURE A.3: Route B - Actual Path

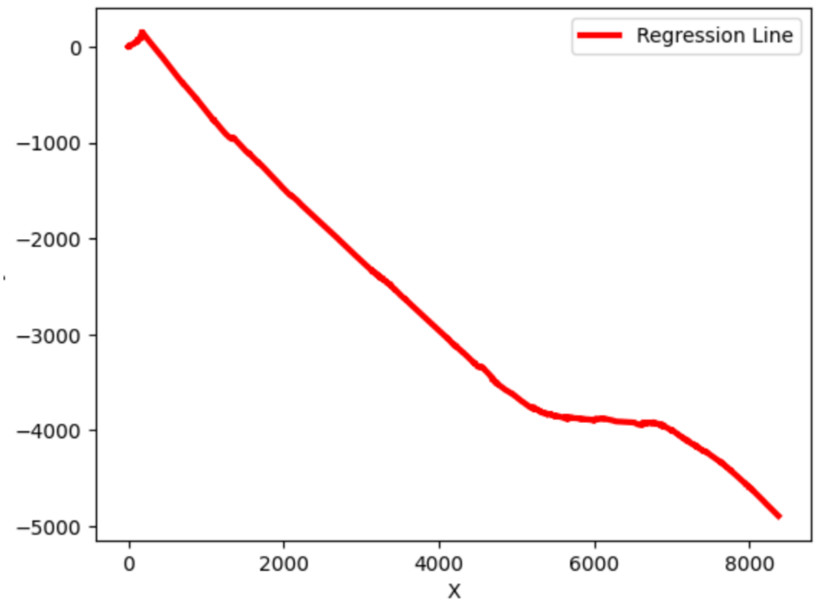


FIGURE A.4: Route B - ML Predicted Path

ROUTE C

- **START**: Twyford Rail Station
- **END** : Reading Rail Station
- **MODE** : Rail - Elizabeth Line

