**Master's Thesis**

# A Design of IoT-based Searching System for Victim Localization

M. Rosyidi

March 14, 2019

Graduate School of Information Science
Nara Institute of Science and Technology

A Master's Thesis
submitted to Graduate School of Information Science,
Nara Institute of Science and Technology
in partial fulfillment of the requirements for the degree of
MASTER of ENGINEERING

M. Rosyidi

Thesis Committee:

| | |
|---|---|
| Professor Kazushi Ikeda | (Supervisor) |
| Professor Keiichi Yasumoto | (Co-supervisor) |
| Assoc. Prof. Takatomi Kubo | (Co-supervisor) |
| Assist. Prof. Hiroaki Sasaki | (Co-supervisor) |
| Assist. Prof. Nishanth Koganti | (Co-supervisor) |

M. Rosyidi

**Abstract**

As a disaster is an unpredictable event, it may be difficult to entirely escape damages even if evacuation drills are carried out periodically. In such case, people requiring evacuation assistance, like aged and disabled people, are prone to be victims. To minimize the damage, fire departments and local governments need to swiftly grasp the situation of the afflicted areas in order to execute rescue operations. In this thesis, to support the rescue operations, we propose the design and implementation of an Internet of Things (IoT) devices based search system to grasp the presence of people requiring evacuation assistance. The IoT device captures Wi-Fi signal and data packets from the smartphones of the victims, and provides an analysis for victims localization in open and closed spaces. Through the experimental results, we show that our closed-space approach became a complement to the open-space approach results. The closed-space approach shows better results than the open-space approach in areas with obstacles, otherwise the open-space approach works better in other areas such as line-of-sight buildings. Also, the system provides a visualization of victims in the disaster areas to support the rescue operations.

**Keywords:**

IoT, Wi-Fi, GPS, victim localization, Smartphones, Disaster

# Acknowledgements

First and foremost, I would like to thank God the Almighty (Allah SWT) for giving me the strength, knowledge, and opportunity to undertake this Master study. Without His blessings, this achievement would not have been possible.

I would like to sincerely thank my supervisor, Professor Kazushi Ikeda, for his guidance and for providing me with an excellent atmosphere for doing research. It is an honor for me to be one of his students in the Mathematical Informatics Laboratory. Thank you to all the thesis committee members, Professor Keiichi Yasumoto, Assoc. Prof. Takatomi Kubo, Assist. Prof. Hiroaki Sasaki , Assist. Prof. Nishanth Koganti for supervising the content of this thesis. Additionally, I send my gratitude to Assistant Professor Shigeru Kashihara and Assistant Professor Doudou Fall for their invaluable guidance, patience, and supporting me in this research. Special thanks to all the members of MI Lab who also supported and helped me in daily life.

I send my deep gratitude to RISET-PRO, KEMENRISTEKDIKTI, Indonesian for supporting me with scholarships to study in NAIST. I send my gratitude to my institution in Indonesia, Badan Pengkajian dan Penerapan Teknologi (BPPT) for giving me the chance to continue my studies in Japan. I give my gratitude to the staff of the International Student Affairs Division of NAIST and Information Science office for their help regarding some administrative matters during my stay in Japan.

I would also like to thank all Indonesian students in NAIST, who were always supporting me during my study. Last but not least, I thank my parents Mahnim and Dahrun, my wife Ratih Destarina, my children Aqil and Eara, and all of my family members for supporting me in my life. They were always supporting me and encouraging me with their best wishes, cheering me up and standing by me through the good and bad times.

# Contents

# List of Figures

# List of Tables

# 1 Introduction

## 1.1 Background

When disaster strikes, fire departments and local governments need to swiftly grasp the situation of the afflicted areas in order to execute rescue operations as 72 hours after the occurrence is known as the critical time window for saving lives [1]. However, as a disaster is an unpredictable event, it may be very difficult to escape damage entirely even if evacuation drills are carried out periodically. In such case, the people requiring evacuation assistance, such as aged and disabled people, are prone to be victims.

In Japan, to minimize disaster victims, it is compulsory for local governments to make a list of the people that require evacuation assistance in order to take necessary measures for saving their lives. However, in a disaster-stricken area, it may be impossible to make all safety confirmation for listed people because the human resource for rescue operations is limited. For this reason, it is highly desired to grasp the presence of the victims promptly.

As a conventional way to make safety confirmation, one may make a phone call to a certain number. However, during disasters, it may be difficult to make it by phone due to congestion. In recent years, with the rapid advancement of smartphones, the penetration rate makes the upward trend. As an alternative for a call, some companies make safety confirmation for employees through web applications and emails on smartphones, but these applications may not be feasible for local governments due to the limitations in terms of cost and other aspects. Also, if the base stations in the affected areas are rendered unusable, the survivors will not be able to connect to the Internet.

This research focuses on promptly grasping the presence of the victims requiring evacuation assistance. However, as described above, although the current way strongly depends on the communication including the Internet, we have no choice but to return to a primitive way, i.e., visiting, when the Internet is not working. To get rid of such a situation, we try to utilize the Wi-Fi signal emitted by smartphones to find victims. A smartphone is commonly equipped with three types of communication media, i.e., LTE, Wi-Fi, and Bluetooth. Wi-Fi holds the following advantages. First, we can easily capture the signal strength and the packet of Wi-Fi by using a small computer like a Raspberry Pi (RasPi)*. We can also analyze the contents of the captured packets. Besides, Wi-Fi medium is installed in a handheld game console like Nintendo 3DS that a child may carry, not only a smartphone. Thus, the utilization of Wi-Fi to find persons has the potential to be extended to practical use. Note that this thesis assumes that a person requiring evacuation assistance has registered the MAC address of his/her smartphone with his/her local government.

In this thesis, we propose the design and implementation of an Internet of Things (IoT) based search system to grasp the presence of victims requiring evacuation assistance. The IoT device captures Wi-Fi signal and packets from smartphones, and performs victims localization. To make it, we first explain how and what information an IoT device captures, and then we explain how the IoT device creates an estimated area from many captured packets. Although the main contribution of this thesis is to show an estimated area by capturing Wi-Fi packets, the IoT device has a great potential as a sensor for finding victims because it can be attached to various objects and living beings, e.g., drones, cars, bikes, bicycles, humans, rescue dogs, etc. Also, we show preliminary results in a real environment.

## 1.2 Motivation and Purpose

Our motivations for this thesis are : 1) To design an IoT system that is easy to attach to any kind of rescuer from various objects and living beings, and 2) To design IoT system that is useful in 72 hours rescue operations, to quickly take action(s) and reduce risk. Our purposes are : 1) To estimate the location of the victims from various disasters, and 2) To make visualization of estimation area

*Raspberry Pi B+

for rescue operations.

## 1.3 Contributions of this thesis

Search and rescue teams need to quickly grasp the situation in a given disaster area, and sometimes it is difficult to find victims in specific locations. The contribution of this thesis is to support rescuers to detect the victims presence area by providing victims location estimation.

## 1.4 Organization

This thesis is organized as follows. Chapter 2 provides the details the literature review for the related work on applications for finding victims and localization method using Received Signal Strength Indicator (RSSI) captured by Wi-Fi sensors. Chapter 3 provides information about received signal strength indicator (RSSI), distance derivation and position, and implementation scenario for victims area estimation. Chapter 4 explains the design and implementation of our IoT-based search system. This chapter contains general system overview, system design, and implementation of the system. Finally, Chapter 5 shows parameter evaluation in open-space and closed-space approaches, data analysis, visualization and evaluation results.

# 2 Related Work

## 2.1 Disaster App for Finding Victims

This research is aimed at grasping the presence of victims requiring evacuation assistance. Many researchers have proposed various attractive applications using smartphones for collecting information of victims. In this section, we first survey the papers related to those applications. Also, we try to indicate a presence area based on Wi-Fi information and introduce the related work.

Literatures [2–4] focus on collecting information of disaster victims. Suzuki et al. [2] proposed SOSCast, an application that can propagate SOS messages over device-to-device (D2D) communication by smartphones. When receiving the SOS messages, a rescuer estimates the location of the victim from the positions where the information was collected. To send a message over D2D communication, Nishiyama et al. [3] proposed message transmission in the architecture of Delay-/Disruption-Tolerant Network (DTN) over Mobile Ad-hoc Network (MANET). Also, in the field experiment, they showed that messages could be relayed through D2D communication by their smartphone application. Lu et al. [4] also presented a smartphone application, i.e., TeamPhone, to collect the information of disaster victims through a D2D communication. Similarly, many researchers have proposed useful smartphone applications for collecting information. However, the fact that the applications are installed in smartphones causes a major issue: the spread of the application.

## 2.2 Location Estimation

The study of localization corresponds to the related work for indicating the presence area of a victim. Many researchers have been studying the topic of localization [5]. For instance, Chen et al. [6] proposed a convolutional neural network (CNN)-based indoor Wi-Fi localization method. Rida et al. [7] proposed indoor localization based on Bluetooth signal strength. The authors set some node points inside the area and using Bluetooth technology inside the smartphone, the application can estimate the distance of people to the three nearest nodes. In general, the studies of indoor localization based on Wi-Fi information are utilizing fixed access points (APs). However, in our scenario, as we are aiming to search a victim without any fixed special equipment like AP, it is difficult to adapt indoor localization techniques.

Sichitiu et al. [8] proposed outdoor localization wireless nodes using range measurements from multiple, sparsely located, beacon stations with known locations. Other researchers, Kuo et al. [9] build a framework for monitoring the signal strength based on the transmission protocol AODV (Ad hoc On-Demand Distance Vector) and the placement of Zigbee is established and used in sensor node positioning, this system not only localizes nodes in indoor areas but also in outside areas. Literatures [8, 9] have similar systems by setting static beacons to detect objects or nodes that enter the area of set. Similarly with indoor localization, it is difficult to adapt these approaches. We do not use static beacons and set areas. We make flexible sensors that are easy to employ in any other situation and area.

