BEACON-BASED INDOOR FIRE EVACUATION SYSTEM USING AUGMENTED REALITY AND MACHINE LEARNING

Bacon Beacon

Index

- Design
- Indoor localization
- Pathfinding algorithm
- Navigation system
- Q&A

Design

- Redesigned

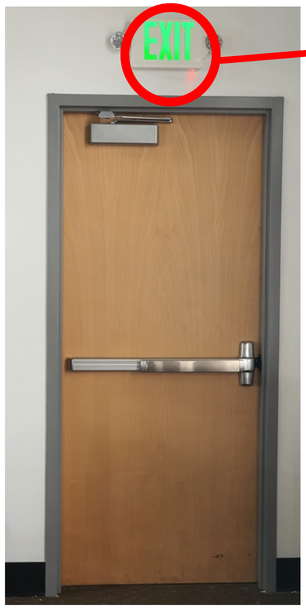
1. Indoor localization
2. Algorithm
3. AR
4. System

- Renewed Overview

Background

Real Situation

- It is hard to escape buildings when **full of smoke**
- People feels hard to know the **location of fire**



Not Enough!

Fig.1 EXIT sign

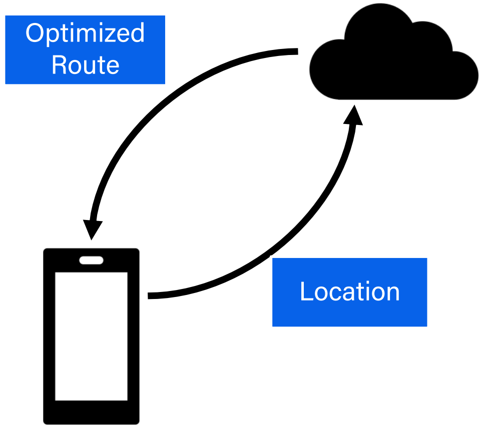
Fig.2 Building full of smoke [1]

Background

Goal of BEST

- **High accuracy** of indoor localization using iBeacon
- Server sends **optimized evacuation route** to exit in **Real-Time**
- **Intuitive Escape Route**: using Augmented Reality (AR) to easily follow shortest path.

Fig. 5 Apple iBeacon. Adapted from “Apple Developer”[2]



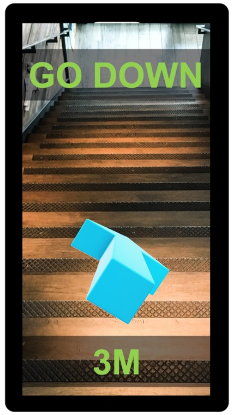


Fig. 6 AR indicator showing route

Redesigned – Kalman Filter

- Problem
  - Kalman Filter is limited to **linear** system
- Previous Solution
  - ~~Extended Kalman Filter(EKF)~~
    - Reason: RSSI is scalar, does not fit to EKF
- New Solution
  - Reset Kalman Filter when fluctuate too much

Design

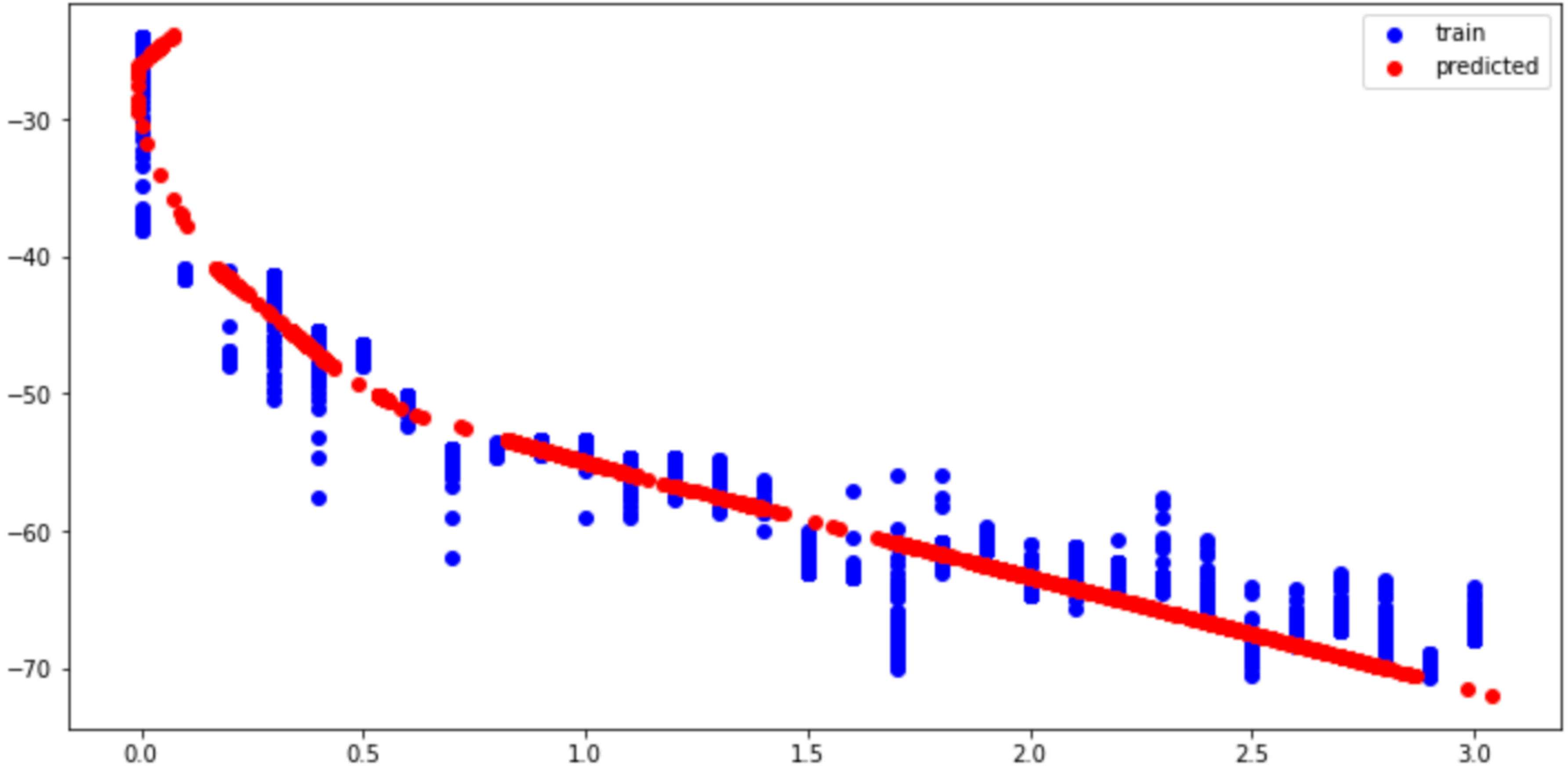
Redesigned – Support Vector Regression

- Regression Model tried to use for localization
  - Not fit at real world. No difference between 1~3m
- Solution
  - Deep Learning techniques were used

Environment Setting

- KSW building
- Device: iPhone 12, Galaxy A30
- Beacon and device are in horizontal alignment

Mid term Experiment Result



Environment Setting

- KSW building
- Device: iPhone 12, Galaxy A30
- Beacon are installed on the ceiling
- Devices are held by person

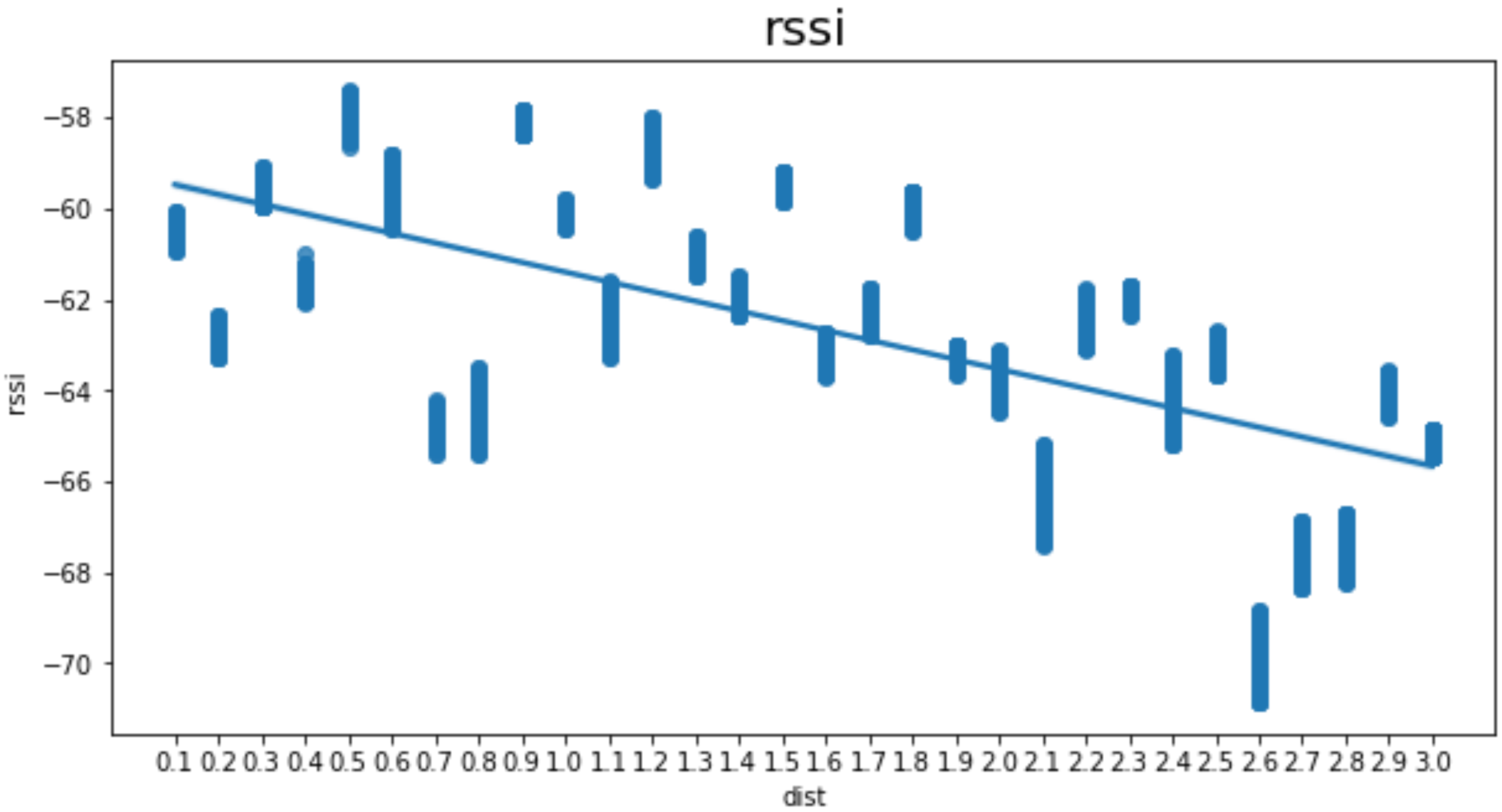
Background

Indoor Localization

- Experiment failed
  - Reason: if the signal is not on same height, the RSSI value changes, not as same as theory
  - Test: Collect RSSI and distance again, but on different height

Distance(m)	Average RSSI(dB)
0.5	-51
1	-52
1.5	-50
2	-59
2.5	-56
3	-60

Experiment Result



Experiment Result

Redesigned – Optimized Route Algorithm

- Previous: Dijkstra or A* Algorithm were considered
  - Needed to find multiple path
- Solution: Reinforced Learning Q-Algorithm
  - It is designed for evacuation system

Design

Redesigned – Augmented Reality(AR)

- Previous: only 3-dimensional arrow used
  - Problem: User cannot know the exact place
  - Solution: Both 2D map and AR used

Redesigned – System

- Database no longer used
  - Not contain useful information
  - System memory can handle temporary information
- ESP32 used for sensing
  - Raspberry Pi is too expensive for place everywhere
- AP was not available
  - Apple blocked the access of AP.

Overview

Overview – Fire detection

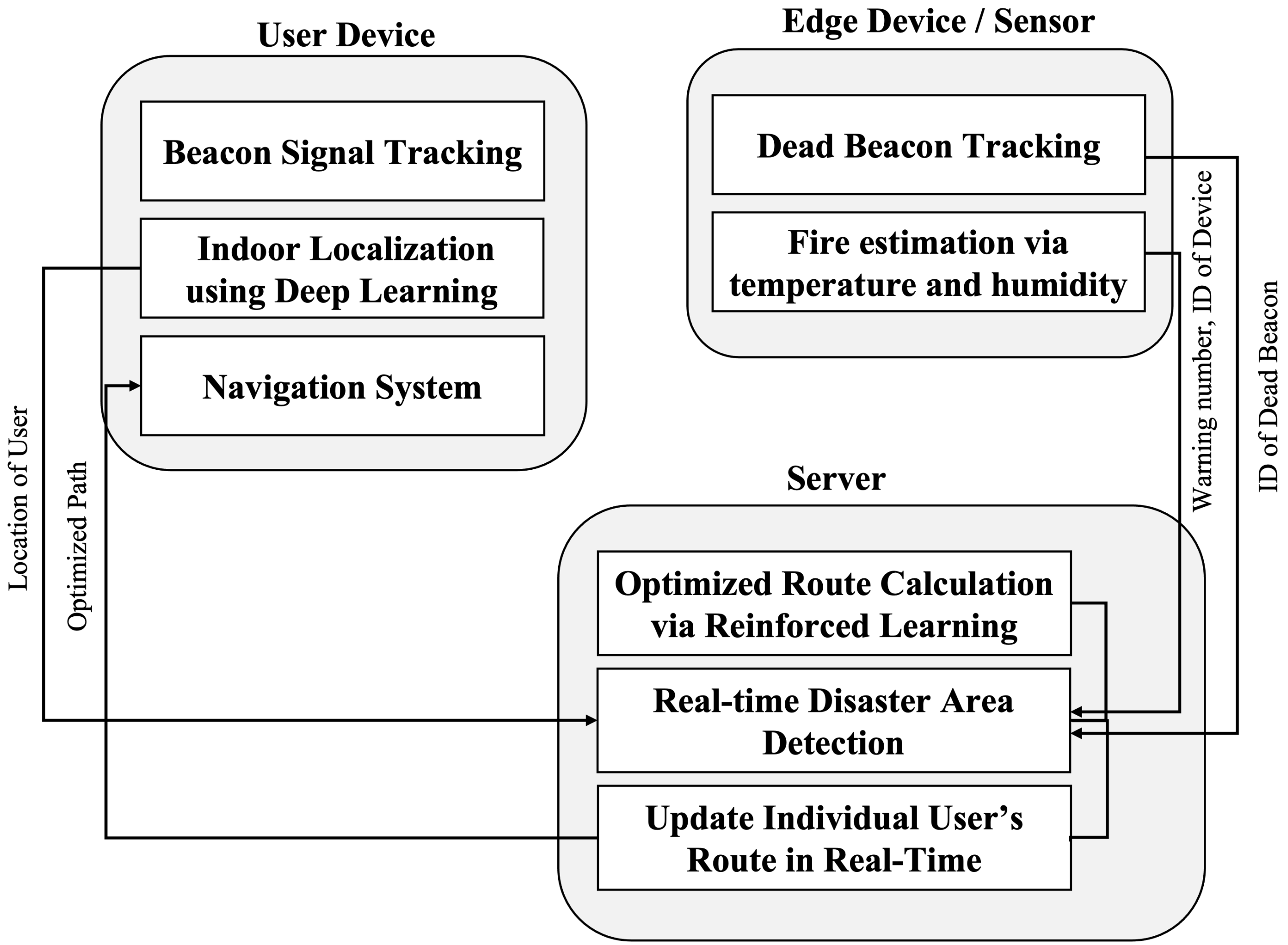
- Temperature and humidity detection
  - Using ESP32, Raspberry Pi as edge device
  - Used DHT11, DHT22 to sense temperature and humidity

