Real-time Localisation System for GPS Denied Open Areas using Smart Street Furniture

Mohamed Ali

This thesis is presented for the Degree of Doctor of Philosophy
Murdoch University

September 2020

Declaration

I declare that this thesis is my own account of my research and contains as its main content work which has not previously been submitted for a degree at any tertiary education institute.

Mohamed Ali

Abstract

Real-time measurement of crowd dynamics has been attracting significant interest, as it has many applications including real-time monitoring of emergencies and evacuation plans. To effectively measure crowd behaviour, an accurate estimate for pedestrians' locations is required. However, estimating pedestrians' locations is a great challenge especially for open areas with poor Global Positioning System (GPS) signal reception and/or lack of infrastructure to install expensive solutions such as video-based systems.

Street furniture assets such as rubbish bins have become smart, as they have been equipped with low-power sensors. Currently, their role is limited to certain applications such as waste management. We believe that the role of street furniture can be extended to include building real-time localisation systems as street furniture provides excellent coverage across different areas such as parks, streets, homes, universities.

In this thesis, we propose a novel wireless sensor network architecture designed for smart street furniture. We extend the functionality of sensor nodes to act as soft Access Point (AP), sensing Wifi signals received from surrounding Wifi-enabled devices. Our proposed architecture includes a real-time and low-power design for sensor nodes. We attached sensor nodes to rubbish bins located in a busy GPS denied open area at Murdoch University (Perth, Western Australia), known as Bush Court. This enabled us to introduce two unique Wifi-based localisation datasets: the first is the Fingerprint dataset called MurdochBushCourtLoC-FP (MBCLFP) in which four users generated Wifi fingerprints for all available cells in the gridded Bush Court, called Reference Points (RPs), using their smartphones, and the second is the APs dataset called MurdochBushCourtLoC-AP (MBCLAP) that includes auto-generated records received from over 1000 users' devices.

Finally, we developed a real-time localisation approach based on the two datasets using a four-layer deep learning classifier. The approach includes a light-weight algorithm to label the MBCLAP dataset using the MBCLFP dataset and convert the MBCLAP dataset to be synchronous. With the use of our proposed approach, up to 19% improvement in location prediction is achieved.

Table of Contents

Acknowledgement

For my dear father, words will not explain how I am grateful for him, he is one of the main factors of success in my entire life. He always encourages me to get higher academic degrees. For my beloved mother, who ultimately supports me in all possible ways she can. Without her support I would never have been able to achieve my goals. For my wife, words cannot describe how lucky I am to have her in my life. She did more than her best. She prepared the suitable environment for me even though we have two little boys who sometimes are very hard to control.

My deepest gratitude and special thanks to my supervisors: Dr Polychronis Koutsakis who has guided me throughout my Ph.D. journey, Mr Peter Cole and Dr Giles Oatley who provided a full support to me and gave me invaluable advices to promote the basic idea of this thesis. I am indebted to Murdoch University and yStop company for granting me a full-time scholarship to undertake a Ph.D. in Information Technology in 2018. Many thanks to Department of Jobs, Tourism, Science and Innovation Western Australia for granting Science Industry Ph.D. Fellowship Award 2018.

List of Publications

Journal Articles

J1. Mohamed A. Nassar, Len Luxford, Peter Cole, Giles Oatley, Polychronis Koutsakis, "Wifi-Based Localisation Datasets for No-GPS Open Areas Using Smart Bins," Computer Networks, Vol. 180, 2020, 107422.

J2. Mohamed A. Nassar, Len Luxford, Peter Cole, Giles Oatley, Polychronis Koutsakis, "The current and future role of smart street furniture in smart cities," IEEE Communications Magazine, Vol. 57, No. 6, 2019, pp. 68-73.

Conference Proceedings

P1. Mohamed A. Nassar, Len Luxford, Peter Cole, Giles Oatley, Polychronis Koutsakis, "Adaptive Low-Power Wireless Sensor Network Architecture for Smart Street Furniture-Based Crowd and Environmental Measurements", in Proceedings of the 20th IEEE International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2019.

P2. Mohamed A. Nassar, Len Luxford, Peter Cole, Giles Oatley, Polychronis Koutsakis, "Secure and Privacy-Preserving Real-time Dynamic Crowd Measurement System Using Smart Street Furniture", selected as one of the nine papers that were presented at the Safeguarding Australia Conference, Canberra, 9 May 2019.

Final paper to be included in Technology Surprise: A Compendium of Papers (awaiting publication) Defence Science and Technology Group.

Summary of Publications' Contribution in the Thesis

Chapter	**Publications**
Chapter 1 Introduction	J1, J2, P1, P2
Chapter 2 Adaptive Low-Power Wireless Sensor Network Architecture for Smart Street Furniture-based Crowd and Environmental Measurements	P1, P2
Chapter 3 Wifi-based Localisation Datasets for GPS denied Open Areas Using Smart Bins	J1
Chapter 5 Conclusion and Future work	J1, J2, P1, P2

List of Abbreviations

Abbreviation	Description
AP	Access Point
API	Application Programming Interface
BDR	Bluetooth Detection Records
CSR	Cellular Service Records
GPS	Global Positioning System
GTM	Geotagged Social Media
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IoT	Internet of Things
MAC	Media Access Control
MBCLAP	MurdochBushCourtLoC-AP dataset
MBCLFP	MurdochBushCourtLoC-FP dataset
PTSC	Public Transportation Smart Card
RFID	Radio Frequency Identification
RP	Reference Point
RSS	Received Signal Strength
SUBVS	Smart Urban Bus Navigation System
SWM	Smart Waste Management
UV	Ultra-violet
WSN	Wireless Sensor Network

List of Figures

List of Tables

Statement of Acknowledgement

The research work/contribution in each chapter has been undertaken by the student. He was responsible for doing the research practically and preparing the papers with advice, suggestions and corrections by the co-authors. The student is the principle author of each paper.

Student	Principal Supervisor
Mohamed Ali	Polychronis Koutsakis

Polychronis Koutsakis
Digitally signed by Polychronis Koutsakis
Date: 2020.09.18 17:30:51 +08'00'

Chapter 1 Introduction

1.1 Overview

The continuous increase in population and the volume of human activities leads to more frequent crowded scenes of pedestrians especially in open areas. Real-time measurement of crowd dynamics has been growing in interest as it has many applications including providing real-time identification and monitoring of potential risks and emergency situations, real-time alert systems for emergency personnel and pedestrians, and real-time effective evacuation plans for pedestrians. However, understanding pedestrians' movement and flow in real-time is a great challenge because it requires estimating pedestrians' locations for open areas with GPS signal reception and/or lack of infrastructure to install expensive solutions such as video-based systems.

There have been many developments in managing urban challenges, broadly subsumed under the concept of "Smart Cities". For instance, many cities have utilized street furniture such as bins to achieve efficient waste management. They have equipped the bins with Internet of Things (IoT) devices which enable them to measure different environmental conditions including the level of garbage, air quality, and Ultra-violet (UV) radiation. Street furniture provides excellent coverage across different locations such as parks, streets, homes, universities, etc.

This chapter describes smart street furniture and its current role in smart cities. We discuss the potential role of smart street furniture in providing real-time crowd measurements. The chapter provides a description of existing solutions that measure crowd behaviour and their limitations. Also, we list our research objectives and develop the basic rationale for the Ph.D. thesis. Finally, the organisation of the thesis is explained.

1.2 Smart Cities

There has been significant progress in recent years in addressing and managing urban challenges, broadly characterised by the idea of “smart cities”. These challenges include significant energy consumption, traffic congestion, environmental pollution, urban planning, and huge waste. According to the IBM definition [1], a smart city is a city that is:

- Instrumented, which tends to refer to using sensors, smart devices, and appliances to collect real data from the surrounding environment
- Interconnected, which refers to the ability to connect and integrate data from different sources to provide information throughout city services
- Intelligent, which can broadly be defined as using modeling, visualization and analytical techniques that improve decision making.

A typical smart city ecosystem includes eight significant components:

1) Smart Economy: Support of urban growth, employment and business development using technology and innovations.

2) *Smart People:* Enhancement of people’s creativity and innovation.

3) Smart Governance: Using technology to facilitate government services (i.e., quick, measurable, affordable and sustainable services).

4) *Smart Transportation/Mobility:* Transportation networks (public and private) with real-time control and monitoring systems (ex. Urban Bus Navigation (UBN) Systems).

5) *Smart Living:* Improvement of quality of life using innovative applications.

6) Smart Services: Exploitation of ICT for city services (health, education, safety, surveillance, etc.).

7) *Smart Environment:* Efficient natural resource management using ICT (waste management system, pollution monitoring, etc.).

8) *Smart Infrastructures:* City infrastructure (streets, building, water networks, etc.) with embedded smart technology (sensors, wireless connectivity, etc.).

Smart cities are intrinsically linked to IoT, the technology paradigm referring to a massive number of objects (electronic devices, mechanical and digital machines) that have a unique identification and are connected (together and to the Internet) using wireless networks. IoT has many applications including monitoring and control, big data, and business analytics; hence, it plays an essential role in all components of smart cities.

1.3 Current Role of Smart Street Furniture in Smart Cities

Street furniture refers to objects installed in streets for various purposes. These include benches, bins, traffic lights and bus shelters. This furniture can be considered smart as it is equipped with environmental sensors, wireless modules, processors and microcontrollers. Accordingly, it is possible to connect it to an IoT infrastructure which will enable it to become one of the drivers of future smart cities. Figure 1 shows an example of a smart rubbish bin used for waste management and advertisement.

Figure 1: An example of a smart rubbish bin [2]

Although street furniture provides excellent coverage across different locations such as parks, streets, and homes, recent studies have focused only on limited applications related to smart waste management (SWM) and smart services [3-10].

For SWM, a recent article [11] highlighted that waste is becoming a severe problem for Australia as recently the cost of waste management has increased by 400–500 percent. On the other hand, there is no optimization in the control of resources, for example, static schedules for trucks to empty rubbish bins in different suburbs regardless of different areas, different filling behaviour, and separate sessions. Consequently, it becomes a big challenge to build a scalable Wireless Sensor Network (WSN) to provide real-time recommendations for SWM.

For smart services, due to the rapid increase in population and mobility, there is an urgent requirement to provide a real-time monitoring and reporting system for different resources including air, water, and power. Moreover, it is crucial to engage community members with governments. Accordingly, smart street furniture is being utilized to provide governments with platforms that provide real-time reporting systems and facilitate the interaction between authorities and people in a responsive manner to provide necessary recommendations.

1.3.1 Smart Waste Management

Many cities are utilizing garbage bins to enhance the process of waste management. These bins are called smart bins because they are equipped with multiple sensors that enable them to measure different environmental conditions including the level of rubbish and the contamination level [3, 4, 6, 7].

The recent research efforts can be classified into two main streams.

1) Dynamic Waste Collection Solution

The research works in this stream focus on proposing systems which achieve dynamic waste collection. The authors in [7] have proposed a system which has two main modules: first, semi-static routing module (offline routing module) in which the module provides the necessary computations to provide the shortest path per suburb for covering all full smart bins, and second, dynamic module (online routing module) in which it finds an alternative route with minimum distance weight in case of any issue in the road segment.

2) Efficient Data Collection Architecture

In this stream, the relevant research efforts are trying to provide an efficient architecture for the data collection tier. The authors in [8] have proposed an architecture for data collection which aims to reduce the power consumption at each smart bin because they run using batteries. For smart bins, it has been noticed that the power consumption for communications is much higher than the power consumption for computations. The suggested architecture is based on using an 802.15.4 low power wireless module on each smart bin. The smart bins are clustered, and each cluster communicates with the gateway using a mesh topology. Moreover, authors have utilized a duty cycle method to reduce the power consumption of sensors by keeping sensors in sleep mode while they are idle.

1.3.2 Smart Services

Several initiatives are underway to utilize street furniture for providing smart services [12, 13]. The authors in [13] provide WSN architecture that can be attached to street furniture to measure air quality. The proposed architecture is scalable and uses cheap air quality sensors that enabled them to perform experiments in the Cambridge area in the United Kingdom. In [12], an interdisciplinary team from the University of New South Wales has been awarded a federal grant. The research project will utilize picnic tables, bins, barbecues, seats, ash receptacles, bubblers, and lights. Smart sensors will be installed on street furniture to monitor and respond in real time. These will measure usage, including water and power consumption, and will provide live messages to the

Council. The work is still in its early stages, as it does not provide a real-time solution to enable the Council to communicate with pedestrians.

1.4 Crowd Measurement and Street Furniture

The continuous increase in population and the volume of human activities leads to more frequent crowded scenes of pedestrians such as travellers, workers, athletes, or students. Efficient management of crowds is essential in achieving customer satisfaction in events, effective urban planning and decision making for governments, efficient public transportation services, active community engagement and education, effective advertisement by companies, safety and security of people. Efficient crowd management requires measurements of crowd dynamics including crowd density and flow. For example, crowd density and crowd flow enable different entities to provide a secure, healthy and safe environment. Also, in life-threatening situations such as fires and explosions, they could be utilized to provide efficient evacuation plans.

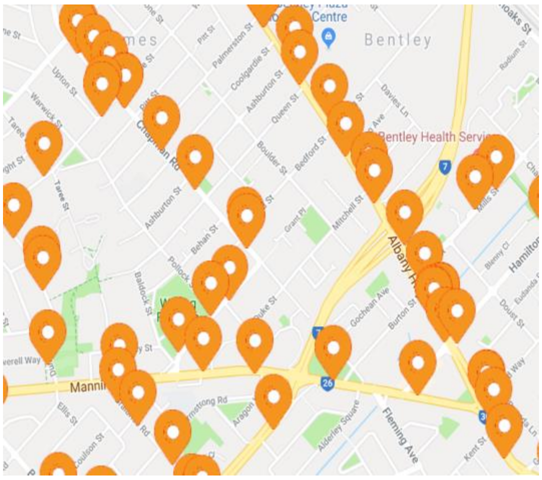
The techniques and methods used for dynamic crowd measurements can be classified into two streams: 1) Simulations and Modelling Stream, and 2) Crowd Behaviour Tracking Stream. In Simulations and Modelling stream, many research works utilize complex simulations and models to measure crowd behaviour [14-18]. The proposed solutions, however, have significant shortcomings as the simulations and models are based on conditions and assumptions which can lead to inaccurate results.

In the second stream, several relevant works have proposed different methods based on crowed video analysis, Geotagged Social Media (GTM), Radio Frequency Identification (RFID), Public Transportation Smart Card (PTSC), GPS, Cellular Service Records (CSR), Received Signal Strength (RSS), and Bluetooth Detection Records (BDR) to understand human mobility [19-22].

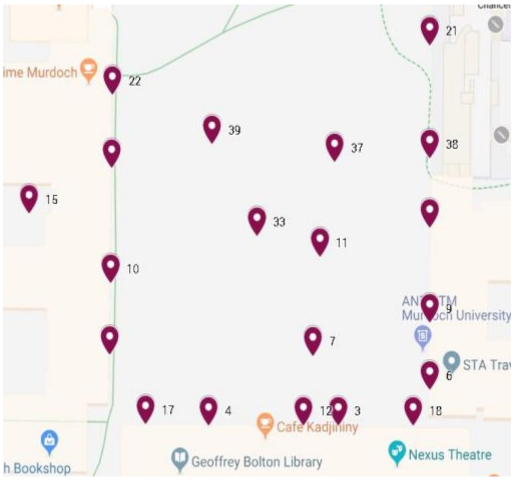
This stream generally faces scalability challenges as it has not been possible until now to apply the methods in large scale due to high setup costs and huge required computational power. This stream uses identification information about people, therefore it also raises privacy-preserving, data security, and ethical issues [23].

RSS-based methods exploit the Wifi-enabled devices standard to measure crowd behaviour in which mobile devices frequently send Wifi probe requests to the surrounding Access Points (APs) to accelerate the Wi-Fi connection process. These probe requests contain Media Access Control (MAC) addresses and RSSs are recorded by the receiver. They outperform other methods because 1) unlike GPS they provide fine-grained spatial and temporal data given that there are suitable amount of APs covering the area [22, 24], 2) they provide high precision localization [24-26], and 3) they can measure crowd behaviour for different types of crowds in different locations and modes such as transportation modes, between buildings, inside buildings, and between city zones [22].

