

Bluetooth Low Energy based Occupancy Detection for Emergency Management

Avgoustinos Filippoupolitis
Computing and Information Systems
University of Greenwich, UK
Email: a.filippoupolitis@gre.ac.uk

William Oliff
Computing and Information Systems
University of Greenwich, UK
Email: ow302@gre.ac.uk

George Loukas
Computing and Information Systems
University of Greenwich, UK
Email: g.loukas@gre.ac.uk

Abstract—A reliable estimation of an area’s occupancy can be beneficial to a large variety of applications, and especially in relation to emergency management. For example, it can help detect areas of priority and assign emergency personnel in an efficient manner. However, occupancy detection can be a major challenge in indoor environments. A recent technology that can prove very useful in that respect is Bluetooth Low Energy (BLE), which is able to provide the location of a user using information from beacons installed in a building. Here, we evaluate BLE as the primary means of occupancy estimation in an indoor environment, using a prototype system composed of BLE beacons, a mobile application and a server. We employ three machine learning approaches (k-nearest neighbours, logistic regression and support vector machines) to determine the presence of occupants inside specific areas of an office space and we evaluate our approach in two independent experimental settings. Our experimental results indicate that combining BLE with machine learning is certainly promising as the basis for occupancy estimation.

1. Introduction

Precise knowledge of the location of people in a building can be highly useful for several applications, from optimised building energy consumption and customised museum experiences to targeted marketing. It may even be crucial in responding to evolving crises, as first responders and emergency personnel can use location information to improve their planning and intervention approach. In fact, the availability of precise location information is often an assumption in related research [1], [2], [3]. This assumption is accurate in only few cases. For instance, during the January 9th 2015 terrorist attack in a French supermarket, an employee helped hide customers inside a cold storage room and used his mobile phone to inform authorities on their locations. On the same day, during a security operation that took place in a printing plant taken over by terrorists, one employee managed to hide from the hostiles and communicate with the police via text messages to provide them with his location. However, this was not the case during the terrorist attack in the Tunis museum on March 18th 2015. Two tourists spent the night hiding in the museum and were found the

next day. They did not use their mobile phones because they were afraid of attracting the attention of the terrorists.

In such cases, modern wireless communication technologies are required. A particularly promising one is Bluetooth Low Energy (BLE), the importance of which for indoor localisation has been acknowledged by the US Federal Communications Commission. The Commission has proposed a roadmap that would use WiFi and BLE to help locate emergency 911 callers inside buildings. The plan includes a call for demonstrations that would use handsets able to detect BLE or WiFi location beacons to communicate with a public safety network and allow the callers location to be identified. Our prototype system progresses along these lines. It uses off the shelf BLE technology to provide information on the locations of occupants inside a building, with primary goal to provide information regarding the real-time utilisation of specific areas in the building. These can include offices, laboratories, conference rooms, communal spaces and other types of spaces depending on the building in question.

2. Related Work

Bluetooth Low Energy is used for a variety of applications related to building occupants, such as remote healthcare monitoring [4], [5], indoor navigation [6] and activity recognition [7]. With respect to indoor localisation and occupancy estimation, there are numerous approaches that target different area types. In [8], the authors present a system based on iBeacons for detecting the occupancy of a building. They have evaluated their system by predicting whether a user was inside or outside of a single room. The authors in [9] have evaluated a system for determining the movements of office occupants and controlling a building management system. They have used an office area to evaluate their BLE based system and although they present building energy consumption results, there is no evaluation of the mobile application’s energy consumption. Another approach presented in [10] aims at estimating a building’s occupancy using Arduino hardware beacons that implement Apple’s iBeacon protocol and Apple mobile phones. The evaluation only addresses the users presence inside or outside a room and does not give details on the individual room occupancy accuracy. An occupancy estimation system for hospitals

		Smartphone Energy Consumption	
		Tested	Not Tested
Occupancy Detection	Room level	[8]	
	Office level		[9]
	Floor level	[10]	[11]
Localisation	Room level		[12], [15]
	Office level		[13]
	Floor level		[14], [16]

TABLE I. COMPARISON OF BLE BASED RELATED APPROACHES FOR OCCUPANCY DETECTION (TOP) AND LOCALISATION (BOTTOM)

is presented in [11], where iBeacons were used inside the floor of a university building. The system’s prediction has a high overall accuracy rate but accuracy results for individual areas are not given while energy consumption is not investigated. In [12], the authors present an indoor localisation system based on BLE beacons. They evaluate the system inside a single room and although they report a high accuracy performance, they provide limited results. An indoor localisation system based on BLE beacons is presented in [13]. The authors evaluate the system inside an office area and achieved a low localisation error (less than 1.8m) for 75% of the time. Although they discuss the energy consumption of the beacons, the mobile phone’s energy consumption is not addressed. In [14], the authors explore the advantage of BLE over WiFi for indoor positioning. They use a dense BLE beacon distribution and have achieved tracking accuracy of less than 2.6m for 95% of the time. While different beacon transmission frequencies have been investigated, their effect on the energy consumption of the mobile device is not addressed.