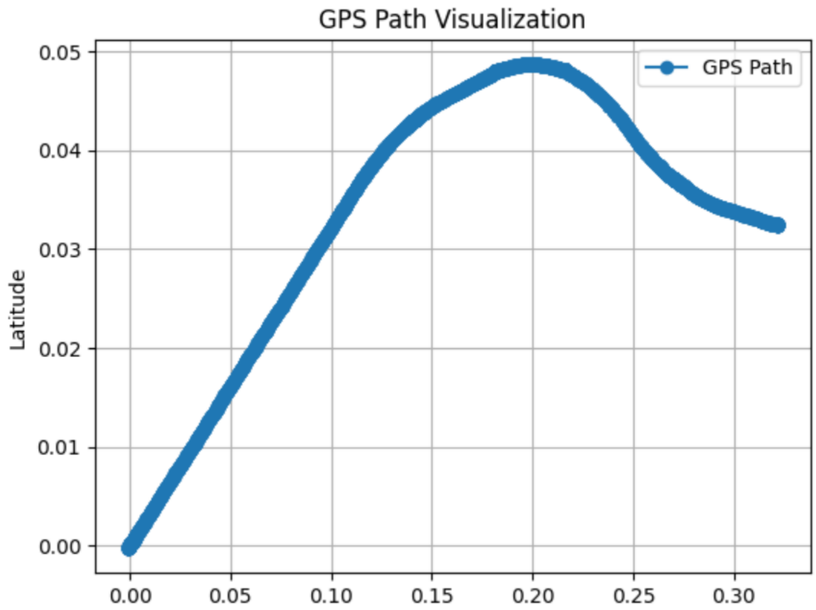


FIGURE A.5: Route C - Actual Path

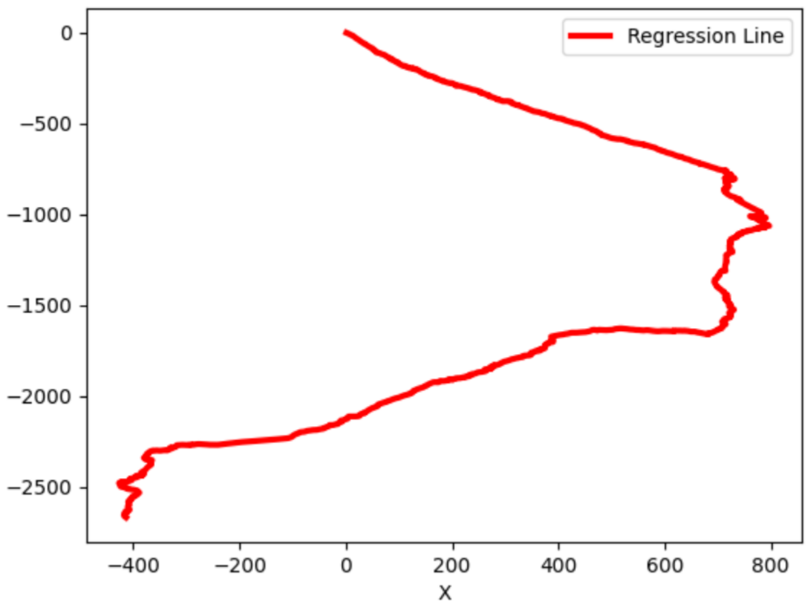


FIGURE A.6: Route C - ML Predicted Path

ROUTE D

- **START**: Reading Rail Station
- **END** : Caversham Bus Stop
- **MODE** : Bus (Double-Decker)

