Towards an Indoor Navigation System using Bluetooth Low Energy Beacons

Fernando Campaña, Adriano Pinargote, Federico Domínguez and Enrique Peláez
Centro de Tecnologías de Información
Escuela Superior Politécnica del Litoral, ESPOL
Vía Perimetral Km. 30.5
Guayaquil, Ecuador
Email: {ferecamp, apinargo, fexadomi, epelaez}@espol.edu.ec

***Abstract*—Indoor navigation, the ability to find a path to an office, a conference room, or the exit in an unfamiliar building, is one of the most well-known applications of indoor positioning technology. This technology remains relatively underdeveloped due to the inherent difficulty of geolocalization in an indoor environment. Currently, we are working in an indoor navigation system based on Bluetooth Low Energy (BLE) Beacons fingerprinting and we created a testbed deployment of 30 beacons in the Center for Information Technologies of ESPOL University. This paper presents a comparison study of three machine learning techniques used to improve BLE fingerprinting accuracy within our testbed. Preliminary results show that Random Forest, 30% more accurate than Naive Bayes, is able to correctly estimate the location of multiple users with room-level accuracy 91% of the time.**

***Index Terms*—eddystone, beacons, indoor localization, artificial intelligence, random forest**

I. Introduction

Geolocalization technologies have become an integral part of our daily lives thanks to the ubiquitousness of smartphones and GPS coverage. We use GPS powered services for requesting directions, sharing our location with friends and family, for giving context to photos and videos, and to trigger actions based on location. Reliable outdoor navigation is however one of the most beneficial applications of geolocalization technologies and providing a similar solution to indoor environments has been so far a long term goal. GPS is not a good choice for indoor localization because the walls and ceilings completely block the signal of the satellites employed by GPS. Moreover, the measurement error of GPS is too large for its usage in indoor environments where meters are significant and can place a user in the wrong room. Many technologies such as BLE (Bluetooth Low Energy), Wi-Fi and Electromagnetic Field have been tested to solve the indoor localization problem. Wi-Fi fingerprinting is known to provide accuracy of a few meters. However, it is a power-hungry protocol and access points are rarely deployed with the required geometry and density. In contrast, BLE has been designed to be a machine-to-machine energy efficient protocol, allowing devices with long battery lives, lower costs and maintenance. [1]

This paper presents a study of an indoor localization methodology using BLE fingerprinting together with three different machine learning algorithms. The final aim is to produce a robust low-cost indoor localization technique for in-building navigation.

Specifically, our study focuses on the impact of beacon density on the accuracy of the system, as well as answering the following questions:

- How well suited are BLE beacons for an Indoor Positioning System (IPS)?
- Which are the best parameters for improving location prediction?
- What can be done to reduce errors due to multi-path propagation and signal fading?
- Which Machine Learning algorithm is more effective for BLE beacons fingerprinting?

We tested the indoor localization system in our building, varying some parameters to improve its accuracy. This article presents this work and it is structured as follows: section II discusses previous and related work, section III describes in detail the design of the solution, section IV describes the experimental results obtained, section V presents a brief discussion, and the conclusion is presented in section VI.

II. Related work

Most approaches found on literature and commercial systems use information obtained from sensors (magnetometer, accelerometer, etc) and Radio Frequency signals (BLE, Wi-Fi). Here we present the most salient examples found on the literature.

There are several approaches which employ Wi-Fi signals for localization. The authors in [2] propose a solution in which a user wears an android based smartwatch, which measures the Wi-Fi signal strength to create a Wi-Fi fingerprint for estimating its location via a machine learning algorithm. They analyzed a pair of algorithms and learning strategies, and compared algorithms such as Random Forest, SVM (Support Vector Machine), Bayesian Networks, Decision Trees and Multilayer Perceptron with walking and static sampling. Bayes networks was the best approach followed by Random Forest, yielding an accuracy of 87%.

Some researchers analyzed BLE signals for feasibility in building indoor location systems. Faragher and Harle analyzed the BLE signals and concluded that 6-8 beacons is a good number for a fingerprint with beacons parameters of -20dBm and 10Hz. They also suggested that Wi-Fi scanning can reduce BLE signal strength measurements. They reached an accuracy of 2.6m, 95 percent of the time [1]. Kavitha. S et al. also analyzed the characteristics of the BLE signals of two models (Estimote and pebBLE) by obtaining the path loss of both models and then using that information for conducting simulations on a random distribution of beacons using KNN fingerprinting [3].