- Sends to server the value of temperature, humidity

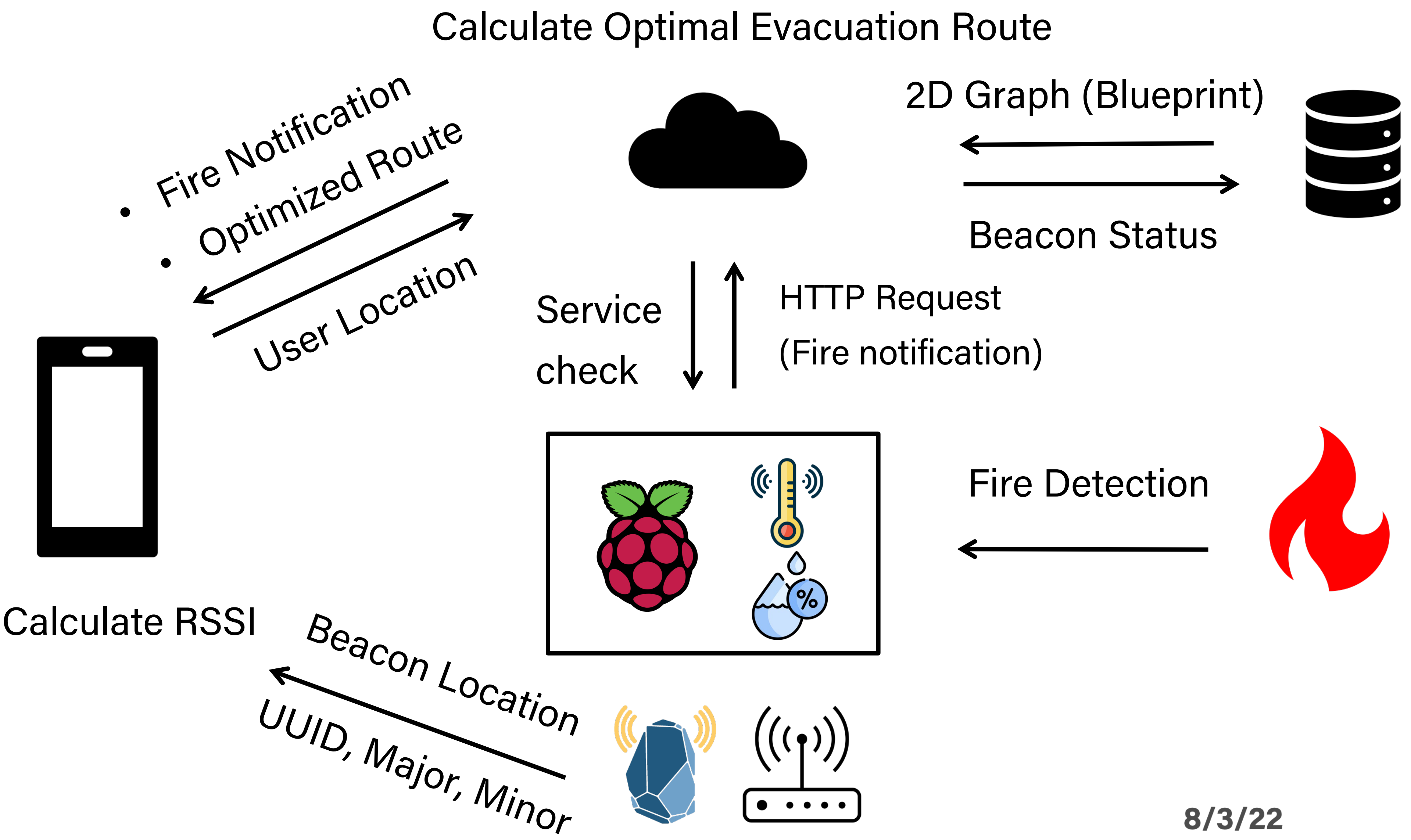
Overview – Fire detection

- Server receives the value, and notify the fire
  - Fire situation: temperature > 60’C, humidity < 20%
  - Change the fire_situation True
- Server sends notification to mobile devices
  - Using Firebase API, sends notification

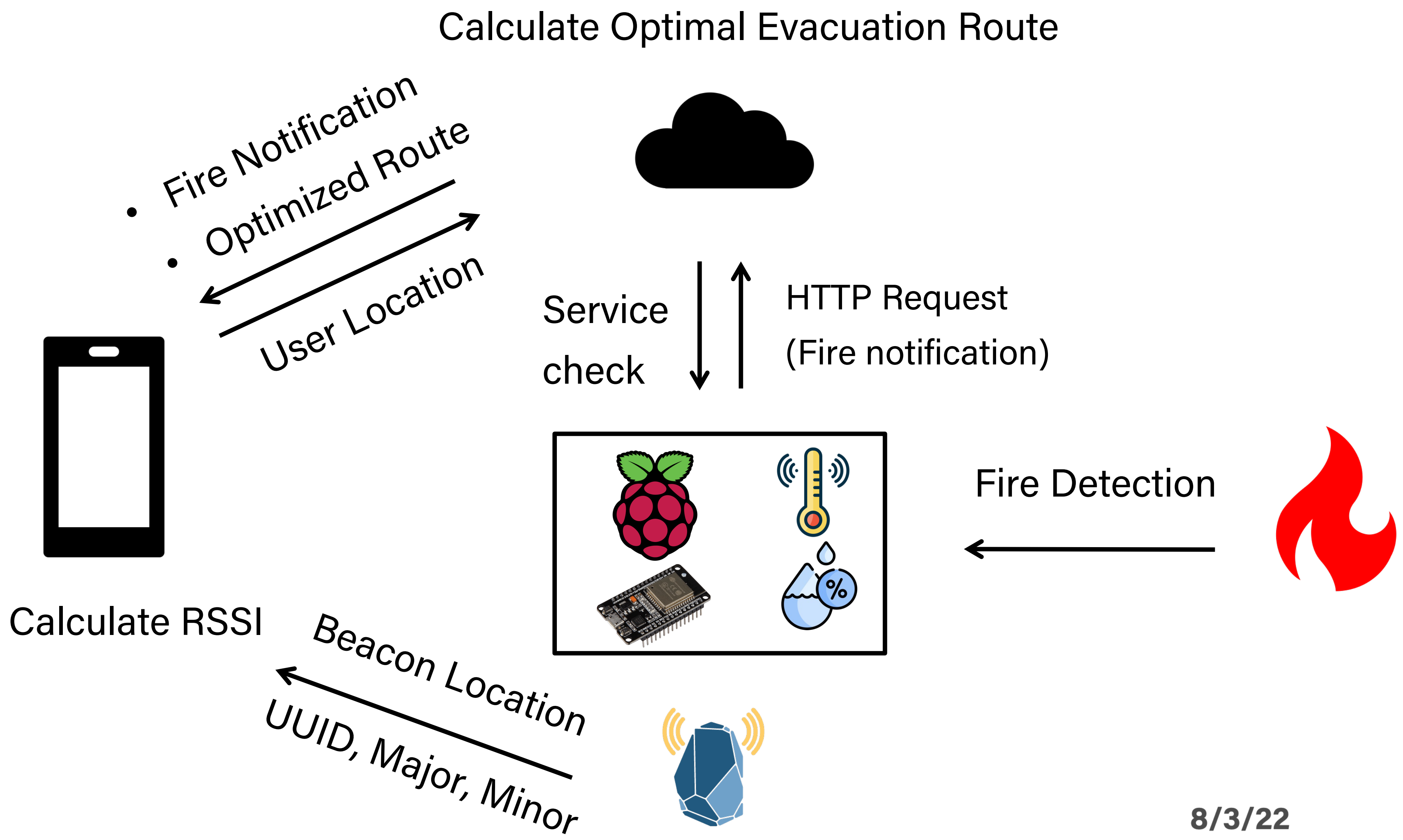
Overview


User Device
Beacon Signal Tracking
Indoor Localization using Deep Learning
Navigation System
Edge Device / Sensor
Dead Beacon Tracking
Fire estimation via temperature and humidity
Warning number, ID of Device
ID of Dead Beacon
Location of User
Optimized Path
Server
Optimized Route Calculation via Reinforced Learning
Real-time Disaster Area Detection
Update Individual User's Route in Real-Time

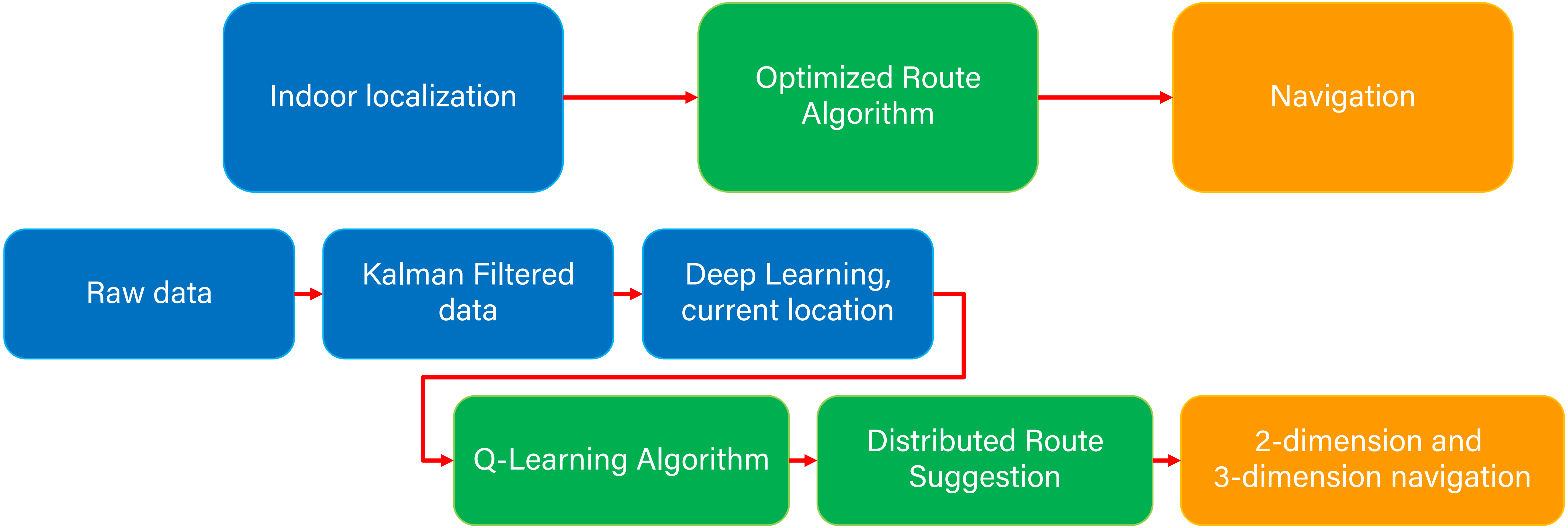
Overview


Calculate Optimal Evacuation Route
2D Graph (Blueprint)
Beacon Status
• Fire Notification
• Optimized Route
User Location
Service check
HTTP Request (Fire notification)
Fire Detection
Calculate RSSI
Beacon Location
UUID, Major, Minor

Overview


Calculate Optimal Evacuation Route
• Fire Notification
• Optimized Route
User Location
Service check
HTTP Request
(Fire notification)
Fire Detection
Calculate RSSI
Beacon Location
UUID, Major, Minor

Overview



Indoor localization
Optimized Route Algorithm
Navigation
Raw data
Kalman Filtered data
Deep Learning, current location
Q-Learning Algorithm
Distributed Route Suggestion
2-dimension and 3-dimension navigation

Indoor Localization

- Classification
- Overview
- Moving Object

Multi class Classification

- Collect the data from each class
- Train to classify the class
- Pros in indoor localization
  - Consideration of specific indoor environment
  - Consideration of building construction
  - Ex) Basement vs Room

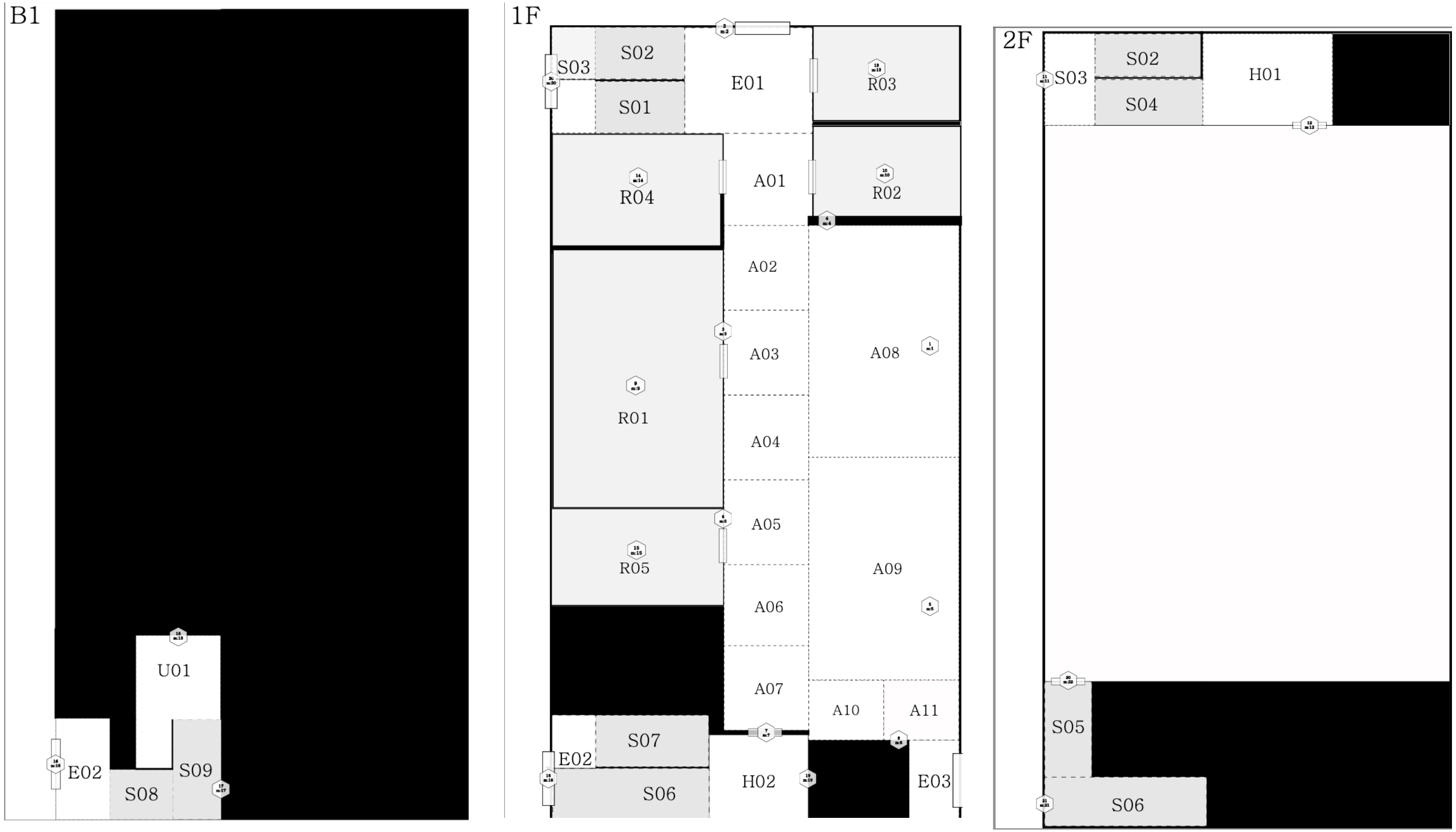
Experiment Setting

- KSW building
- Device: iPhone 12, Galaxy A30
- 22 Beacons are installed on the ceiling or at the top of the wall
- Devices are held by person