Annadata et al. [10] proposed a system that automatically detects the people in a building using multiple channels such as Wi-Fi and motion detection. They presented a mathematical model to detect identifiers that belong to actual persons in a privacy preserving manner using IoT technologies. This research is similar to our scenario, the problem is that they just propose some techniques that focus on the people inside building, they did not test real life scenarios.

# 3 Victims Localization Method

## 3.1 Problem

Rescue operation plans for finding victims must be executed quickly. Sometimes, problems occur with expanding search areas or due to the complexity of environment. In the worst case, inside the disaster area, a base station is unusable. The victims would be isolated and it would be difficult for them to ask for help. They cannot get access the internet for sending email and cannot send messages directly. Even if the base station is still usable, but the victims search area is very complex, then it will take a longer time to find and identify the locations of the victims.

## 3.2 Requirements

### 3.2.1 Assumption in Disaster Area

In the case of a real emergency, emergency response coordinators (ERCs) ensure that the disaster is reported to the proper emergency responders, like fire figthers, as soon as possible. Some corporations also suggest that ERCs should have current attendance list on hand and take it with them to the assembly point. Once in the assembly point, they perform the task of accounting for every person that was in the building prior to the disaster as indicated per the attendance list [10].

Attendance lists are usually held by local governments as the lists of their resident members. With the assumption that every person carries at least one device such as a smartphone, a game console, a laptop, etc., local governments can make a list using the registered MAC address of every resident member.

### 3.2.2 Wi-Fi Measurements

Wi-Fi signals can be viewed in different kinds of networking layers. When viewed as radio waves in the physical layer, it is unnecessarily difficult to identify the transmitter since there is no easily accessible MAC address in this layer. When viewed on the data link layer, it is easy to identify the transmitter by their MAC address [11]. The disadvantage of using the data link layer is that you will no longer be able to get the exact signal strength in decibel meters. By using the data link layer, you will end up with the signal strength defined in RSSI which is short for Received Signal Strength Indicator. RSSI values are representations of signal strength. The range in which the values may vary change from device to device based on the receiving device's wireless networking card [12].

The signal strength is measured in dBm (decibel in milliwatt). A Wi-Fi station has an EIRP (Equivalent Isotropically Radiated Power) of 100mW - 1000mW (20dBm - 30dBm) [13].

$$Lp(dBm) = 10 * \log \frac{P}{1mW} \tag{3.1}$$

### 3.2.3 Internet of Things (IoT) Devices

Internet of Things (IoT) is a concept that aims to expand the benefits of continuously connected internet connectivity. As for capabilities such as data sharing, remote control, and so on, including also objects in the real world. Examples of food, electronics, collectibles, any equipment, including living beings that are all connected to local and global networks through embedded and always on censorship. Essentially, the IoT refers to objects that can be uniquely identified as virtual representations in an Internet-based structure. Many predict that the IoT is "the next big thing" in the world of information technology. This is because a lot of potential that can be developed with the technology of the IoT [14].

An IoT device has a radio that can send and receive wireless communications. The IoT wireless protocol is designed to meet several basic services, operate with low power and bandwidth, and work in mesh networks. Some devices work at 2.4 GHz field frequencies, which are also used by Wi-Fi and Bluetooth, and sub-GHz coverage. The sub-GHz frequencies including 868 and 915 MHz have an advantage in low interference. The method used by the IoT is wireless or automatic control without distance. Implementation of the IoT itself usually always follows the developer's desire in developing an application that he created, if the application was created to help to monitor a room then the implementation of the IoT itself must follow the flow of programming diagrams about the sensor in a house, how far the distance room can be controlled, and internet network speed used. The development of network and Internet technologies such as the presence of IPv6, 4G, and WiMAX, can help the implementation of the IoT become more optimal, and increase the allowed distance for communication, making it easier for us to control things [14].

## 3.3 Approach

### 3.3.1 Distance Derivation

To estimate victim localization, the first step needed is deriving the distance. There are two approaches used in this research and each approach will employ different calculation processes of distance.

#### 3.3.1.1 Open-space Distance Estimation

To obtain the distance, we conducted short range measurements [15] without estimated standard deviation and mean of the group data by distance. The measurements of the RSSI values of a smartphone from 1 meter until 70 meters. Measurement will be conducted in indoor and outdoor, and the results will be treated equally without worrying about obstacles and environment factors.

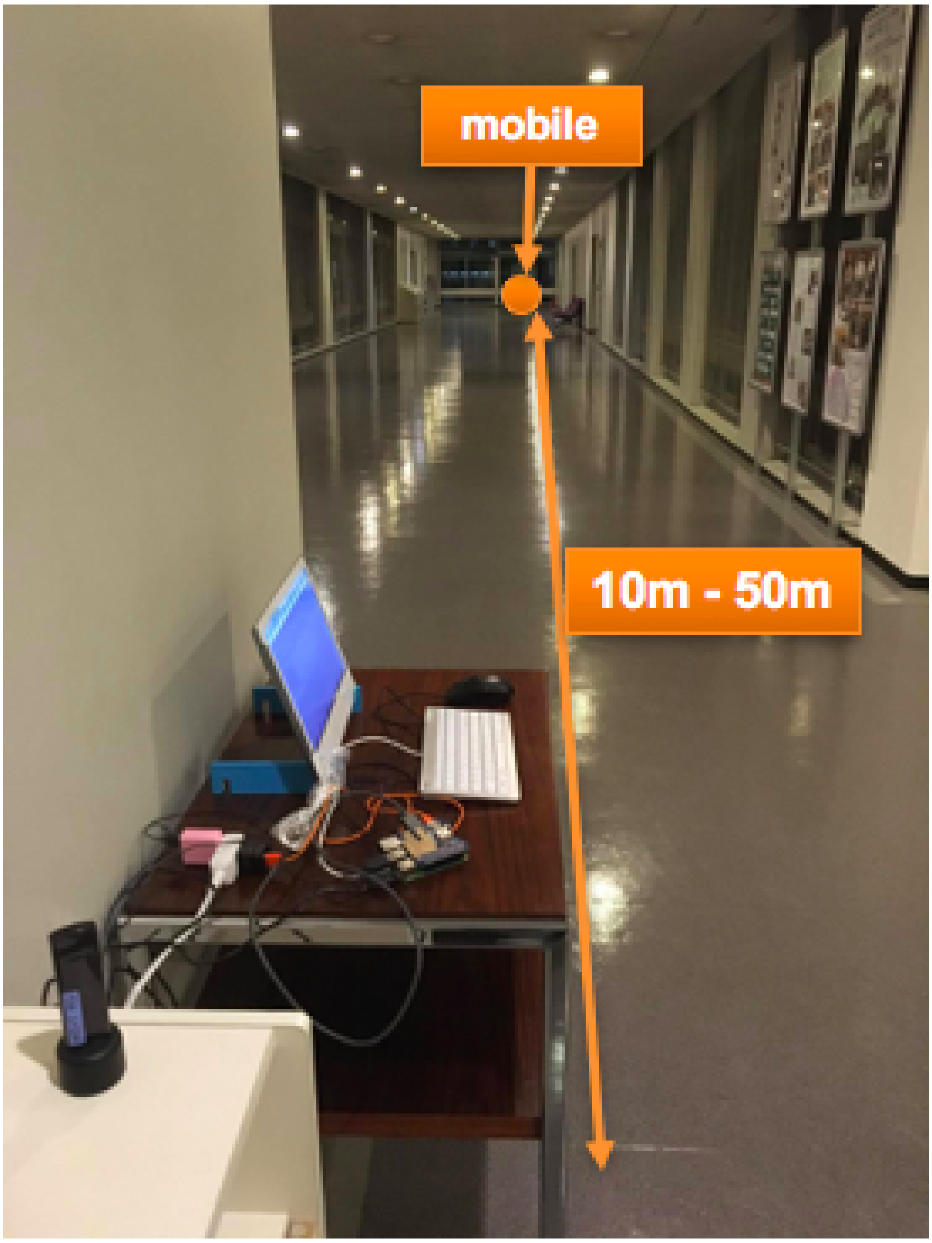


Figure 3.1: Distance derivation - Indoor RSSI measurements.