Some approaches use only Radio Frequency signals for localization. Estel and Fischer introduced a system composed of BLE beacons in which the location is estimated by trilateration, yielding an accuracy of about 5m, they argued that it is not enough for indoor localization [4].

There exists more error on BLE signals than on Wi-Fi, caused by fast fading multi-path interference [1], therefore some solutions employ some filtering for reducing the error. Park et al. propose a trilateration location based system by employing an algorithm with Kalman filtering. The systems was composed of Android devices and Estimotes BLE beacons. The average location tracking error was about 1.77 m [5].

The implementation of the Kalman Filter in [6] presents an improvement for Bluetooth signals, and using the trilateration through beacons improved the accuracy calculation of the positioning.

Other approaches employ device sensors and Radio Frequency signals. Duco [7] uses pedestrian dead reckoning along with iOS wireless location estimation (CoreLocation) which uses a combination of GPS, cellular, Wi-Fi and Bluetooth signals to estimate location. However, CoreLocation's specific implementation is unknown. iBeacons are deployed in Wi-Fi dead-zones, stairs and elevators. By using trilateration, dead reckoning and Kalman filtering, Rbesaat et al. [8], developed a system that achieves an accuracy of less than one meter by employing BLE beacons.

In [9] the improvement for the RSSI measurements is ruled by two steps, averaging and smoothing, the test had two phases, with and without calibration, the calibration consists on taking the path loss values and measure them with the path signals gathered before. These steps reduce errors caused by conditions such as humidity and frequency interferences.

In [10] there are three methods implemented, the nearest-beacon, the Gaussian-noise, and a particle filter. The authors proposed a mapping scheme to represent the environment as a node graph with the user's path, and using the positioning algorithms to constrain the search space. Another implementation of two commercial solutions are explained in [11], Navigine and GoIndoor, and an online content management system (CMS) for the map scaling and measurement. Both solutions use a smart-phone device for recognizing and placing the beacons. These solutions allow the map to separate through walls for better positioning.

III. Methodology

We propose a solution composed by a fingerprinting Android App, a server, and a beacon infrastructure. The beacon infrastructure was designed to obtain a good distribution of beacons across the building. The beacons broadcast Eddystone packets. According to the Eddystone's Google site [12]: "Eddystone is an open beacon format developed by Google and designed with transparency and robustness in mind". The App's purpose was to read the Eddystone packets over a period of time and send the information of the beacons found to the server for further processing. The server was responsible for saving and processing the fingerprints, so they can be used in a machine learning algorithm to predict a location.

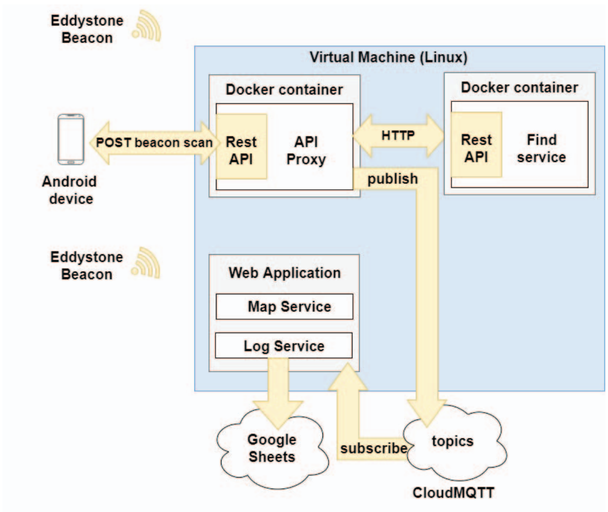

Fig. 1. Architecture of the system

For the beacon infrastructure, we used two beacons types, Estimote and iBeeks. The architecture was composed of 6 Estimote Beacons and 24 iBeeks. The beacons were placed throughout the second floor of our office building as shown in (Fig. 2). They were placed on the wall at a height of 2.50 meters to minimize signal absorption, diffraction, interference and multi path propagation, caused by the materials and objects encountered through its path [13]. The transmission power of the beacons was selected so the signal was able to cover the biggest of the rooms. The selected power was -12dBm for the Estimotes and -16dBm for the iBeeks. According to each manufacturer the selected powers for the beacons are able to reach 15m and 18m respectively. The information was obtained from each of manufacturer's beacons configuration App.