Experiment Setting

- KSW building is divided into 31 cells
- Default cell size: 2.5M * 2.5M
- Others: various cell size
  - Rooms, Stairs, Area do not influence to exit path, Small to divided into default cell size

Experiment Setting

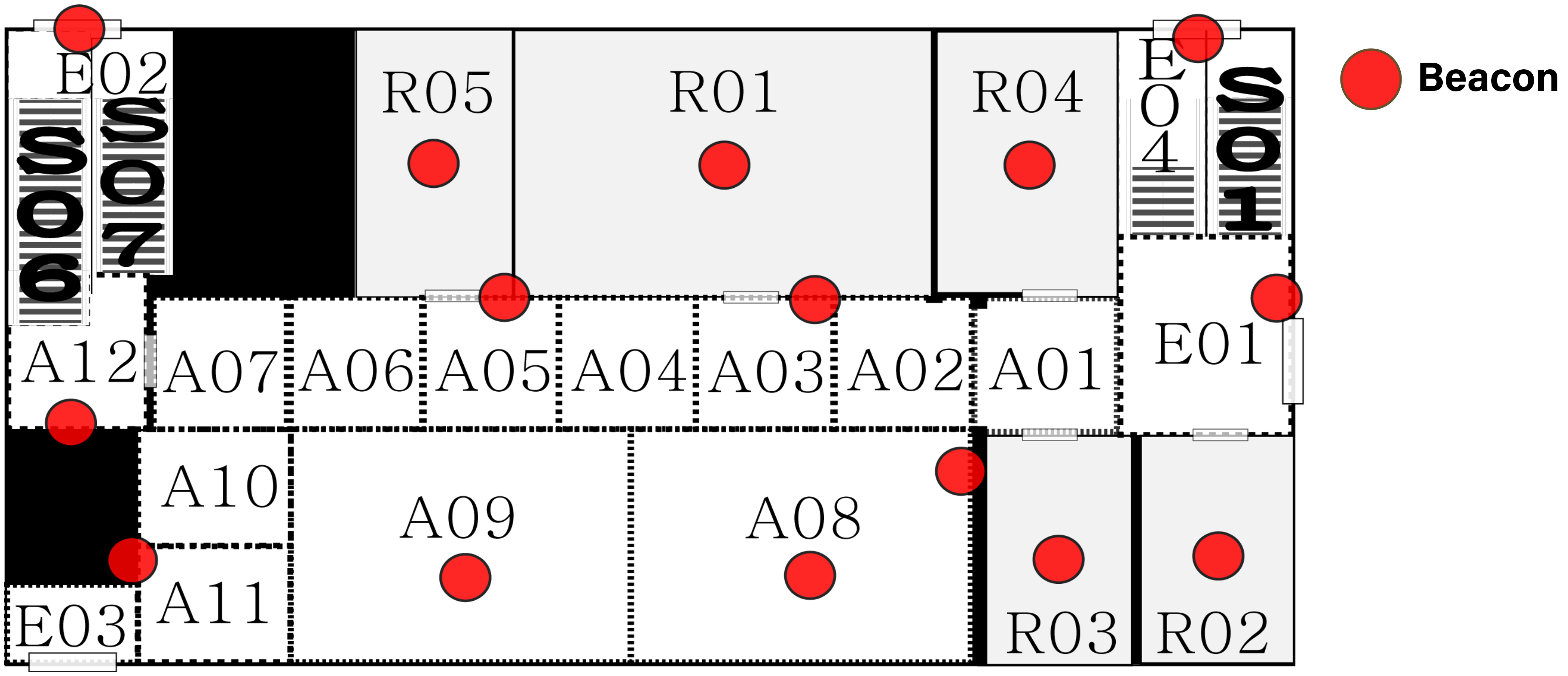


- A: Normal
- S: Stair
- R: Room
- E: Exit

Experiment Setting

- 22 Beacons are installed in the KSW building
- Method to determine the location of beacons
  - Checkpoints (Exit, Middle of the stair)
  - Inside the rooms
  - Place beacons in every 6m

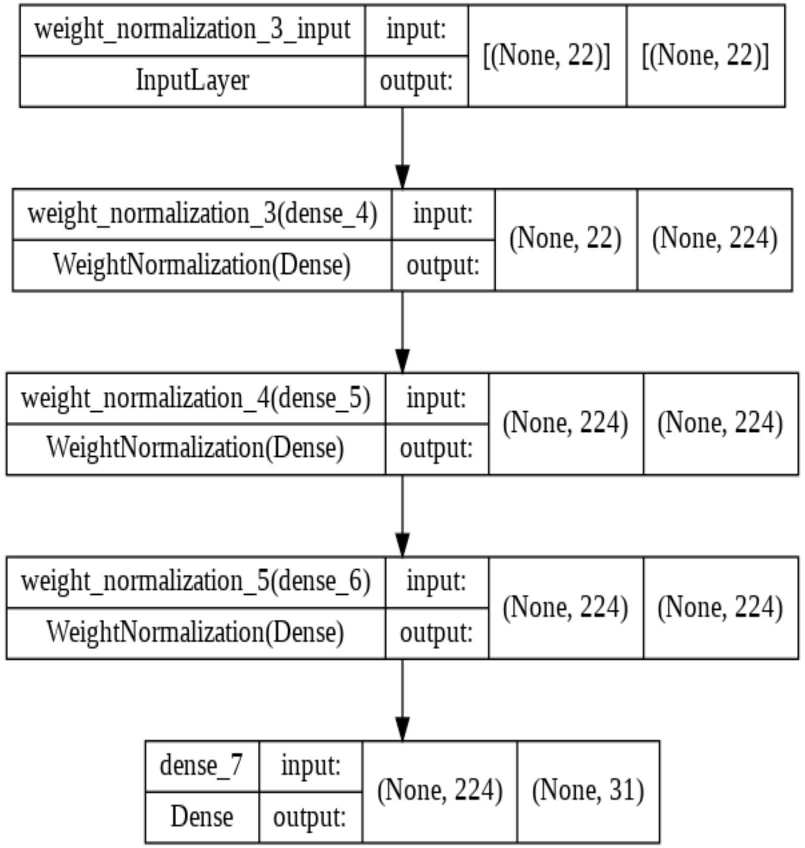
Experiment Setting



Experience – collecting training data

- 16 mins / cell
- Device is held by person
- Collected in various directions and locations.
- 950 ~ 1000 filtered RSSI vectors / cell
- Total data: 30806

DNN classification Model



- Input: filtered RSSI vector
- Normalization method
  - He Initialization
  - Weight normalization
- ReLU activation function
- Softmax activation function
- Output: probabilities of 31 classes

Experience – collecting test data

- 3 mins / cell
- Device is held by person
- Collected via various heading direction and location inside the cell.
- 180 ~ 200 filtered RSSI vector / cell
- Total data: 5735

Experiment Result – preprocessing X

Training		Validation		Test	
Accuracy	0.9152	Accuracy	0.8493	Accuracy	0.1874
Precision	0.9270	Precision	0.8654	Precision	0.1263
Recall	0.9031	Recall	0.8374	Recall	0.1263
F1-score	0.9137	F1-score	0.8478	F1-score	0.1093

Why?

Hypothesis 1

- Beacon's transmitting frequency and the mobile's receiving frequency are different
  - Beacon's advertising interval: $100ms$
  - iOS/AOS receiving frequency: $1000ms$ - $\partial(0 \leq \partial \leq 100ms)$

Hypothesis 1

- Packet loss as -200 dbm
- Apple document and Alt beacon document instruct the loss of beacon packets due to the mismatch of transmitting and receiving

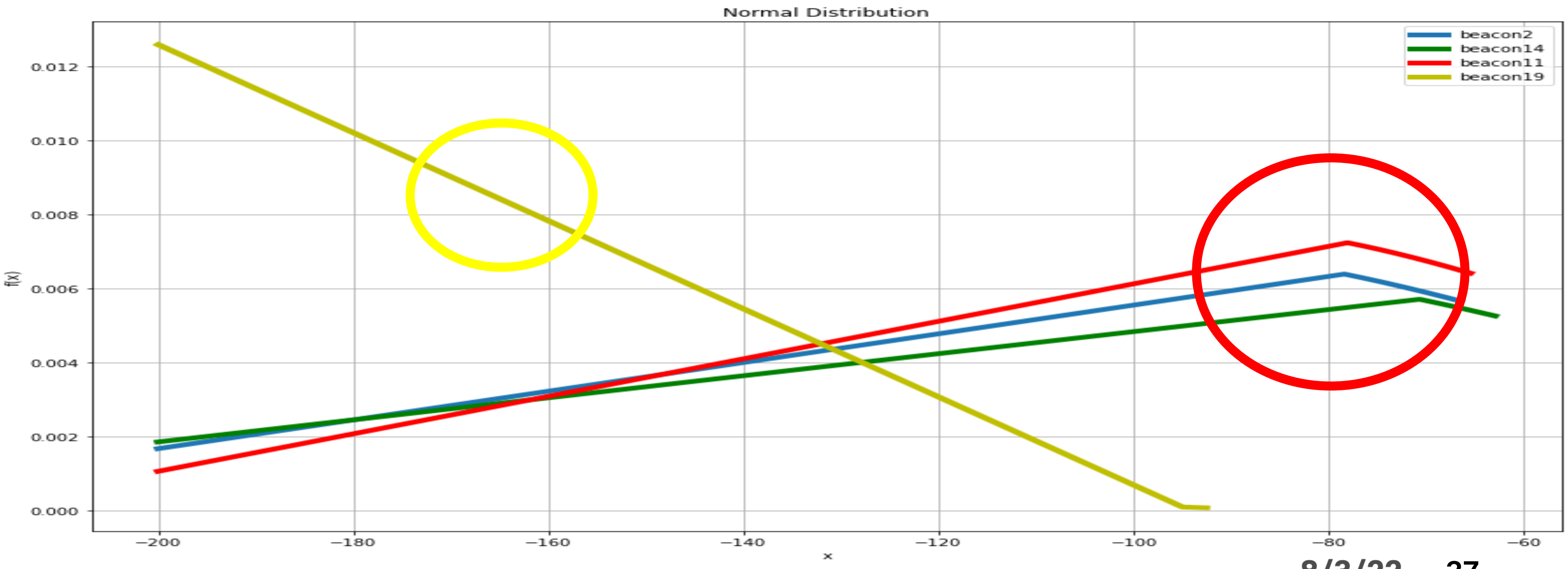
Beacon1	Beacon2	Beacon3	...	beacon22
-200	-75.33	-200	...	-200
-78.25	-74.24	-69.60	...	-200
-76.17	-73.58	-69.18	...	-200
-75.63	-73.14	-68.95	...	-200
-74.37	-200	-68.76	...	-200
-200	-73.56723	-200	...	-200

Hypothesis 2

- If we use whole 22 beacon RSSI as an input, the model will be confused.
- As the distance between beacon and device increases, the fluctuation pattern of RSSI is not comparable.
- Need to find the optimal top N of beacons.

Solution – Hypothesis 1

- Use Median value of N-th beacon RSSI
- Why Median?



Solution – Hypothesis 1

- Use Median value of N-th beacon RSSI
- Why Median?

Beacon1	Beacon2	...
-200	-75.33	...
-78.25	-74.24	...
-76.17	-73.58	...
-75.63	-73.14	...
-74.37	-200	...
-200	-73.56	...

Beacon1	Beacon2	...
-75.63	-75.33	...
-78.25	-74.24	...
-76.17	-73.58	...
-75.63	-73.14	...
-74.37	-73.58	...
-75.63	-73.56	...