Although smart street furniture has recently attracted some research work, there is still much to do for solving urban problems especially crowd measurement and urban planning. It goes without saying that street furniture provides excellent coverage across open areas such as parks, streets and universities. Figure 2 shows two examples of street furniture distribution: 1) macro level distribution as in a part of city of Canning, Western Australia (Figure 2 (a)), and 2) micro level distribution as in a central open area at Murdoch University (Western Australia), known as Bush Court (Figure 2 (b)).



(a) Distribution of street furniture in a part of the city of Canning, Western Australia [2]



(b) Distribution of street furniture in Bush Court of Murdoch University

Figure 2: Two examples of street furniture distribution in macro and micro levels

1.5 Thesis Objectives

This thesis aims to propose a novel localisation approach based on a suitably augmented smart street infrastructure rather than depending on a device-based profiling-data collection for localization. The proposed approach ensures scalability, less cost, privacy preservation and real-time crowd localisation. This is done by achieving the following objectives:

1) **Proposal of an innovative WSN for smart street furniture**. There are many concerns which limit the smart street furniture role to particular applications such as waste management. The biggest challenge is power supply, with smart furniture such as smart bins working on batteries which makes it hard to add more sensors. Other issues are related potential risks associated with data security and privacy, and possible adverse events related to ethical issues. Consequently, we aim to design an adaptive real-time low-power wireless sensor network for crowd and environmental measurements. The proposed architecture includes a real-time low-power design for wireless sensor nodes which efficiently communicate the updated data instead of

sending the same data on a regular basis. The proposed solution maintains data privacy and security in different system tiers.

2) **Generation of the first Wifi-based localisation datasets for GPS denied open areas, which will be made publicly available.** Over the past decade, many datasets have been introduced to enable researchers to compare different localisation techniques. Existing datasets, however, have failed to cover open areas such as parks in cases where GPS is still unavailable, and there is a lack of Wifi access points. Also, the existing datasets only focus on getting Wifi fingerprint collected and labelled by users. To the best of our knowledge, no dataset provides RSSs collected by Wireless Access Points (APs) for GPS denied open areas. We aim to scale our innovative wireless sensor node architecture across all rubbish bins in Bush Court of Murdoch University. This will allow us to offer two datasets publicly: 1) the Fingerprint dataset MurdochBushCourtLoC-FP (MBCLFP) in which four users generate accurate and consistently labelled Wifi fingerprints for all available Reference Points (RPs), 2) the APs dataset MurdochBushCourtLoC-AP (MBCLAP) that includes auto-generated records received from all users' devices, including those who generated MBCLFP.

3) **Proposal of a real-time localisation approach using both MBCLFP and MBCLAP.** MBCLFP is a labelled dataset but requires users to participate using a mobile application and stand at each RP to report RSS [26-31]. On the other hand, MBCLAP is quite large because it is an auto-generated dataset by APs for all users in Bush Court, i.e., it doesn't require human intervention, but it is not labelled. We aim to provide a lightweight real-time accurate localisation approach that utilises the two datasets and a deep learning classifier. This approach includes: 1) developing a light-weight algorithm to label the MBCLAP from MBCLFP for the four users' smartphones, 2) homogeneity test for the two datasets, 3) data cleaning and feature engineering and 4) the four-layer deep learning classifier.

1.6 Thesis Organization

The remainder of the thesis is organised as follows.

Chapter 2 starts with existing techniques and methods for crowd measurements, and their limitations. We present in detail our innovative wireless sensor network designed for street furniture and how it allows street furniture to measure crowd density and environmental conditions. At the end of the chapter, experiments and results are discussed.

Chapter 3 explains the process of generating the unique localisation datasets (MBCLFP and MBCLAP) for Bush Court of Murdoch University as an example of GPS denied open area. Then, the developed light-weight algorithm to label the MBCLAP from MBCLFP is explained. Finally, we present some facts, statistics and visualisations for both datasets including a homogeneity test using t-test.

Chapter 4 proposes a localisation approach that converts MBCLAP from asynchronous format to synchronous. By converting MBCLAP to synchronous form, we are able to utilise the data received from APs to improve the localisation model accuracy. The chapter applies feature engineering and four-layer deep learning classifier for MBCLAP and MBCLFP. Finally, we demonstrate that by using this approach we achieve higher prediction accuracy with up to 19% improvement, compared with the case of using only MBCLFP.

Chapter 5 provides the conclusions of our research work. We also outline areas for future work where it is possible to utilise our proposed wireless sensor network architecture to provide a real-time crowd measurement system including pedestrians' flow. Moreover, we explain that based on crowd movements, it is possible to build a real-time context-aware recommendation system to provide pedestrians with targeted contents using digital signage.

Chapter 2 Adaptive Low-Power Wireless Sensor Network Architecture for Smart Street Furniture-based Crowd and Environmental Measurements

2.1 Overview

Street furniture such as bins, seats and bus shelters can become “smart” with the inclusion of wireless sensor nodes, which consist of environmental sensors, wireless modules, processors and microcontrollers. One of the most crucial challenges for smart street furniture is how to manage power consumption efficiently without affecting data freshness. In this work, we propose a novel WSN architecture for smart street furniture which has the following main features:

- **Two-way communication model.**

The WSN extends the one-way communication model, where wireless sensor nodes push the data to the server periodically, to a two-way communication model. Thus, the server manages, controls and synchronises all wireless sensor nodes. This includes the dynamic adaptation of the time interval for measurements for each wireless sensor node to balance between power consumption and data freshness.

- **Adaptive real-time low-power design for wireless sensor nodes.**

Generally, each wireless sensor node is designed to have two modes: (1) active mode where it communicates with the server frequently and (2) sleep mode to save battery power. Our approach reduces the power consumption in the active mode, by using a customised configuration for each wireless sensor node, for the frequency of sensing and communicating the data with the server. The server is responsible for this customised configuration which is based on the context of each wireless sensor node's operation characteristics (date/time, location, weather, session, event places, etc.)

- **Network and storage optimisation.**

Wireless sensor nodes do not send redundant environmental data and only send the updated data to achieve lower network traffic and less database storage.

- **Crowd level measurement.**

To the best of our knowledge, this is the first work in the relevant literature which extends the functionality of the wireless module to act not only as a station sending environmental data but also as a soft AP sensing MAC addresses and Wifi signal strengths from surrounding Wifi-enabled devices. Accordingly, the wireless sensor nodes are not only utilized to communicate data to and from the server, but also to measure levels of crowd density and flow in real-time.

- **Access the internet using any existing wireless network.**

Wireless sensor nodes do not require any special requirements to access the internet as they can use any existing wireless network with different security protocols including WPA2-Enterprise, which is available in large organizations and universities.

- **Secure and privacy-preserving.**

No personal information is gathered from devices (only the MAC address, similarly to what is done by any AP), and the proposed solution maintains privacy and security in different system tiers, following the Australian Privacy Principles [53] and the European Union General Data Protection Regulation [54].

We have conducted experiments on Bush Court of Murdoch University campus and our results show that our proposal improves lifetime of wireless sensor nodes up to 293% compared to static architectures similar to the ones that have been proposed in the literature. Moreover, network traffic is improved up to 38% without affecting data freshness. Finally, storage space for the database at the server is reduced up to 99%.

2.2 Existing Work

2.2.1 Three-tier Architecture of Internet of Things

Generally, applications of smart cities are based on a three-tier architecture of IoT [10]. Figure 1 shows this architecture. The Data Generation tier contains wireless sensor nodes which consist of groups of embedded sensors that are connected wirelessly. These embedded sensors measure environmental conditions such as pollution level, temperature, etc. This data is sent to the second tier, Centralized Computing (Middleware). It is important to mention that the communication between wireless sensor nodes and Centralized Computing could be extended to be a two-way communication including feedback from the central processing location to control some sensors and actuators. Generally, wireless sensor nodes work on distributed and/or harsh environments on batteries. Accordingly, they have limited power, computational and network capabilities. According to recent work [3, 4, 7, 10, 32-34], the different devices that have been used for WSNs are summarized in Table 1.

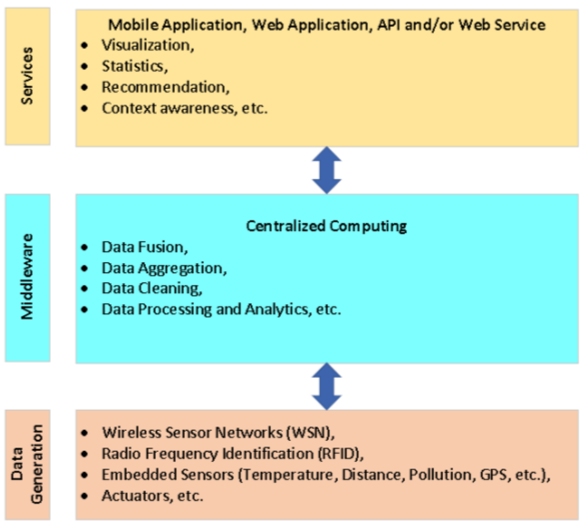


Figure 3: Three-tier architecture of IoT

Table 1: Different devices used for WSNs

Device Name	Usage
RFID reader (i.e., RC522)	• Identifying the type of wastes which are tagged by RFID tags. • Identifying and tracking WSN nodes by labelling them with RFID tags.
Actuator	• Locking the WSN node. For example, locking the smart bin until it is emptied.
GPS/ General Packet Radio Service (GPRS) module (i.e., SIM808)	• Identifying the location information where the system becomes able to keep track of the current location of WSN nodes. • Communicating with the central computing service directly using Subscriber Identity Module (SIM).
Microcontroller board (i.e., Arduino Uno R3)	• Converting raw readings from different sensors to data packets. • As a bridge between sensors and the computing unit (i.e., RaspberryPi). • Sending the data packets to the wireless communication module.
Sparkfun Air Quality Sensor (i.e., SEN-14193)	Measurement of air quality including total volatile organic compounds, equivalent carbon dioxide, and metal oxide.
Ultrasonic Sensor (HC-SR04)	Measurement of the distance from the sensor to an object. For example, it is used to measure the garbage level in smart bins.
Wireless communication module	Creating wireless networks between WSN nodes and centralized computing service. Four different modules have been used: • **LoRaWaN** supports low power, low data rate, and long-range wireless networks. • **IEEE 802.15.4 wireless module (ZigBee)** supports low power, low data rate, and short-range wireless networks. • IEEE **802.11 b/g/n wireless module (ESP8266)** supports low power, high data rate, and short-range wireless networks. • **WI-SUN** supports low-power, low data rate, and long-range wireless networks.
Single-board computer (i.e., RaspberryPi)	• Processing data packets (i.e., filtering and aggregating data packets) which are sent by a microcontroller (i.e., Arduino Uni R3). • Sending the processed data to the wireless • communication module (i.e., LoRaWaN).

2.2.2 Smart Environment

A recent article [11] showed that waste is becoming a severe problem for Australia as recently the cost of waste management has increased by 400%-500%. On the other hand, there is no optimization in the control of resources, e.g., static schedules for trucks to empty rubbish bins in different suburbs regardless of different areas, different filling behaviour, separate sessions.

Consequently, several works are focusing on providing solutions based on smart bins.

The relevant research efforts have focused on providing an efficient architecture for the data collection tier. Authors in [8] have proposed an architecture for data collection which aims to reduce the power consumption at each smart bin because they run using batteries. For smart bins, it has been noticed that the power consumption for communications is much higher than the power consumption for computations [35]. The suggested architecture is based on using a ZigBee wireless module on each smart bin. The module consumes low power and covers short-range communications. Therefore, the smart bins are clustered, and each cluster communicates with the gateway using a mesh topology to ensure availability. Moreover, the authors have utilized a duty cycle method to reduce the power consumption of sensors by keeping sensors in sleep mode while they are idle. Although the architecture in [8] has an excellent contribution to the power consumption, it also suffers from significant weaknesses as follows:

- The sensors measure the level of rubbish in a static time interval. The static period could be studied more deeply to adjust it dynamically, e.g., by taking into consideration different seasons and holidays.
- Experiments have been performed based on the synthetic structure of bins, i.e., in actual suburbs with different barriers such as building, the architecture might not perform well. Using LoRaWaN, which provides low-power and long-range wireless connectivity, might reduce the complexity of mesh topologies and give a broader range for wireless connectivity.

Kristanto et al. [6] proposed a dynamic polling algorithm for low power consumption. The proposed algorithm has two main phases. The first phase is the data collection where data is collected over time for a static time interval and sent to the server. The second phase is the analysis where the server performs computations to find the highest probability for rubbish to reach a threshold level. The proposed algorithm is based on the assumption that bins are filled in a fixed time interval. The server adapts the time interval by sending the sleep time to smart bins. The proposed algorithm used in [6] has the following challenges:

- It doesn't specify the length of the first phase.
- The experiments were limited to just one smart bin and for a period of five days in the same environmental conditions.
- The solution is not scalable for different smart bins with different filling behaviour.
- The suggested algorithm is based on the assumption that the rubbish level reaches a threshold in a fixed time interval. However, in public holidays certain areas like schools and universities become less crowded, hence this assumption is not always valid.

2.2.3 Crowd Measurements

Over the past few decades, measurement of crowd dynamics has gained much interest. The techniques and methods of measuring crowd dynamics are generally classified into two main categories as follows.

1) Simulations and Modelling Stream

In this stream, many methods have been proposed to build complex simulations and models to measure crowd behaviour [14-17]. Many of these methods include multi-models to simulate the crowd behaviour of a specific scenario such as pedestrians walking on zebra-crossing [36]. The proposed models, however, have the following significant shortcomings [37]:

- The simulations and models are based on limited data, conditions, and assumptions which leads to inaccurate results, especially in emergency cases.
- It is impossible to build a model and/or a simulation that works for different crowd types and scenarios.

2) Crowd Behaviour Tracking Stream

In this stream, several relevant works have been proposed based on different techniques. The works in [19-21] introduced video surveillance solutions. These solutions have many processing layers including object classification, motion detection and motion analysis. Moreover, Neural Networks are employed for feature extraction and tracking [20]. Although many techniques and algorithms have been developed, current video surveillance-based solutions have the following shortcomings:

- They require a significant amount of computational power, especially when it comes to a high density of population [20, 38].
- They are a non-scalable and expensive solution as setup cameras, high-speed internet connection, a massive amount of space to store data from different places, and powerful servers to process data cannot be applied in all areas inside a city [21, 36, 38-40].
- They are based on limited data, conditions, and assumptions regarding the environment.

Another crowd behaviour tracking technique is Wifi-based tracking, which has gained much interest due to the dense Wifi coverage [22, 24-26, 41-43]. Accordingly, Wifi-based techniques are more scalable and lower cost. Wifi-enabled devices send probe requests to surrounding APs. Each probe request contains the signal strength and MAC address of the device. The signal strength is an excellent indicator for distance [24, 44].

In [10], the authors have proposed a Smart Urban Bus Navigation System (SUBVS) which provides crowd-aware route recommendation and predicts crowd levels on bus journeys. Consequently, the system suggests less crowded routes, and eventually people become encouraged to use public transportation. In the system, buses are equipped with Wifi APs to estimate the number of passengers within the bus. The crowd estimation is based on the fact that probe requests are sent periodically from passengers' Wifi-enabled devices. The server receives real-time occupancy data from buses and uses the data for providing enhanced recommendations for bus users, to help them avoid large crowds. However, the crowd measurement is only performed inside the bus, which means that although the bus may not be crowded, the road can be very busy which may delay the journey more than expected. Also, the system provides only non-personalised recommendations and it does not provide any user profiling and/or modelling for passengers' behaviour. The proposed system could be extended to different means of transportation if street furniture is equipped with Wifi devices. For example, using spatial-temporal analysis for Wifi signals received from pedestrians' devices by distributed smart bus shelters and bins could give a better understanding for crowd dynamics of passengers and hence better recommendations for passengers and councils.

In [22], a sensing device (called SENSg) with environmental sensors and a Wifi module was designed. The devices were given to students to study their mobility patterns in large scale. These devices send probe requests every 13 seconds to the surrounding Wifi APs. Thanks to Wifi APs coverage, a commercial localization service called Skyhook [45] was used to analyse all received data. The work in [22] provides different data analyses including extracting staying points, identifying places of interest and trips, identifying students' homes and schools and identifying transportation modes.

The work in [22] has two significant challenges. The first is a scalability issue as it is impossible to provide everyone with a SENSg device due to the high costs and maintenance overheads. Moreover, the solution cannot be generalised in different cities where there is limited Wifi coverage. The second is a lack of real-time responses, as the proposed work does all these analyses offline after gathering all data from different APs. In addition, the device sends data every static period (13 sec) which may affect the freshness of the data.