We used fingerprinting to estimate position. Fingerprinting consists of scanning the Eddystone packets over a period of time, process each packet as they come, obtain the values associated with each beacon, create a message containing the beacon MAC and chosen RSSI for that beacon, and finally log the collected information (a fingerprint) to a server for further processing. In the tracking stage of fingerprinting, the application can use those fingerprints to return a prediction

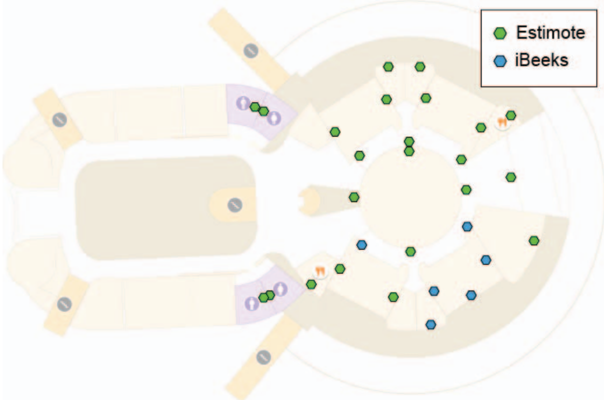

Fig. 2. All beacons were evenly placed in the laboratory section of the second floor.

comparing the fingerprint sent to the model created by the fingerprints used for learning. The data representation are shown in Fig. 3. The fingerprinting technique creates a map in a radio of an area based on the data from several access points (Beacons) and generates a probability distribution of values for a given location. Then the values are compared to the fingerprint to find the closest match and generate a predicted location.

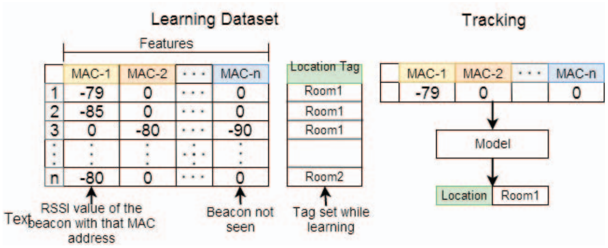

Fig. 3. Data representation for learning and tracking

The mobile application and the server were based on an open-source project called Find [14]. However Find was designed for Wi-Fi fingerprinting, so we adapted the mobile application to scan Eddystone packets and send the MAC address and RSSI of nearby BLE beacons instead of the MAC address and RSSI of Wi-Fi Access Points. We also used three machine learning algorithms implemented in FIND: Bayes Classifier, Random Forest, and SVM. SVM's performance was inferior compared to the other two and was discarded immediately. We also increased the number of estimators that Random Forest used, this improvement allowed the algorithm to fit the classifier with more data gathered in the fingerprinting to obtain a better precision in the positioning. The parameters that allowed us to improve the estimation were:

- *n estimators*: The number of trees in the forest.
- *max depth*: The number of depth in the tree to be explored.
- *min samples split*: The minimum number of samples to perform an internal split in the node.
- *random state*: The random generator criterion.
- *max features*: The maximum number of samples to perform an internal split in the node.

In our experiments we used 50% of the collected data for training and 50% for tracking.

The mobile application runs on Android 5.1.1 (API 21) and makes use of the location and Bluetooth services. The application allows the user to set the learning period, as well as the learning and scanning rate. It also lets the user learn new locations by providing its name and training the location over a period of time, the process is shown in Fig. 4. After a location has been learned, it can be used as a location for tracking the user. The tracking is made using the training information of the Eddystone packets of the beacons.