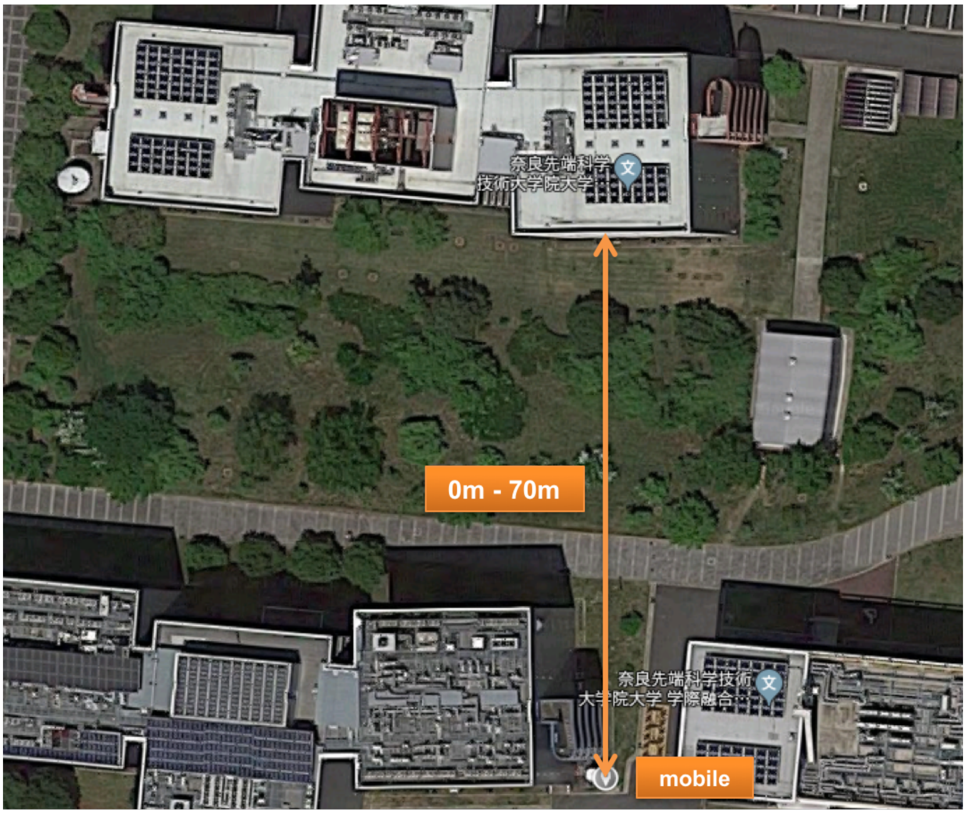


Figure 3.2: Distance derivation - Outdoor RSSI measurements

After collecting the RSSI, we decide the distance between the Wi-Fi sensor and the target smartphone for each RSSI value. We obtained the value of RSSI in certain distance set, and then averaged the RSSI by applying linear regression, which is stated as in equation 3.2.

$$Y = a + bX \tag{3.2}$$

$Y$ is the predicted dependent variable of distance, $X$ is an independent variable of RSSI value, $a$ is the $y$-intercept and $b$ is the slope of the line.

$$a = \bar{Y} - b\bar{X} \tag{3.3}$$

$$b = \frac{\sum(X_i - \bar{X})(Y_i - \bar{Y})}{\sum(X_i - \bar{X})^2} \tag{3.4}$$

$\bar{Y}$, $\bar{X}$ are the average values of Y, X and $X_i$, $Y_i$ are the paired of sample (i=1,....,n).

#### 3.3.1.2 Closed-space Distance Derivation

Received signal strength indicator (RSSI) indicated the energy loss in the process of signal transmission. The value of RSSI indicated the distance value between receiver and transmitter. In an open-space environment, we can define the path loss model [16] as in Equation 3.5

$$Loss = 32.44 + 10n\log(d) + 10n\log(f) \tag{3.5}$$

Loss indicates the path loss of signal energy, $d$ indicates the signal transmission distance (m), $f$ indicates the wireless signal frequency (MHz), and $n$ indicates the path attenuation factor in the actual environment.

However, in the actual environment such as buildings sites, indoor buildings, forest, water; we need to consider shade and absorbance by obstacles and the interference of scattered reflection, and so forth. In such cases, the path loss model is described as in equation 3.6 [17].

$$P_L(d) = P_L(d_0) + 10n\log(d/d_0) \tag{3.6}$$

$P_L(d)$ indicates the path loss of receiving sign when the measuring distance is $d(meters)$ and it indicates absolute power value. $P_L(d_0)$ indicates the path loss of receiving sign when reference distance is $d_0$. $n$ indicates the path loss index in specific environment, it indicates the speed of the path loss, which increases as the number of obstacles increases. RSSI receiving nodes follows equation 3.7 :

$$RSSI = P_t - P_L(d) \tag{3.7}$$

$P_t$ indicates the signal transmission power and $P_L(d)$ indicates path loss when the distance is $d$. If we set A as reference signal strength, then

$$A = P_t - P_L(d) \tag{3.8}$$

The path loss model, which is measured at the real distance $d$(m), is as follows:

$$P(d) = P(d_0) - 10n\log(d/d_0) \tag{3.9}$$

$P(d)$ indicates the received signal strength when the measured real distance is $d$(meters). $P(d_0)$ indicates received signal strength when reference distance is $d_0$. If we take the reference distance $d_0 = 1$ m; it can be obtained from formulas 3.7, 3.8, and 3.9

$$RSSI = A - 10n\log(d) \tag{3.10}$$

$d$ is regarded as the undetermined distance.

$$d = 10^{(A-RSSI)/10n} \tag{3.11}$$

#### 3.3.1.3 Estimation of Reference Signal Strength (A)

Reference (A) must be defined before calculating the distance in the closed-space approach. To optimize formula 3.9, experiments considering 1 meter for each device must be considered. Experiment for deriving reference can be taken in indoor or outdoor, and to optimize the results both the Wi-Fi sensor and the smartphone are placed horizontally at 1 meter from each other, and vertical to the ground..

#### 3.3.1.4 Estimation of Path Loss Index (n)

The values of n (path loss index) that are satisfied by equation 3.11 depend on the obstacles in the environment of the experiment, more complex experiments will make it challenging to measure the values of $n$. Some researchers captured thousands of data to define the values of n in specific environments. In this thesis, we hard-code the values of $n$ as in Table 5.1

### 3.3.2 Center Area Estimation

We assume that planet earth follows the spherical form, every point on the surface of earth will have same distance to its center (R). Based on the spherical theory, the rescuer position — latitude and longitude information — must be converted

Table 3.1: Path Loss Exponents for different environments

| Environment | Path Lost Exponent,n |
|---|---|
| Free Space | 2 |
| Urban area cellular radio | 2.7 to 3.5 |
| Shadow urban cellular radio | 3 to 5 |
| In building line of sight | 1.6 to 1.8 |
| Obstructed in building | 4 to 6 |
| Obstructed in factories | 2 to 3 |

to the earth reference system, axis values (x,y,z), in the Cartesian coordinates system as follow [18]:

$$\begin{aligned} x &= R * \cos(lat) * \cos(lon) \\ y &= R * \cos(lat) * \sin(lon) \\ z &= R * \sin(lat) \end{aligned} \tag{3.12}$$

R is the approximate radius of the earth (e.g., 6371km). We will use two formulas to define localization of victims, area estimation for the open-space and closed-space approaches. The center area estimation for smartphone target in open-space approach is defined in equation 3.13 :

$$2 \begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 2R^2 - d1^2 \\ 2R^2 - d2^2 \\ 2R^2 - d3^2 \end{bmatrix} \tag{3.13}$$

$(x1, y1, z1), (x2, y2, z2), (x3, y3, z3)$ represent the Cartesian coordinates of Wi-Fi sensor against a mobile device, while $(x, y, z)$ are the Cartesian variables of the possible position of a mobile device.

In the analysis process for the open-space approach, we clean the data carefully after obtaining the Cartesian coordinates system. By setting the tentative center of each data, we delete the outlier and change the candidate data if the obtained distance is very far from the center. We cannot deliver this system in the close-space approach. We utilize all data without cleaning and replacing. The analysis process in the closed-space approach is more simple than in the open-space approach. For the closed-space approach, we can derive a center area estimation by using Trilateration Techniques [19] :

$$x = \frac{d_1^2 - d_2^2 + x_2^2}{2x_2} \tag{3.14}$$

$$y = \frac{d_1^2 - d_3^2 + x_3^2 + y_3^2}{2y_3} - \frac{x_3}{y_3}x \tag{3.15}$$

$$z = \sqrt{d_1^2 - x^2 - y^2} \tag{3.16}$$

To obtain the actual position that represents the latitude and the longitude we must use the following formulas:

$$p = (180/\pi) * \arcsin(\mathrm{z/R}) \tag{3.17}$$

$$q = (180/\pi) * \arctan(\mathrm{y/x}) \tag{3.18}$$

where $p$ is the latitude of center area estimation and $q$ is the longitude of center area estimation.

### 3.3.3 Visualization

The system employs Google Maps APIs* for the visualization. The position of the smartphone is not the center of the area, because it is difficult to estimate the accurate position. Even if the actual position is different with the estimated position, a rescuer estimates the position if the actual position is within the area. Also, as the visualization has a circle with three colors for every radius of 10 m, a rescuer can estimate the situation clearly.

*https://developers.google.com/maps/

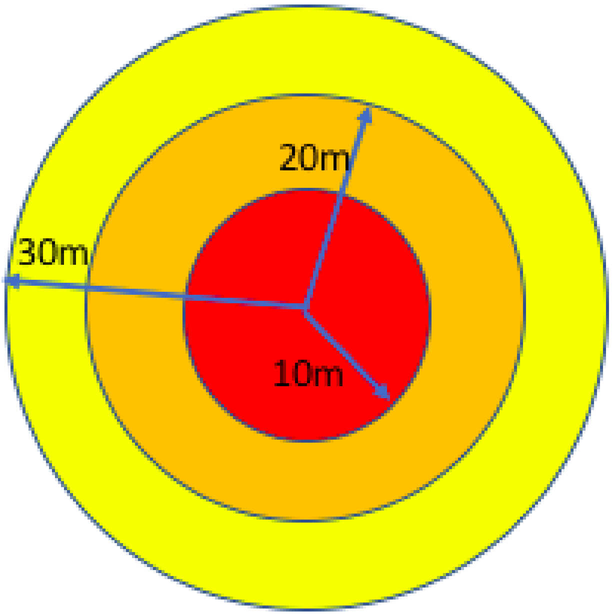


Figure 3.3: Three-layer visualization - 10 m (first layer), 20 m (second layer),and 30 m (third layer)

Figure 3.3 shows a three-layer visualization. The analysis will be satisfied if the target smartphone is inside the localization area. Three layers are needed to determine the rescuers priorities. The first layer will become the priority area before inspection of the second layer and so on. But in practice, the rescuer can estimate the position by evaluating the condition of the area. Firstly, the rescuer can inspect the building or a specific object inside the area or near the area estimation.