Additionally, to video surveillance and Wi-Fi based tracking, other works in the literature have proposed systems which measure crowd dynamics using RFID technology, cellular service records and GPS [46-48]. The systems based on RFID are limited to structured crowd scenes. Those using cellular service records suffer from large spatial granularity which leads to inaccurate localisation [22]. Moreover, the GPS-based systems are not able to measure crowd dynamics in hyperlocal locations, such as inside buildings, parks or within a small area [49].

2.3 Our Proposed System Architecture

2.3.1 Overall System Architecture

In our approach, we take into consideration both the feasibility and the cost of the proposed solution. Unlike existing solutions (presented in the previous section), our proposal does not require any special hardware or protocols. Figure 4 provides the overall system architecture to achieve an adaptive real-time WSN for both environmental and crowd measurements as follows:

1) Wireless sensor nodes are built in street furniture assets such as bins, seats and bus shelters.
2) Each wireless sensor node senses probe requests from surrounding Wifi-enabled devices and environmental conditions including temperature, humidity, garbage level (the terms “garbage” and “rubbish” are being used interchangeably in this thesis), air quality, etc.
3) The designed wireless sensor nodes access the internet through any existing wireless network infrastructure.
4) Each wireless sensor node directly communicates with the server, unlike other solutions that depend on complex topologies such as mesh topology. These solutions require high cost of setup and maintenance. The direct communication reduces communication overhead and provides better control for wireless sensor nodes.
5) The server sends each wireless sensor node a specific configuration to use. The configuration includes the frequency of sensing data and period of

sensing, sleep time, etc. The configuration depends on the context of operation of each wireless sensor node (date/time, areas, events, etc.).

6) Wireless sensor nodes send periodically environmental data and probe requests as raw data to the server.

7) The server stores this data in the database for further processing.

8) Many real-time and offline applications, including real-time crowd measurements, can be implemented using the stored data.

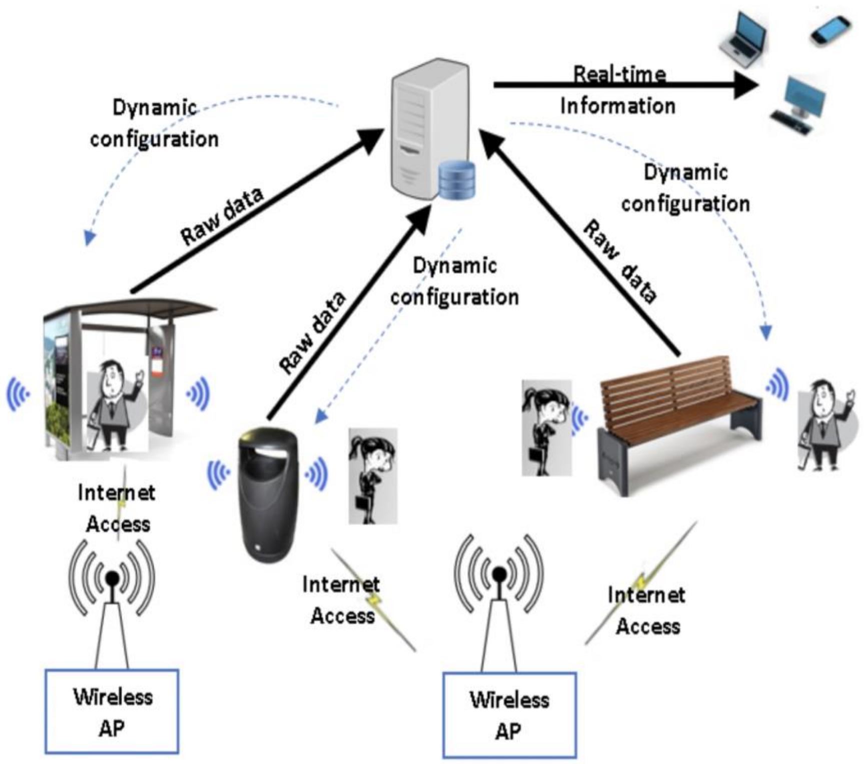


Figure 4: Overall system architecture

Figure 5 presents the two-way communication model between the server and the wireless sensor node. Using interactive communication, we can achieve network and power efficiency. The idea behind the proposed communication model is that each wireless sensor node gets a specific configuration from the server based on its context of operation such as events, places, data and time. In certain situations, the server responds with two parameters, Freq (1/f) and Period (p), measured in seconds^{-1} and seconds, respectively (Active scenario, Figure 5(a)). Consequently, the wireless sensor node every f seconds sends raw data to the server for a period of p sec (Sending raw data scenario, Figure *1*5(c)).

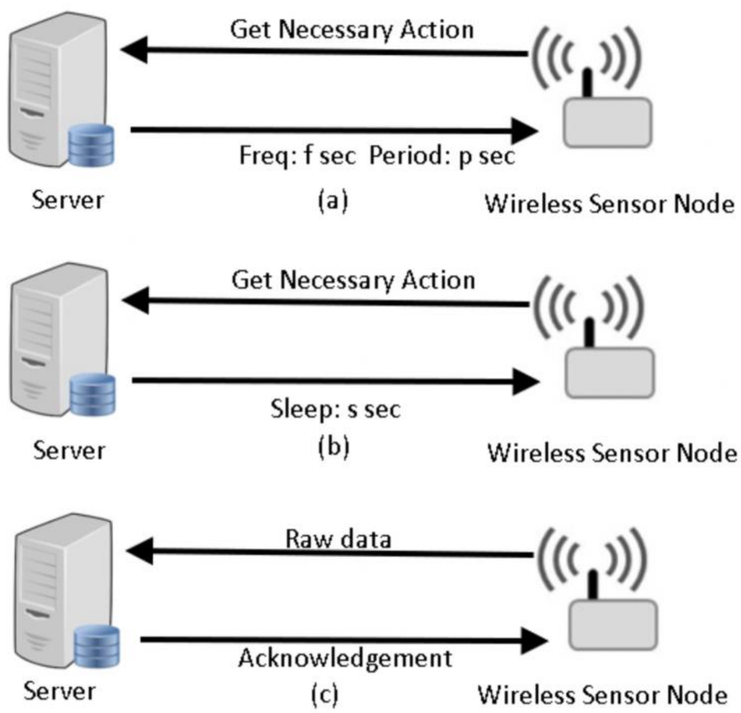


Figure 5: Communication model (a) Active scenario; (b) Sleep scenario; (c) Sending raw data scenario

It is important to mention that the *f* and *p* values vary from area to area based on the situation. For example, in a school area between 8 am and 10 am, f could be a small number because this place is very active at this time. When no events are expected to take place, there is no need to activate the wireless sensor nodes. Accordingly, the server responds with the Sleep parameter s, also measured in seconds (Sleep scenario, Figure 5(b)), through which it instructs the wireless sensor node to go into in sleep mode for s seconds to save power. An example of this scenario is a University area at night, where there may be no need to keep the nodes active. Finally, for security reasons, each request from a wireless sensor node is associated with authentication details. Hence, there is no overhead on the server to maintain a separate session for each wireless sensor node.

2.3.2 Design of Hardware and Software of Wireless Sensor Nodes at each Node

As the wireless sensor node gathers not only environmental conditions, but also identifies pedestrians using Wifi probe requests sent from their Wifi-enabled devices, we depend on two main components: 1) Arduino Uno R3 which is the most used and documented board of the whole Arduino family [50]. There are many sensors which are compatible with Arduino. Table 2 summarizes relevant sensors that can be connected to Arduino, and 2) ESP-13 Wifi shield which is a famous version of ESP8266 family. It enables Arduino to access the internet, and it can act as a soft AP or web server [51].

Table 2: A set of relevant sensors that can be connected to Arduino

Sensor	**Usage**
Sound Sensor Module	detect loudness of the sound
Infrared Emission Sensor Module	detect near-infrared light
Temperature Sensor (LM35 Sensor)	detect temperature
Vibration Switch Module	allows detection of shock and vibration
Photoresistor/Light sensor/Photosensitive resistance element sensor)	set to detect light intensity
Flame Sensor	set to detect the density of flame
UV Sensor	set to detect UV
Temperature sensor module	set to detect the temperature.
Temperature and humidity sensor module	set to detect the temperature and humidity
Air Quality Sensor	set to detect air quality including total volatile organic compounds, equivalent carbon dioxide, and metal oxide
Ultrasonic Sensor	set to detect the distance
Motion Detector	set to detect the motions

Figure 6 presents the flowchart for the operation of the Wifi module and Figure 7 the flowchart for efficient communication between ESP-13 and Arduino Uno R3, and between ESP-13 and the server. These flowcharts are applicable for any Wifi module and any microcontroller. In Figure 6, ESP-13 module requests the configuration from the server which is mainly one of two configurations as follows: 1) sleep for s seconds (accordingly, the module goes to deep sleep mode which is the most power efficient option as it stops everything except Real Time Clock [52]), 2) sense every f sec for a period of p sec. In this case and to achieve power, storage, and network efficiency, the module communicates with the server only if within every f sec there are one or more sensed probe requests and/or environmental conditions updates. For example, if the difference between current waste level and last sent waste level for a bin to the server is less than or equal to a threshold, then there is no need to resend the data to the server and consume more storage at the server database.

The ESP-13 module can work as a soft AP to sniff probe requests using an interrupt which can be seen at the right of Figure 6. It also acts as a station accessing any existing wireless networks with different security protocols including WPA2 and WPA2-Enterprise. Moreover, the ESP-13 module can communicate with Arduino Uno R3 using serial communication in which ESP-13 interrupts Arduino Uno R3 requesting updated environmental data.

Arduino Uno R3 (Figure 7 flowchart) does not sense any data until getting interrupted by ESP-13. Consequently, it gets new readings from different sensors and it only communicates with ESP-13 the updated data. The right part of Figure 7 shows the ESP-13 interrupt which makes Arduino Uno R3 sense environmental conditions.

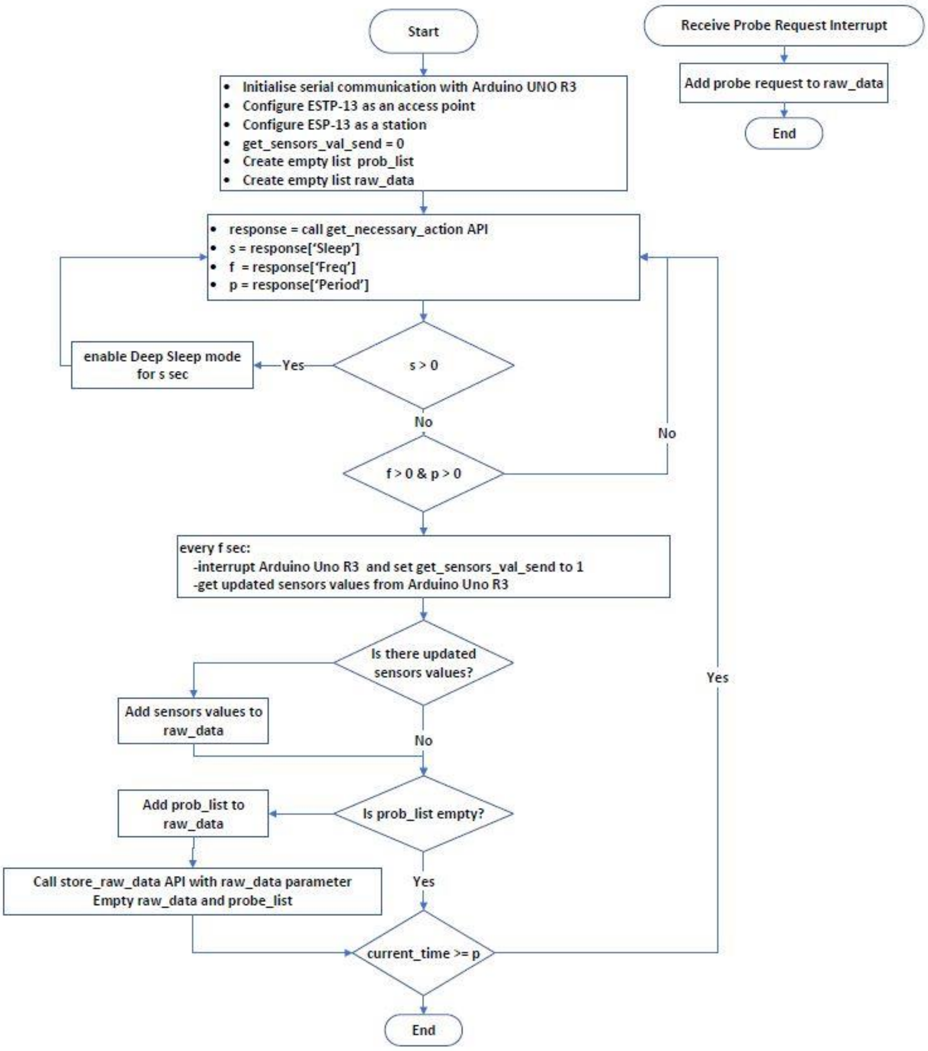


Figure 6: Flowchart for ESP-13 Wifi module

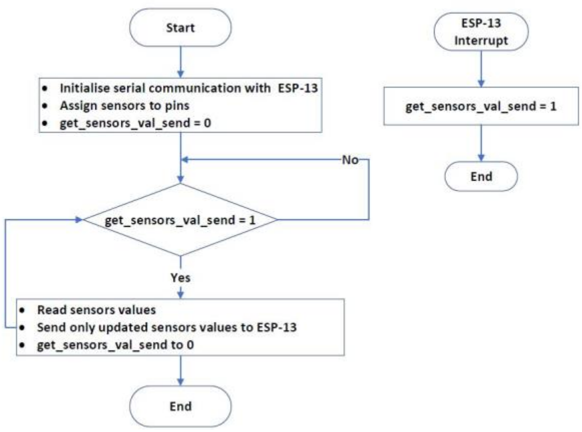


Figure 7: Flowchart for Arduino Uno R3

At the server side, there are two main Application Programming Interfaces (APIs). The first one is get_necessary_action which provides the required configuration as discussed in the previous subsection. It is worth mentioning that in long periods expected to be idle (e.g., summer holidays at schools/universities) the server will ask wireless sensor nodes to sleep for a predefined maximum period t hour instead of sleeping for the whole period. This choice is made to avoid cases where special events may be organised during that period.

The second API is called store_raw_data to store raw data to the database. It is worth mentioning that the server is fully responsible for scheduling tasks for different wireless sensor nodes because it holds all information about different circumstances including rush hours, holidays, working dates/hours, future events, residential zones, industrial zones, etc.

2.3.3 Data Security and Privacy-Preserving Aspects

Even though Wifi probe requests can provide excellent special and temporal data, until now there is no solution for measuring dynamic crowds in real-time due to issues associated not only with infrastructure, design and power, but also privacy-preserving, ethical and data security issues as follows.

1) Privacy-Preserving and Ethical Issues

MAC addresses and RSS sent from Wifi-enabled devices are used to identify people's locations, which raises privacy and human ethical concerns. Accordingly, to apply any RSS-based solution, it is required to follow these guidelines:

- the Australian Privacy Principles [53].
- the EU General Data Protection Regulation [54].
- Avoid publishing any part of the data.
- Provide awareness for people about the solution including how data are gathered and processed.

- Put clear static signs describing the solution around sensor nodes and periodically checking that they are in place.

2) Security Challenges

Generally, WSNs face security issues at different levels:

1) Securing Physical Sensor Nodes

It is one of the most challenging tasks because if someone captures a sensor node, the possibility emerges to reverse engineer the software and receive sensitive information from it including the encryption algorithms used [55]. Therefore, it is important to avoid this scenario. If it happens, it is crucial to minimise the risk of getting confidential information by reducing key-sharing between neighbouring nodes and efficiently structure the topology [56].

2) Routing and Network Security

It is important to limit the role of intermediate sensor nodes so that they cannot remove or add sensor nodes [56]. Besides, a secure routing protocol guarantees the integrity, authenticity, and availability of data. Insecure wireless communication can give adversaries the ability to manipulate the traffic from and to the server. A secure solution can identify manipulated data and report the issue for further investigation [57].

3) Intrusion Detection

Intrusion detection is important to analyse data within a server or a network. It identifies any security breaches, which include both types of attacks: 1) attacks from outside the system, and 2) attacks within the system. Moreover, intrusion detection includes the ability to recognize patterns of attacks and tracking of user policy violations.