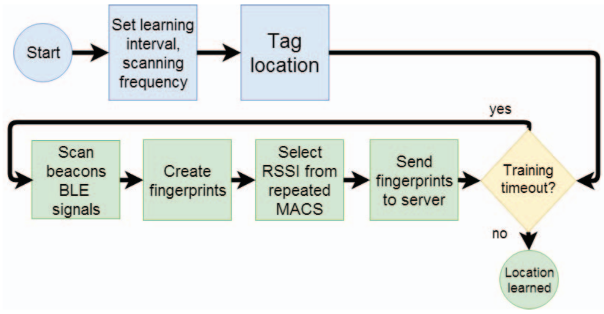

Fig. 4. Training process

The Find server comes with various machine learning algorithms by default. We tested Bayes Classifier, Support Vector Machine (SVM) and Random Forest. A proxy was placed between the App and the server, so the fingerprints can be intercepted and sent to a pub/sub service over MQTT. A proxy server was also placed between the server and the tracking devices, to catch the information directed to the server and resent it to an MQTT Broker. We also developed a map, a

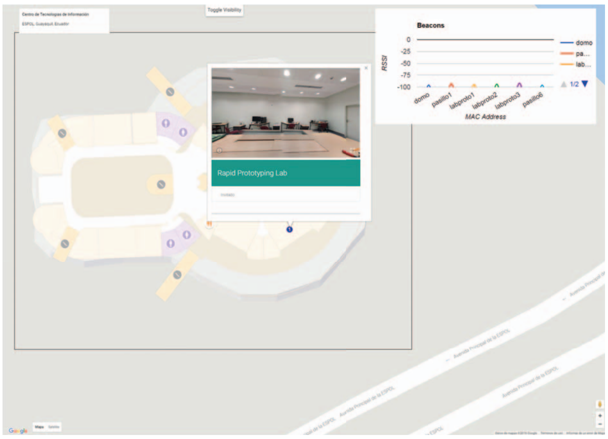
Fig. 5. A localization test using the map interface is shown here together with a debug window of scanning results.

chart and a logging utility. The map was intended to show the real time location of all users using the App. It uses MQTT and Websockets to show real-time information. It displays an internal map of our building superimposed over Google Maps, each room was registered as a location. When at least one user was located in a room, a marker appears indicating the number of users in that room. Clicking the marker displays a pop-up showing a 360 degrees photo of the room, the name of the room and the names of the users in the room. A web page consumes the data sent to the proxy server and logs the fingerprints into Google Sheets for further processing as shown in Fig. 6. The chart was also used for debugging purposes such as viewing the amount of beacons scanned during a period, the highest beacon scanned and the signal's change over time. Some filtering and ordering was made, to ensure the beacons appeared in the same place every time and the comparisons can be made correctly.

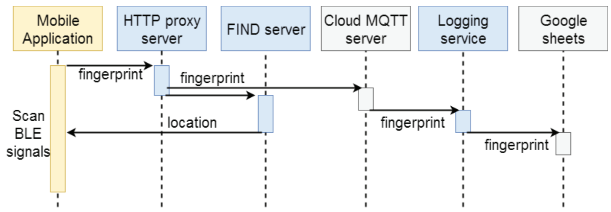

Fig. 6. Tracking process

IV. Results

We developed some experiments to analyze the accuracy of the system. Some experiments were done varying the transmission and sampling rate of the beacons, the RSSI's selection of the repeated occurrences of the same MAC address, and defining proximity zones in the application.

For the beacons infrastructure, initially we started with a transmission rate of 1 second, later on it was changed to 100ms for better performance. The transmission rate's increase allows us to capture more beacons during the same scanning window, creating better fingerprints for learning.

We tested the system in a static setting, the beacons were transmitting at 1Hz. In this test, we trained the locations using the App while walking parallel to the walls inside the room, at a steady pace of a normal person walking velocity. Then we verified the accuracy, by placing three different brands of mobile devices in separated positions within the room and using the tracking App. The characteristics of the devices used in the test are detailed in Table I.

TABLE I
Smart-phone's Characteristics

Device	RAM	Bluetooth	Android	Processor
1	3 GB	4.0	6.0.1	Qualcomm MSM8974AB
2	3 GB	4.2	7.0	Snapdragon 810
3	2 GB	4.0	5.0	Intel Atom Z2560

During the experiment, the cellphone 1 and cellphone 2 were placed in the laboratory labeled 'lab1', and the cellphone 3 was placed in the laboratory labeled lab2. We collected the fingerprints over enough time to obtain more than 100 records for doing an evaluation of successes and errors.

The results obtained for the Bayes Classifier are shown on Table II.