# 4 Design and Implementation of the IoT-based Search System

## 4.1 System Overview

The proposed system employs Wi-Fi signal strength and probe request. Based on assumption, a person holds smartphone, and the MAC address of the smartphone is registered to a list of persons requiring evacuation assistance. For instance, as depicted in Figure 4.1, in a disaster scenario, a rescuer collects Wi-Fi packets by using the IoT-based search system while moving.

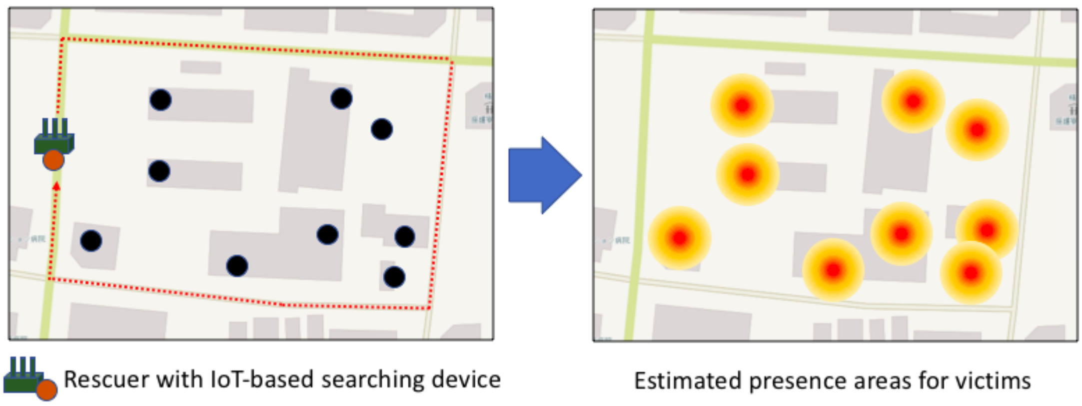


Figure 4.1: Use Case in Disaster Scenario

## 4.2 System Design

IoT components in this research consist of Unit board Rasberry Pi, Wi-Fi sensor*, a GPS device†, and a Camera Sensor. A Battery is attached to this system for powering the board and all of the sensors. The camera is not used yet in this research, but we attach it for the future development of our proposal.

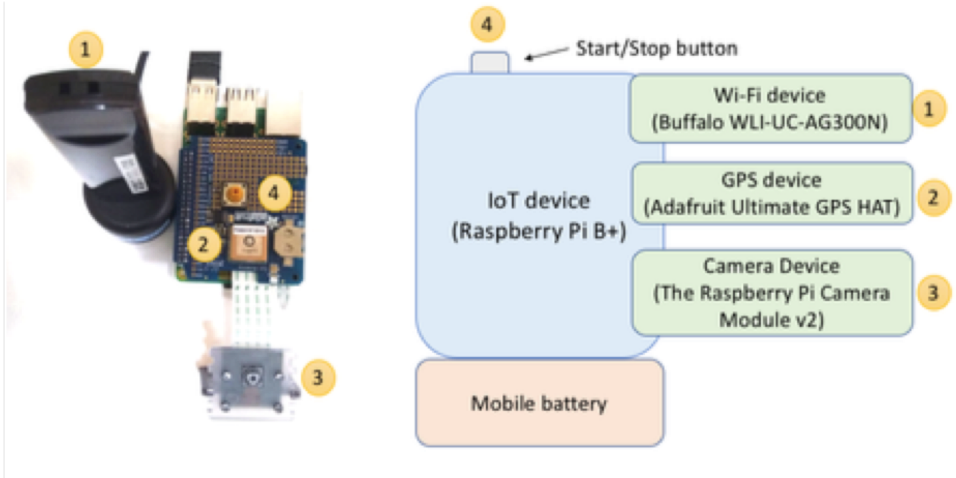


Figure 4.2: System Design

### 4.2.1 Raspberry Pi

Raspberry Pi is a single board computer for docking and processing Wi-Fi sensor, camera, and GPS. It is a low-cost computer capable of running a GNU/Linux desktop environment on a low-power ARM processor. All ports on a Raspberry Pi are capable of running three sensors that will be used in this research. RPi port can handle camera and Wi-Fi sensors and GPS will communicate by using a GPS Hat that is compatible with Rasberry PI. Sensor data will be collected simultaneously and the battery is powerful enough to handle the Raspberry PI movement of a drone flying around the location of the disaster area. The position of the GPS module paired to Raspberry Pi, it has GPS module support tool used as GPS signal capture to obtain a coordinate that will be displayed on the

*Buffalo WLI-UC-AG300N
†Adafruit Ultimate GPS HAT for Raspberry Pi A+/B+/Pi 2 - Mini Kit

map in the form of tracking results. NMEA data from a GPS sensor stored in the Raspberry Pi. The processed NMEA formats are GPGGA and GPRMC. the GPGGA format contains position data, i.e., latitude, longitude, altitude, and UTC Time, whereas the GPRMC format contains RTC (Real Time Clock) update data.

### 4.2.2 Wi-Fi Sensor

We use the Wi-Fi sensor Bufallo WLI-UV-AG300 for capturing probe requests from smartphones. This Wi-Fi sensor can capture devices until 100 meters. Along with the Wi-Fi sensor, we use tcpdump to capture probe requests data from target smartphones. Timestamp of probe requests data will be validated by GPS satellite time.

***tcpdump -I -i [target wifi] type mgt subtype probe-req -w [name of file].pcap***

The PCAP file of the probe requests data contains information of the target devices: Mac address of the, execution time, RSSI value, and device manufacturer.

### 4.2.3 Adafruit Ultimate GPS Hat for Rasberry PI

GPS Hat takes over the Raspberry Pi's hardware UART to send/receive data to and from the GPS module. The specifications of this GPS Hat are : -165 dBm sensitivity, 10 Hz updates, 66 channels, only 20mA current draw, built in Real Time Clock (RTC) - slot in a CR1220 backup, PPS output on fix by default connected to pin 4, internal patch antenna which works quite well when used outdoors. UFL connector for external active antenna for when used indoors or in locations without a clear sky view, Fix status LED blinks to let you know when the GPS has determined the current coordinates. The accuracy of this GPS Hat is 10 meters, so our experiment area is within 10 meters. It is possible to obtain a different RSSI value with same position. Like usual, GPS needs warming up before showing the fix position. Spare time for warming is very important in merged data process.

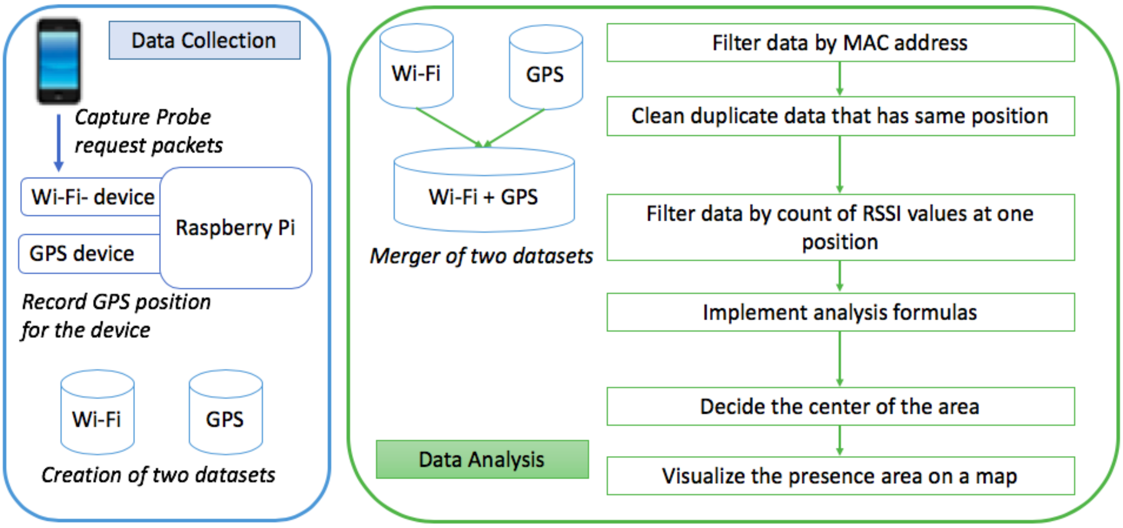


Figure 4.3: System process - data collection and data analysis processes

## 4.3 Implementation

### 4.3.1 System Process

The process is divided into two: data collection and data analysis. In the data collection process, the IoT-based searching system, that is equipped with Wi-Fi and GPS devices, collects probe requests data from a targeted smartphone. Besides, the data analysis process estimates a presence area where there may be the targeted person. The following sections make the detailed explanations. Figure 4.3 illustrates the process of the system.

### 4.3.2 Experimental Scenarios

We conducted six experiments in four different places. The location of the experiments 1, 2, and 3 has a similar situation: there are buildings and trees as obstacles. The location for experiments 4, 5, and 6 was a wide empty area for the most part with grass and few trees as obstacles. Figure 4.5 shows the experimental scenarios. We dispatched the IoT device to the area and carried them by walking around the area setup. We set the targeted smartphone inside the area with the power turned on and no access to connection. We measured the actual position of the smartphone to calculate the error distance in the analysis process.


Figure 4.4: Experimental areas (satellite view)


Figure 4.5: Scenario for the experiments 4, 5, and 6

# 5 Evaluation

## 5.1 Parameter Evaluation

### 5.1.1 Open-space distance measurement

After estimating the center position of a presence area, the presence area is depicted on a map. In this section, we first discuss the relationship between RSSI and distance. Although our method collects Wi-Fi RSSI and packets sent from a smartphone, each smartphone has a different signal power for sending packets. We then investigate RSSI and distance of two smartphones, i.e., iPhone 4s and Xiaomi Mi Max.