Table 1 presents a comparison of the aforementioned approaches in terms of the type of problem they address (occupancy detection or localisation), the scale of the experimental evaluation (at the level of a room office or floor) and whether they have evaluated their approaches’ energy costs. Our approach is geared towards emergency management and the level of detail we aim at is to provide a snapshot of the number of occupants inside building areas such as offices, laboratories and conference rooms. Even if our system stops functioning (e.g. due to an earthquake that affects communication infrastructure or due to hostiles that have confiscated the occupants mobile devices), it still can provide invaluable information related to the most recent spatial distribution of the occupants before an incident took place. Since energy consumption plays an important role in emergency situations, as it affects how long a mobile device can remain functional, we have also investigated this aspect of our system’s performance.

3. Description of the System

In this section we give the details of our system design, including an overview of the architecture and a description of the system components.

3.1. Overall architecture

The main concept of our approach is based on the use of BLE beacons installed inside the building which communicate with a mobile application and allow our system to infer the user’s location.



Figure 1. Commercial system architecture

More specifically, Figure 1 illustrates how such a system operates in a commercial setting. The beacons use the BLE advertising communication channel, to periodically broadcast packets. This is a non-connectable mode where all the information of interest (such as beacon’s unique ID) is included in the advertisement packet. These packets are received by mobile phones in the vicinity of the beacon, and are further processed by a corresponding mobile application. The assumption is that the mobile application has information regarding the location of the beacons inside the building (such as a mapping between beacons and rooms). Using this information, the mobile application calculates which is the beacon closest to it. Finally, the mobile application communicates its location to a remote server to receive contextual information (such as a targeted advertisement).

Figure 2 illustrates how our system will work in an emergency situation taking place inside a building. The mobile application receives BLE packets from the beacons to collect information regarding their unique IDs. Then the mobile application sends the respective RSSI values and beacon IDs to the server located at the remote control center. A software running on the remote control center receives the information and uses it to calculate the occupancy of the building. The main differences of our approach are:

- We do not assume knowledge of the mapping between beacon ID and location of beacons inside the building
- The mobile phone does not process the received beacon packets to calculate its location
- The server does not send information to the mobile phone

Since our system does not require any localisation processing to take place in the mobile phone, it is more flexible as the mobile devices can have low computational and memory requirements. The server is responsible for processing the data sent by the mobile phone and calculate the building occupancy. This is achieved by conducting one data gathering phase for the system during which the the data gathered are used to train a classifier. Section 4.1 provides

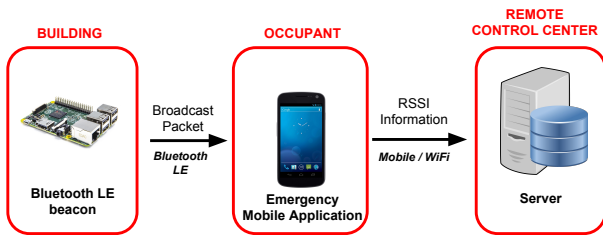


Figure 2. Proposed emergency system architecture

TABLE 2. OVERVIEW OF COMPONENT FUNCTIONALITY DURING NORMAL SYSTEM OPERATION

Component	Functionality
BLE beacon	These devices are deployed in the building and use BLE in advertising mode to broadcast their unique ID to mobile phones in their vicinity.
Mobile App	The Mobile Application installed on the building occupants phones will receive BLE messages from multiple beacons and send their RSSI values and respective beacon IDs to the remote control center
Remote Control Center	When the server at the remote control center receives information from a mobile device, it uses a trained classifier to update the building occupancy estimation

further details on the data gathering process. After this phase has taken place, the system can operate in normal mode, as depicted in Table 2.

3.2. BLE Beacons

Our beacons are based on a Raspberry Pi 2 with an attached Bluetooth 4 BLE module. Using the Bluez package, we can successfully simulate an iBeacon, which is the BLE beacon implementation proposed by Apple. The iBeacons are transmitters only and broadcast a set Bluetooth Low Energy advertising data packets. The data packets can then be picked up by mobile devices that are listening for Bluetooth data packets being transmitted by iBeacons. Figure 3 depicts our Raspberry Pi based beacon with a BLE USB interface.