Table III shows the results when the Random Forest Classifier was used.

As it is shown the Bayes Classifier failed almost half of the time it tried to classify, resulting in several sudden "jumps" of locations. However, the Random Forest Classifier resulted in 25% of the times failing, resulting in a decrease on the number of jumps, but still not eliminated at all.

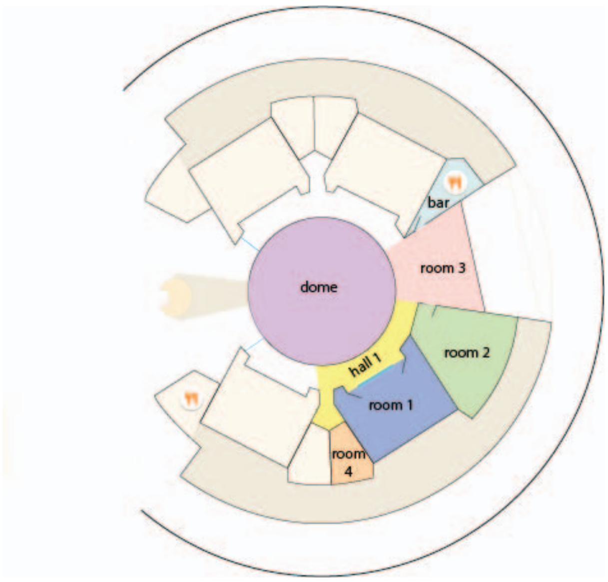

Fig. 7. Location names

For the subsequent experiments, we used the Random Forest Classifier. The next experiment was intended for verifying which selection strategy was better between median, average and the latest RSSI obtained. We called selection strategy, the act of selecting one RSSI from various packets scanned for the same beacon, and then use that RSSI as the RSSI for the beacon for an scanning interval. The experiment consisted on doing a simultaneous learning with the three strategies while walking through the same path. The learning period was set to 5 min, and the scanning period for both tracking and learning was set to 3 seconds. Then the accuracy of each one was logged individually while traversing the rooms. We learned four locations labeled as room 1, room 2, room 3 and hall 1, as shown in Fig. 7 . Obtaining the confusion matrix shown on Table IV.

TABLE II
Bayes Classifier's Algorithm Results.

Bayes					
Smart-phone	HITS	FAILS	TOTAL	HITS (%)	FAILS (%)
1	2	72	74	2.70	97.30
2	46	8	54	85.19	14.81
3	26	26	52	50.00	50.00
ALL	74	106	180	41.11	58.89

TABLE III
RANDOM FOREST'S ALGORITHM RESULTS.

smart-phone	HITS	FAILS	TOTAL	HITS (%)	FAILS (%)
1	149	22	171	87.13	12.87
2	89	32	121	73.55	26.45
3	68	34	94	63.83	36.17
All	298	88	386	77.20	22.80

TABLE IV
CONFUSION MATRIX

Median	room 1	room2	room 3	hall 1
room 1	88	5	0	3
room 2	5	89	3	1
room 3	2	0	91	11
hall 1	5	6	6	85

The confusion table shows that the prediction makes mistakes more frequently with the locations that are adjacent to them, something that we were expecting. But, occasionally it made mistakes with places which were far apart, due to multi-path signal propagation.

From here on, we changed the transmission rate of the beacons to 100ms. We also wanted to verify which one of the three learning processes was better; for instance, the system learned by walking around the room parallel to the walls, by walking around randomly and by placing the learning devices statically over a surface on the middle of the location. We learned three locations labeled as room 1, room 4 and hall 1. The results are shown in table V. This shows that when surveying a point and tracking that same point, the results were outstanding when the learning time was short, failing completely when increasing the time, caused by overfitting. Walking through the location provided the best results, because it covered almost every place of the location and the quantity was good enough for estimation.

TABLE V
LEARNING STRATEGIES

	5 min	10 min	30 min
Walking	95	96.6	
Walking parallel to wall	87	97.1	87.8
Static on center	100	100	34.5

An experiment was developed dividing the selection process, partitioning the tracking period and creating 3 or more fingerprints over the time interval. These fingerprints were sent to the server and a location was selected using a voting system. The results are presented in tableVI. Where a 'Bad' is shown, it means that the prediction was totally incorrect, like switching fast between locations and picking the wrong location too often. Descriptions were used because the results couldn't be logged with that frequency due to limitations on the API. The best was the one with the tracking of every 1.5 seconds and learning of 1 second, which tells us that partitioning and voting didn't have a major impact and reducing the scanning window actually led to bad results due to fewer beacons present.