As shown in Figure 5.1, as the actual RSSI has variable values due to various invisible factors like multi-path fading; the linear regression model calculates the relationship between RSSI and distance. We also obtain the following two equations for the smartphones. $x$ means RSSI, while $y_{(smartphone)}$ indicates the distance.

$$y_{(iPhone)} := -2.2x - 99 \tag{5.1}$$

$$y_{(Xiaomi)} := -2.5x - 150 \tag{5.2}$$

From the graph, we can see that each smartphone has a different distance for RSSI. Also, as the experimental result is based on our environment that is an open-space, it may be changed if obstacles exist.

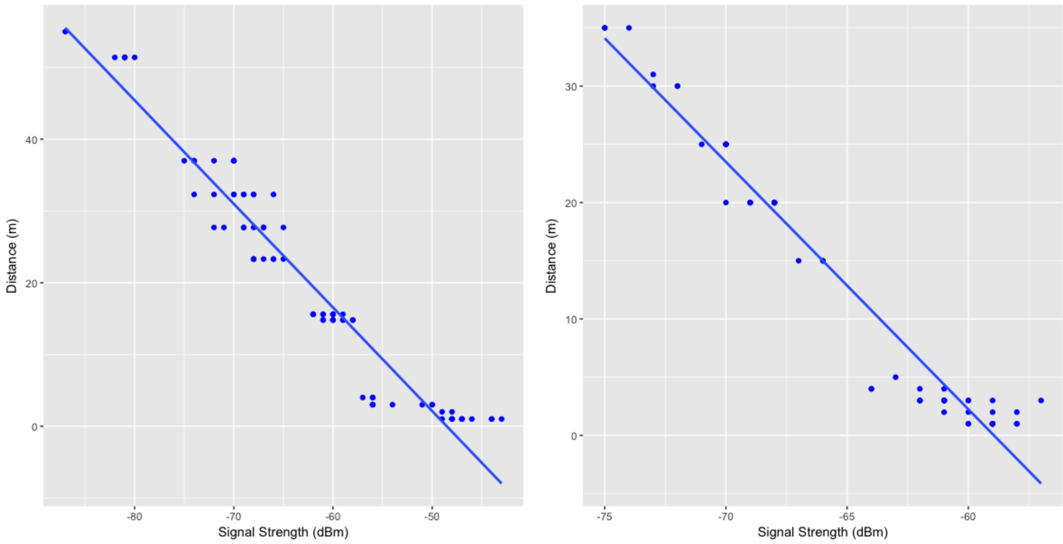


Figure 5.1: Relationship Between RSSI and Distance - iPhone (left side) and Xiaomi (right side)

### 5.1.2 Reference Signal Strength Estimation

Reference (A) is one parameter to calculate distance in closed-space approach. To get the optimum value reference, we conduct 1 meter horizontal experiment between smartphones and Wi-Fi sensor, and 1 meter vertical with both the smartphones and the Wi-Fi from ground in indoor line of sight.

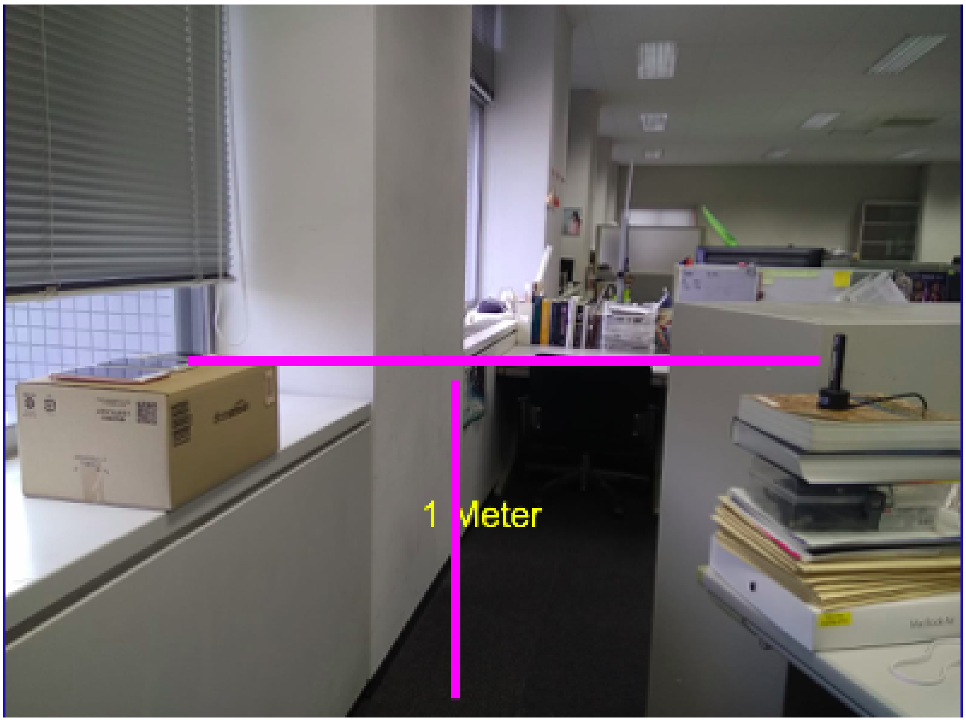


Figure 5.2: one meter experiment for reference

We calculate both references between iPhone 4s and Xiaomi Mi Max. Figure 5.3 shows the data distribution of iPhone 4s with a mean value of -45.66809 and figure 5.4 for Xiaomi Mi Max with a mean value of -44.2175. The red lines on the figures indicate the values of the mean.

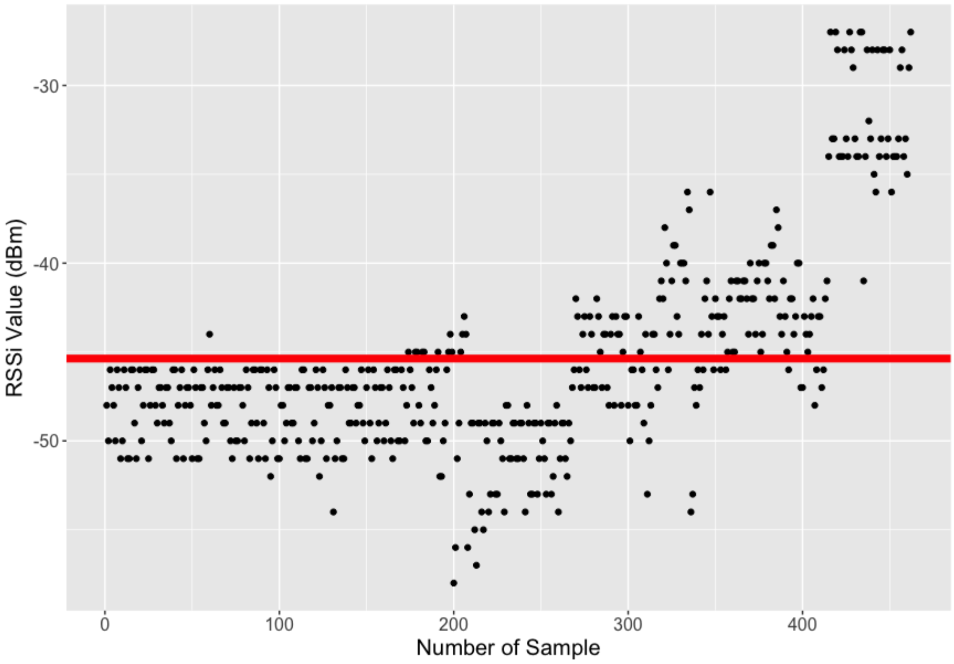


Figure 5.3: Reference for iPhone 4s

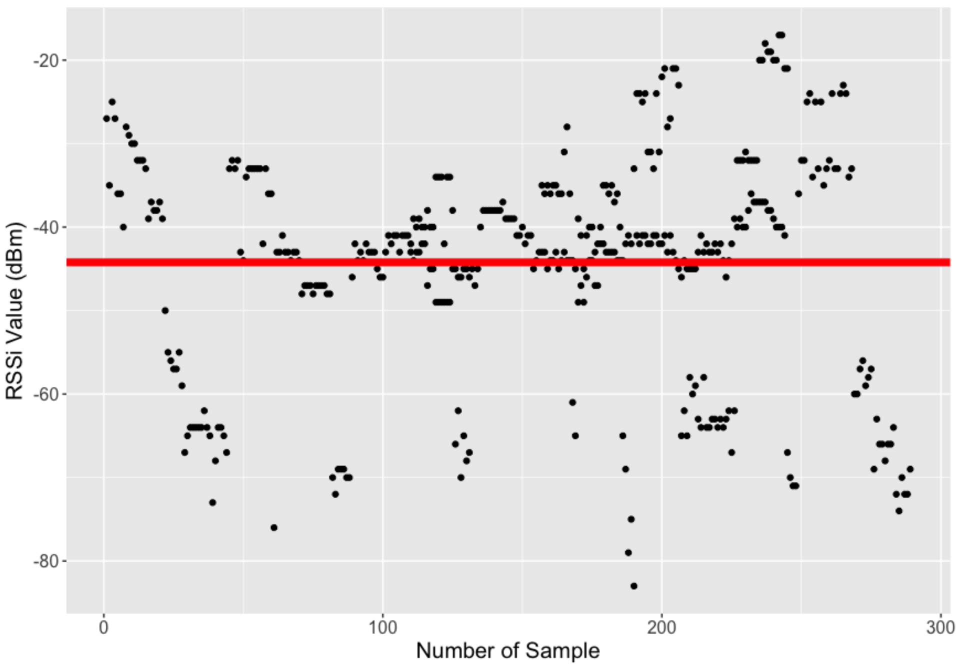


Figure 5.4: Reference for Xiaomi Mi Max

### 5.1.3 Path loss index estimation

It is difficult to directly define the value of n because it is related to the condition of the environment. Mostly, we conduct experiment in area with open-space and urban condition. The value of path loss index (n) is between 1.5 - 2.5 as shown in table 5.1