Figure 3. One of our Raspberry Pi based prototype BLE beacons

We should note that this approach is more flexible than using a commercial BLE beacon, since our system is not

tied to a specific beacon manufacturer. Figure 4 illustrates the structure of the BLE advertising packet.

Prefix [9 bytes]	UUID [16 bytes]	Major [2 bytes]	Minor [2 bytes]	Tx [1 byte]
---------------------	--------------------	--------------------	--------------------	----------------

Figure 4. iBeacon Bluetooth Advertising Packet Structure

To identify the iBeacons individually, a small hierarchy using the Universally Unique Identifier (UUID), major number and minor number has been put in place. The UUID is used to define the universal group. Hence, all the iBeacons have the same set UUID in the Bluetooth advertising data packet. Thus, allowing the separation of the iBeacons being used in our experiments from other unassociated Bluetooth devices. Then, the major number is used to define local groups of iBeacons. For example, iBeacons belonging to a certain building or floors in a building depending on size. Lastly, the minor number is used to define each individual iBeacon within the local group it belongs to.

3.3. Mobile application

Our Android mobile application can be used for both the data gathering face and the normal operation. In the data gathering mode, the application detects BLE advertising data packets being transmitted from iBeacons that the mobile device is in range of and logs all activity of the iBeacons being detected. The application does filter the received Bluetooth advertising data packets and only stores the iBeacons that are being used in this project. When the mobile application does detect a Bluetooth advertising data packet from an iBeacon, it is able to extract the individual pieces of data from the payload of the data packet, such as the UUID, major number, minor number and transmission (Tx) power. Also, the application logs the received signal strength indicator (RSSI) of each Bluetooth advertising data packet detected. Then the application is able to store all the needed information about each iBeacon in a linked list data structure. Additionally, if the application detects another Bluetooth advertising data packet from an iBeacon that it has already detected, it will add the new RSSI value to all of the RSSI values that are associated with that iBeacon.

The mobile application does have multiple features and functions that can be used during the data gathering phase, as shown in Figure 5. The combo box (spinner) at the bottom left of the application is used to set the room name (or general location of the mobile device) when gathering data, which is appended to the logging information. The two text fields (X Position and Y Position) allow the Cartesian coordinates to be entered for the mobile device location within a building when gathering data. The coordinates information is also appended to the logging information. The two buttons either side of the text files can be used to increment and decrement the respective X or Y coordinate values. We should also note that using the Background Switch the user can enable background scanning, meaning that when the application is running in the background it

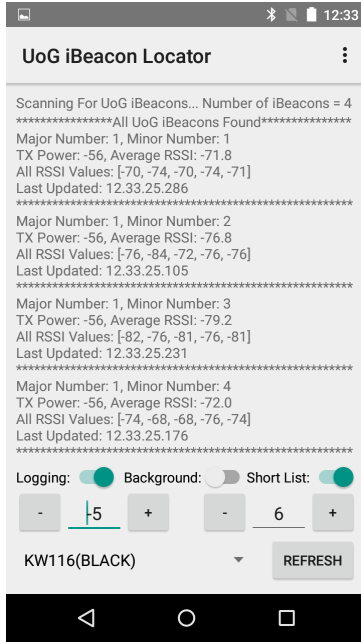


Figure 5. User Interface of the mobile application

will still continue to scan and log Bluetooth advertising data packets being transmitted from the iBeacons.

When run in normal operation mode, the mobile application disables the logging functionality. It simply receives BLE messages from multiple beacons and sends their RSSI values and respective beacon IDs to the server.

3.4. Server

As discussed in Section 3.1, the server is responsible for processing the data sent by the mobile phone and calculate the building occupancy. When operating in normal mode, the server receives information from a mobile device and uses a trained classifier to update the building occupancy estimation. With respect to the initial data gathering phase, it is not necessary for the training of the classifier to take place in the server. The only requirement is the result of the training (i.e. the trained classifier model) to be stored on the server so that it can be used during normal operation.

4. Experimental evaluation

This section elaborates on the experimental methodology we adopted and the experimental results we produced. We first give a description of the method we followed for our experiments and we continue by presenting our classification approach and related results.