TABLE VI
RESULTS DUE TO DIVIDING AND VOTING

	Learning scanning		
Tracking (s)	0.5 s	1 s	1.5 s
0.5	Bad	Bad	Bad (41.9%)
1	Almost good	Good	Good (87.5%)
1.5	Almost good	Good (92.1%)	Good (88.35%)
0.5 divided by 3	Really bad	Bad	Bad
1 divided by 3	Bad	Bad	Bad
1.5 divided by 3	Bad	Bad	Bad

Next, with the goal of obtaining a better localization resolution, we split the lab into two areas (labeled as lab1 and lab2) and set 3 seconds for tracking and learning scanning intervals. Using the latest RSSI obtained as selection method and learning for 5 minutes in each area, we obtained at most 95.5% correct predictions.

The table VII also shows that the median gave the best results, followed by the latest and the average in last place. The failing of the average was caused by the rapid signal variability, and as it is shown, they differ between each other.

- Experiment A: 3 s scanning, 5 minutes, 4 locations
- Experiment B: 1.5 s scanning, 2 locations
- Experiment C: 3s scanning, lab split into 2 locations

TABLE VII
SELECTION

	Learning			
Selection	A	B	C	Total
Median	85.425	87.5	79.3	84.075
Average	79.875	78.15	60	72.675
Latest	88.025	83.75	95.5	89.092

Lastly, we tried converting the fingerprints into proximity zones, and sending the zones to the learning algorithm to be able to detect proximity. We used a number to represent each zone; hence, if the RSSI of a beacon was between a range, the application detected it, and sent a number representing the zone to the learning algorithm. This approach yielded a 91% of success. The zones were selected by measuring the beacons RSSI from different distances.

Zones:

- Between -10 and -55: -40 (Immediate)
- Between -56 and -79: -60 (Near)
- Between -80 and -89: -80 (Far)
- Under -90: -100 (Unknown)

V. DISCUSSION

The Random Forest Classifier worked particularly well in our system, compared to the Shadowing Bayes Classifier. At first we used a transmission rate of 1s for making battery efficiency and to allow the system infrastructure to be used on real situations, but then we changed the transmission rate to 100ms, the fastest rate possible, and the results improved considerably. The downside is that it is not battery efficient and cannot be implemented on a real system without having to replace all beacons twice a year.

Find was originally developed for Wi-Fi fingerprinting, but in this experiment it was adapted to work with Bluetooth beacons fingerprinting. Moreover, the smart-phones used in the experiments have different Bluetooth versions, affecting for example the beacon's timestamps, these issues may introduce some errors in the fingerprinting stage.

The use of electromagnetic waves in indoor localization has always been challenging due to reflection, path propagation, and interference with materials, which can introduce noise in the survey. Noise can be reduced in this case by using filters, such as a Mean filter or a Median filter in the collected data. After filtering, we determined that good results were obtained by using the Median. A Kalman filter could be used to improve the system.

The parameters that improved the localization prediction the most were the transmission rate of the beacons and the scanning window, due to the presence of more beacons and more signals and data available for training the algorithm.

Find uses the Naive-Bayes [15] classifier for Wi-Fi systems by default, the reason for its usage is because it assumes that all features are conditionally mutually independent, this assumption is aggressive compared to other Bayes Algorithms that may assume dependency within their variables by using a Bayesian Network[16]. This assumption cannot be done in Wi-Fi or BLE systems, because we need to prove the dependencies between the variables to find the best position estimation. Both algorithms don't overfit [17], allowing them to predict the location with RSSI values not fingerprinted. Random Forest proved to be a better choice against the Bayes classifier because Bayes can't handle interactions between features as good as RF, where the features are the MAC addresses [18].