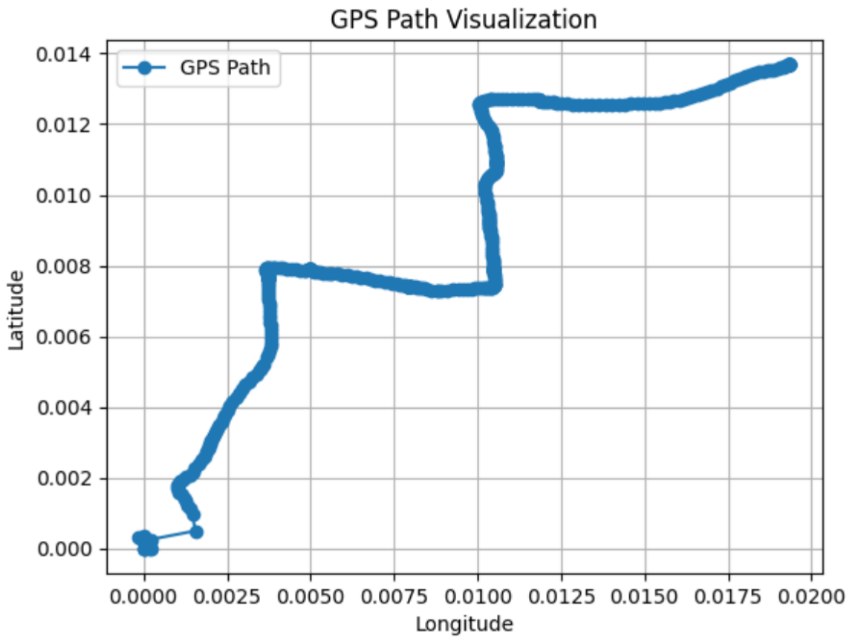


FIGURE A.7: Route D - Actual Path

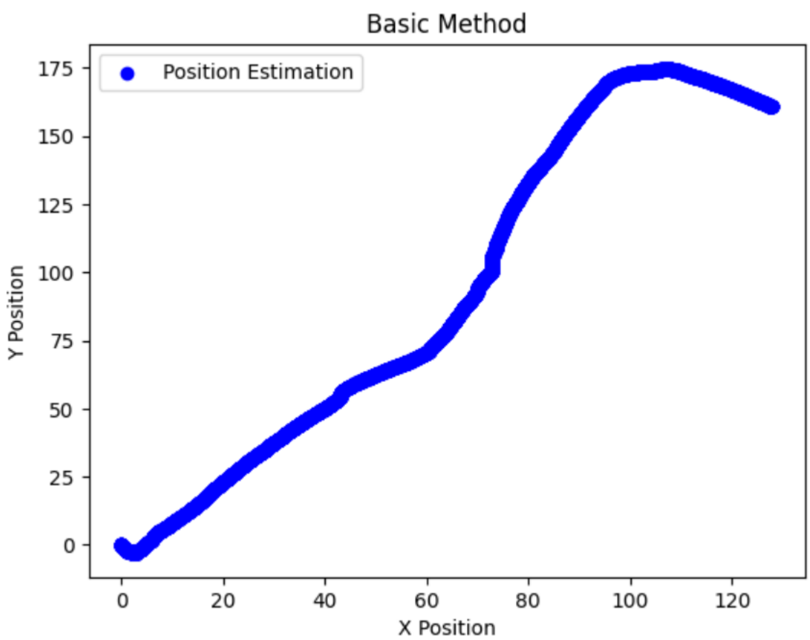


FIGURE A.8: Route D - Mathematically Calculated Path

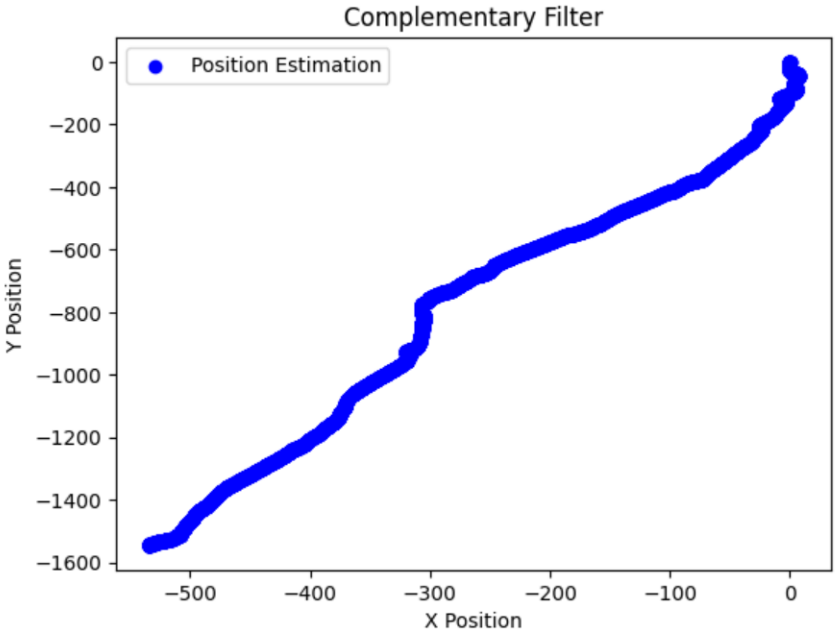


FIGURE A.9: Route D - ML Predicted Path

Appendix B

Meeting Logs

Discussions, conclusions, and action items during project meetings are documented in meeting logs with a supervisor. They serve as a vital record, assisting in tracking progress, identifying issues, and ensuring team members' alignment. These records help to keep clear communication, accountability, and a thorough grasp of project advancements.

Record of Supervisory Meetings

Student Name: Elvis Tony	**Programme:** MSc. AIT
Supervisor: Dr. Saber Farag	

Date of Meeting:	18/10/2023 15 Minutes
Meeting Number:	1
Mean of the meeting:	Online (Microsoft Teams)
Brief Summary of Discussion Explained the project idea and approach during the second week	
Agreed Actions: Next Task was decided to work on the Interim report and begin initial programming	

Date of Meeting:	25/10/2023 15 Minutes
Meeting Number:	2
Mean of the meeting:	Online (Microsoft Teams)
Brief Summary of Discussion Presented the initial study of existing approaches and provided a status update about the programming.	
Agreed Actions: Next task was to fill out the Research ethics form and receive approval to proceed.	

Date of Meeting:	01/11/2023 15 Minutes
Meeting Number:	3
Mean of the meeting:	Online (Microsoft Teams)
Brief Summary of Discussion Presented the partial source code and initial findings about the dataset and discussed possible approaches in calculating the proximity of a device in a network.	
Agreed Actions: Test out various machine learning algorithms like KNN to estimate the WiFi Neighbourhood.	

Date of Meeting:	07/11/2023 15 Minutes
Meeting Number:	4
Mean of the meeting:	Online
Brief Summary of Discussion Reviewed the Interim Report and shared possible improvements to the Literature Survey and Methodology.	
Agreed Actions: Expand the Methodology section and refer more research studies linked to the concept.	