Solution – Hypothesis 2

- Find out the Optimal top N beacons to use as valid value
- If not, Make -200 dbm as an invalid value

N = 4

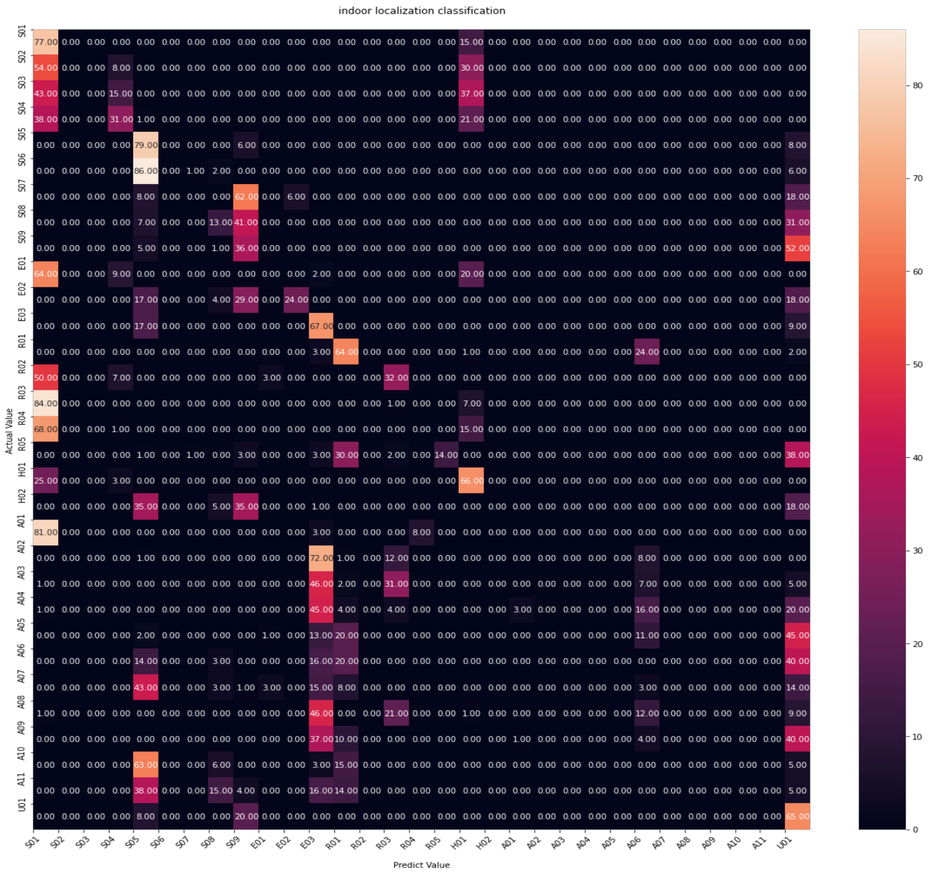
Beacon1	Beacon2	Beacon3	Beacon4	Beacon5	Beacon6	...	beacon19	beacon20	beacon22
-75.63	-75.33	-68.95	-200	-200	-200	...	-200	-200	-56.95
-78.25	-74.24	-69.60	-200	-200	-200	...	-200	-200	-54.14
-76.17	-73.58	-69.18	-200	-200	-200	...	-200	-200	-56.95
-75.63	-73.14	-68.95	-200	-200	-200	...	-200	-200	-58.27
-74.37	-73.58	-68.76	-200	-200	-200	...	-200	-200	-59.56
-75.63	-73.56	-68.95	-200	-200	-200	...	-200	-200	-57.11

Result – Optimal N for valid beacon

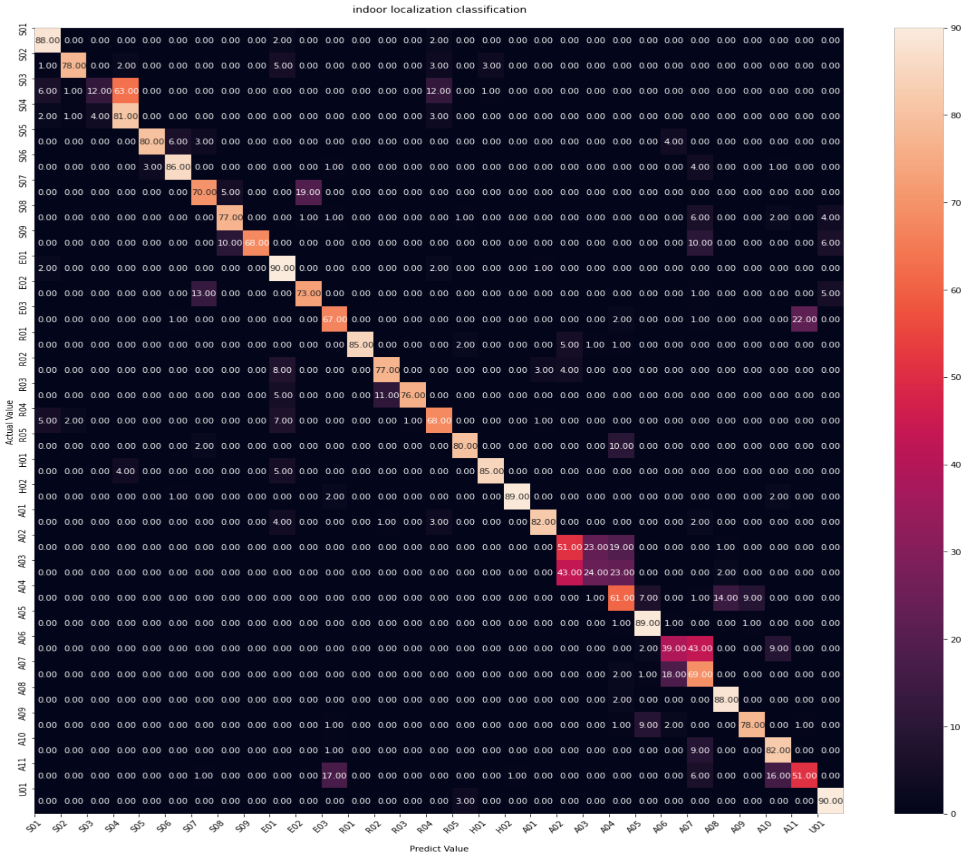
N	Training		Validation		Test	
22	Accuracy	0.9152	Accuracy	0.8493	Accuracy	0.1874
	Precision	0.9270	Precision	0.8654	Precision	0.1263
	Loss	0.9031	Loss	0.8374	Recall	0.1263
7	Accuracy	0.9551	Accuracy	0.9515	Accuracy	0.4699
	Precision	0.9574	Precision	0.9535	Precision	0.4701
	Loss	0.1199	Loss	0.1564	Recall	0.4699
5	Accuracy	0.8995	Accuracy	0.8998	Accuracy	0.7200
	Precision	0.9051	Precision	0.9074	Precision	0.7350
	Loss	0.2451	Loss	0.2544	Recall	0.7200
4	Accuracy	0.9140	Accuracy	0.9148	Accuracy	0.7797
	Precision	0.9237	Precision	0.9236	Precision	0.7945
	Loss	0.2222	Loss	0.2348	Recall	0.7797

Indoor Localization

Result – With median vs Without median

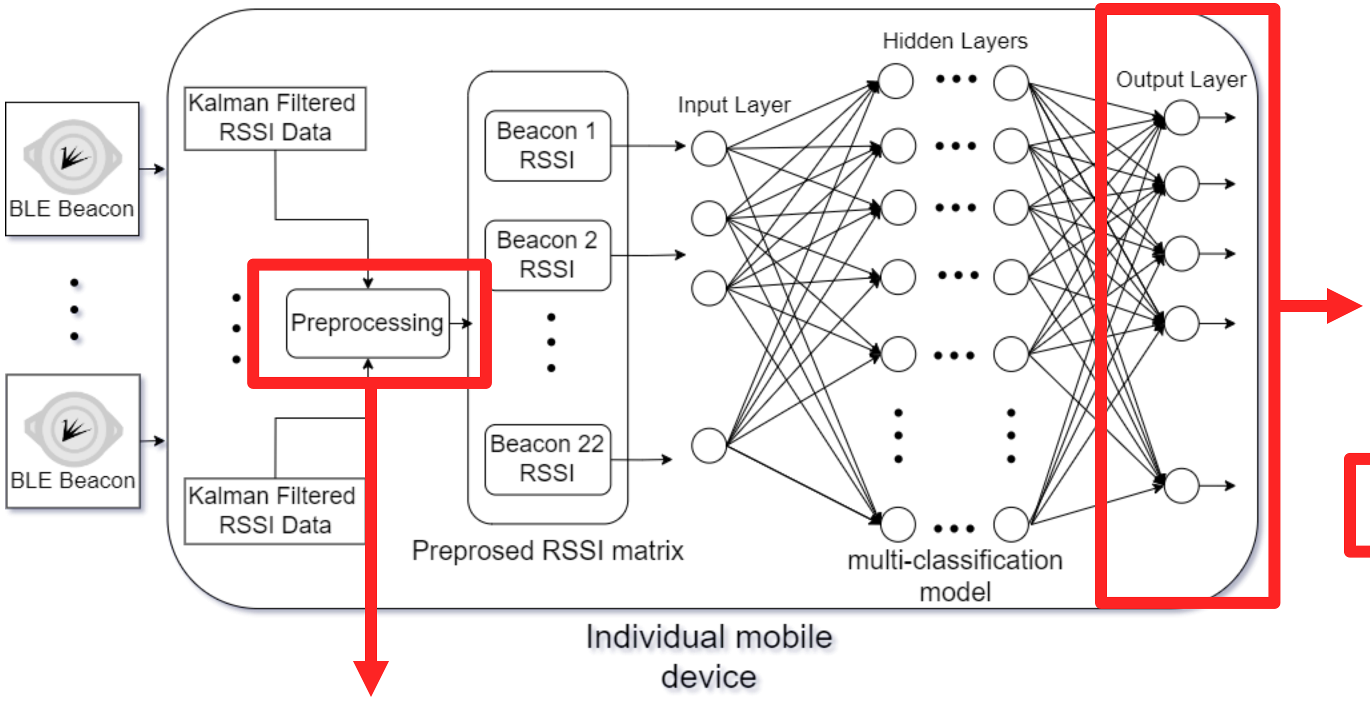


Without median value



With median value

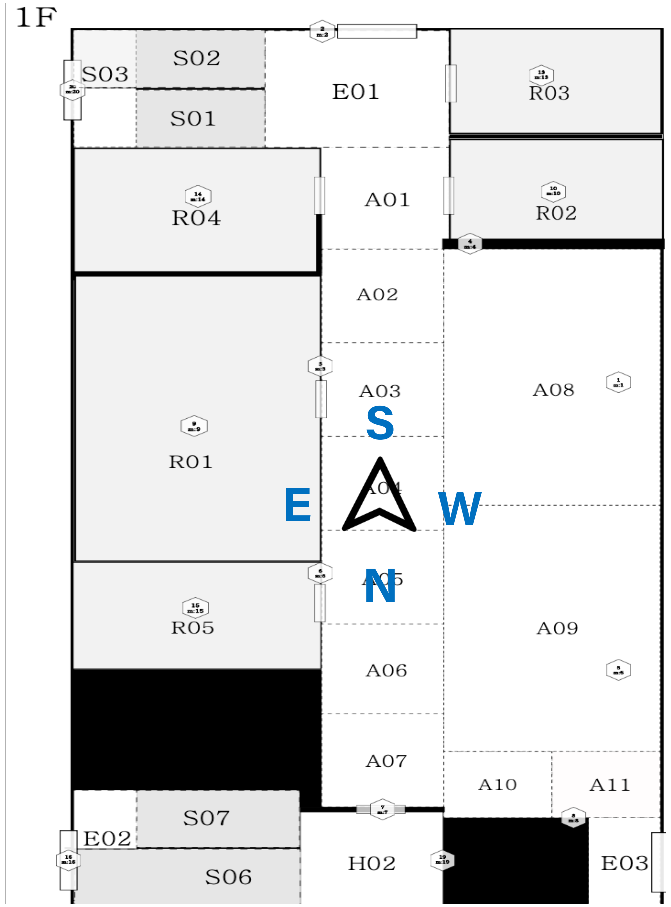
Overview of indoor localization

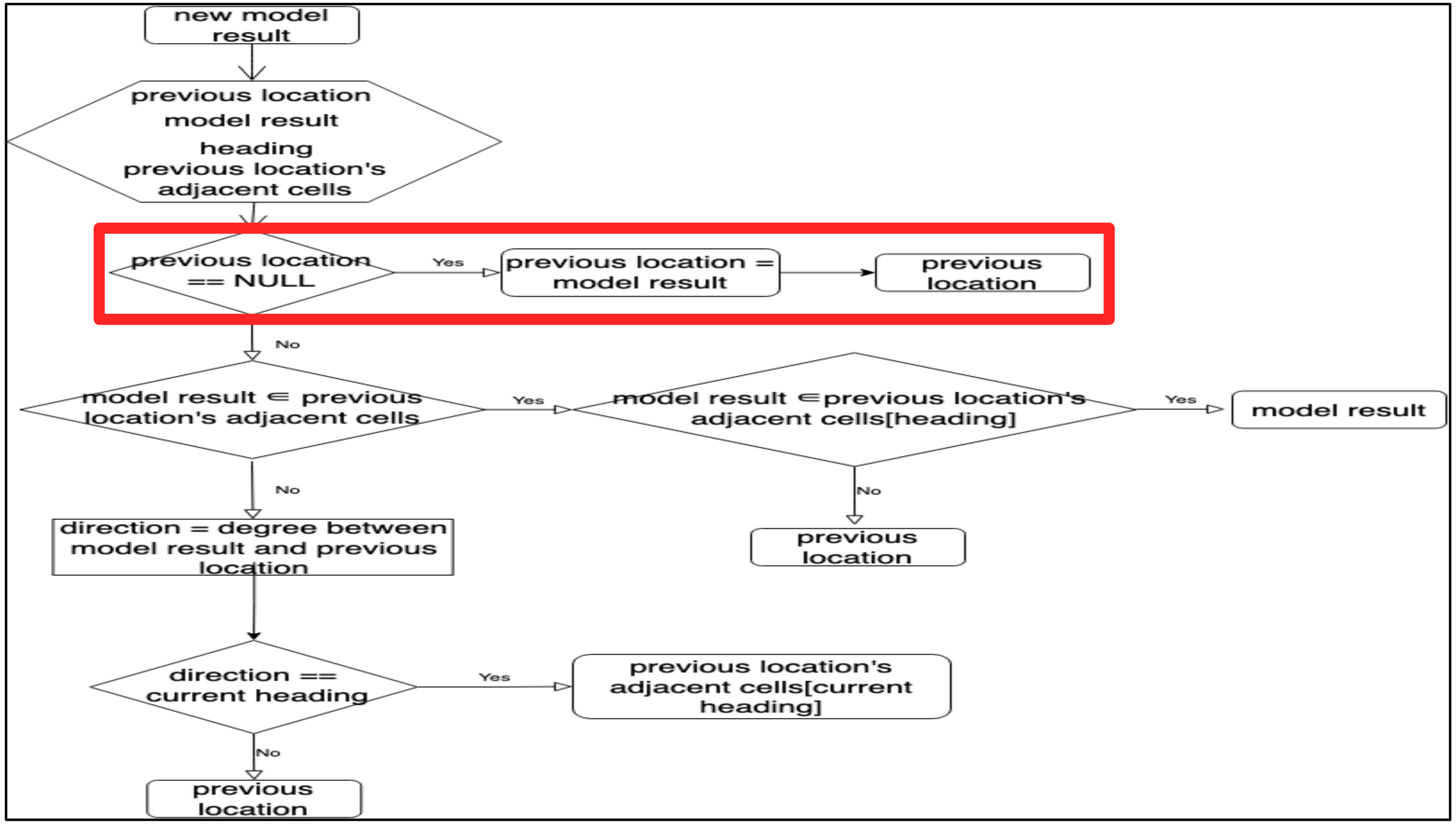


class	prob
A01	0.00
A02	0.00
A03	0.02
A04	0.04
…	…
E03	0.96

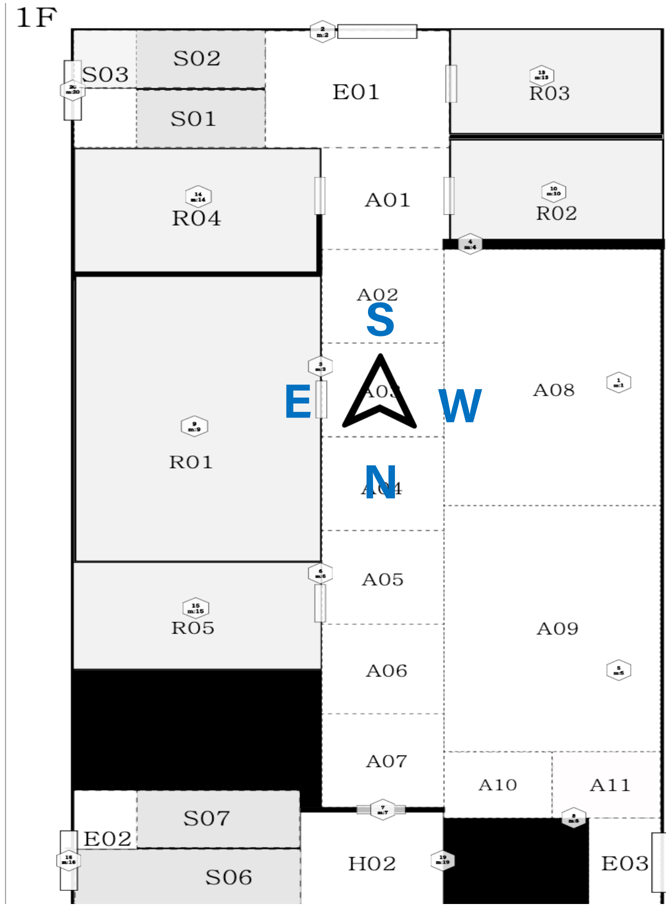
N	Beacon1	Beacon2	Beacon3	…	beacon22
N_{old}	-75.63	-75.33	-68.95	…	-200
$N + 1_{old}$	-78.25	-200	-69.60	…	-200
N_{new}	-75.63	-75.33	-68.95	…	-200