Our solution is:

1) Privacy-preserving and ethical

The solution assures privacy by applying the following actions:

- At **sensor node level**: Only a minimal set of data is gathered (RSS and MAC address). MAC addresses are encrypted using md5, a one-way encryption algorithm, at sensor nodes' level. Accordingly, encrypted MAC addresses are sent through the network securely.
- At **network level**: Hypertext transfer protocol secure (HTTPS) protocol is used for communication between sensor nodes and the server to allow secured data transfer, and Eduroam is used to assure secure and scalable transfer of data from/to the server.
- At **application and server level**: No personal information is gathered that might help to identify people and MAC addresses are stored anonymously.
- Avoiding publishing for any part of the raw data.

According to the actions taken, our solution covered the Australian Privacy Principles [53] and the European Union General Data Protection Regulation [54]. Furthermore, we considered human ethics issues even before starting the project by getting Human Ethics approval from Murdoch University, which included the following actions. We believe that the following actions are required from any entity that wants to implement the proposed solution including councils, governments, stadiums, etc:

- Before starting the project:

Announce to all potential participants about the project through different media. In our case, participants are University students and staff and we used the website and admin email of the University to announce the relevant information about the project. For a system like ours to be applied widely, by local councils or central governments, a public communications campaign would be needed, which would enable the public to understand and hopefully accept the trade-offs between limited loss of privacy and larger safety/better quality of life.

- At/After starting the project:

Add static signage on the smart street furniture (in our case, five bins) which contain information about the project. Figure 8 shows a bin with signage.

Figure 8: A bin with signage

The email, announcement, and signage contain the following information:

- Project title.
- Project purpose.
- Project summary.
- Sensor node components.
- Images show what the sensor nodes look like.
- How students/staff can avoid participating in the project. We provide them with two options: 1) disable Wifi of their devices within the area of sensor nodes or 2) submit their MAC address to the contact person to disable tracking their devices.

- Contact Person and Phone.
- Duration of the project.

2) secure

To achieve **physical sensor node security**, we attached the sensor nodes firmly on the ceiling of the bin so that it is very hard to remove them. To ensure **routing and network security**, we avoided using intermediate sensor nodes and/or complex topology. Moreover, HTTPS protocol is used for the data encryption through the network, and Eduroam is used to assure secure and scalable transfer of data from/to the server. To avoid any **outside intrusion**, we configured each sensor node to send credentials (id and password) at each time it communicates with the server. Accordingly, the server authenticates the sensor node. In the case of failed authentication, all details are recorded for further investigation.

2.4 Experiments and Results

We have deployed the system at Bush Court of Murdoch University because it is considered as a hub for staff and students due to the many facilities surrounding it, including the main library, restaurants and student centre. The deployment includes five bins distributed across Bush Court. Figure 9 shows the hardware components used in this design: (a) Arduino Uno R3, (b) ESP-13 Wifi shield, (c) Temperature sensor to provide environment temperature, and (d) Ultrasonic sensor to calculate the garbage level. We implemented the two flowcharts in Figure 6 and Figure 7 using Arduino IDE [58]. Figure 10 shows one of the bins after adding the hardware prototype.

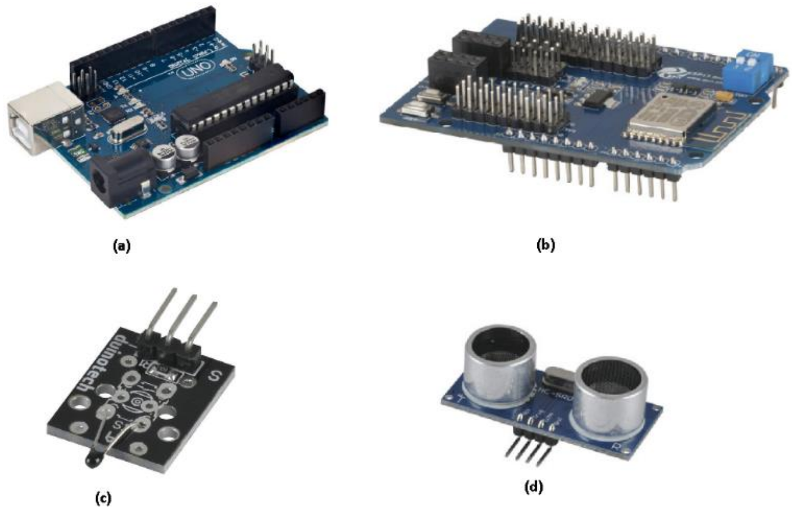


Figure 9: (a) Arduino Uno R3;(b) ESP-13 WiFi shield; (c)Temperature sensor; (d) Ultrasonic sensor

Figure 10: A bin after adding the hardware

On the server side, we developed the two APIs using PHP programming language, and MySQL is used as a relational database management system. We created three tables in MySQL to store three different data types. Figure 11 shows the sample data from three tables. The first table (RSS) stores encrypted MAC addresses and signal strengths received from different bins. The second table (GRB) stores the garbage level collected from different bins. Finally, the third table (TEMP) stores the sensed temperature from different bins.

id	bin	device	rss	time ▾ 1
145695	1017	e7bada2e878705c14aa68ee839defd54	87	2019-01-24 18:46:19.000
145694	1017	71bb5d33d7d97e50e9b5a72684d7d509	89	2019-01-24 18:45:51.000
145693	1017	9ddb91d56b36c046136d6d26c5dee20a	90	2019-01-24 18:45:48.000
145692	1017	e7bada2e878705c14aa68ee839defd54	86	2019-01-24 18:44:19.000
145691	1017	1483f75f02ad6dbb9f46a55dd2be44d6	83	2019-01-24 18:42:56.000
145690	1017	200669ff2e5dedbab487a86ac4fef1d2	81	2019-01-24 18:42:33.000
145689	1017	e7bada2e878705c14aa68ee839defd54	87	2019-01-24 18:42:21.000
145688	1017	1483f75f02ad6dbb9f46a55dd2be44d6	84	2019-01-24 18:40:40.000

(a) RSS Table

id	bin	grb	time
1	1010	148	2018-12-12 11:00:34
2	1010	12	2018-12-12 11:01:35
3	1010	124	2018-12-12 11:01:45
4	1010	147	2018-12-12 11:01:55
5	1010	112	2018-12-12 11:02:25

(b) GRB Table

id	bin	temp	time ▾ 1
1092	1010	211	2019-01-21 17:59:42
1091	1010	197	2019-01-21 17:59:36
1090	1010	216	2019-01-21 17:50:43
1089	1010	231	2019-01-21 17:34:32
1088	1010	220	2019-01-21 17:33:21

(c) TEMP Table

Figure 11: Sample data from different tables

To allow wireless sensor nodes to access the server, an Apache Hypertext Transfer Protocol (HTTP) web server is installed on the server. Also, each wireless sensor node is configured to access the internet using Eduroam which is covering the research and education sector in Australia [59] and in several other countries. Eduroam is a secure and scalable roaming wireless network which uses WPA2 Enterprise for secure, reliable and fast internet access.

To compare different WSNs including our proposal, we prepared the following three different setups:

- ***Static_20s***: In this setup, wireless sensor nodes send data every 20 sec.
- ***Static_10s***: In this setup, wireless sensor nodes send data every 10 sec.

The intervals of 10 and 20 seconds, respectively, were chosen because they are close to the static 13 seconds intervals used in [22].

- ***Dynamic_20s_10s:*** In this setup, which is our proposed approach, the server controls wireless sensor nodes to operate only between 10 am and 4 pm in weekdays but in different frequencies. First, the server instructs wireless sensor nodes to sense (and send) data every 20 sec for three consecutive hours from 10 am to 1 pm. Then, it instructs them to sense data every 10 sec from 1 pm to 4 pm. The different frequencies are related to the fact that Bush Court is significantly more populated from 1pm to 4pm. Finally, the server requests wireless sensor nodes to enter sleep mode until 10 am the next day. The t parameter is selected to be equal to one hour which means that wireless sensor nodes go to sleep for an hour before they communicate again with the server to get a new configuration if needed (e.g., in the case of a special event in the area). This process is repeated until 10 am the next day. Please note that the only reason for using intervals of 20 and 10 seconds in our experiments is to have the same intervals as the static setups, and hence to make the comparison “fairer” to the static setups. In our proposed approach, the server can change the intervals either based on the information it is receiving from the nodes (e.g., the intervals could change when very high or low temperatures are reached, when the waste level is considered to be excessive or more generally when there are reasons for concern in the system recognized through the violation of certain thresholds in the measurements) or, as mentioned above, in the case of special events in an area. The dynamic nature of this setup makes it excel conceptually in comparison with the static ones and as it will be shown in the following results, it also clearly excels in all of the metrics we used to evaluate them.

It is worth to mention that we used in these experiments 5V rechargeable battery with 10,000mAh capacity[60].

2.4.1 Network Performance

We used the number of generated requests from wireless sensor nodes to the server as an indicator for network traffic. To ensure fairness in the comparison, we performed our computations within the working hours from 10am until 4pm for all setups. In Static_20s and Static_10, all bins communicate every 20 and 10 seconds, respectively. Accordingly, the number of requests per hour is the same across different bins (180 and 360 request/hour respectively). The situation is different for Dynamic_20s_10s, where each bin communicates only when there are: 1) new probe requests showing the

existence of new devices (MAC addresses and signal strengths), and/or 2) changes in temperature and rubbish levels (the chosen
threshold in our experiments was 10 cm for waste level and degree difference in temperature; other values can be chosen with no qualitative difference in the results). Accordingly, we computed the average number of requests per hour sent by each bin for five consecutive working days. Figure 12 shows the average number of requests sent per hour by each bin in Dynamic_20s_10s. As shown in the figure, the first three hours have roughly half of the requests of the last three hours due to the different frequencies of sensing data in this setup.

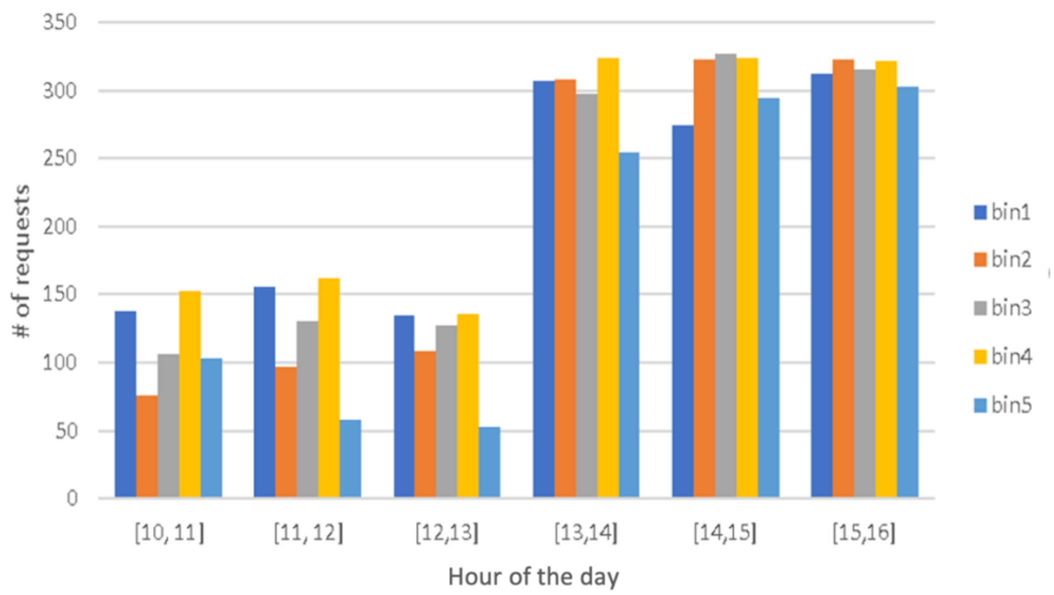


Figure 12: Average number of requests sent by each bin for Dynamic_20s_10s

Table 3 provides a comparison between Static_20s, Static_10s and Dynamic_20s_10s in terms of the number of requests across different hours. We computed the average across all bins in each hour for Dynamic_20s_10s. As shown in the table, Dynamic_20s_10s is able to decrease the network traffic up to 38%.

Table 3: Number of requests per hour by different setups and Dynamic_20s_10s network traffic improvement

	[10, 11]	**[11, 12]**	**[12,13]**
Dynamic_20s_10s	115	120	111
Static_20s	180	180	180
Network Traffic Improvement (%)	36	33	38

(a)

	[13, 14]	**[14, 15]**	**[15,16]**
Dynamic_20s_10s	298	308	315
Static_10s	360	360	360
Network Traffic Improvement (%)	17	14	13

(b)

2.4.2 Database Storage

In the same environment, the RSS table is the same for all three setups because wireless sensor nodes are going to report the same signal strength and MAC addresses. On the other hand, for TEMP and GRP tables, each setup has its own behaviour. In Static_20s and Static 10s, all bins communicate regularly every 20 and 10 seconds respectively. Accordingly, the number of generated records per hour per each across different bins is the same in each table (180 and 360 records per hour respectively). The situation is different for Dynamic_20s_10s, where each bin communicates only when there are changes in temperature and/or garbage levels.

Table 4 and Table 5 show the number of records stored in GRB and TEMP tables per hour in each setup respectively. In Dynamic_20s_10s setup, we computed the average of all records stored by all bins per hour. For the GRB table, our proposed architecture saved space up to 93% and 96% compared with the Static_20s and Static_10s, respectively. For the TEMP table, our proposed architecture saved space up to 97% and 99% compared with the Static_20s and Static_10s, respectively. Our proposal excels because it does not depend on the frequency of sensing data, but rather communicates the data only when there is something new to report.

Table 4: Number of records stored in GRB table by different setups and Dynamic_20s_10s storage utilisation

	[10, 11]	[11, 12]	[12,13]	[13,14]	[14,15]	[15,16]
Dynamic_20s_10s	19	20	13	26	28	28
Static_20s	180	180	180	180	180	180
Storage Utilisation Improvement (%)	89	89	93	85	85	85
Static_10s	360	360	360	360	360	360
Storage Utilisation Improvement (%)	95	95	96	93	92	92

Table 5: Number of records stored in TEMP table by different setups and Dynamic_20s_10s storage utilisation

	[10, 11]	[11, 12]	[12,13]	[13,14]	[14,15]	[15,16]
Dynamic_20s_10s	6	4	5	6	5	6
Static_20s	180	180	180	180	180	180
Storage Utilisation Improvement (%)	97	98	97	97	97	97
Static_10s	360	360	360	360	360	360
Storage Utilisation Improvement (%)	98	99	99	98	99	98

2.4.3 Power Consumption

To measure power consumption for Dynamic_20s_10s, we charged the batteries of the five bins and computed the total number of hours until batteries were fully discharged. Table 6 shows the lifetime in hours for each bin. There is no significant difference in lifetime between different bins because they send relatively the same number of requests to the server (as shown in Figure 12).

Table 6: Lifetime in hours for different bins

Bin #	**Lifetime in Hours**
bin1	147
bin2	149.5
bin3	149
bin4	151
bin5	150.5
Mean	149.4
Standard Deviation	1.6

Static_20s and Static_10s were also applied on the system in order to compare the power consumption of all three setups. Table 7 presents this comparison in terms of lifetime in hours.

Dynamic_20s_10s shows 158% and 293% improvement over Dynamic_20s and Dynamic_10s, respectively. Dynamic_20s_10s uses adaptive behaviour based on the circumstances in different areas. Accordingly, it improves the power consumption without affecting data freshness.

Table 7: Lifetime for different setups

	Dynamic_20s_10s	**Static_20s**	**Static_10s**	**Dynamic_20s_10s Improvement (%)**	
				Static_20s	**Static_10s**
Lifetime in hours	149.4	58	38	158	293

2.4.4 Real-time Environmental Condition and Dynamic Crowd Measurements

We developed two screens using Google Data Studio [61]. First, we connected it with the MySQL database engine to access the data received from sensor nodes. We gave these identifications: 12, 13, 17, 18 and 19 to the five bins. We activated the sensor nodes only from 10am until 4pm. General Statistics screen (Figure 13) shows the number of records that were reported from each bin for temperature, garbage level, and RSS within a selected period. The position and identification of each bin are shown on the map at the top of the screen. As we can see, bin #12 captured the largest amount of RSS and the largest amount of changes in garbage levels because this bin is in front of the main library entrance which experiences huge traffic and contains a coffee shop.