Date of Meeting:	08/11/2023 15 Minutes
Meeting Number:	5
Mean of the meeting:	Online (Microsoft Teams)
Brief Summary of Discussion Discussed the challenges faced when programming the algorithm to calculate the positions of the device across the given time frame. The graphical visualizations produced needs improvement.	
Agreed Actions: Check out other similar implementations and study their approach.	

Date of Meeting:	15/10/2023 15 Minutes
Meeting Number:	6
Mean of the meeting:	Online (Microsoft Teams)
Brief Summary of Discussion Algorithm still fails to produce correlating graphs for the given dataset. Although the graphs have improved, there exists a significant deviation to the path due to noise in the dataset.	
Agreed Actions: Continue working on the algorithm and explore other methods to estimate the location using IMU sensor data.	

Date of Meeting:	22/11/2023 15 Minutes
Meeting Number:	7
Mean of the meeting:	Online (Microsoft Teams)
Brief Summary of Discussion Attempted to use Machine learning but faced accuracy issues. The graphs did not produce the target journey path.	
Agreed Actions: Explore the possible ML algorithm alternatives to solve the problem.	

Date of Meeting:	29/11/2023 15 Minutes
Meeting Number:	8
Mean of the meeting:	Online (Microsoft Teams)
Brief Summary of Discussion Explored and tested SVR and Multi-output SVR and produced better results. Results however were not reliable.	
Agreed Actions: Explore the influence of the dataset on the Algorithm and attempt to change the Input features.	

Date of Meeting:	30/12/2023
Meeting Number:	9
Mean of the meeting:	Email
Brief Summary of Discussion Breakthrough! Feature engineered columns in the dataset to replace actual positions with relative positions to improve the dataset's robustness. This Approach proved fruitful with much better visualizations that corresponded to the target journey path.	
Agreed Actions: Await Supervisor's feedback and continue drafting the Implementation section of the report.	

Date of Meeting:	02/01/2024
Meeting Number:	10
Mean of the meeting:	Email
Brief Summary of Discussion Mentioned potential concerns regarding the similarity scores when testing for plagiarism (Majority of the similarity pointed towards the previously submitted Research Methods Assignment).	

Date of Meeting:	07/01/2024
Meeting Number:	11
Mean of the meeting:	Email
Brief Summary of Discussion Source code uploaded to GitHub. Asked if the source code can be hosted on GitHub and linked to the Research project due to the presence of multiple files in the source code.	

Date of Meeting:	10/01/2024 15 Minutes
Meeting Number:	12
Mean of the meeting:	Online (Microsoft Teams)
Brief Summary of Discussion Discussed the Breakthrough and explained the change in approach. Went through the entire written report and submitted the report for supervisor feedback.	

Date of Meeting:	11/01/2024
Meeting Number:	13
Mean of the meeting:	Email
Brief Summary of Discussion Final Report drafted and sent to the Supervisor for Feedback.	

Appendix C

Source Code

The source code for the project, which is critical to our Wi-Fi and IMU sensor fusion, is available on GitHub with an MIT License. GitHub, a collaborative version control platform, provides transparency into our methods and methodologies. This open-source model encourages academics, developers, and enthusiasts to investigate, participate, and contribute to the progress of the project. The GitHub code repository acts as a platform for collaborative learning and knowledge exchange. Join us on this coding adventure as we explore the potential of Wi-Fi and IMU sensor fusion.

Source Code is available on GitHub

github.com/elvistony/imu-wifi-fusion

Click here to open the repository

Link to Source Code: github.com/elvistony/imu-wifi-fusion

Bibliography

Antsfeld, L., Chidlovskii, B. and Sansano-Sansano, E. (2020), 'Deep smartphone sensors-wifi fusion for indoor positioning and tracking', *arXiv preprint arXiv:2011.10799* .

Bao, F., Mazokha, S. and Hallstrom, J. O. (2021), Mobintel: Passive outdoor localization via rssi and machine learning, *in* '2021 17th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)', IEEE, pp. 247–252.

Bi, J., Zhao, M., Yao, G., Cao, H., Feng, Y., Jiang, H. and Chai, D. (2023), 'Psosvrpos: Wifi indoor positioning using svr optimized by pso', *Expert Systems with Applications* **222**, 119778.