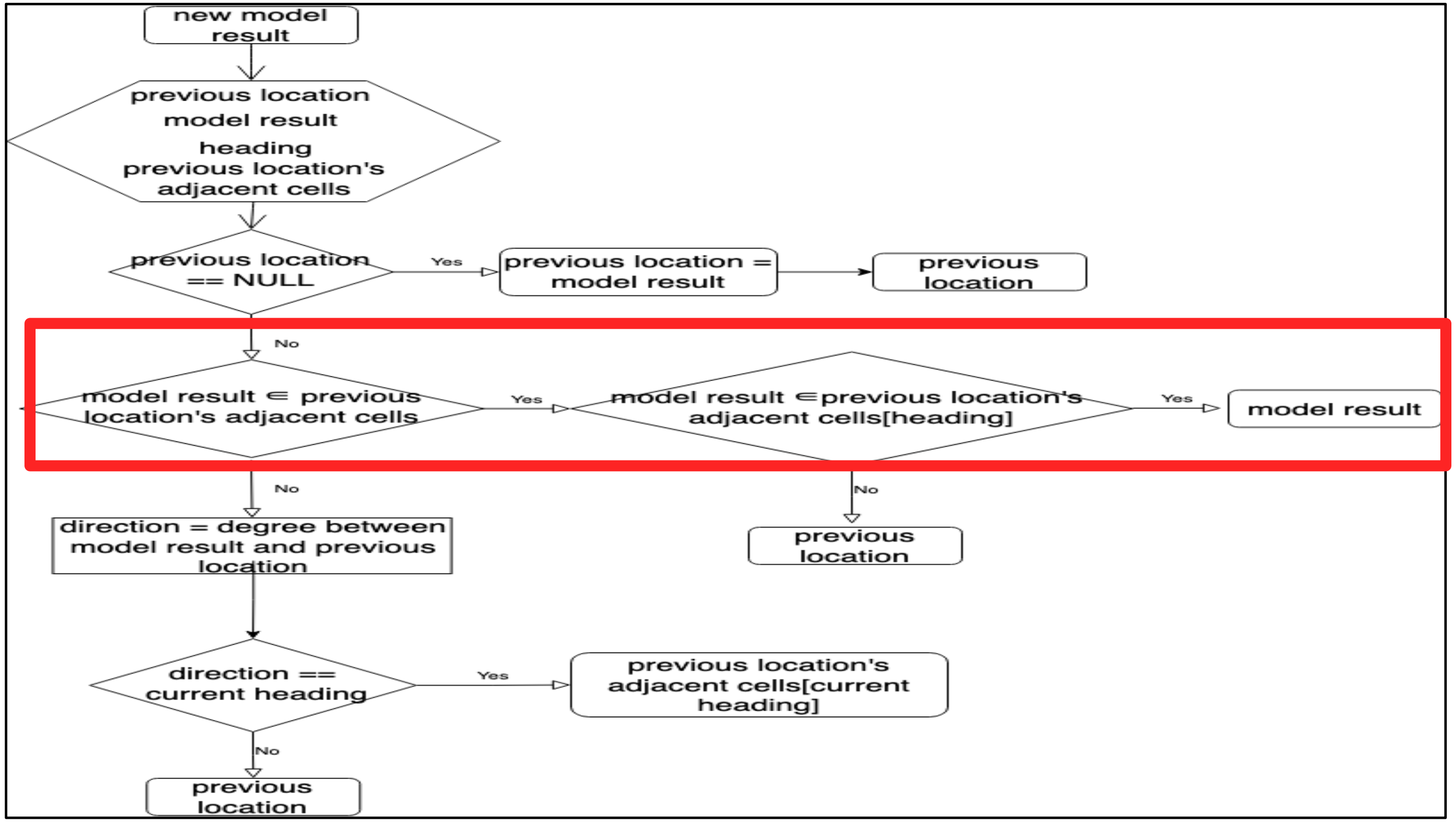
Indoor Localization for moving person



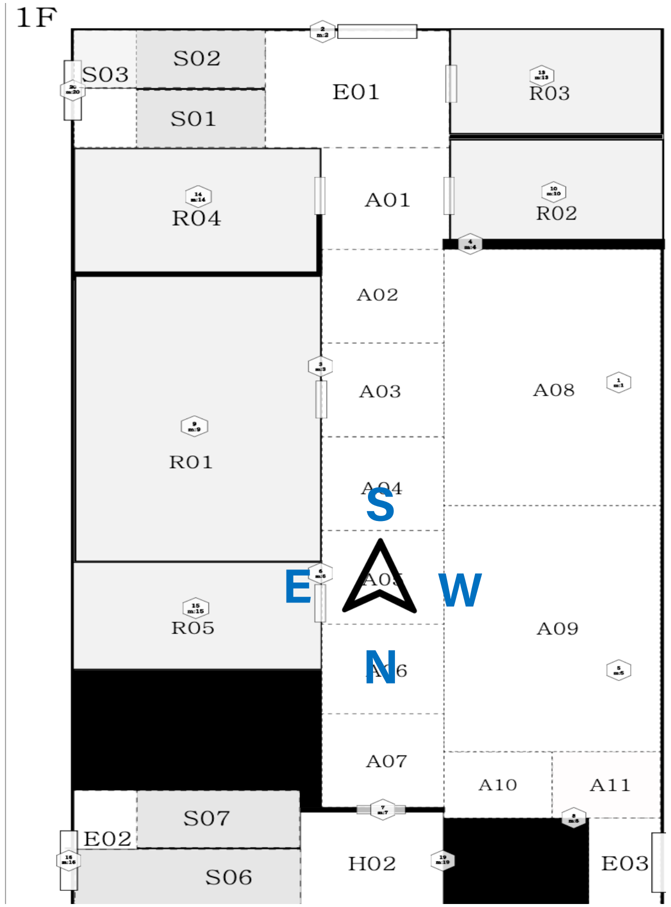


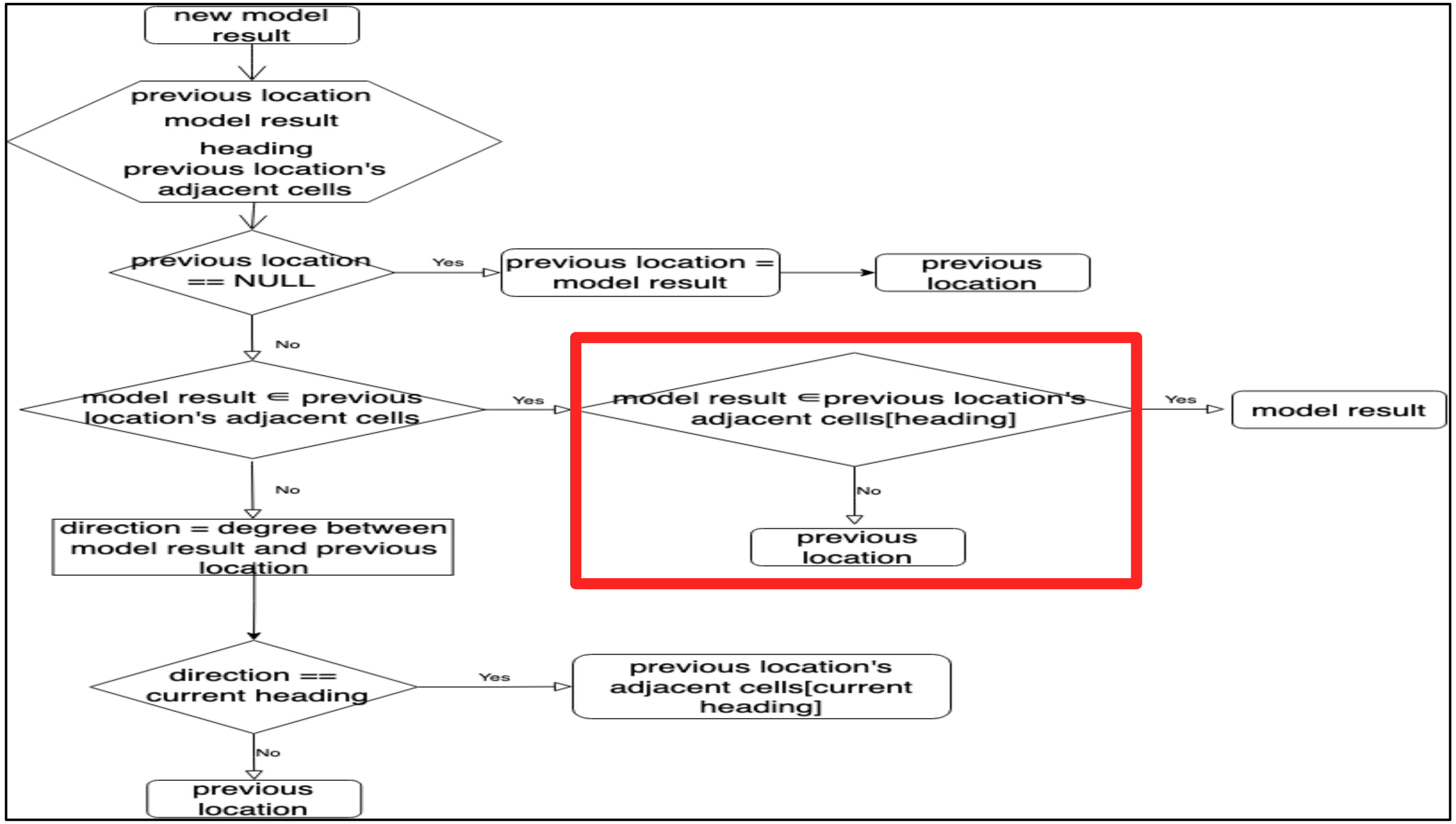
Indoor Localization for moving person



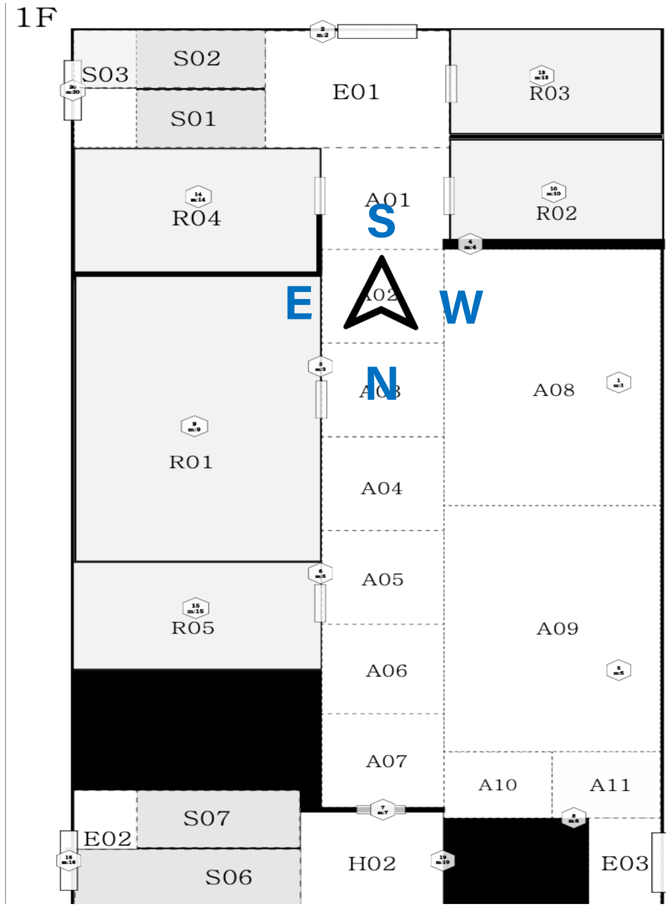


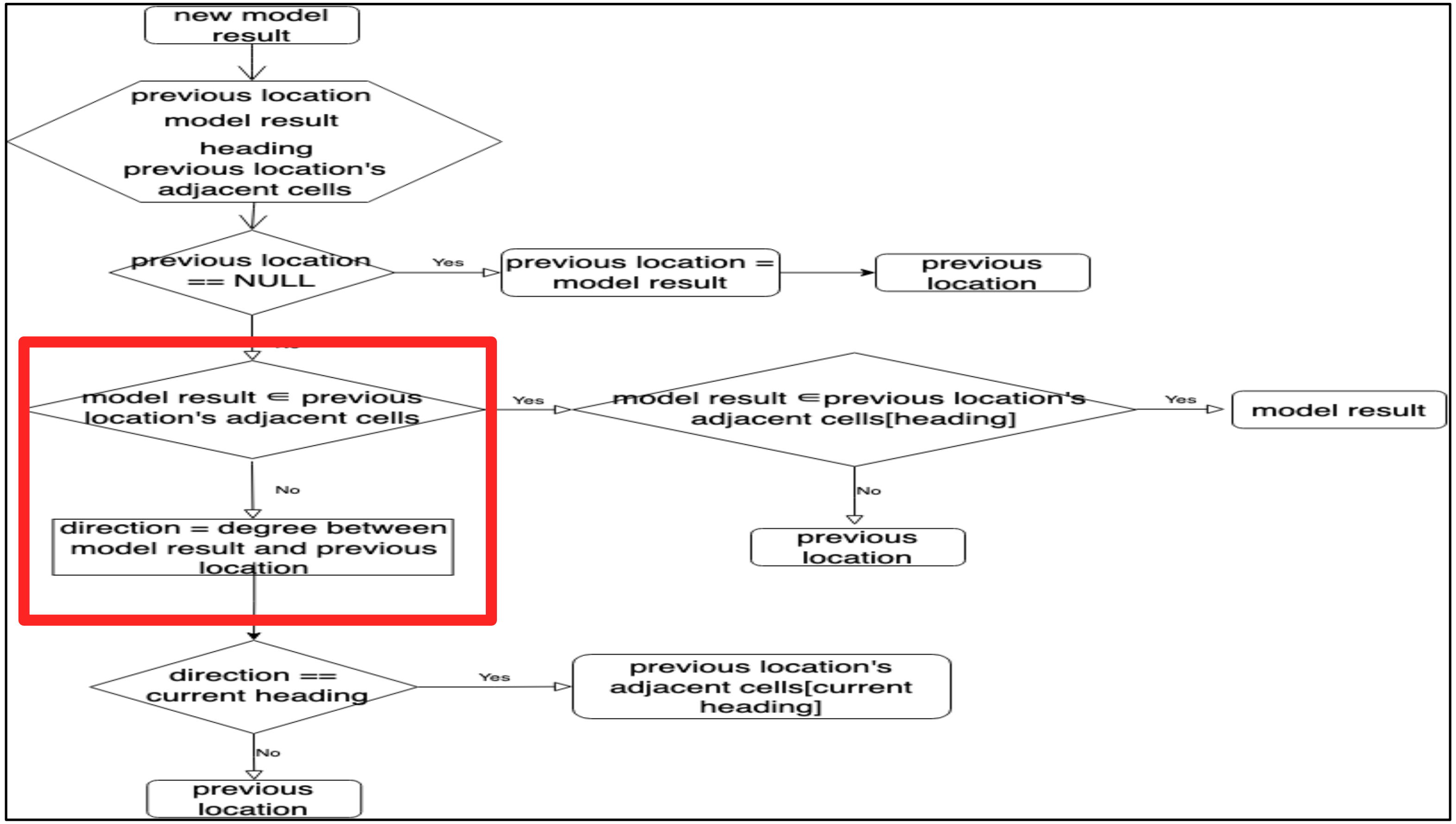
Indoor Localization for moving person



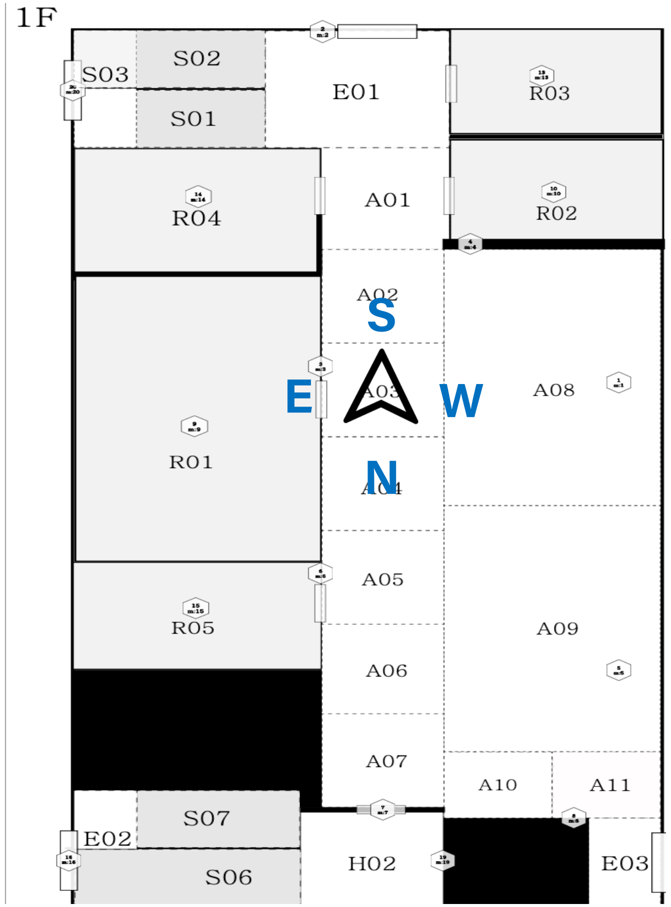


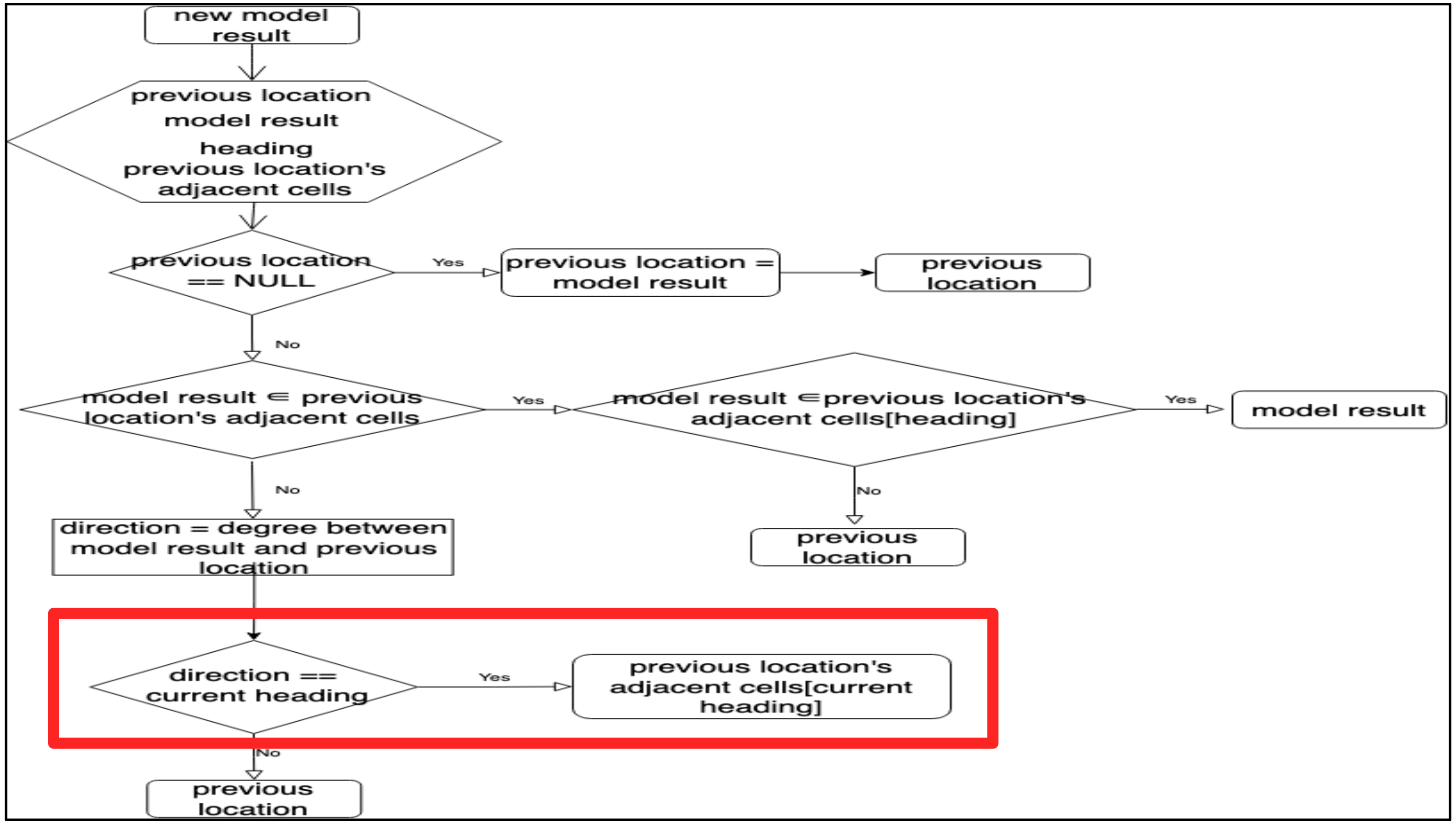
Indoor Localization for moving person



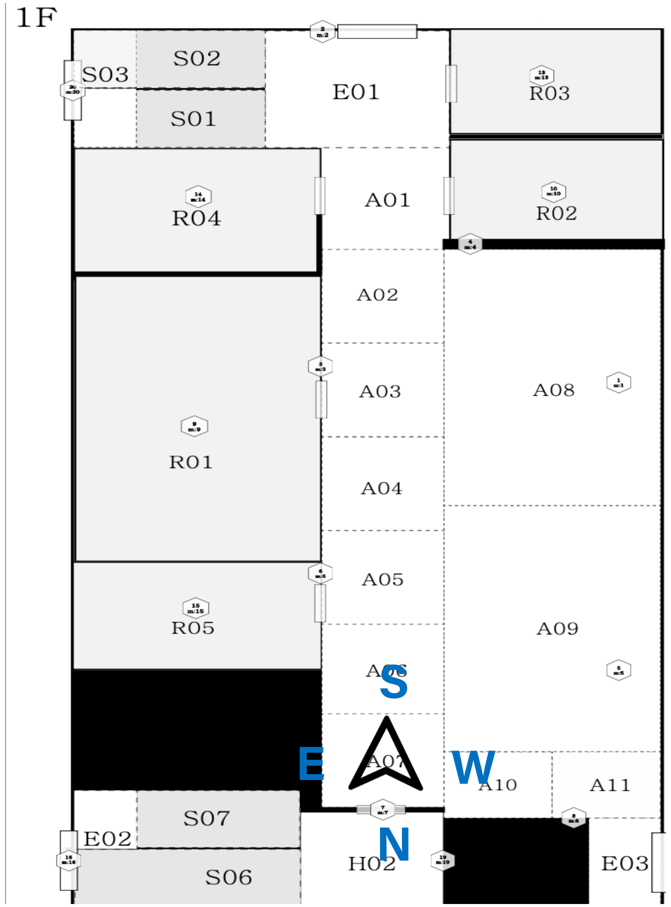


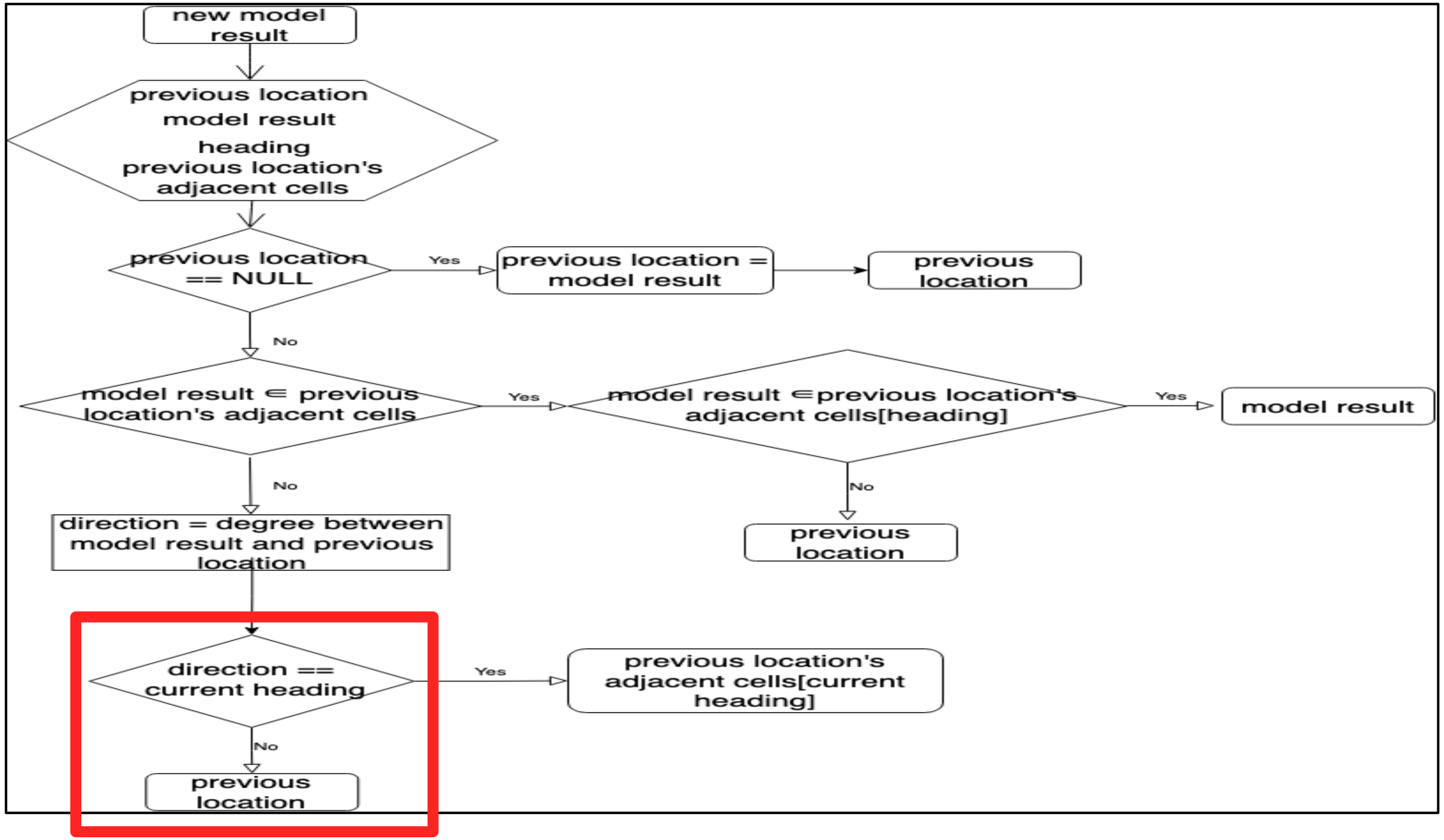
Indoor Localization for moving person



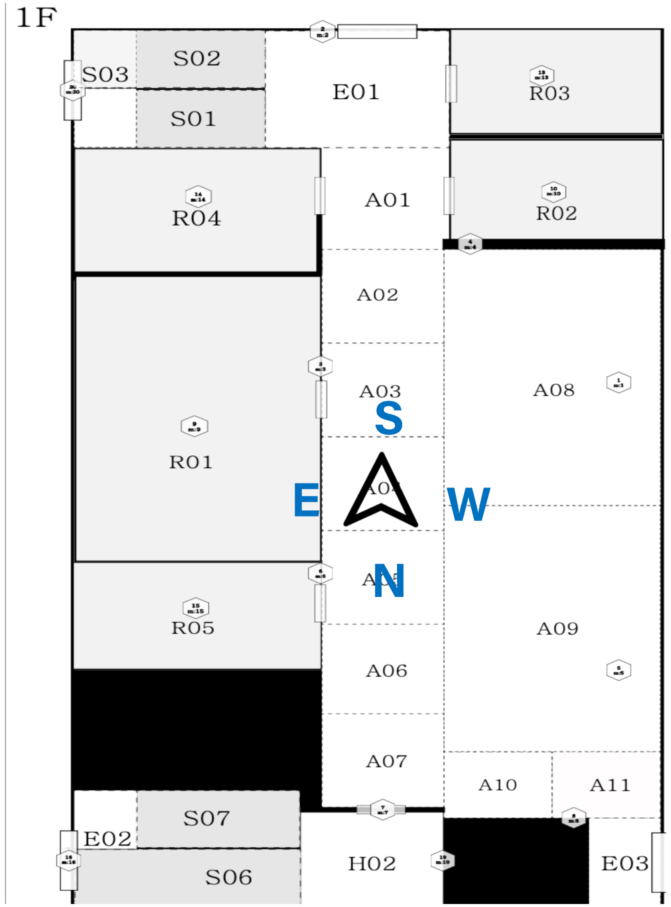


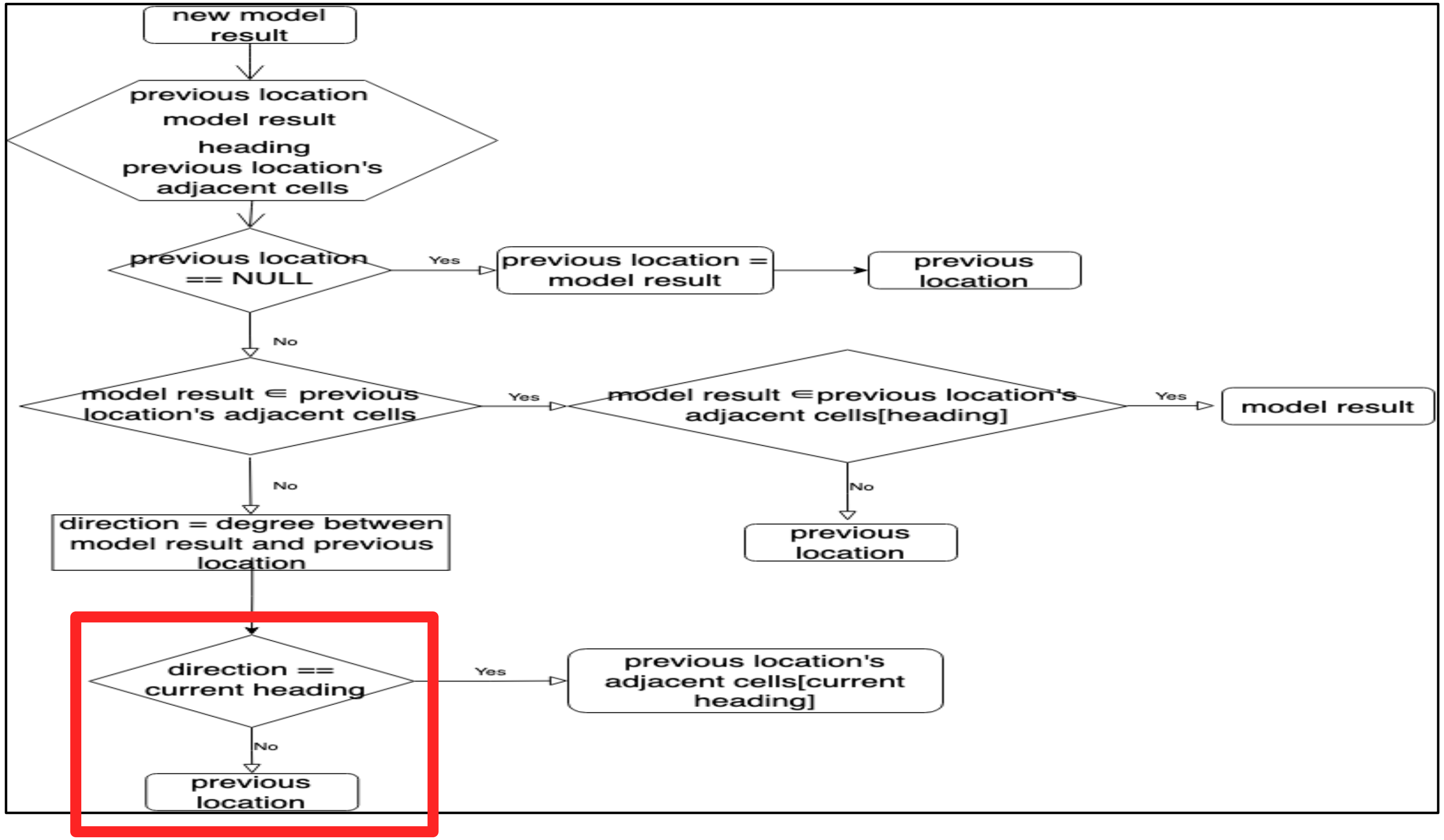
Indoor Localization for moving person





Indoor Localization for moving person

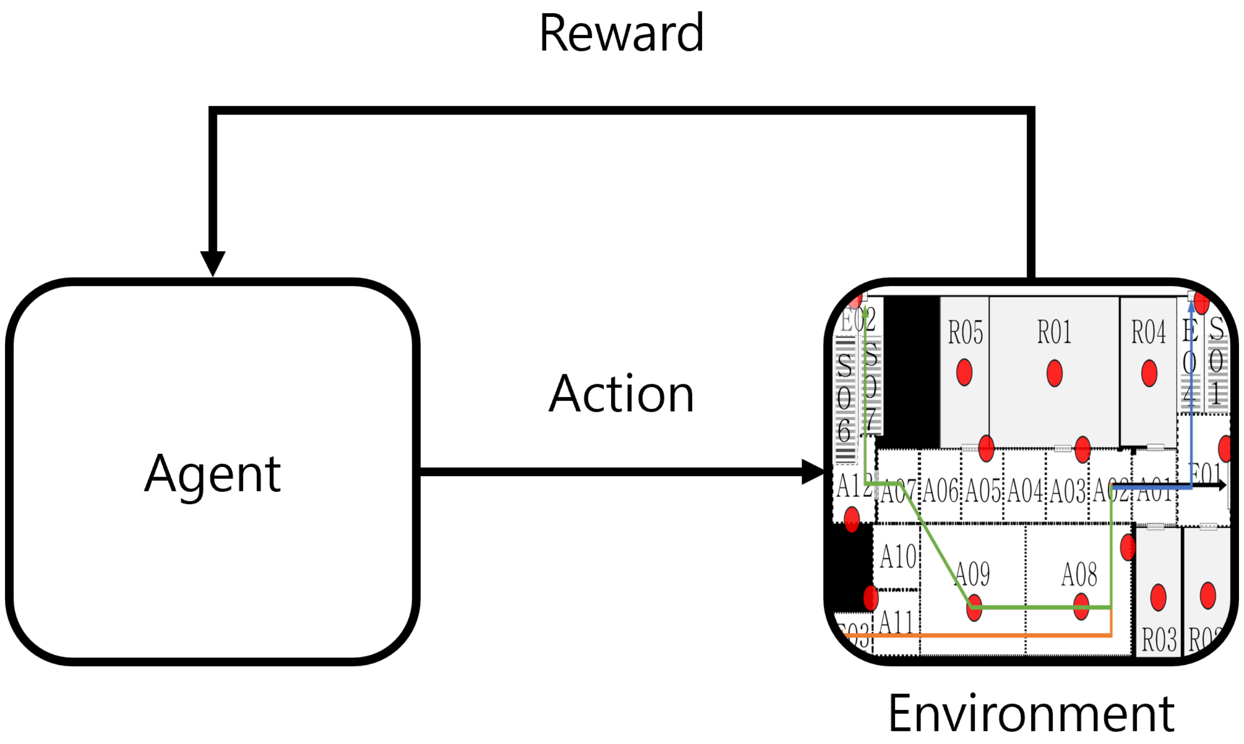




Pathfinding Algorithm

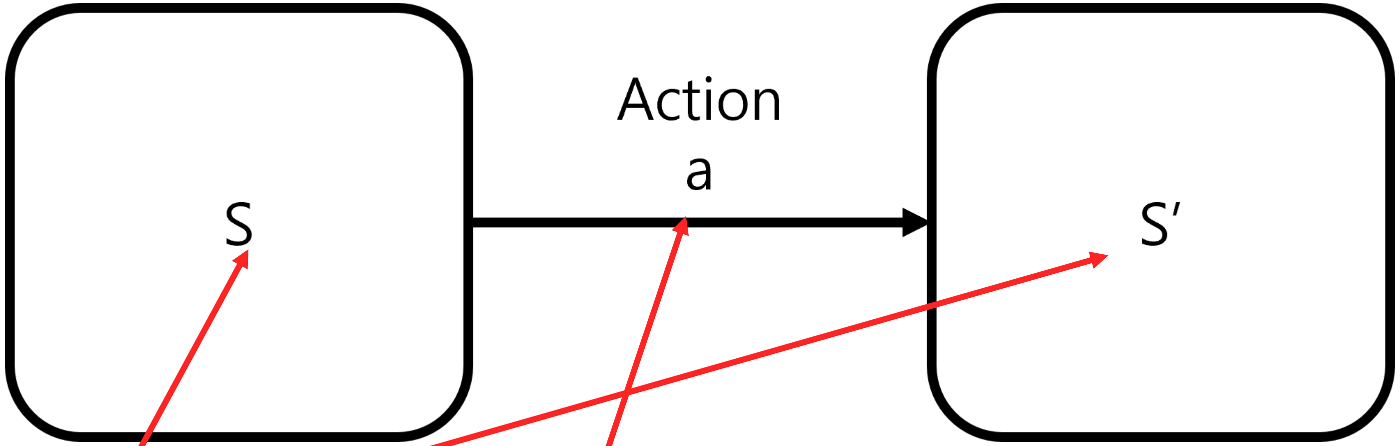
- Design of Q-learning algorithm
- Evacuation Simulation

Proposed Q-learning algorithm