Figure 13: General statistic screen

The General Trend screen (Figure 13) is the second screen which provides the behaviour of RSS, garbage level and temperature over time for different bins. It is important to mention that we converted garbage level to percentage scale from 0 to 100. The screen provides filters which enable the user to compare all bins or focus on a specific bin. They enable the user also to filter certain data/hour. Bin #12 encounters the highest no. of RSS which indicates the highest crowd in this area at the specified date/time. Bin #13 encountered the lowest no. of RSS which indicates the lowest crowd at this area at the specific date/time. There was a continuous increase in the number of RSS which indicates a continuous increase in the crowd at the area of bin #17 over time. Regarding garbage level, Bin #12 was almost full over the measurement period. However, the levels of Bin #17 and Bin #13 showed fluctuations over time.

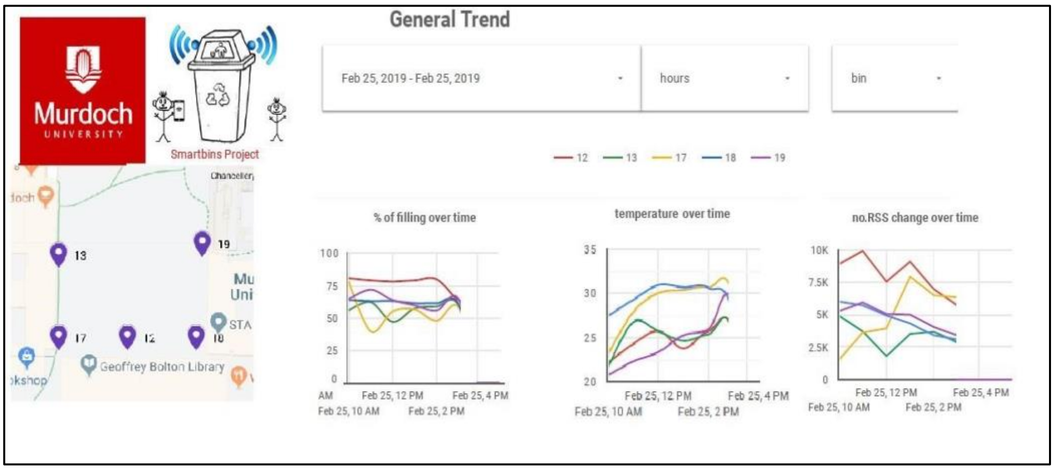


Figure 14: General trend screen

2.5 Summary

In this chapter, we gave an overview of the IoT architecture and relevant existing work including current challenges. We proposed an efficient WSN architecture for smart street furniture which utilises power consumption, storage, and network performance without affecting data freshness. Our architecture is not only used for the measurement of environmental conditions but also for sensing wireless devices which can be utilised for real-time crowd measurements.

We conducted experiments at the Murdoch University campus after getting human ethics approval, and we developed a dashboard using Google Data Studio to provide real-time dynamic crowd and environmental measurements. In the next chapter, we show how we generated unique localisation datasets after scaling our system to cover all bins in the Bush Court of Murdoch University.

Chapter 3 Wifi-based Localisation Datasets for GPS denied Open Areas Using Smart Bins

3.1 Overview

In this chapter, the process of generating the unique localisation datasets (MBCLFP and MBCLAP) for the Bush Court of Murdoch University after deploying our sensor nodes across all existing bins is explained. MBCLFP is a labelled dataset generated using four users' smartphones and MBCLAP is an unlabelled dataset autogenerated by our deployed sensor nodes. Moreover, we provide detailed specifications of the datasets. Finally, we present some statistics and visualisations for both datasets including homogeneity test using t-test.

The value of the datasets can be summarised as follows:

- All existing datasets are generated using users' smartphones only [26, 27, 29, 31, 62]. The process of generating them requires users to participate using smartphones which make them limited in terms of size and scalability. Our proposed datasets are the first datasets produced using two different Wifi sources for the same RPs for an open area. The first source is the four users' smartphones which is limited compared with the second source. The second source is APs which generated a huge amount of data for all users in the Bush Court including the four users.
- These datasets allow networking/communications researchers to study the signal variation between devices' fingerprints and AP RSSs for the same devices.
- These datasets allow data scientists to provide different visualisations and analyses to test homogeneity between the two datasets.

- Using the two datasets, more efficient models can possibly be developed [29].
- The datasets open the door for providing new techniques to label the huge amount of auto-generated data received from APs using the limited amount of labelled data.

3.2 Process of Generating the Datasets

We nominated Bush Court of Murdoch University to be the area of study because it 1) is central to the campus, and 2) is considered as a hub for staff and students given that it is surrounded by many facilities including the main library and restaurants. We tackled the problem of APs coverage in Bush Court by scaling our previously designed sensor node across all existing garbage bins. Figure 15 shows the distribution of garbage bins and their IDs in which 20 bins provide excellent coverage for Bush Court. ArcGIS is used to visualise bins locations [63].

Figure 15: Distribution of rubbish bins across the Bush Court of Murdoch University

Before starting to deploy our sensor nodes across the smart bins, we simulated the coverage of the Wifi signal. Wifi can cover more than 100m distance, however, we limited the Wifi coverage of each bin to be only 30m to avoid excessive signal decay due to blocking or noise problems. Figure 16 shows a heatmap that represent the number of bins that covers any location within the Bush Court. Accordingly, we are sure that smart bins provide excellent coverage for the Bush Court. We deployed 20 sensor nodes inside the available smart bins.

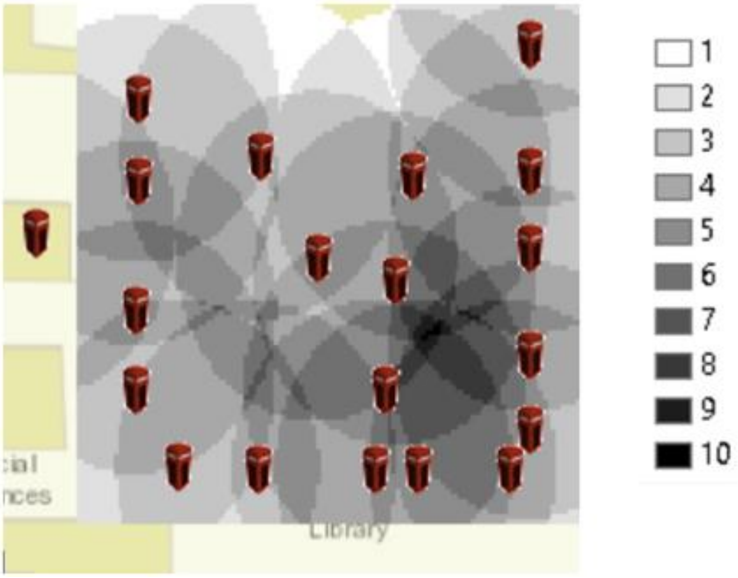


Figure 16: Number of bins that covers any location within the Bush Court

We gridded the area into 31 rows and 33 columns, resulting in 1023 RPs with each RP occupying 3x3 m^2. Each user installed the *Wifi Analyzer* application [64] with the settings set at the shortest scan interval. Table 8 lists the users' smartphones details. This application enables users to get fingerprints with timestamp at each RP in CSV format. Users scanned RPs row by row during one-month period. Each user started at the middle of a RP and record four fingerprints for four direction (North, East, South, West). Figure 17 illustrate the process of scanning. Among 1023 RPs, 21 RPs cannot be visited because they have huge trees. These indices of RPs are (7,21), (8,31), (12,28), (14,19), (15,22), (16, 17), (16,18), (17,17), (17,18), (18,17), (23, 11), (21, 21), (21, 22), (22, 21),(22,22), (23,21), (23,22), (23,23), (24,21), (24, 22) and (22,23).

Table 8: Users' smartphones details

User id	Smartphone Type	Operating System
1	PRIMO GH7I	Android 8.1
2	LG G6	Android 7.0
3	Samsung S8	Android 7.0
4	Oppo F1 Plus	Android 6.0

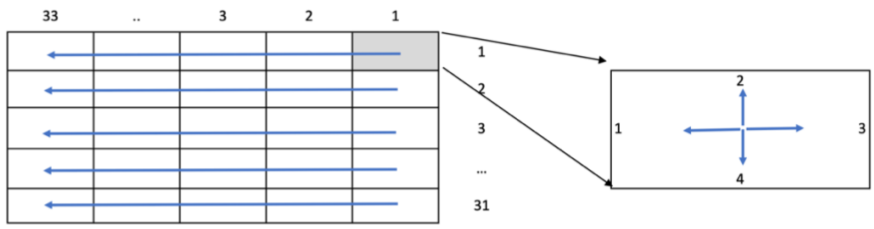


Figure 17: Process of scanning reference points

3.3 Datasets Specifications

The MBCLFP dataset contains 16032 rows (2 MB) where each user generates 4008 fingerprints in form of CSV files. We developed a python script to consolidate all these files into one single CSV file. Figure 18 shows some records from the dataset. Each record shows 20 APs RSSs at (x,y) RP in a given timestamp for a user. RSS values are from -31 to -100 where the bigger value indicate closer proximity to a given AP and -100 indicate out-of-range RSS.

Access Points																				timestamp	RP		user
11	12	13	15	17	18	19	10	21	22	3	33	37	38	39	4	6	7	8	9		x	y	
-80	-79	-100	-100	-63	-100	-100	-72	-100	-100	-100	-100	-100	-100	-100	-79	-100	-67	-67	-85	2019-09-25-184140	0	0	1
-80	-79	-100	-100	-63	-100	-100	-72	-100	-100	-100	-100	-100	-100	-100	-79	-100	-67	-67	-85	2019-09-25-184142	0	0	1
-83	-83	-100	-100	-100	-100	-100	-81	-100	-100	-100	-100	-85	-100	-100	-100	-100	-67	-78	-86	2019-09-25-184152	0	0	1
-100	-100	-100	-100	-71	-100	-100	-81	-100	-85	-100	-100	-100	-100	-100	-100	-100	-67	-85	-100	2019-09-25-184159	0	0	1
-84	-80	-86	-100	-68	-85	-100	-75	-100	-87	-100	-78	-88	-100	-100	-71	-89	-76	-81	-88	2019-09-25-184209	0	1	1
-83	-85	-89	-100	-60	-83	-100	-86	-100	-86	-82	-100	-84	-100	-100	-70	-100	-85	-73	-86	2019-09-25-184315	0	1	1
-100	-86	-100	-100	-65	-100	-100	-84	-100	-100	-88	-100	-100	-100	-100	-78	-87	-73	-81	-100	2019-09-25-184321	0	1	1
-100	-100	-100	-100	-73	-100	-100	-81	-100	-82	-89	-100	-100	-100	-89	-85	-100	-100	-72	-100	2019-09-25-184329	0	1	1
-86	-83	-100	-100	-65	-100	-100	-87	-100	-84	-81	-100	-100	-100	-100	-77	-100	-84	-81	-100	2019-09-25-184337	0	2	1
-83	-100	-100	-100	-51	-83	-90	-76	-100	-83	-100	-100	-85	-100	-80	-76	-91	-65	-64	-100	2019-09-25-184432	0	2	1
-100	-78	-100	-100	-51	-84	-100	-100	-100	-100	-79	-100	-100	-88	-100	-100	-89	-65	-67	-80	2019-09-25-184440	0	2	1
-87	-100	-82	-100	-100	-100	-100	-85	-87	-82	-100	-100	-100	-100	-100	-77	-100	-78	-71	-100	2019-09-25-184447	0	2	1
-82	-78	-84	-100	-100	-87	-100	-79	-100	-100	-77	-100	-81	-100	-100	-73	-84	-74	-61	-100	2019-09-25-184455	0	3	1
-81	-100	-100	-100	-64	-100	-100	-73	-100	-100	-76	-100	-100	-100	-100	-68	-83	-71	-69	-79	2019-09-25-184532	0	3	1
-77	-84	-100	-100	-64	-100	-100	-100	-86	-85	-76	-100	-86	-88	-100	-70	-100	-69	-70	-78	2019-09-25-184539	0	3	1
-100	-81	-75	-100	-64	-100	-100	-80	-100	-83	-100	-74	-84	-100	-100	-100	-100	-75	-74	-86	2019-09-25-184549	0	3	1

Figure 18: A sample from MBCLFP dataset

For MBCLAP, we configured sensor nodes to communicate probe requests associated with RSSs with a server entry recorded every 1 second. We recorded 2450865 rows (72 MB) during the same period of collecting MBCLFP. Figure 19 shows a sample from the MBCLAP dataset. Each record shows RSS from a user recorded by a bin in a given timestamp.

Bin/AP	User	RSS	timestamp
13	1	-90	2019-09-25-184209
11	1	-83	2019-09-26-184209
22	1	-89	2019-09-27-184209
4	1	-79	2019-09-28-184209
17	1	-72	2019-09-29-184209
7	1	-84	2019-09-30-184209
12	1	-90	2019-10-01-184209
8	1	-86	2019-10-02-184209
11	1	-90	2019-10-03-184209
4	1	-81	2019-10-04-184209
10	1	-84	2019-10-05-184209
17	1	-73	2019-10-06-184209

Figure 19: A sample from MBCLAP dataset. The data is part of the data recorded for User 1, and the RSS signals are shown for the access points in bins 13,11,22 etc.

3.4 Data Analysis and Visualisation

We visualise the number of rows for each RP (Figure 20). The figure shows consistency in the number of fingerprints provided by four users in which exactly 16 fingerprints for each RP except for the aforementioned 21 unavailable RPs (grey area). Figure 21 is another way to show the consistent number of fingerprints provided by users across different dates.

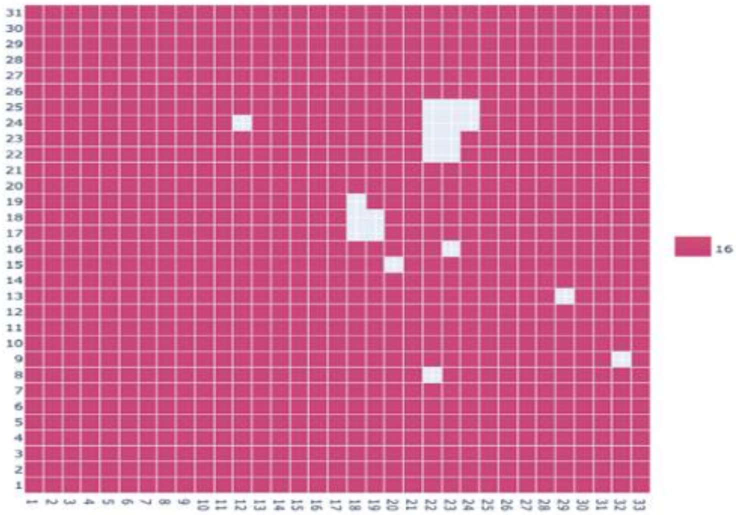


Figure 20: No. of fingerprints per reference point

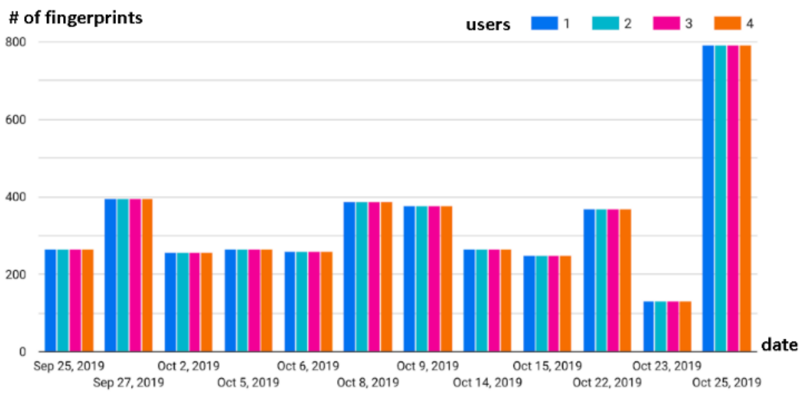


Figure 21: No. of fingerprints per users across different dates

Correlation is an excellent metric to show how two variables are associated [65-68]. Pearson correlation coefficient is computed using the following equation:

$$r = \frac{\mathrm{N}\sum xy - (\sum x)(\sum y)}{\sqrt{[\sum x^2 - (\sum x)^2][\sum y^2 - (\sum y)^2]}}$$

where N = number of pairs of values

$\sum xy$ = sum of the products of paired values

$\sum x$ = sum of x values

$\sum y$ = sum of y values

$\sum x^2$= sum of squared x values

$\sum xy^2$= sum of squared y values

Table 9 shows the Pearson correlation between each AP RSSs and RPs locations (rows and columns). Most of AP RSSs are either positively or negatively correlated with RPs locations. The lowest contribution comes from AP 15 since it is outside the area of Bush Court (see earlier Figure 15). For improved understanding regarding coverage, we generated heatmaps for a sample of APs RSSs including AP 15 across the area (Figure 22). It is obvious that the closer we are to the AP location, the stronger signal strength we receive from it. Moreover, no RSS is received for unavailable RPs. One more observation is that AP 15 shows a different behaviour with respect to others because it is not in the Bush Court area but rather located between two buildings (please refer to Figure 15).

