Lawrence Technological University

Research Day, April 20, 2022

The Prototype Development of IoT Based Wheelchair Tracking System for Detroit Metro Airport (DTW)

Gerard Bringard, MS Computer Science Candidate, CAS; CJ Chung, PhD, Advisor, Professor, CAS

Lawrence Tech — Theory and Practice 1932 — Arts and Sciences

Abstract

Detroit Metro Airport (DTW) has a need for tracking wheelchairs throughout the North terminal, parking structures, and some outdoor spaces of the airport. We are working on the development of Wi-Fi-based geolocation systems using ESP32 and ESP8266 microcontrollers. We are comparing different Machine Learning algorithms such as Decision Trees, Naive Bayes, and Random Forest along with a Deep Learning model using TensorFlow Lite for microcontrollers. The Machine Learning algorithms and the Deep Learning model determine the current location by signal strengths of nearby Wi-Fi hotspots and then send the predicted location information to the gateway hosted on ThingPark Enterprise. The number of input dimensions for the ML/DL model will be the number of known Wi-Fi signals in the terminal. And the label will be the location ID we decide to set up, such as "Gate 1", "Bag Pickup" and so on. In addition, Bluetooth Low Energy (BLE) based geolocation system is being experimented with. This consists of 3 parts, trackers, beacons, and gateways. The current analysis results show that Wi-Fi-based systems are low-cost solutions working both indoors and outdoor, but accuracy depends on the number of known Wi-Fi signals at the airport. However, there is an issue with mobile phone hotspots and other non-permanent Wi-Fi signals are being read in and negatively impacting the accuracy. To combat this issue, we scrubbed those out and only used permanent Wi-Fi signals (such as McDonald's and Starbucks). BLE systems ensure very low power consumption and better accuracy however, BLE solutions are expensive as the trackers and beacons needed are a lot more expensive than the ESP32 and ESP8266 microcontrollers.

Design and Approach

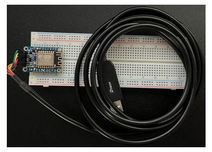
Wi-Fi Based:

The base of the Wi-Fi based solution with machine learning is the ESP8266 microcontroller. This microcontroller has a Wi-Fi module (see Figure 1) that can read the surrounding networks and their strengths. An Arduino sketch was written that allows the user to enter the place they are in, then scan the surrounding networks and record their name and strength in the serial monitor. The results in the serial monitor are fed into a python notebook which takes the networks and their strengths and generates Classifier and Converter header files in C for each algorithm. The algorithms used were Decision Tree, Gaussian Naïve Bayes, and Random Forest. Another Arduino sketch was written that uses both of the header files to predict the location. Figure 3 shows an example layout for a Wi-Fi based solution. The red dot represent the microcontroller. It receives the surrounding Wi-Fi signals and their strengths and runs a ML algorithm to determine it's position. It sends that position to the gateway.

Bluetooth Low Energy:

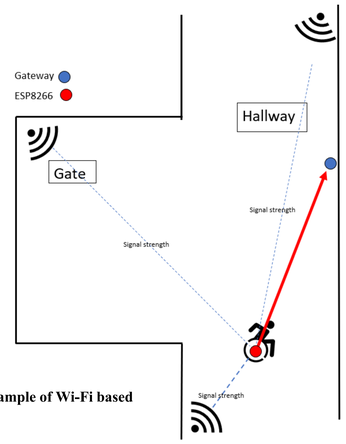
This approach has 3 main parts: the tracker, the beacons, and the gateway (See Figure 2). The beacons continuously emit a signal that has identifying information about what beacon it is that is picked up by the tracker. The tracker picks up the signal from the nearest beacon and sends an encoded message containing the identifying information for the beacon to the gateway. The gateway then displays the encoded payload on the ThingPark website. Figure 4 shows an example layout for a BLE based solution.

Design and Approach Cont.

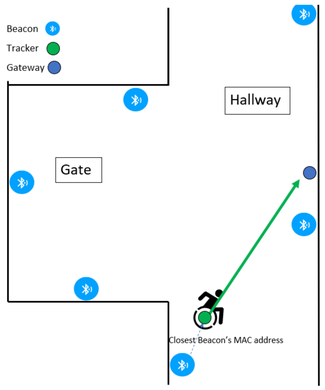


[Figure 1: ESP8266]

[Figure 2: Tracker, Beacon, and Gateway]



[Figure 3: An example of Wi-Fi based solution]



[Figure 4: BLE Example] The blue cicles represent BLE beacons which continuously emit a signal that is then received by the tracker, represented by the green dot. The tracker then sends the information it received to the gateway, represented by the blue dot.