Pathfinding algorithm

Proposed Q-learning algorithm



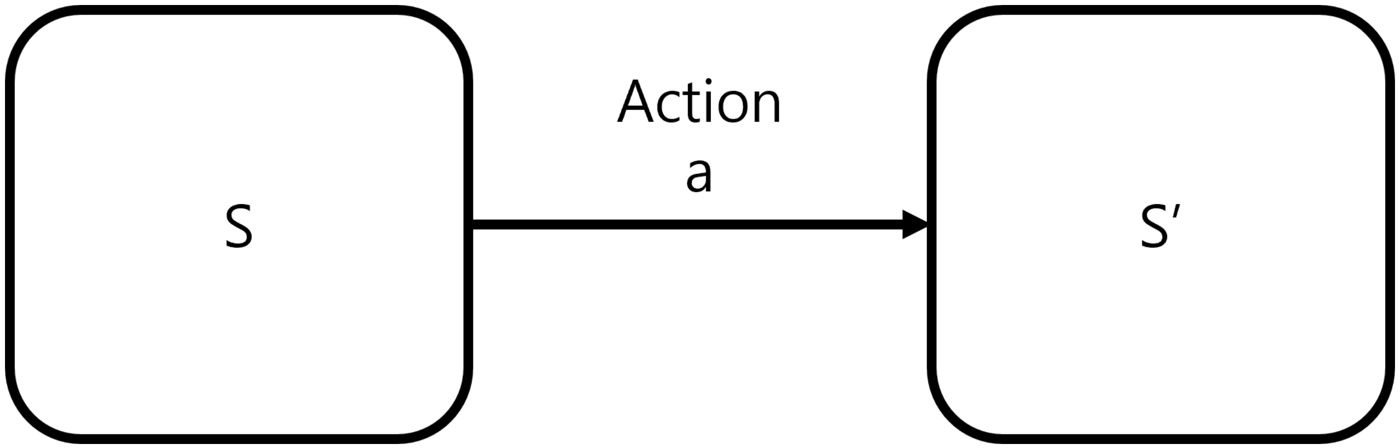
$$S = \{cell_1, cell_2, cell_3, \ldots, cell_N\}$$

$$A = \{A_1, A_2, A_3, \ldots, A_N\}$$

$$A_i = \{a_i \mid a_i \text{ is adjacent cell of } cell_i\}$$

Pathfinding algorithm

Proposed Q-learning algorithm



$$Q(s,a) = (1-\alpha)Q(s,a)+\alpha[R(s,a)+\gamma\times \max Q(s',a')]$$

Pathfinding algorithm

How the Q-learnig is trained

Algorithm 2: Proposed Q-Learning for Evacuation Pathfinding

Input: Cells topology, Number of episodes N_{epi}, Destination cell

Output: Q-table for Destinaion cell

for *iteration* $\leftarrow 1$ **to** N_{epi} **do**

  current cell $\leftarrow$ random cell

  $epsilon \leftarrow 1 -$ iteration / N_{epi}

  while *current cell* $\neq D$

    pick next cell by ϵ - greedy approach

    Update Q(s, a) by using Equation 7.

    currenct cell $\leftarrow$ next cell

  end

end

Pathfinding algorithm

Calculating path