Table 5.1: Error Distance (ED) for 4 experiments with the path loss index varying between 1.5 to 2.5

| Path Loss Index,n | ED 1 | ED 2 | ED 3 | ED 4 |
|---|---|---|---|---|
| 1.5 | 26.39438141 | 43.63988431 | 40.20129585 | 28.52223245 |
| 1.6 | 14.48908509 | 30.91875911 | 17.82570988 | 15.14713383 |
| 1.7 | 8.550772447 | 21.6966277 | 9.35746564 | 14.03341205 |
| 1.8 | 4.333191332 | 14.77521724 | 11.93539475 | 16.02487258 |
| 1.9 | 4.920006793 | 9.382366038 | 15.13183745 | 17.62983364 |
| 2.0 | 1.397037923 | 5.31873454 | 17.18249875 | 18.61647008 |
| 2.1 | 2.810286669 | 2.517570704 | 18.42605658 | 19.4923382 |
| 2.2 | 3.331079398 | 2.253501735 | 19.58498549 | 20.68087303 |
| 2.3 | 3.360495691 | 3.410306903 | 20.00133808 | 21.42099144 |
| 2.4 | 3.631601731 | 4.618995785 | 20.77396128 | 21.42099144 |
| 2.5 | 3.847895603 | 5.912341234 | 21.13335683 | 21.74570097 |

We setup 4 experiments in different places around Takayama-cho area. These experiment are for estimating the value of path loss index (n). The situation in first place and second place is similar, so is the environment of third and fourth places. In the first and second places, the obstacles are mostly buildings, trees, and cars. The places are dominated by grass and some trees, and they are far from buildings. Table 5.1 shows that the minimum error of distance obtained by using path loss index is around 1.8—2.5, for the third and fourth places with path loss index around 1.6—2.0.

Based on the result, the first and second places can be assumed as free spaces to urban areas, the third and fourth places are in building line of sight to free space. We will use this path loss index value as a parameter in the analysis process.

## 5.2 Data Analysis

### 5.2.1 Open-space Analysis

To estimate the center of an area, the data will be selected appropriately. We here explain the whole processes for the estimation.

In our assumption, as we know the MAC address of a targeted user's smartphone, we can filter data by MAC address. However, in a wireless environment, since values of RSSI are fluctuating even if a smartphone does not move, the captured data may have outliers. That is, as we may obtain various values of RSSI in one location, it may indicate an incorrect location if poor values of RSSI are selected as representation. To alleviate the impact of the estimation by outliers, we need to remove as many of them as possible.

First, the captured data is filtered through equations (5.1, 5.2) and (3.12). For instance, we can estimate the distance for the value of RSSI by equations (5.1) and (5.2). When the distance calculated is less than 0 m, the value of RSSI has a high possibility to be an outlier. The outliers are removed from the data.

We then need to extract an RSSI that seemed to be accurate, among plural values of RSSI at one location. For selecting one data for one position, we count the number of each value of RSSI at the same position. We select the value that the count number is the largest as a representative RSSI on the location.

Next, to remove outliers carefully and estimate the center of an area, we add the following process in addition to the above filtering. At present, the data filtered may have more than three positions. By using position data, we calculate position $\mu$, i.e., $(x_\mu, y_\mu, z_\mu)$ as a tentative center position as equation 5.3.

$$x_\mu = \sum_{i=0}^{n} x_i, \quad y_\mu = \sum_{i=0}^{n} y_i, \quad z_\mu = \sum_{i=0}^{n} z_i \tag{5.3}$$

After this, we select the three positions from the data and estimate the center $C_j$, i.e., $(x_{Cj}, y_{Cj}, z_{Cj})$, by using equation 3.13 based on all combinations. That is, when we have seven positions, the 35 candidates for the center position are obtained. Here, to remove outliers carefully, we calculate distances $(D_j)$ between $\mu$ and $C_j$. If $D_j$ is between 50 m and 100 m, $(x_{Cj}, y_{Cj}, z_{Cj})$ is updated to $(x_\mu, y_\mu, z_\mu)$. Also, if $D_j$ is larger than 100 m, $C_j$ is removed as an outlier.

From the above process, as we obtain plural candidates for the center position, we need to select one center position. In this paper, we discuss the following three selection methods. The first one is a random selection. It selects one among them randomly. The second one is based on average values. It calculates the average of the positions like equation (5.3). Lastly, the third one is to select median among them. If we have multiple medians, we calculate their average.

### 5.2.2 Closed-space Analysis

Estimating the center area in closed-space analysis is simpler than in open-space analysis process. First, we combine probe requests data with position data and then we filter the data using the list of MAC addresses of the victims. We clean the data that has same position by selecting max count RSSI value as candidate. We finish selecting the data in this process, the next process is analysis of the data.

In the analysis process, first we calculate the distance using equation 3.11. Environment factor (A) and path loss index (n) will impact the value of the distance. We set the value of reference for iPhone 4s to -45.66809 and Xiaomi Mi Max to -44.2175. The path loss index selected is between 1.8-2.3 in semi-urban area and 1.6 - 2.0 in line of sight or open-space area. Negative value of distance will be counted as outlier of data and removed.

The next process is to estimate the center area using all the values of the path loss index and reference for each smartphone. After obtaining the distance, we will estimate the center area using trilateration with n points of data through equations 3.14, 3.15, and 3.16. We will obtain the plural results of center area estimation by the number of path loss index that we used. To select one candidate of center area estimation, we calculate the mean and the median of all the positions that are obtained from the trilateration process.

## 5.3 Visualization

Visualization is the last process that shows the victims search area on base-maps, by obtaining the center area estimation, we will set a three-layer visualization as reference in rescue operation. Each layer represents the priority for searching victims, but rescuers can decide another option by investigating a building or a specific object inside or near the search area.

Figure 5.5 shows the visualization for single devices and figure 5.6, for multiple devices. Three layers are set as priority searching areas, the targeted smartphone inside first layer is detected with good accuracy whereas in the third layer or outside, the detection accuracy is poor. The purple line in multiple devices visualization indicates the rescuer route.