VI. Conclusions

In this paper we presented a study of an indoor positioning system implemented using Bluetooth Low Energy beacons. In this implementation, we used a fingerprinting technique for estimating location with three machine learning algorithms: Naive Bayes, SVM, and Random Forest. Random Forest gave us the best results with a 30% increase accuracy over Bayes and a correct location identification (room-level accuracy) of around 91%. A robust indoor positioning system is the back-end of an indoor navigation system, the final aim of our work, and at this moment we are able to locate multiple users with room-level accuracy in the second floor of our office building. We currently have a prototype mobile App that uses the smart-phone's magnetometer sensor and Dijkstra's shortest path algorithm to find the best route between two users within our building. Room-level accuracy is sufficient for a useful indoor navigation application, however it is still necessary to improve the 91% location estimation to have a better user experience.

Acknowledgment

The authors would like to thank Google for their contribution to this research via the IoT Research Award.

References

[1] R. Faragher and R. Harle, "An analysis of the accuracy of bluetooth low energy for indoor positioning applications," in *Proceedings of the 27th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2014), Tampa, FL, USA*, vol. 812, 2014, p. 2.

[2] Ó. Belmonte-Fernández, A. Puertas-Cabedo, J. Torres-Sospedra, R. Montoliu-Colás, and S. Trilles-Oliver, "An indoor positioning system based on wearables for ambient-assisted living," *Sensors*, vol. 17, no. 1, p. 36, 2016.

[3] M. Ji, J. Kim, J. Jeon, and Y. Cho, "Analysis of positioning accuracy corresponding to the number of ble beacons in indoor positioning system," in *Advanced Communication Technology (ICACT), 2015 17th International Conference on*. IEEE, 2015, pp. 92–95.

[4] M. Estel and L. Fischer, "Feasibility of bluetooth ibeacons for indoor localization," *Digital Enterprise Computing (DEC 2015)-GI-Edition: Lecture Notes in Informatics (LNI). P-244. Bonn: Gesellschaft für Informatik, printed by Köllen Druck+ Verlag GmbH*, pp. 97–108, 2015.

[5] J. Park, J. Kim, and S. Kang, "Ble-based accurate indoor location tracking for home and office."

[6] S.-H. Lee, I.-K. Lim, and J.-K. Lee, "Method for improving indoor positioning accuracy using extended kalman filter," *Mobile Information Systems*, vol. 2016, 2016.

[7] C. Surmeli and T. Serif, "Duco-hybrid indoor navigation," in *International Conference on Mobile Web and Information Systems*. Springer, 2016, pp. 256–267.

[8] J. Röbesaat, P. Zhang, M. Abdelaal, and O. Theel, "An improved ble indoor localization with kalman-based fusion: An experimental study," *Sensors*, vol. 17, no. 5, p. 951, 2017.

[9] N. Kuxdorf-Alkirata and D. Brückmann, "Improved indoor localization approach based on bluetooth low energy."

[10] P. Dickinson, G. Cielniak, O. Szymanezyk, and M. Mannion, "Indoor positioning of shoppers using a network of bluetooth low energy beacons," in *Indoor Positioning and Indoor Navigation (IPIN), 2016 International Conference on*. IEEE, 2016, pp. 1–8.

[11] D. Torstensson, "Indoor positioning system using bluetooth beacon technology," 2017.

[12] "Eddystone format — beacons — google developers," https://developers.google.com/beacons/eddystone, (Accessed on 08/24/2017).

[13] "Best practices for installing estimote beacons estimote community portal," https://community.estimote.com/hc/en-us/articles/202041266-Best-practices-for-installing-Estimote-Beacons, (Accessed on 07/21/2017).

[14] Z. Scholl, "Find," https://www.internalpositioning.com/, (Accessed on 07/20/2017).

[15] "High precision wifi indoor positioning framework: Find cyberpunk," https://n0where.net/high-precision-wifi-indoor-positioning-framework/, (Accessed on 08/19/2017).

[16] D. Kelly, S. McLoone, and T. Dishongh, "A bluetooth-based minimum infrastructure home localisation system," in *Wireless Communication Systems. 2008. ISWCS'08. IEEE International Symposium on*. IEEE, 2008, pp. 638–642.

[17] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct 2001. [Online]. Available: https://doi.org/10.1023/A:1010933404324

[18] "Choosing a machine learning classifier," http://blog.echen.me/2011/04/27/choosing-a-machine-learning-classifier/, (Accessed on 08/28/2017).