Algorithm 3: Calculates path for Evacuation

Input: Cells topology, Starting cell, Destination cell
Output: Path from Starting cell to Destination cell
currenct cell $\leftarrow$ Starting cell
path $\leftarrow$ [Starting cell]
while *current cell* $\neq$ *Destination cell* **do**
    pick next cell by greedy approach
    currenct cell $\leftarrow$ next cell
    append current cell to path
end
return path

Pathfinding algorithm

$$Cost\ of\ path = Length\ of\ path + \sum_{i}^{f,a,c} W_i N_i$$

$$Efficiency\ of\ path = \frac{1}{cost\ of\ path}$$

W: Weight of
N: Number of
f: fire cell
a: fire adjacent cell
c: congested cell

Pathfinding algorithm

M: Most efficient path

S: Second most efficient path

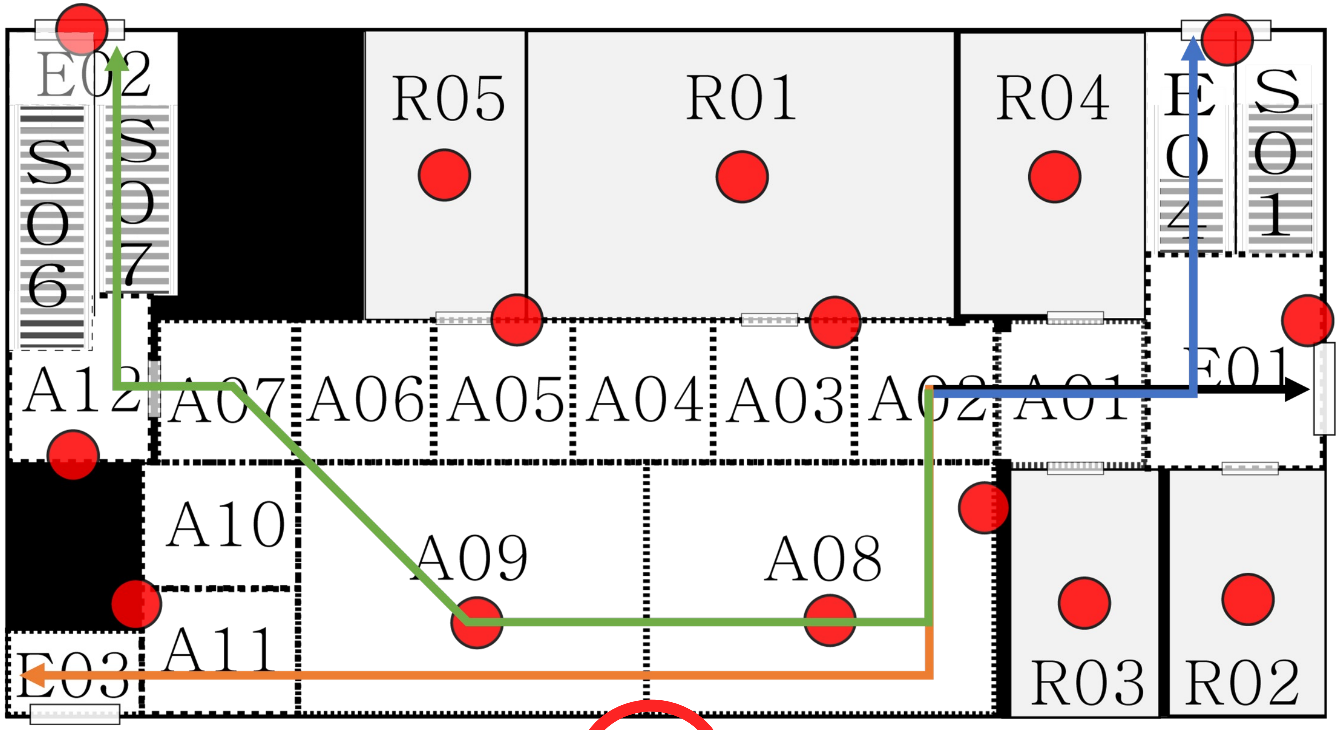
EM: Efficiency of M

ES: Efficiency of S

$$\text{Path} = \begin{cases} \text{M} & \text{with probablity } \dfrac{EM}{EM+ES} \\ \text{S} & \text{with probablity } \dfrac{ES}{EM+ES} \end{cases}$$