Table 9: Pearson correlation between APs RSSs and RPs locations

AP	**x**	**y**
3	-0.71	0.18
4	-0.68	-0.49
6	-0.60	0.63
7	-0.70	0.02
8	-0.61	-0.48
9	-0.49	0.73
10	0.10	-0.57
11	0.23	0.51
12	-0.74	0.25
13	0.41	-0.70
15	**0.16**	**-0.21**
17	-0.64	-0.56
18	-0.56	0.67
19	0.09	0.78
21	0.78	0.39
22	0.61	-0.44
33	0.17	-0.35
37	0.72	0.11
38	0.59	0.55
39	0.66	-0.43

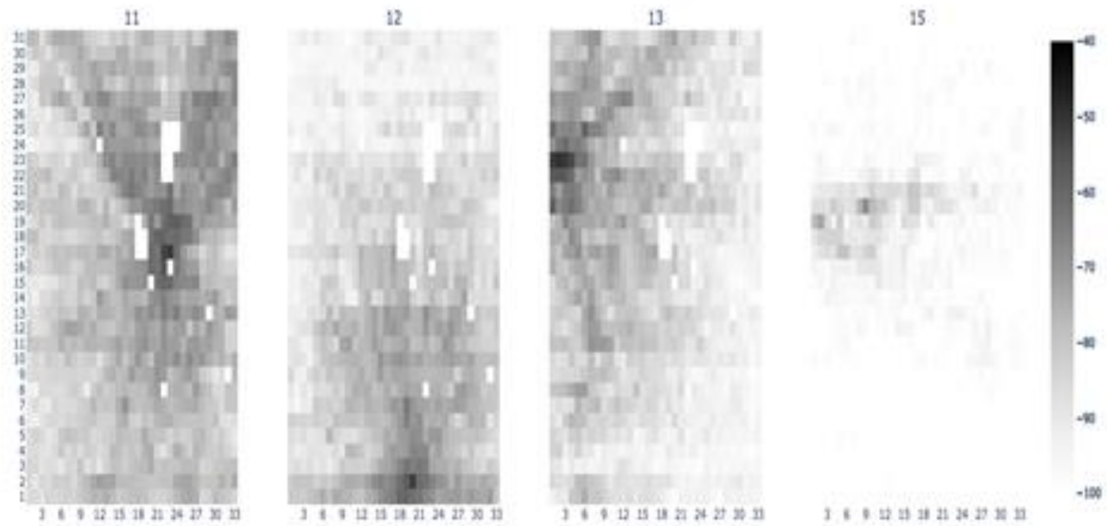


Figure 22: Heatmaps for APs 11, 12, 13, 15 showing respective coverage

3.4.1 Algorithm of Generating Labelled MBCLAP

APs send encrypted MAC addresses associated with RSSs gathered from existing devices to the server (including the four users' smartphones). The collected dataset from APs has 2450865 rows from 1167 unique devices. It is worth mentioning that we replaced encrypted MAC addresses with sequence ID. We reserved IDs 1, 2, 3 and 4 for the four users' smartphones. Table 10 shows the proposed algorithm for labelling MBCLAP dataset from MBCLFP dataset. First, MBCLAP-Labelled is defined as an empty dataset (line 1). For each user, iterations over RPs one by one is performed (lines 2, 3 and 4). Then, getting minimum and maximum timestamps per RP per user from MBCLFP dataset is performed (lines 5, 6). After that, filtering MBCLAP dataset by user and minimum and maximum timestamps and appending the filtered records to MBCLAP-Labelled (lines 7, 8). Finally, MBCLAP-Labelled is returned (line 12).

Table 10: The proposed algorithm for generating four users MBCLAP-Labelled

Generate_Four_Users_Aps_Dataset (MBCLFP, MBCLAP)
1. Initialise dataset MBCLAP-Labelled =[]
2. Start loop over four users u ∈[1,2,3,4]
3. Start loop over rows x ∈ [0...32]
4. Start loop over columns y ∈ [0…30]
5. Set from_time_stamp = minimum timestamp from MBCLFP for a user u and a RP (x, y)
6. Set to_time_stamp = maximum of timestamp from for a given user u and a RP (x, y)
7. Set filtered_data = get all records from MBCLAP for a given user u and timestamp between from_time_stamp and to_time_stamp
8. Add filtered_data to MBCLAP-Labelled
9. End columns loop
10. End rows loop
11. End users loop
12. return MBCLAP-Labelled

3.4.2 Homogeneity Test between MBCLFP and MBCLAP

Before studying the homogeneity of the MBCLAP-Labelled and MBCLFP datasets, we performed two steps: First, we filtered the APs RSSs to remove all values with -100 because we added them to indicate out-of-range and they were not recorded by users (applied for MBCLFP only). Secondly, we used Z-score normalisation [69] to normalise the users' data per AP for both datasets.

Table *11* shows the t-test results with 95% confidence between both datasets for the users across different APs. The last row of the table shows the t-test after merging all APs data for different users. Given that all p-values were larger than 0.05, we accepted the null hypothesis that both datasets are from the same population and homogenous.

Table 11: p-values from the t-test between MBCLAP-Labelled and MBCLFP datasets where columns are users and rows are Access Points

AP \ User	1	2	3	4
11	0.9631 (Accept)	0.8138 (Accept)	0.8274 (Accept)	0.7863 (Accept)
12	0.4344 (Accept)	0.2741 (Accept)	0.9202 (Accept)	0.1880 (Accept)
13	0.8508 (Accept)	0.8276 (Accept)	0.9732 (Accept)	0.7292 (Accept)
15	0.8624 (Accept)	0.8417 (Accept)	0.8943 (Accept)	0.8408 (Accept)
17	0.923 (Accept)	0.5321 (Accept)	0.4236 (Accept)	0.2791 (Accept)
18	0.7699 (Accept)	0.9195 (Accept)	0.6728 (Accept)	0.9325 (Accept)
19	0.5677 (Accept)	0.9139 (Accept)	0.8986 (Accept)	0.3545 (Accept)
10	0.2247 (Accept)	0.3435 (Accept)	0.6859 (Accept)	0.9112 (Accept)
21	0.8602 (Accept)	0.8699 (Accept)	0.4699 (Accept)	0.8549 (Accept)
22	0.7437 (Accept)	0.7089 (Accept)	0.6853 (Accept)	0.8799 (Accept)
3	0.8399 (Accept)	0.8674 (Accept)	0.5309 (Accept)	0.9083 (Accept)
33	0.7886 (Accept)	0.7102 (Accept)	0.7864 (Accept)	0.8889 (Accept)
37	0.8484 (Accept)	0.7444 (Accept)	0.661 (Accept)	0.6908 (Accept)
38	0.6046 (Accept)	0.926 (Accept)	0.4888 (Accept)	0.4663 (Accept)
39	0.7651 (Accept)	0.8239 (Accept)	0.999 (Accept)	0.8577 (Accept)
4	0.4791 (Accept)	0.266 (Accept)	0.5585 (Accept)	0.6688 (Accept)
6	0.5324 (Accept)	0.9322 (Accept)	0.663 (Accept)	0.3761 (Accept)
7	0.93 (Accept)	0.9662 (Accept)	0.9388 (Accept)	0.5621 (Accept)
8	0.7899 (Accept)	0.6455 (Accept)	0.8245 (Accept)	0.7133 (Accept)
9	0.6862 (Accept)	0.6252 (Accept)	0.7698 (Accept)	0.9265 (Accept)
All APs	0.3776 (Accept)	0.1969 (Accept)	0.3975 (Accept)	0.8714 (Accept)

3.5 Summary

In this chapter, we introduced the first Wifi-based localisation datasets for a GPS denied open area. We selected a central and busy open area at Murdoch University

(Western Australia), known as Bush Court, to generate the datasets using so-called "smart bins". We enabled the rubbish bins to work as access points by attaching our previously designed sensor nodes. The data gathered, referred to as MurdochBushCourtLoC, consists of two different datasets: Firstly, the MurdochBushCourtLoC-FP (MBCLFP) dataset in which four users generated labelled Wifi fingerprints for all available Reference Points (RPs) using four different smartphones. Secondly, the MurdochBushCourtLoC-AP (MBCLAP) dataset that includes 2450865 auto-generated rows received from more than 1000 devices.

In addition, we performed visualisations, statistics and analyses which can be generalised for any fingerprint dataset. Then we developed a light-weight algorithm to label the MBCLAP from MBCLFP for the four users' smartphones. Finally, we showed that the two datasets are homogeneous using a t-test. In the next chapter, we propose a localisation approach that converts MBCLAP from asynchronous format to synchronous. The conversion process enables us to utilise the AP dataset to provide more accurate model. i.e. it enables us to compare the accuracy of the same machine learning model using two different datasets (MBCLAP and MBCLFP). In addition, the chapter applies feature engineering and a four-layer deep learning classifier for MBCLAP and MBCLFP. Finally, we demonstrate that by using this approach we achieve higher prediction accuracy, up to 19% improvement, compared with MBCLFP only.

Chapter 4 Real-time Localisation Approach using Data from Wifi Fingerprints and Access Points

4.1 Overview

There are many methods to estimate a users' location based on RSS. These methods are generally classified as attenuation-based methods [70-72] and fingerprint-based methods [26, 41, 43, 72-74]. Attenuation-based methods estimate a user's location based on 1) RSSs received by APs, and 2) signal degradation due to obstacles in environment such as walls. Fingerprint-based methods estimate location of users based on 1) RSSs from their smartphones, and 2) historical RSSs collected for different RPs by different users. The attenuation-based methods are difficult to work in complex environments which have many obstacles, walls and users [27]. On the other hand, fingerprint-based methods show higher accuracy, robustness and scalability. Many datasets are publicly available as a benchmark to compare different methods [25, 28, 29, 41, 75]. Moreover, several researchers have proposed machine learning models which eliminate the noise associated with RSS to improve localisation accuracy, [25, 74, 75].

In summary, Fingerprint methods suffer from the following limitations:
Generation of fingerprint datasets is a complex and error-prone process. To create a fingerprint dataset, users are required to participate using smartphones. Users are required to download an application and stand at each RP to report RSS [26-31]. It is a complex process and depends on human intervention which can lead to errors. For example, a user must be standing at the exact RP that he/she is reporting.

Smartphones' RSSs are cached, which affects the currency of RSS readings. Wifi fingerprint datasets depend on Android devices [26, 28, 31, 62] due to the rich and well-documented Software Development Kit (SDK) [76] and the availability of readymade software [64]. According to [77], the resulting data from a Wifi scan might not contain current values which will in turn affect the accuracy of any prediction method.

Fingerprint methods are not scalable/flexible. Wifi fingerprint datasets/methods use only Android devices [26, 28, 31, 62, 72] and require users to have the relevant applications on their smartphones. Accordingly, they target a small percentage of users. Moreover, all fingerprint datasets mainly target indoor areas. There are open areas that are not covered by GPS and require study.

The above-mentioned limitations lead us to study the data collected by APs. The advantages of utilising AP data include:

More scalability and flexibility. It is not required to have applications installed on users' smartphones. APs sniff any RSSs from any surrounding device so data collection is automated, no human intervention is required.

Higher accuracy. IoT APs come with basic firmware and SDK that enables developers to control the hardware unlike smartphones [51].

The core challenges which limit AP data from being utilised for location estimates include the following:

Collected data are asynchronous. In fingerprint data, all available RSSs from APs are captured at a given timestamp (synchronous behaviour). On the other hand, in AP data, each AP records RSSs without coordination with other APs (asynchronous behaviour) and different APs may encounter a delay in reporting RSSs which makes it difficult to have synchronous data.

Lack of publicly available datasets. All publicly available datasets are fingerprint datasets, which makes it hard for researchers to build new methods based on AP data. Additionally, AP data is not labelled.

To enable new research and development of more scalable and accurate localisation methods, we developed MurdochBushCourtLoC, the first Wifi dataset that contains both fingerprint and AP datasets for a GPS denied open area based on Wifi (the Bush Court of Murdoch University) [28]. MurdochBushCourtLoC consists of two datasets: First, MurdochBushCourtLoC-FP (MBCLFP) dataset in which four users generated labelled Wifi fingerprints for all available RPs using four different smartphones. Secondly, MurdochBushCourtLoC-AP (MBCLAP) dataset that includes 2450865 auto-generated rows received from more than 1000 devices. There are semi-supervised deep reinforcement learning methods that can be applied to utilise the relatively massive dataset (MBCLAP) by labelling it using MBCLFP [30].

In this chapter, we propose a localisation approach that converts MBCLAP from asynchronous format to synchronous, applies feature engineering and a deep learning classifier for MBCLAP and MBCLFP. Finally, we demonstrate that by using this approach we achieve higher prediction accuracy, with up to 19% average improvement, compared with using only MBCLFP.

4.2 Converting the Asynchronous MBCLAP-Labelled dataset to Synchronous

It is important to have a general method that can work not only with our dataset but with any other dataset. The method also must tackle the following challenges:

- Some APs cannot report RSSs due to either a connectivity issue or because they did not receive any RSSs from smartphones. For example, there can be a gap between two strong RSSs value.
- There is a delay in reporting the RSSs from some APs due to connectivity delay or server-side bottleneck.

Table 12 shows the steps performed to convert the dataset from asynchronous format to synchronous. For each RP per user, all records are sorted in ascending order (step 1). Then, the records are divided into a number of partitions, each partition of *Partition_Size* (steps 2 and 3)**.** Step 4 is used to merge each partition into a single row with all APs. If more than one RSS are recorded by the same AP per partition, we simply compute the average.

It is worth mentioning that Partition_Size should be carefully selected based on parameters such as people's activities in the area and AP settings. For example, if people are using the area for crossing only, then for a given time window, a small number of RSSs are expected. Accordingly, the Partition_Size is expected to be relatively small. On the other hand, if APs are set to communicate RSSs every short period and people stay in the area for long time, then a massive number of RSSs is expected to be received. Accordingly, the Partition_Size is expected to be relatively large.

Figure 23 shows an example of partition merging. In partition three, two records are received from bin 2 and accordingly, the average is calculated. In the last step (Step 5), for a missed RSS between two RSSs, the average is calculated. In Figure 23, the missed RSS for bin 2 in Synchronous MBCLAP-Labelled will be replaced with the average of the previous RSS and the next RSS. In the example, it will be -70. The aforementioned steps can be implemented online, they do not require offline processing. In step 3, no_records and no_of_partitions can be calculated once a new record arrives in the system. Steps 4 and 5 can be performed incrementally.

Table 12: Method of converting MBCLAP-Labelled to be synchronous

For each user:

For each RP

1- sort records by timestamp in ascending order
2- no_of_partitions = no_records/Partition_Size
3- divide records into no_of_partitions
4- merge each partition to a single row. If more than 1 RSSs are received from the same AP per partition, compute the average
5- if RSS is missed between two known RSSs, average is calculated, otherwise, replace it with (-100)

End RP loop

End user loop

Asynchronous MBCLAP-Labelled

	bin	user	rss	time_stamp	x	y
partition 1	1	1	-60	t1	0	0
	2	1	-70	t2	0	0
partition 2	3	1	-50	t3	0	0
	1	1	-30	t4	0	0
partition 3	2	1	-65	t5	0	0
	2	1	-75	t6	0	0

Synchronous MBCLAP-Labelled

	bin 1	bin 2	bin 3	user	x	y
partition 1	-60	-70		1	0	0
partition 2	-30		-50	1	0	0
partition 3		-70		1	0	0

Figure 23: Example of partition merging

4.3 Cleaning data, Feature Engineering and Prediction Model

We applied the following rules to clean the data.

1- Under certain circumstances (e.g., limited/no Internet connection), APs could not send data, which affects the quality of the prediction model accuracy. Therefore, the records not containing MIN_NO_RSSs greater than -100 will be excluded.
2- We did not use the actual RSSs values, but instead normalised the values using Z-score and removing any outliers.

For feature engineering, our aim was to remove any AP that does not add any value to the location (x and y) to gain higher accuracy and reduce the complexity of the prediction model. Based on the Pearson correlation values in Table 9, we excluded AP 15.