Location \ Router name	ACTor	actor2	Robofest	iWheels	Phantom
Canoe	-67	-43	-77	-78	-78
Canoe	-71	-52	-80	-78	-76
...	...	...	...	...	...
Café	-70	-72	-78	-56	-72
Café	-72	-70	-78	-54	-80
...	...	...	...	...	...
AAC	-50	n/a	-88	-75	-86
AAC	-46	n/a	-85	-76	-84
...	...	...	...	...	...
M216	-82	-84	-90	-88	-70
M216	-87	-85	-90	-91	-70
...	...	...	...	...	...

[Table 1: Some examples of actual training data acquired in Figure 5 test environment]

Testing

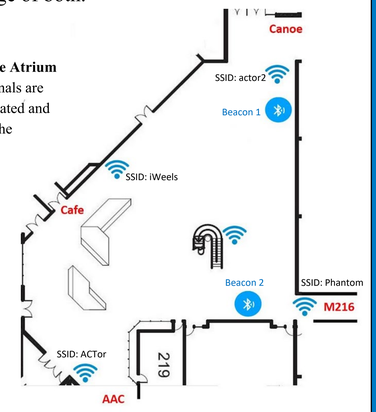
Wi-Fi Based:

Training data was collected by running the Arduino sketch that scans the Wi-Fi signals and strengths at each location chosen. 50 scans were taken at each location. We chose by the Canoe, in front of the café, in front of the AAC, and in front of room M216. Testing was done by running a different Arduino sketch and going to each location and recording the output of the serial monitor. 10 tests were ran at each location. The layout for testing is shown below in figure 5. Table 1 shows some examples of training data.

Bluetooth Low Energy:

The testing for this method was done by placing the beacons at opposite sides of the atrium and moving around to different points in the atrium to see which beacon the tracker would pick up. We tested with points close to Beacon 1, close to Beacon 2, equidistant from both, and out or range of both.

[Figure 5: Testing Layout of the Atrium on LTU campus] The Wi-Fi signals are where the Wi-Fi routers were located and the Bluetooth signals are where the beacons were located.



Findings and Results

Wi-Fi Based:

The testing result of each trained model is as follows:

Decision Tree:	92.5%
Gaussian Naïve Bayes:	95.0%
Random Forest:	97.5%
Deep Learning:	98.3% (This was not physically tested in the Figure 5 environment)

Bluetooth Low Energy:

When the testing closer to Beacon 1, the payload would contain the MAC address of beacon. When the testing closer to Beacon 2, the payload would contain the MAC address of Beacon 2. When testing equidistant from both, the results were split on which MAC address to return. When the testing outside of the range of either beacon, the payload would not contain a MAC address.

[Figure 6: Beacon 1 MAC]

```
"payload_hex": "2901000000000000000045b0609fa63fd6fe34034c02ec0034",
```

[Figure 7: Beacon 2 MAC]

```
"payload_hex": "2901000000000000000001a708165569bd5001affc6fc4c00b0",
```

[Figure 8: No MAC]

```
"payload_hex": "02",
```

Discussion

Based on the results of the testing and the price of the materials needed for each method, we recommend Wi-Fi based tracking. The highest accuracy for the Machine Learning algorithms was 97.5% with the Random Forest algorithm. This accuracy is excellent and can be replicated at DTW due to the various Wi-Fi networks such as Starbucks and McDonald's. The ESP8266 microcontroller is also a lot less expensive than the Bluetooth tracker ($10 vs. $60) and you need one for each cart. Also for the Wi-Fi based, you can use the existing Wi-Fi networks in the area, there is no need for buying more routers where as with the Bluetooth method you have to buy the beacons to place every 10-15m to ensure accuracy ($11 each). Both methods make use of the gateway so that will be a fixed cost for each method.

Future Steps

Wi-Fi Based:

- Run the DL model on the ESP32 board
- Test at DTW airport
- Send results to gateway
- Create user interface for the locations of the wheelchairs
- Test with multiple microcontrollers
- Improve ML accuracy
- Research security of microcontrollers

Bluetooth Low Energy:

- Test at DTW airport
- Test with multiple trackers and gateways
- Create user interface for the locations of the wheelchairs
- Research security of Bluetooth trackers and beacons

Acknowledgements

This project requirements were provided by Ryan Daniels & Robert Strong of the Wayne County Airport Authority. Ryan Daniels gave some technical support. The project materials were supported by the Dept. of Math and Computer Science and Dr. Chung's robotics research account.

References

[1] Eloquent Arduino: https://eloquentarduino.com/projects/arduino-indoor-positioning

[2] CETech Videos: https://www.youtube.com/watch?v=H07p6zvBavg

[3] TensorFlow Lite for Microcontrollers: https://www.tensorflow.org/lite/microcontrollers

[4] Yuri R Videos: https://www.youtube.com/watch?v=5IuZ-E8Tmhg&t=380s

[5] Kirby's SQL Talk: https://www.youtube.com/watch?v=jaM4E-zKcoE&t=71s

[6] Dmytro Korablyov: https://medium.com/@dmytro.korablyov/first-steps-with-esp32