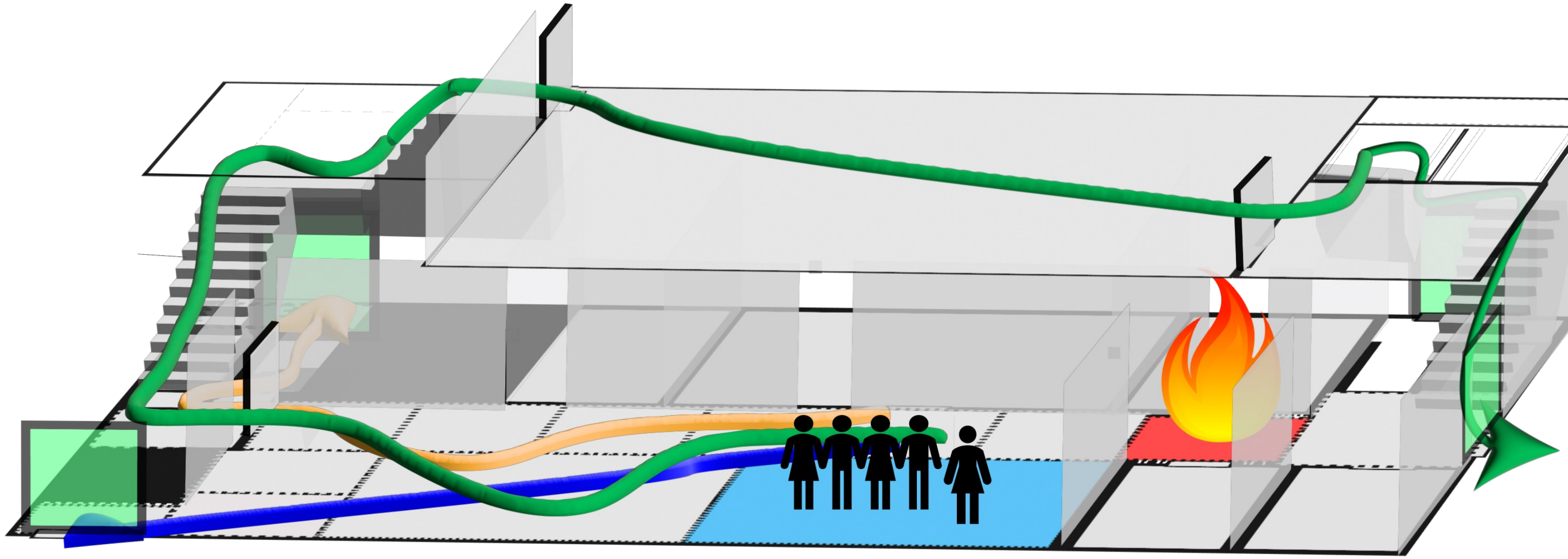
Pathfinding algorithm

Process of how the algorithm choose the path



- A02 to E01: A02 A01 E01 & 0.33
- A02 to E02: A02 A08 A09 A07 A12 S07 E02 & 0.14
- A02 to E03: A02 A08 A09 A11 E03 & 0.2
- A02 to E04: A02 A01 E01 E04 & 0.25

Pathfinding algorithm



- A02 to E01: A02 A03 A04 A09 A07 A12 S06 S05 A13 S04 S03 S02 E01 & 0.058
- A02 to E02: A02 A03 A04 A09 A07 A12 S07 E02 & 0.1
- A02 to E03: A02 A03 A04 A09 A11 E03 & 0.125
- A02 to E04: A02 A03 A04 A09 A07 A12 S06 S05 A13 S04 S03 S02 E01 E04 & 0.055

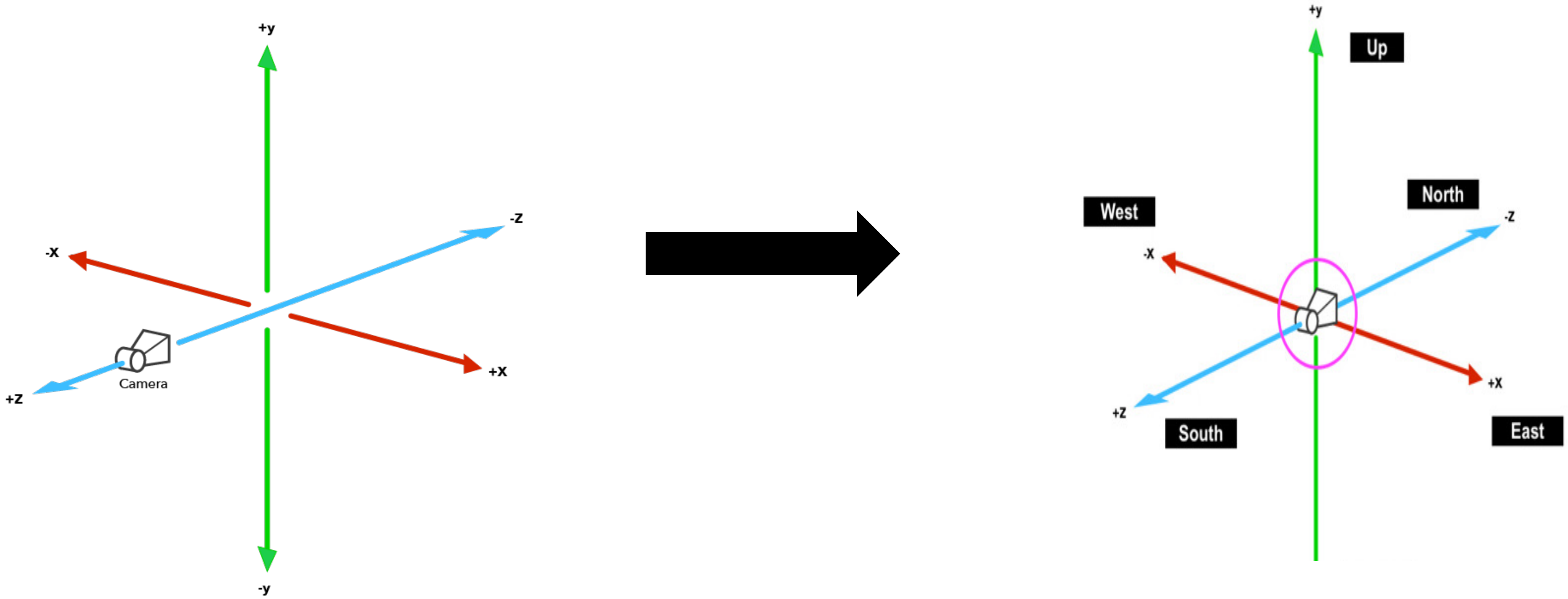
Pathfinding algorithm

	Split exit	Single path
# of people in each cell	Time for evacuation (sec)	
Normal situation		
1	11.2	12.0
3	29.45	36.0
5	46.5	60.0
8	75.05	96.0
10	91.95	120.0
Disaster situation		
1	13.85	13.0
3	37.9	39.0
5	60.65	65.0
8	100.65	104.0
10	119.95	130.0

Navigation system

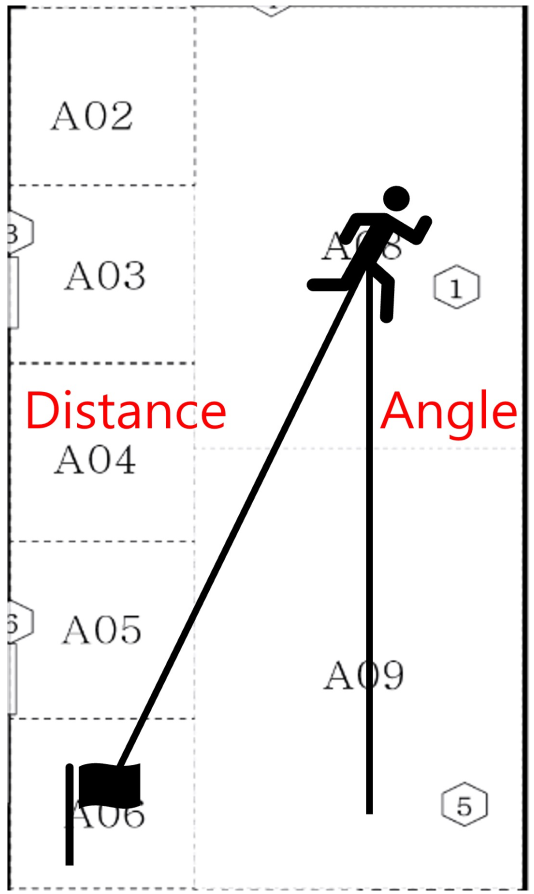
- Design of navigation system
- Demo video

AR; Orientation and Position



Navigation system

2D map



$$atan2(y,x) = arctan(\frac{y}{x})[x \neq 0] + (1 - 2[y < 0])$$

$$(\pi[x < 0] + \frac{\pi}{2}[x = 0]) + \textbf{undefined}[x = 0 \wedge y = 0]$$

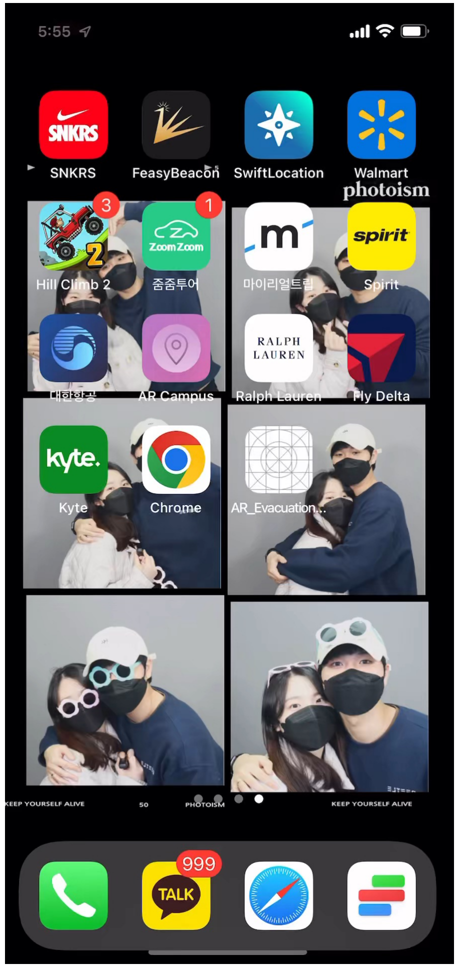
Navigation system

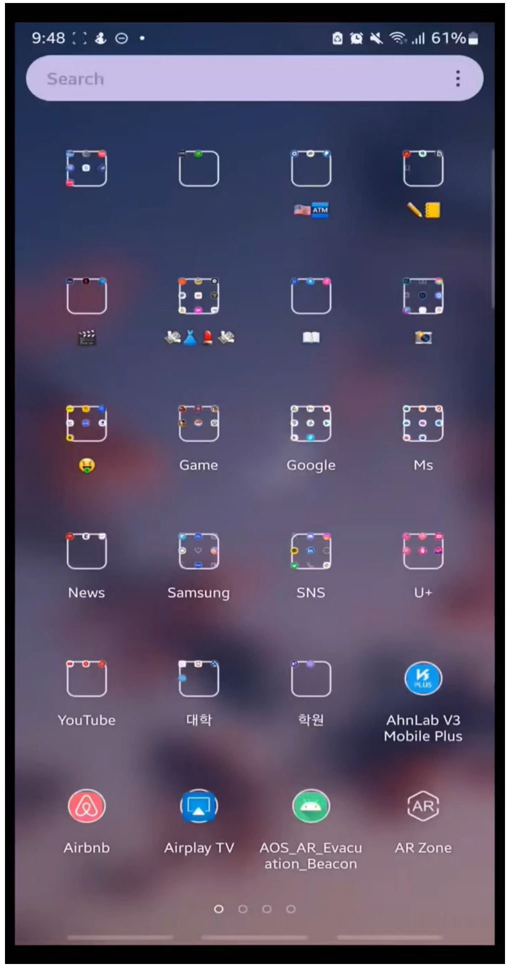
2D map

Color of Cell	Cell Status
White	Normal area cell
Green	Evacuation path cell
Yellow	Congestion cell
Orange	Risky area cell
Red	Fire area cell

Navigation system

Demo video





Q & A

References

- [1] S. RANGANATHAN, “Fire Protection Research Foundation publishes a literature review report on firefighter exposure to fireground carcinogens | NFPA,” *www.nfpa.org*, Jul. 01, 2021. https://www.nfpa.org/News-and-Research/Publications-and-media/Blogs-Landing-Page/NFPA-Today/Blog-Posts/2021/07/01/Fire-Protection-Research-Foundation-publishes-a-literature-review-report-on-firefighter-exposure (accessed May 25, 2022).
- [2] Apple iBeacon. Adapted from “Apple Developer” by Apple, retrieved from https://developer.apple.com/ibeacon/
- [3] S. Sadowski and P. Spachos, “Optimization of BLE Beacon Density for RSSI-Based Indoor Localization,” *2019 IEEE International Conference on Communications Workshops (ICC Workshops)*, May 2019, doi: 10.1109/iccw.2019.8756989.
- [4] S. Sadowski and P. Spachos, “RSSI-Based Indoor Localization With the Internet of Things,” *IEEE Access*, vol. 6, pp. 30149–30161, 2018, doi: 10.1109/access.2018.2843325.
- [5] C. H. Lam and J. She, “Distance Estimation on Moving Object using BLE Beacon,” *2019 International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, vol. 2019 International Conference on Wireless and Mobile Computing, Networking and Communication, no. 19222518, Oct. 2019, doi: 10.1109/wimob.2019.8923185.
- [6] K. Shimizu and D. Kushida, “Evacuation Guidance System Using Beacon Information and Dijkstra’s Algorithm,” *2021 IEEE 3rd Global Conference on Life Sciences and Technologies (LifeTech)*, Mar. 2021, doi: 10.1109/lifetech52111.2021.9391946.
- [7] A. Satan, "Bluetooth-based indoor navigation mobile system," 2018 19th International Carpathian Control Conference (ICCC), 2018, pp. 332-337, doi: 10.1109/CarpathianCC.2018.8399651.
- [8] “Building Map | PMU Purdue University,” *union.purdue.edu*. https://union.purdue.edu/event-services/venues/krach/floor-plan.html

THANK YOU