Many neural network techniques have been proposed [25, 26, 30, 74, 75, 78] for location prediction. These techniques show higher accuracy compared to classic machine learning techniques [79]. Our problem is a classification problem as we need to predict x and y of a RP given a fingerprint and x and y are integer values between [1, 33] and [1, 31] respectively.

We developed a four-layer deep neural network classifier. The classifier is used twice in parallel to predict x and y. Figure 24 shows the general architecture of the classifier. The input layer has the no_AP nodes, which is the number of APs that contribute in fingerprint records. Then, three hidden layers with a power of 2 number of nodes. Finally, the output layer has only one node. To improve model generalisation and reduce overfitting, we used dropout regularisation method [80] in which during training, in each layer, a random number of outputs are dropped out. More details on our experiments can be found in Section 4.4.

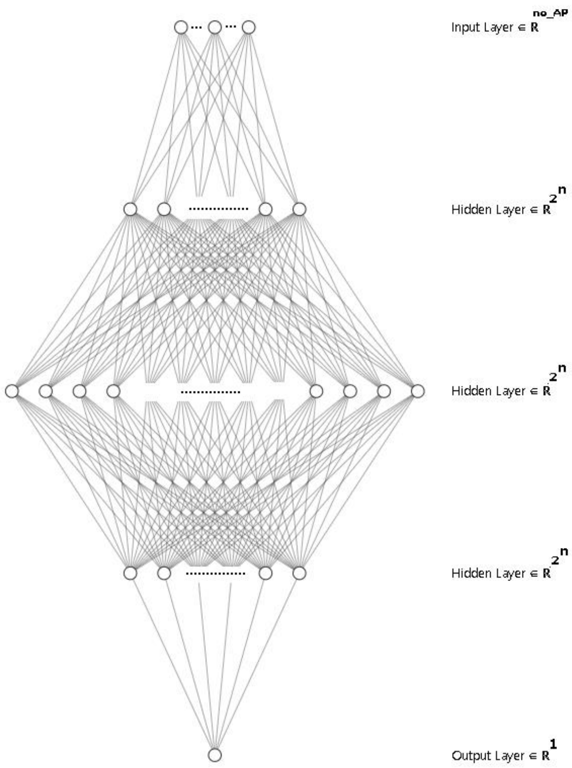


Figure 24: Four-layer deep neural network classifier

4.4 Experiments and Results

For generating the synchronous MBCLAP-Labelled dataset, we tried different values for Partition_Size and MIN_NO_RSSs. Among different values, 10 and 8 show the best accuracy respectively. Table *13* Table shows the training accuracy for x location accross different Partition_Size and MIN_NO_RSSs values after 10 iterations.

Table 13: Training Accuracy for x location across different Partition_Size and MIN_NO_RSSs values

Min_NO_RSS / Partition_Size	4	6	8	10
5	36.7%	37.0%	36.8%	36.9%
10	38.0%	97.2%	97.4%	37.9%
15	37.8%	38.0%	38.0%	38.0%
20	38.0%	38.2%	38.0%	38.1%

Accordingly, every 20 records are merged to generate one synchronised record per user per RP. In addition, we removed records that do not have at least 8 RSSs greater than (-100). After cleaning the data, the total number of records for the MBCLAP-Labelled and MBCLFB datasets are 5222 and 13952 respectively. We used 90% for training and 10% for testing the models to avoid overfitting and poor performance (the 80%-20% split provided clearly inferior results). To best represent the entire population in model training and testing, we used stratified random split to generate the training and testing datasets. We ran the experiments ten times and computed the average for the results. For the four-layer neural network model, we selected the following parameters:

1- Input layer contains 19 nodes that represent the 19 APs (i.e., excluding bin 15).
2- For the hidden layers, rectified linear (RELU)[81] as the activation function for hidden units. We tried different sizes for hidden layers, and we found that 256, 512, and 256 for the first, second and third hidden layers respectively show higher prediction accuracy.

3- For the output layer, softmax [81] as the activation function and sparse categorical cross entropy as the loss function.
4- We used ADAM optimizer[82] and dropout regularization with 0.20 rate.

We used Tensorflow for our experiments [36] because it provides a collection of workflows to develop and train model using Python. For reproducing the results, we provided the source code and datasets in the datasets repository[28]. The users scanned Bush Court row by row. For example, for a given row x, users scanned y from 0 to 33 sequentially. This means that all columns are covered within a short period and due to the caching behaviour smartphones mentioned in the introduction [26, 28, 31, 62], the accuracy of y prediction is relatively poor compared with the x prediction. On the other hand, the rows are scanned for a relatively long period. Accordingly, we reduced the y granularity to reduce the confusion in the prediction of the y location. This can be achieved by performing integer division of the y label by an integer number greater than one. It is worth to mention that the greater the integer number, the accuracy of the model will be higher, and the precision of the localisation prediction will be lower. For example, if we have an area of 10*10 meters, and we grid the area with blocks 1m*1m, the x/y classes will be 10. If we decide to make each block 2m*2m, the x/y classes will be 5. Accordingly, it becomes easier for the model to predict x/y (5 classes instead of 10) which leads to higher accuracy, but the model is only able to localise a user within 2m*2m instead of 1m*1m, which leads to lower precision of the localisation prediction.

For our experiments, we chose the integer number to be three to achieve higher accuracy without significantly affecting the precision of localisation. Accordingly, the area became a 31 x 11 block.

Figure 25 and Figure 26 show the convergence speed for the MBCLAP-Labelled and MBCLFP datasets for the x location prediction. Figure 27 and Figure 28 show the convergence speed for the MBCLAP-Labelled and MBCLFP datasets for the y location prediction. Generally, the accuracy of the prediction of the x location is better than the accuracy of the prediction of the y location because of the caching behaviour of smartphones, and the aforementioned way of scanning RPs.

The MBCLAP-Labelled dataset shows better performance in accuracy and speed of convergence compared with the MBCLFP dataset for both x and y location prediction. For the x location after 26 iterations, the MBCLAP-Labelled dataset achieves 99.2% and 95.1% for training and testing respectively, compared with 96.4% and 80.1% for MBCLFP. Due to our proposed method for converting the MBCLAP- Labelled dataset to be synchronous, we achieved a 19% average improvement in accuracy for the testing dataset compared with the MBCLFP dataset for x location prediction.

Another problem with the MBCLFP dataset that does not exist in the MBCLAP-Labelled dataset is that of overfitting. For the y location after 34 iterations, the MBCLAP-Labelled dataset achieves 95.5% and 81.8% accuracy for training and testing respectively compared with 90.7% and 77.6% for the MBCLFP. The MBCLAP- Labelled dataset shows, overall, a 6% improvement in accuracy for the testing dataset compared with the MBCLFP dataset for the y location prediction. Even though the MBCLFP dataset contains an almost tripe number of records than the MBCLAP-labelled dataset, the MBCLAP-labelled dataset shows better performance, which renders APs is better than smartphones as a source of RSSs.

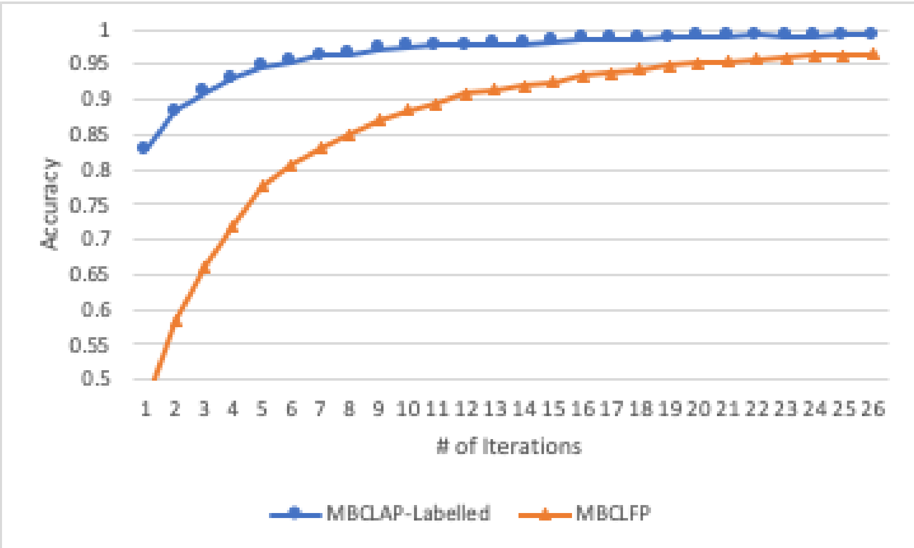


Figure 25: Training convergence speed for the datasets for the x location prediction

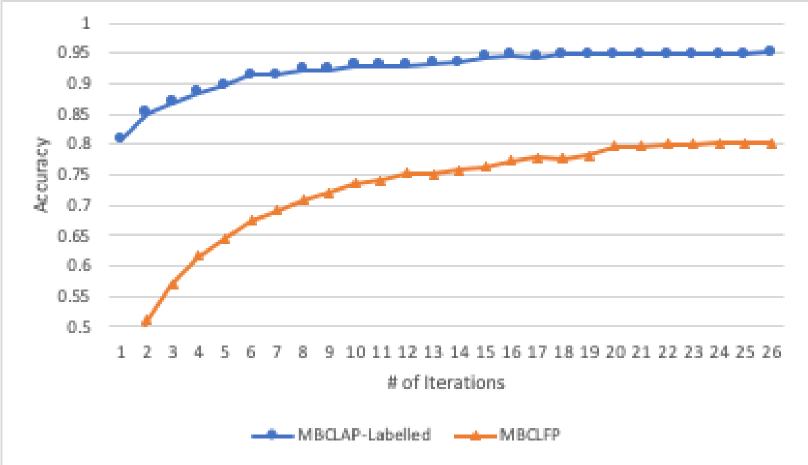


Figure 26: Testing convergence speed for the datasets for the x location prediction

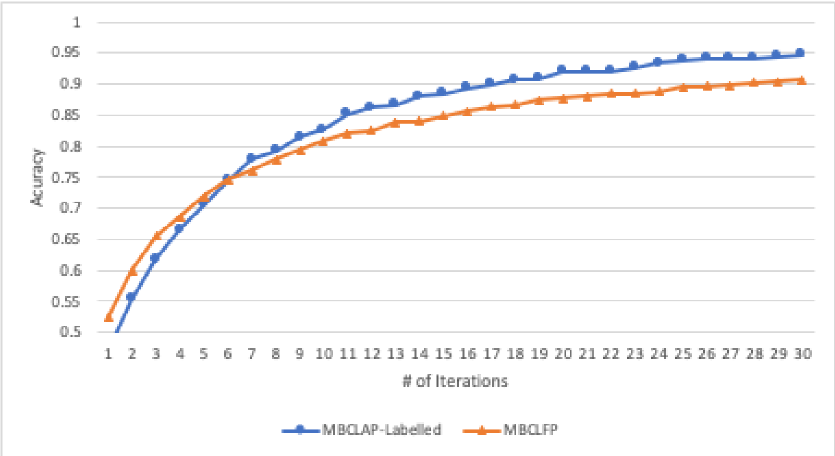


Figure 27: Training convergence speed for the datasets for the y location prediction

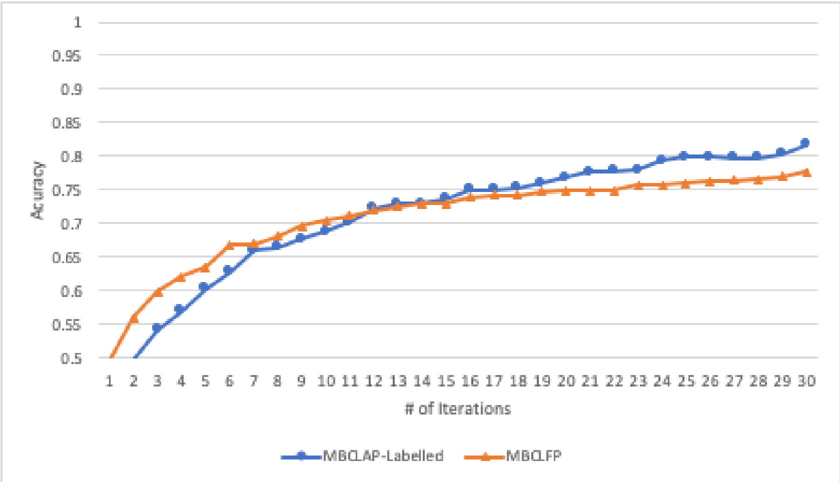


Figure 28: Testing convergence speed for the datasets for the y location prediction

4.5 Summary

In this chapter, we have summarised the advantages and disadvantages of APs and smartphones as a source of RSSs. We showed that APs are more accurate in reporting RSSs as it doesn't engage any human intervention and/or caching behaviour, however, APs data are not synchronised. Accordingly, we proposed a simple method for converting Asynchronous MBCLAP-Labelled dataset to Synchronous.

We provided a new localisation approach that engages data cleaning, feature engineering and a four-layer deep neural network classifier. The MBCLAP-Labelled dataset shows better performance in accuracy and speed of convergence compared with the MBCLFP dataset for location prediction. The MBCLAP-Labelled dataset shows 19% and 6% average improvement in the classification accuracy compared with the MBCLFP dataset for x and y location prediction, respectively. In the next chapter, we conclude this work and provide recommendations for future work.

Chapter 5 Conclusion and Future work

5.1 Conclusion

This thesis explained the importance of real-time crowd measurement, which has many applications including real-time monitoring of potential risks and emergency situations. RSS-based methods outperform other methods when it comes to the scalability, cost, availability and precision of data. It explained how smart cities can utilize existing street furniture to provide better understanding of pedestrians' movements and flow in real-time using RSS. The major challenges which prevent real-time dynamic crowd measurement solutions from being applied was listed in detail.

An overview of the IoT architecture and relevant existing work including current challenges relevant to power consumption, scalability and data security and privacy was presented. The thesis proposed an efficient WSN architecture for smart street furniture which utilises power consumption, storage, and network traffic without affecting data freshness. The architecture is not only used for the measurement of environmental conditions but also for sensing wireless devices which can be utilised for real-time crowd measurements. Experiments have been conducted on the Murdoch University campus with results showing that our/the proposal improves the lifetime of wireless sensor nodes by up to 412% compared to existing architectures. Moreover, network traffic is improved by up to 38% without affecting data freshness. Finally, storage space for the database at the server is reduced by up to 99%.

It was shown that WSN architecture maintains data security and privacy across different levels including sensor, network and server levels. The project received human ethics approval from Murdoch University. A dashboard using Google Data Studio to provide real-time dynamic crowd and environmental measurements was developed.

Two Wifi-based datasets were introduced, the first Wifi localisation datasets generated from two different sources to study localisation and mobility in GPS denied open areas. The first dataset, MBCLFP, has Wifi fingerprints collected for available RPs from four different users' smartphones. The second dataset, MBCLAP, contains auto-generated data from 20 different APs. It contains 2450865 records received from more than 1000 devices. We proposed an algorithm for generating a labelled dataset for the four users (MBCLAP-Labelled dataset). The algorithm uses the MBCLFP dataset which is labelled for generation process. We showed that the two datasets are homogeneous using t-test.

A summary about the advantages and disadvantages of APs and smartphones as a source of RSSs was provided. APs are more accurate in reporting RSSs; however, the APs data are not synchronised. Accordingly, a simple method for converting Asynchronous MBCLAP-Labelled dataset to Synchronous was proposed. A localisation approach that uses data cleaning, feature engineering and four-layer deep neural network classifier was provided. MBCLAP-Labelled dataset shows better performance in accuracy and speed of convergence compared with the MBCLFP dataset for location prediction. MBCLAP-Labelled dataset shows 19% and 6% improvement in the classification accuracy compared with MBCLFP dataset for x and y location prediction, respectively.

5.2 Future Work

This thesis proposed a new localisation approach based the adaptive real-time low-power wireless sensor network. The proposed solution maintains data privacy and security in different system tiers. We recommend extending the experimental studies to cover more complex environments that could introduce new challenges. For example, city centres are full of buildings and mobile objects that can introduce noises. Based on that, we recommend proposing a more generalised system that can adapt to different environments.

Real-time group-based recommendations can be achieved based on measuring crowd flow. It can be achieved by changing the contents displayed on digital signages in real-time. The real-time recommendations can include targeted advertising, events, announcements, evacuation procedures and context-aware information.

Figure 29 shows the envisioned overall system's components. The following list shows the expected basic functionalities of the system:

- Wi-Fi-equipped smart bins will continuously monitor wireless signals from pedestrians' mobile devices surrounding them. It is important to mention that no additional overhead on pedestrians' devices will be required as wireless modules in mobile devices periodically send probe requests to surrounding access points. Also, they will continuously sense environmental conditions such as the garbage level.