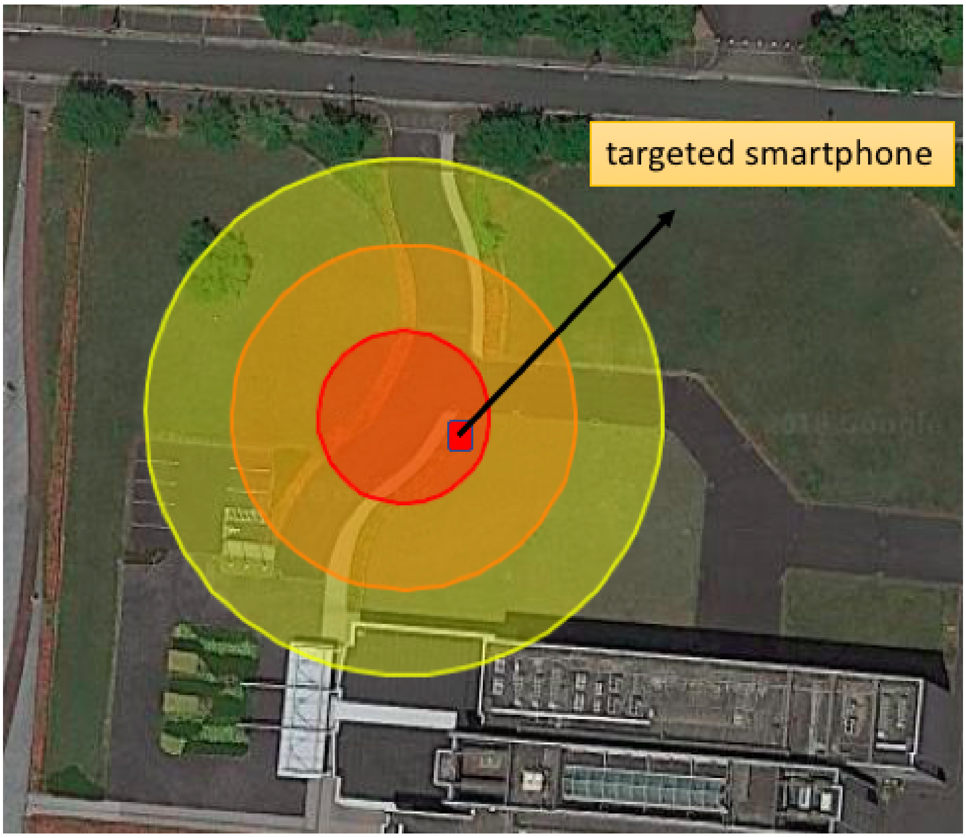


Figure 5.5: Visualization - single device

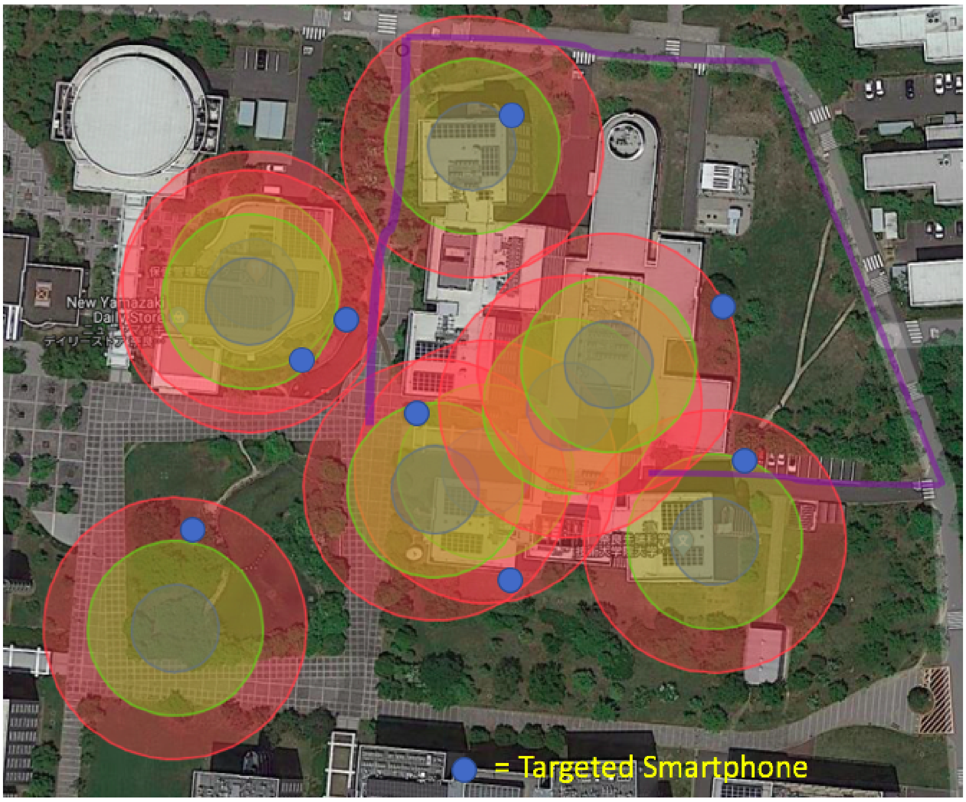


Figure 5.6: Visualization - multiple devices

## 5.4 Evaluation Results

In the experiment, a smartphone (iPhone 4s) is located on the ground, and then our system collects RSSI from the smartphones while we are on movement. The estimation process for the open and close-space has two ways to select the center of an area, i.e., by calculating the mean and the median of the data. Mean and median selection almost have the same results, then we can do comparison between them. To compare their accuracy, we ran the same experiment six times and compared the selection processes. The environment factor for experiment 1-3 is semi-urban area and 4-5 is line of sight open-space area.

### 5.4.1 Open-space Approach Evaluation

Figure 5.7 shows the distances between the actual position and the estimated position. We obtained poor results for the experiments 1, 2, and 3 because of the obstacles in the area. However, we obtained good results from experiments 4, 5, and 6 in the open-space area. From figure 5.7, we can see that the median selection is slightly better than the average selection. We suspect that median is better than mean because of the number of the data is not large enough. However, as the results are strongly depending on various factors, at present, we cannot identify which one is superior.

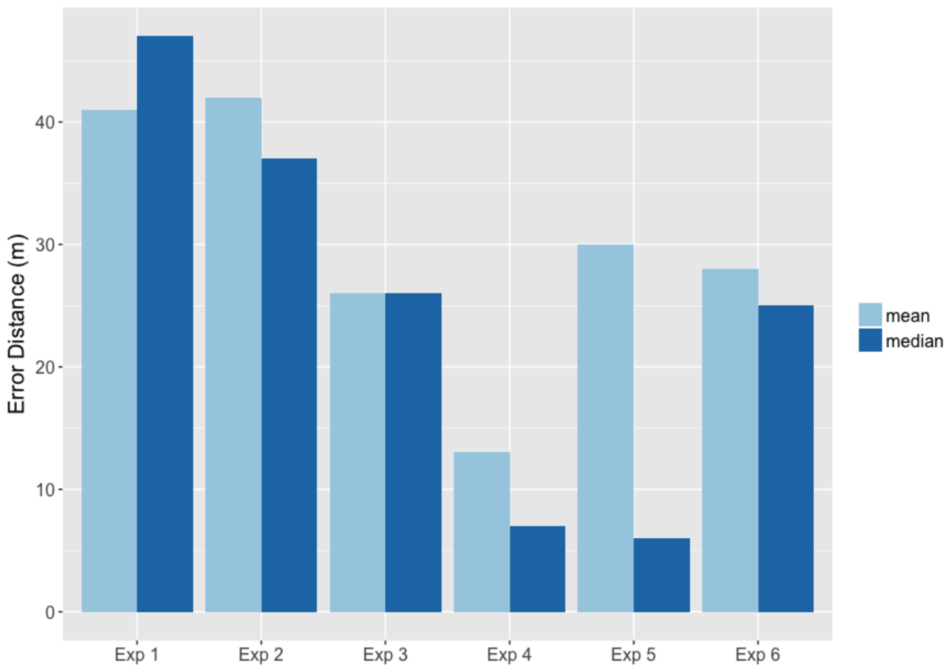


Figure 5.7: Comparison of accuracy of mean and median selection in open-space approach

### 5.4.2 Closed-space Approach Evaluation

We will show the results of 6 experiments which have been analyzed using closed-space approach. Figure 5.8 shows that the error distance for semi-urban area data form experiment 1 to experiment 3 decreased, because this approach is suitable for this environment situation. Otherwise the error distance for line of sight open-space increased.

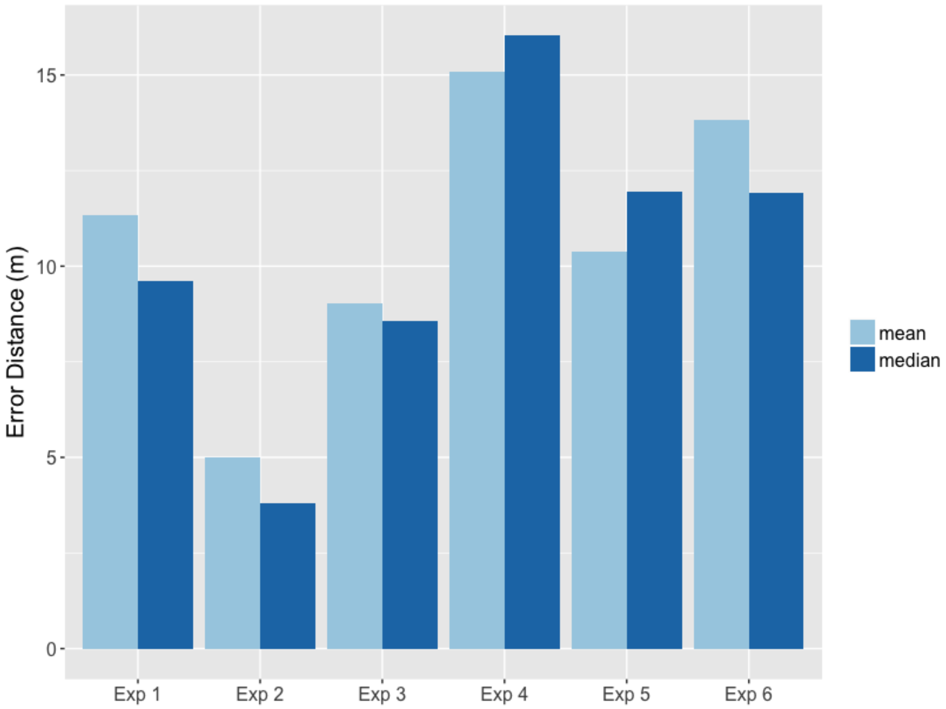


Figure 5.8: Comparison of accuracy of mean and median selection in closed-space approach

# 6 Conclusions and Future Research

## 6.1 Conclusions

In this thesis, we present an IoT-based search system that can estimate a presence area where there may be a victim by using Wi-Fi probe request packets and GPS information. We explained two system processes, i.e., data collection and data analysis. The data collection process collects Wi-Fi RSSI and GPS information. In the data analysis process, based on the data collected, we explain two approaches to analyze the data, open-space approach and close-space approach. We use two ways to select the center area estimation. The experimental results showed that for the semi-urban area, close-space approach shows better results than open-space approach. The selection for both results shows that median selection is slightly better than mean selection. Also, we showed that the prototype system could visualize an estimated area on base-map.

## 6.2 Future Research

In this research, we used two smartphones for experimentation. As we know that different smartphones will send different signal strength, to build a big IoT system we need to increase the number of smartphones for experiment. Additionally, we entirely conducted our experiments on the ground and totally ignored the altitude factors hence, we need to consider the aforementioned factors in the future. We consider that the rescuer velocity factor must be inspected because it will impact the number of probe requests data that are received from the smartphones.

The Camera sensor can be activated to show the actual position of the disaster area. Not only actual situation, but captured location can help to interpret the value of path lost index. We can use image processing techniques to estimate the value of path lost index in a disaster area.

# References

[1] W. H. Organization *et al.*, "Emergency response framework (erf)," 2017.

[2] N. Suzuki, J. Zamora, S. Kashihara, and S. Yamaguchi, "Using SOS message propagation to estimate the location of immobilized persons," in *Proceedings of the Annual International Conference on Mobile Computing and Networking, MOBICOM*, 2012.

[3] H. Nishiyama, M. Ito, and N. Kato, "Relay-by-smartphone: realizing multihop device-to-device communications," *IEEE Communications Magazine*, vol. 52, no. 4, pp. 56–65, 2014.