Bin Uzayr, S. (2022), *Mastering Visual Studio Code: A Beginner's Guide*, CRC Press.

Brida, P., Machaj, J., Racko, J. and Krejcar, O. (2021), 'Algorithm for dynamic fingerprinting radio map creation using imu measurements', *Sensors* **21**(7).
URL: *https://www.mdpi.com/1424-8220/21/7/2283*

Choi, K. T. H. (2022), 'Awesome sensor logger applicaton'.
URL: *https://github.com/tszheichoi/awesome-sensor-logger/*

Crozet, Y. and Coldefy, J. (2021), Mobility as a Service (MaaS): a digital roadmap for public transport authorities, PhD thesis, CERRE.

Faragher, R. and Harle, R. (2015), 'Location fingerprinting with bluetooth low energy beacons', *IEEE journal on Selected Areas in Communications* **33**(11), 2418–2428.

Ferrari, L., Berlingerio, M., Calabrese, F. and Reades, J. (2014), 'Improving the accessibility of urban transportation networks for people with disabilities', *Transportation Research Part C: Emerging Technologies* **45**, 27–40.

Franček, P., Jambrošić, K., Horvat, M. and Planinec, V. (2023), 'The performance of inertial measurement unit sensors on various hardware platforms for binaural head-tracking applications', *Sensors* **23**(2), 872.

Hou, C. (2020), A study on imu-based human activity recognition using deep learning and traditional machine learning, *in* '2020 5th International Conference on Computer and Communication Systems (ICCCS)', IEEE, pp. 225–234.

Hussain, A., Akhtar, F., Khand, Z. H., Rajput, A. and Shaukat, Z. (2021), 'Complexity and limitations of gnss signal reception in highly obstructed enviroments', *Engineering, Technology & Applied Science Research* **11**(2), 6864–6868.

Innes, E. (2013), ''digital distraction' causes 20 million passengers to miss their bus or train stop every year'.

Jalilian, H., Najafi, K., Monazzam, M. R., Khosravi, Y. and Jamali, J. (2017), 'Occupational exposure of train drivers to static and extremely low frequency magnetic fields in tehran subway', *Jundishapur Journal of Health Sciences* **9**(4).

Jambrosic, K., Krhen, M., Horvat, M. and Jagust, T. (2020), Measurement of imu sensor quality used for head tracking in auralization systems, *in* 'Forum Acusticum', pp. 2063–2070.

Jin, M., Koo, B., Lee, S., Park, C., Lee, M. J. and Kim, S. (2014), Imu-assisted nearest neighbor selection for real-time wifi fingerprinting positioning, *in* '2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN)', pp. 745–748.

John Hunter, Darren Dale, E. F. M. D. and Team (2023), 'Matplotlib - documentation'.
URL: *https://matplotlib.org/stable/index.html*

JupyterTeam (2023), 'Jupyter - documentation'.
URL: *https://docs.jupyter.org/en/latest/*

Kim, K. S., Wang, R., Zhong, Z., Tan, Z., Song, H., Cha, J. and Lee, S. (2018), 'Large-scale location-aware services in access: Hierarchical building/floor classification and location estimation using wi-fi fingerprinting based on deep neural networks', *Fiber and Integrated Optics* **37**(5), 277–289.

Kotaru, M., Joshi, K., Bharadia, D. and Katti, S. (2015), Spotfi: Decimeter level localization using wifi, *in* 'Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication', pp. 269–282.

Lee, J., Kang, M., Park, Y., Park, D. and Choi, S. (2019), 'Evaluation of intensity of extremely low frequency magnetic fields (elf-mf) inside of cabins as generated during subway operation', *Journal of Korean Society of Occupational and Environmental Hygiene* **29**(2), 185–194.

Leitch, S. G., Ahmed, Q. Z., Abbas, W. B., Hafeez, M., Laziridis, P. I., Sureephong, P. and Alade, T. (2023), 'On indoor localization using wifi, ble, uwb, and imu technologies', *Sensors* **23**(20), 8598.

Li, S., Qin, Z., Song, H., Si, C., Sun, B., Yang, X. and Zhang, R. (2020), 'A lightweight and aggregated system for indoor/outdoor detection using smart devices', *Future Generation Computer Systems* **107**, 988–997.