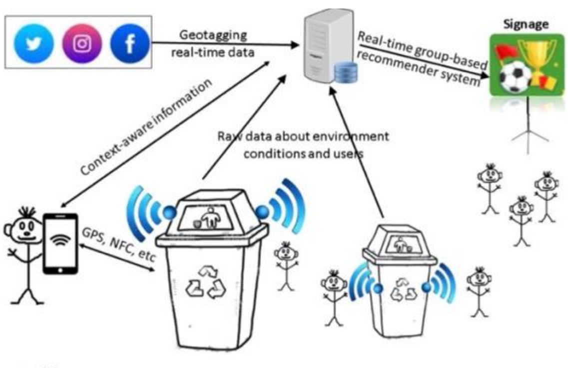


Figure 29: Overall system components

- Smart bins will periodically send all raw data to the server. More precisely, smart bins will not waste time in processing the data received from sensors or network module, but instead will send the data in a periodical manner to avoid any delay and improve overall system performance.
- The server will collect real-time geotagging data from social media and process them. The process of collecting geotagging data from social media will be performed using social media APIs.
- The server will process real-time data from different sources (sensors, social media and wireless signals from mobile devices) to measure crowd dynamics.
- Pedestrians will get real-time context-aware recommendation using location-enabled devices where the exact location of the user will be determined with high accuracy.
- The system will profile pedestrians based on their activities and display relevant content on digital signage in real-time. For example, if it recognises that some pedestrians always enter a school zone early morning and leave it afternoon, then it will surmise that they most likely are parents together with their children. Accordingly, the digital signage will display a targeted content in real-time. In addition to that, The system should consider data protection and ethics.

References

1. Greenstein, B. *The Internet of Things: Intelligent, instrumented, interconnected and*. 2015, October 20; Available from: http://www.ibmbigdatahub.com/blog/internet-things-intelligent-instrumented-interconnected-and-real-time.
2. *yStop Recommender System*. 2018; Available from: https://recsys.ystop.com.au/.
3. Mirchandani, S., et al. *IoT enabled dustbins*. in *In Proceedings of the International Conference on Big Data, IoT and Data Science*. 2017. Pune: IEEE.
4. Anagnostopoulos, T., et al. *Top--k Query Based Dynamic Scheduling for IoT-enabled Smart City Waste Collection*. in *In Proceedings of the 16th IEEE International Conference on Mobile Data Management (MDM)*. 2015. IEEE.
5. Fedchenkov, P., A. Zaslavsky, and I. Sosunova. *Enabling smart waste management with sensorized garbage bins and low power data communications network*. in *In Proceedings of the Seventh International Conference on the Internet of Things*. 2017. Linz, Austria: ACM.
6. Kristanto, S., et al. *Dynamic Polling Algorithm for Low Energy Garbage Level Measurement in Smart Trash Bin*. in *Proceedings of the Second International Conference on IoT in Urban Space*. 2016. Tokyo, Japan: ACM.
7. Anagnostopoulos, T.V. and A. Zaslavsky. *Effective waste collection with shortest path semi-static and dynamic routing*. in *In Proceedings of the International Conference on Next Generation Wired/Wireless Networking*. 2014. Springer.
8. Folianto, F., Y.S. Low, and W.L. Yeow. *Smartbin: Smart waste management system*. in *In Proceedings of the Tenth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*. 2015. IEEE.
9. Apsara .S, R.G.A., Mohankumari K .V, Anitha .L, Jyothi .B, *Smart Public Transport System Using Internet-Of-Things.* International journal of advance research, ideas and innovations in technology, 2017. **3**(2): p. 1163-1170.
10. Handte, M., et al., *An Internet-of-things enabled connected navigation system for urban bus riders.* IEEE internet of things journal, 2016. **3**(5): p. 735-744.
11. Green, D.J. *Waste Woes: How Blockchain And An Enzyme Could Clean Up The Mess We're In*. 2018; Available from: https://www.forbes.com/sites/jemmagreen/2018/04/20/how-blockchain-can-help-with-our-waste-woes/.
12. *Smart Street Furniture Will Revolutionise the Public Realm*. 2018.
13. Mead, M.I., et al., *The use of electrochemical sensors for monitoring urban air quality in low-cost, high-density networks.* Atmospheric Environment, 2013. **70**: p. 186-203.
14. Sarmady, S., F. Haron, and A.Z. Talib, *Simulation of pedestrian movements using fine grid cellular automata model.* arXiv:1406.3567, 2014.
15. Zhao, M., J. Zhong, and W. Cai. *A role-dependent data-driven approach for high density crowd behavior modeling*. in *In Proceedings of the 2016 ACM SIGSIM Conference on Principles of Advanced Discrete Simulation*. 2016.
16. Fridman, N., *Modeling crowd behavior*. 2012, http://u.cs.biu.ac.il/~galk/Publications/Papers/natalie-phd.pdf, Ph.D. Dissertation, Bar Ilan University.

17. Kaminka, G.A. and N. Fridman, *Simulating Urban Pedestrian Crowds of Different Cultures.* ACM Transactions on Intelligent Systems and Technology (TIST), 2018. **9**(3): p. 27.
18. Rhee, I., et al., *On the levy-walk nature of human mobility.* IEEE/ACM Trans. Netw., 2011. **19**(3): p. 630-643.
19. Amirgholipour, S., et al., *A-CCNN: adaptive ccnn for density estimation and crowd counting.* arXiv:1406.3567, 2018.
20. Szczodrak, M., et al. *Behavior analysis and dynamic crowd management in video surveillance system*. in *In Proceedings of the 22nd International Workshop on Database and Expert Systems Applications (DEXA)*. 2011. IEEE.
21. Ke, S.-R., et al., *A review on video-based human activity recognition.* Computers, 2013. **2**(2): p. 88-131.
22. Zhou, Y., et al., *Understand Urban Human Mobility through Crowdsensed Data.* arXiv preprint arXiv:1805.00628, 2018.
23. Yao, X., et al., *Representation and Analytical Models for Location-based Big Data*. 2018.
24. Ying, J., K. Pahlavan, and X. Li. *Precision of RSS-based indoor geolocation in IoT applications*. in *In Proceedings of IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), 2017*. 2017. IEEE.
25. Ibrahim, M., M. Torki, and M. ElNainay. *CNN based Indoor Localization using RSS Time-Series*. in *In Proceedings of Computers and Communications Symposium*. 2018. IEEE.
26. Zhong, Z., et al., *XJTLUIndoorLoc: A New Fingerprinting Database for Indoor Localization and Trajectory Estimation Based on Wi-Fi RSS and Geomagnetic Field.* arXiv preprint arXiv:1810.07377, 2018.
27. Chen, J., et al., *An INS/WiFi Indoor Localization System Based on the Weighted Least Squares.* Sensors (Basel, Switzerland), 2018. **18**(5): p. 1458.
28. Mohamed A. Nassar, P.K., Peter Cole , Len Luxford, Mahmud Hasan , Ibrahim Khan , Azifa Rumy, Md Mesbahul Hasan, Giles Oatley, *MurdochBushCourtLoC: Wifi-Based Localisation Datasets for No-GPS Open Areas Using Smart Bins*. 2020: Mendely Data.
29. Lohan, E., et al., *Wi-Fi crowdsourced fingerprinting dataset for indoor positioning.* Data, 2017. **2**(4): p. 32.
30. Mohammadi, M., et al., *Semisupervised deep reinforcement learning in support of IoT and smart city services.* IEEE Internet of Things Journal, 2017. **5**(2): p. 624-635.
31. Torres-Sospedra, J.n., et al. *UJIIndoorLoc: A new multi-building and multi-floor database for WLAN fingerprint-based indoor localization problems*. in *2014 international conference on indoor positioning and indoor navigation (IPIN)*. 2014. IEEE.
32. Poddar, H., et al. *Design of smart bin for smarter cities*. in *Power and Advanced Computing Technologies (i-PACT), 2017 Innovations in*. 2017. IEEE.
33. Thaker, T. *ESP8266 based implementation of wireless sensor network with Linux based web-server*. in *Colossal Data Analysis and Networking (CDAN), Symposium on*. 2016. IEEE.
34. Ferdoush, S. and X. Li, *Wireless Sensor Network System Design Using Raspberry Pi and Arduino for Environmental Monitoring Applications.* Procedia Computer Science, 2014. **34**: p. 103-110.
35. Chhabra, S. and D. Singh, *Data Fusion and Data Aggregation/Summarization Techniques in WSNs: A Review.* International Journal of Computer Applications, 2015. **121**(19): p. 21-30.

36. Rodriguez, M., S. Ali, and T. Kanade. *Tracking in unstructured crowded scenes*. in *In Proceedings of the 12th International Conference on Computer Vision* 2009. IEEE.
37. Tatrai, A., T. Semmens, and A. Tran. *Dynamic Crowd Management | Facilitating effective decision making in the management of crowds*. 2018; Available from: http://www.dynamiccrowdmanagement.com/dcm2017/wp-content/uploads/2017/08/DCMPaper_ACES_2017-05-16.pdf.
38. Swathi, H., G. Shivakumar, and H. Mohana. *Crowd Behavior Analysis: A Survey*. in *In Proceedings of the International Conference on Recent Advances in Electronics and Communication Technology (ICRAECT)*. 2017. IEEE.
39. Li, T., et al., *Crowded scene analysis: A survey.* IEEE transactions on circuits and systems for video technology, 2015. **25**(3): p. 367-386.
40. Smeulders, A.W., et al., *Visual tracking: An experimental survey.* IEEE Transactions on Pattern Analysis & Machine Intelligence, 2013(1): p. 1442-1468.
41. Khalili, A., A.-H. Soliman, and M. Asaduzzaman, *A Deep Learning Approach for Wi-Fi Based People Localization.* Preprints, 2018.
42. Gharghan, S.K., et al., *Adaptive Neural Fuzzy Inference System for Accurate Localization of Wireless Sensor Network in Outdoor and Indoor Cycling Applications.* IEEE Access, 2018. **6**: p. 38475-38489.
43. Chen, H., et al., *ConFi: Convolutional neural networks based indoor wi-fi localization using channel state information.* IEEE Access, 2017. **5**: p. 18066-18074.
44. *Wi-Fi Signal Strength: What Is a Good Signal And How Do You Measure It*. 2018 [cited 2018; Available from: https://eyesaas.com/wi-fi-signal-strength/.
45. Skyhook Wireless, I. *Skyhook | Location Data Provider*. 2019; Available from: https://www.skyhook.com.
46. Mitchell, R.O., et al. *Hajj crowd management and navigation system: People tracking and location based services via integrated mobile and rfid systems*. in *In Proceedings of the International Conference on Computer Applications Technology (ICCAT)*. 2013. IEEE.
47. Blanke, U., et al. *Capturing crowd dynamics at large scale events using participatory gps-localization*. in *In Proceeding of the Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*. 2014. IEEE.
48. Draghici, A. and M.V. Steen, *A survey of techniques for automatically sensing the behavior of a crowd.* ACM Computing Surveys (CSUR), 2018. **51**(1): p. 1-40.
49. *Bringing neighbors back to neighborhoods – Sidewalk Talk – Medium*. 2018 2018-01-19; Available from: https://medium.com/sidewalk-talk/bringing-neighbors-back-to-neighborhoods-5f7b2b41e1a6.
50. *Arduino Uno Rev3*. 2019; Available from: https://store.arduino.cc/usa/arduino-uno-rev3.
51. Grokhotkov, I. *ESP8266 arduino core documentation*. ESP8266, 2017.
52. @anaptfox. *Making the ESP8266 Low-Powered With Deep Sleep*. 2019; Available from: https://www.losant.com/blog/making-the-esp8266-low-powered-with-deep-sleep.
53. *Australian Privacy Principles| Office of the Australian Information Commissioner - OAIC*. 2019; Available from: https://www.oaic.gov.au/privacy-law/privacy-act/australian-privacy-principles.
54. *EUGDPR*. 2019; Available from: https://eugdpr.org/.
55. Agrawal, S., M.L. Das, and J. Lopez, *Detection of Node Capture Attack in Wireless Sensor Networks.* IEEE Systems Journal, 2018(99): p. 1-10.

56. Kifayat, K., et al., *Security in wireless sensor networks*, in *Handbook of information and communication security*. 2010, Springer. p. 513-552.
57. Karlof, C. and D. Wagner. *Secure routing in wireless sensor networks: Attacks and countermeasures*. in *Proceedings of the First IEEE International Workshop on Sensor Network Protocols and Applications, 2003.* 2003. IEEE.
58. *Arduino IDE*. 2018 [cited 2018; Available from: https://www.arduino.cc/en/main/software.
59. *AARNet Eduroam*. 2019; Available from: https://www.eduroam.edu.au/.
60. *PLE Computers Promate 10000mAh Dual USB Power Bank Gadgets VOLTAG-10.BLACK Promate*. 2019; Available from: https://www.ple.com.au/Products/629011/Promate-10000mAh-Dual-USB-Power-Bank.
61. *Google Data Studio*. 2019; Available from: https://datastudio.google.com/u/0/navigation/reporting.
62. Mendoza-Silva, G., et al., *Long-term WiFi fingerprinting dataset for research on robust indoor positioning.* Data, 2018. **3**(1): p. 3.
63. *ArcMap | ArcGIS Desktop*. 2019; Available from: https://desktop.arcgis.com/en/arcmap/.
64. *Check the best WiFi analyzer apps for Android*. 2019; Available from: https://www.netspotapp.com/wifi-analyzer-apps-android.html.
65. Naseribafrouei, A., et al., *Correlation between the human fecal microbiota and depression.* Neurogastroenterology & Motility, 2014. **26**(8): p. 1155-1162.
66. Xiao, C., et al., *Using Spearman's correlation coefficients for exploratory data analysis on big dataset.* Concurrency and Computation: Practice and Experience, 2016. **28**(14): p. 3866-3878.
67. Georgievska, S., et al., *Detecting high indoor crowd density with Wi-Fi localization: a statistical mechanics approach.* Journal of Big Data, 2019. **6**(1): p. 31.
68. Zheng, Y. and X. Zhou, *Computing with spatial trajectories*. 2011: Springer Science & Business Media.
69. *Normalization | Codecademy*. 2019; Available from: https://www.codecademy.com/articles/normalization.
70. Shue, S. and J.M. Conrad. *Reducing the effect of signal multipath fading in RSSI-distance estimation using Kalman filters*. in *Proceedings of the 19th Communications & Networking Symposium*. 2016. Society for Computer Simulation International.
71. Huang, Q., et al., *Refining Wi-Fi based indoor localization with Li-Fi assisted model calibration in smart buildings.* arXiv preprint arXiv:1602.07399, 2016.
72. Chen, C., et al., *Indoor GPS with Centimeter Accuracy using WiFi.*
73. Dere, M. *Predicting location from RSSI signals*. 2019.
74. Abbas, M., et al. *WiDeep: WiFi-based accurate and robust indoor localization system using deep learning*.
75. Mittal, A., S. Tiku, and S. Pasricha, *Adapting Convolutional Neural Networks for Indoor Localization with Smart Mobile Devices*, in *In Proceedings of the Great Lakes Symposium on VLSI*. 2018, ACM: Chicago, IL, USA. p. 117-122.
76. *Documentation for app developers*. 2020 [cited 2020 13/04/2020]; Available from: https://developer.android.com/docs.
77. *Wi-Fi scanning overview*. 2020 [cited 2020 13/04/2020]; Available from: https://developer.android.com/guide/topics/connectivity/wifi-scan.
78. Zhang, W., et al., *Deep neural networks for wireless localization in indoor and outdoor environments.* Neurocomputing, 2016. **194**: p. 279-287.

79. BelMannoubi, S. and H. Touati. *Deep neural networks for indoor localization using WiFi fingerprints*. in *International Conference on Mobile, Secure, and Programmable Networking*. 2019. Springer.

80. Pham, V., et al. *Dropout improves recurrent neural networks for handwriting recognition*. in *2014 14th International Conference on Frontiers in Handwriting Recognition*. 2014. IEEE.

81. Chung, H., S.J. Lee, and J.G. Park. *Deep neural network using trainable activation functions*. in *2016 International Joint Conference on Neural Networks (IJCNN)*. 2016. IEEE.

82. Kingma, D.P. and J. Ba, *Adam: A method for stochastic optimization.* arXiv preprint arXiv:1412.6980, 2014.