[4] Z. Lu, G. Cao, and T. L. Porta, "TeamPhone: Networking SmartPhones for Disaster Recovery," *IEEE Transactions on Mobile Computing*, vol. 16, no. 12, pp. 3554–3567, 2017.

[5] A. Yassin, Y. Nasser, M. Awad, A. Al-Dubai, R. Liu, C. Yuen, R. Raulefs, and E. Aboutanios, "Recent Advances in Indoor Localization: A Survey on Theoretical Approaches and Applications," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 1327–1346, 2017.

[6] H. Chen, Y. Zhang, W. Li, X. Tao, and P. Zhang, "ConFi: Convolutional Neural Networks Based Indoor Wi-Fi Localization Using Channel State Information," *IEEE Access*, vol. 5, pp. 18066–18074, 2017.

[7] M. E. Rida, F. Liu, Y. Jadi, A. A. A. Algawhari, and A. Askourih, "Indoor Location Position Based on Bluetooth Signal Strength," in *2015 2nd International Conference on Information Science and Control Engineering*, pp. 769–773, IEEE, apr 2015.

[8] M. L. Sichitiu, V. Ramadurai, and P. Peddabachagari, “Simple Algorithm for Outdoor Localization of Wireless Sensor Networks with Inaccurate Range Measurements,” *International Conference on Wireless Networks*, pp. 300–305, 2003.

[9] W. H. Kuo, Y. S. Chen, K. T. Cheng, and T. W. Lu, “Signal strength based indoor and outdoor localization scheme in Zigbee sensor networks,” *IAENG International Journal of Computer Science*, vol. 43, no. 1, pp. 55–64, 2016.

[10] P. Annadata, W. Eltarjaman, and R. Thurimella, “Person Detection Techniques for an IoT Based Emergency Evacuation Assistance System,” *Proceedings of the 13th International Conference on Mobile and Ubiquitous Systems: Computing Networking and Services - MOBIQUITOUS 2016*, pp. 77–82, 2016.

[11] W. Pattanusorn, I. Nilkhamhang, S. Kittipiyakul, K. Ekkachai, and A. Takahashi, “Passenger estimation system using wi-fi probe request,” in *2016 7th International Conference of Information and Communication Technology for Embedded Systems (IC-ICTES)*, pp. 67–72, March 2016.

[12] G. Lui, T. Gallagher, B. Li, A. G. Dempster, and C. Rizos, “Differences in rssi readings made by different wi-fi chipsets: A limitation of wlan localization,” in *2011 International Conference on Localization and GNSS (ICL-GNSS)*, pp. 53–57, June 2011.

[13] Y. Huang, L. T. Hsu, Y. Gu, H. Wang, and S. Kamijo, “Database calibration for outdoor Wi-Fi positioning system,” *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E99A, no. 9, pp. 1683–1690, 2016.

[14] Z. Kamal, A. Mohammed, E. Sayed, and A. Ahmed, “Internet of Things Applications , Challenges and Related Future Technologies Internet of Things Applications , Challenges and Related Future Technologies,” *World Scientific News*, vol. 67, no. February, pp. 126–148, 2017.

[15] S. Barai, D. Biswas, and B. Sau, “Estimate distance measurement using NodeMCU ESP8266 based on RSSI technique,” *2017 IEEE Conference on*

*Antenna Measurements and Applications, CAMA 2017*, vol. 2018-January, no. December 2017, pp. 170–173, 2018.

[16] T. Rappaport, *Wireless Communications: Principles and Practice.* Upper Saddle River, NJ, USA: Prentice Hall PTR, 2nd ed., 2001.

[17] F. Shang, W. Su, Q. Wang, H. Gao, and Q. Fu, “A location estimation algorithm based on rssi vector similarity degree,” *International Journal of Distributed Sensor Networks*, vol. 10, no. 8, p. 371350, 2014.

[18] S. Gao and S. Prasad, “Employing Spatial Analysis in Indoor Positioning and Tracking Using Wi-fi Access Points,” in *Proceedings of the Eighth ACM SIGSPATIAL International Workshop on Indoor Spatial Awareness*, ISA ’16, (New York, NY, USA), pp. 27–34, ACM, 2016.

[19] V. Cantón Paterna, A. Calveras Augé, J. Paradells Aspas, and M. A. Pérez Bullones, “A bluetooth low energy indoor positioning system with channel diversity, weighted trilateration and kalman filtering,” *Sensors*, vol. 17, no. 12, 2017.

# Appendix

**Source Code for capturing position data, put inside /etc/rc.local file :**

```
gpsd /dev/ttyAMA0 -F /var/run/gpsd.sock
sleep 60
sudo gpspipe -r -d -l -o /data/data.nmea
```

**Source Code for capturing probe requests data :**

```
Start tcpdump at $start
Start tcpdump at $start >>sudo time.txt
tcpdump -I -i wlan0 type mgt subtype probe-req -w probe.pcap
stop= date + T
Finish tcpdump at stop
Finish tcpdump at stop >>sudo time.txt
```

**Source Code for cutting position data timestamp base on probe request data reference time :**

```
gpsbabel -i nmea -f /data/data.nmea -o garmin-txt,utc=+9 -F
    /data/dataUTC.csv
gpsbabel -i garmin-txt -f /data/dataUTC.csv -o
    csv,style=/data/MY.STYLE -F flight-log-all.csv
sudo python cut.py flight-log-all.csv start stop
sudo tac data.csv > sudo flight-log.csv
```

**Source Code for converting pcap file :**

```
tshark -t a -r probe.pcap > output.txt
```

**Source Code for merging and cleaning data :**

```
f = open("output.txt","r")
f2 = open("data-position.csv", "r")
for line in output:
    if ('vendor1' in line or 'vendor2' in line or 'vendor3' in line):
        calculate x,y,z;
        save file to csv;
drop duplicate data;
save clean data;}}
```

---

### Source Code for analysing data in open-space approach :

---

```
read clean data;
group data byMax MAC adress;
define tentative centre;
for x in range(0,len(combination)):
    if(len(rlist)>2):
        combination 3 data;
        calculate distance from combination to tentative center;
            if(distance<=100):
                        if(distance>50):
                            change data to tentative center;
                        else :
                           take original data;
            else
                delete data;
apply randomization, mean, and median process;
```

---

### Source Code for analysing data in closed-space approach :

---

```
library(geosphere)

locations <- read.table("data.csv", sep=",", header=TRUE)

latitude = locations$Latitude
longitude=locations$Longitude
distance=locations$distance
```

```
fit <- nls(
  distance ~ distm(
    data.frame(Longitude, Latitude),
    c(fitLongitude, fitLatitude)
  ),
  data = locations,
  start = list(
    fitLongitude=mean(locations$Longitude),
    fitLatitude=mean(locations$Latitude)
  ),
  control = list(maxiter = 1000, tol = 1e-02)
)

latitude <- summary(fit)$coefficients[2]
longitude <- summary(fit)$coefficients[1]
```

**Source Code for creating visualization on the maps :**

```
dfFinalJS = pd.read_csv('final.csv')
print(len(dfFinalJS))

f = open('EstimatedArea.html','w')

message0 = """<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="initial-scale=1.0, user-scalable=no">
    <meta charset="utf-8">
    <title>Circles</title>
    <style>
      /* Always set the map height explicitly to define the size of the
          div
       * element that contains the map. */
      #map {
        height: 100%;
```

```
      }
      /* Optional: Makes the sample page fill the window. */
      html, body {
        height: 100%;
        margin: 0;
        padding: 0;
      }
    </style>
  </head>
  <body>
    <div id="map"></div>
    <script>
      var devicePosition = {\n"""
temp1=''
for x in range(0,len(dfFinalJS)):
    message1='device'+str(x+1)+': {'
    message2='center: {lat:'+str(dfFinalJS['Latitude'][x])+',
        lng:'+str(dfFinalJS['Longitude'][x])+'},'
    message3='},'
    temp1
        +='\t\t\t\t'+message1+'\n\t\t\t\t'+message2+'\n\t\t\t\t'+message3+'\n'
message4='\t\t\t};\n'

message5='\t\t\tfunction initMap() {\n\t\t\t\t var map = new
    google.maps.Map(document.getElementById(\'map\'), {\n'
message6='\t\t\t\t zoom: 19,\n\t\t\t\t center:
    {lat:'+str(dfFinalJS['Latitude'].mean())+',
    lng:'+str(dfFinalJS['Longitude'].mean())+'},\n\t\t\t\t mapTypeId:
    \'hybrid\'\n\t\t\t});\n'

message7 = """\t\t\tfor (var devpos in devicePosition) {
          // Add the circle for this city to the map.
          var cityCircle = new google.maps.Circle({
            strokeColor: '#013ADF',
            strokeOpacity: 0.8,
            strokeWeight: 2,
```

```
        fillColor: '#013ADF',
        fillOpacity: 0.35,
        map: map,
        center: devicePosition[devpos].center,
        radius: 10
      });

      var cityCircle1 = new google.maps.Circle({
        strokeColor: '#40FF00',
        strokeOpacity: 0.8,
        strokeWeight: 2,
        fillColor: '#40FF00',
        fillOpacity: 0.35,
        map: map,
        center: devicePosition[devpos].center,
        radius: 20
      });
        var cityCircle2 = new google.maps.Circle({
        strokeColor: '#FA5858',
        strokeOpacity: 0.8,
        strokeWeight: 2,
        fillColor: '#FA5858',
        fillOpacity: 0.35,
        map: map,
        center: devicePosition[devpos].center,
        radius: 30
      });

    }
  }
</script>
<script async defer
src="https://maps.googleapis.com/maps/api/js?key=[google_map_key]">
</script>
</body>
</html>
```

```
"""

message = message0+temp1+message4+message5+message6+message7;
f.write(message)
f.close()
```