Li, Z. and Zhang, Y. (2022), 'Constrained eskf for uav positioning in indoor corridor environment based on imu and wifi', *Sensors* **22**(1), 391.

Low, W.-Y., Cao, M., De Vos, J. and Hickman, R. (2020), 'The journey experience of visually impaired people on public transport in london', *Transport Policy* **97**, 137–148.

Macealois, J. (2023), 'Everything you should know about gps tracking battery life'.
URL: *https://www.workyard.com/employee-time-tracking/gps-tracking-battery-life*

Mok, E. and Retscher, G. (2007), 'Location determination using wifi fingerprinting versus wifi trilateration', *Journal of Location Based Services* **1**(2), 145–159.

NumpyDevelopmentTeam (2023), 'Numpy - documentation'.
URL: *https://numpy.org/doc/*

Nurpeiissov, M., Kuzdeuov, A., Assylkhanov, A., Khassanov, Y. and Varol, H. A. (2022), End-to-end sequential indoor localization using smartphone inertial sensors and wifi, *in* '2022 IEEE/SICE International Symposium on System Integration (SII)', IEEE, pp. 566–571.

PandasDevelopmentTeam (2023), 'Pandas - documentation'.
URL: *https://pandas.pydata.org/docs/*

Prikhodko, I. P., Bearss, B., Merritt, C., Bergeron, J. and Blackmer, C. (2018), Towards self-navigating cars using mems imu: Challenges and opportunities, *in* '2018 IEEE International Symposium on Inertial Sensors and Systems', IEEE, pp. 1–4.

PythonDevelopmentTeam (2023), 'Python pickle library - documentation'.
URL: *https://docs.python.org/3/library/pickle.html*

Sarshar, M., Polturi, S. and Schega, L. (2021), 'Gait phase estimation by using lstm in imu-based gait analysis—proof of concept', *Sensors* **21**(17), 5749.

Schmutz, A., Chèze, L., Jacques, J. and Martin, P. (2020), 'A method to estimate horse speed per stride from one imu with a machine learning method', *Sensors* **20**(2), 518.

ScikitLearn-Developers (2023), 'User guide-scikit learn'.
URL: *https://scikit-learn.org/stable/user$_g$uide.html*

Statistics, O. (2022), 'Population estimates - office of national statistics'.
URL: *https://www.ons.gov.uk/peoplepopulationandcommunity/*

Sun, M., Wang, Y., Joseph, W. and Plets, D. (2022), 'Indoor localization using mind evolutionary algorithm-based geomagnetic positioning and smartphone imu sensors', *IEEE Sensors Journal* **22**(7), 7130–7141.

Tavmen, G. (2020), 'Data/infrastructure in the smart city: Understanding the infrastructural power of citymapper app through technicity of data', *Big Data & Society* **7**(2), 2053951720965618.

Toy, I., Durdu, A. and Yusefi, A. (2022), Improved dead reckoning localization using imu sensor, *in* '2022 International Symposium on Electronics and Telecommunications (ISETC)', IEEE, pp. 1–5.

VSCodeDevelopmentTeam (2023), 'Visual studio code - documentation and user guide'.
URL: *https://code.visualstudio.com/docs*

Wang, Q., Li, J., Luo, X. and Chen, C. (2022), Fusion algorithm of wifi and imu for indoor positioning, *in* '2022 3rd International Conference on Information Science, Parallel and Distributed Systems (ISPDS)', IEEE, pp. 349–354.

Wright, T. (2022), 'Transport for london is here to support you on your accessible journeys', *Intelligent transport: Reporting from the cutting-edge of urban transport technology* (3), 38–39.

Zhuang, Y., Sun, X., Li, Y., Huai, J., Hua, L., Yang, X., Cao, X., Zhang, P., Cao, Y., Qi, L. et al. (2023), 'Multi-sensor integrated navigation/positioning systems using data fusion: From analytics-based to learning-based approaches', *Information Fusion* **95**, 62–90.

Zou, H., Chen, Z., Jiang, H., Xie, L. and Spanos, C. (2017), Accurate indoor localization and tracking using mobile phone inertial sensors, wifi and ibeacon, *in* '2017 IEEE International Symposium on Inertial Sensors and Systems', IEEE, pp. 1–4.