4.1. Design and methodology

We have conducted our experiments inside the computer laboratory of the University of Greenwich. We have run two

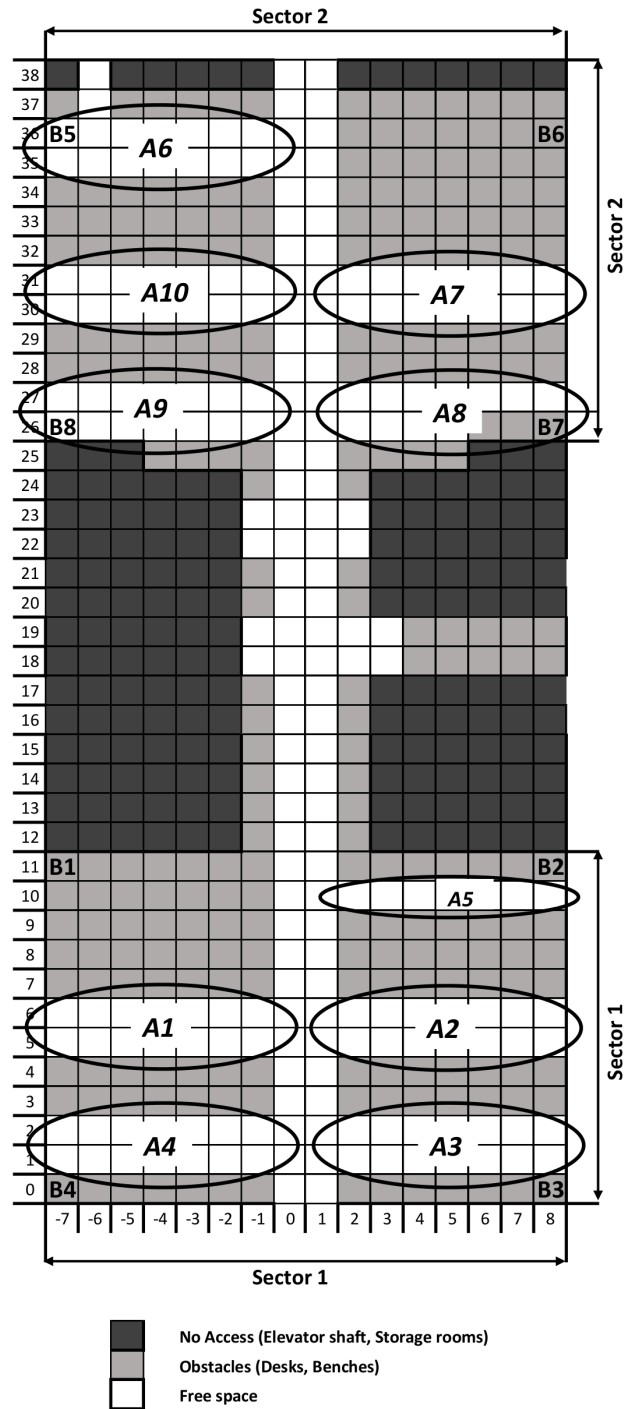


Figure 6. Floorplan of experiment area and beacon locations

independent experiments, in the two different sectors of the laboratory. This is essentially an office space comprising objects such as desks, benches, computers, panels and chairs. We have used an orthogonal grid to map the experimental area, as shown in Figure 6. The dimensions of each grid

square are $1\text{m} \times 1\text{m}$.

In each sector, we identified five areas for which we would estimate their occupancy. Table 3 illustrates the experimental setup for each laboratory sector.

TABLE 3. EXPERIMENTAL SETUP

Sector	Areas	Beacons
1	A1, A2, A3, A4, A5	B1, B2, B3, B4
2	A6, A7, A8, A9, A10	B5, B6, B7, B8

For the data gathering phase, we used the mobile application while walking along the grid. While gathering data, we used the mobile application’s labeling functionality to label the incoming BLE packets. More specifically, for each BLE packet received we logged the respective unique ID and RSSI and we assigned an area label (A1 to A10) based on our actual location. The beacons’ transmission frequency was set to 8 Hz. The walking pattern we chose involved spending 10 seconds on each grid point before moving to the next one. We conducted two runs of this data gathering phase. This approach allows us to gather data for all the locations inside each of the areas A1 to A10, while also taking into account the movement of occupants.

4.2. Classification based occupancy estimation

We used our dataset to train a classifier for the five areas in each of the two independent sectors. As we mentioned in Section 4.1, our raw dataset contained individual packets coming from specific beacon IDs, with a respective RSSI value and a label for each of them (depending on the area they were received in). Since each sector contains four beacons, as shown in Table 3, a data point describing an area contains the four RSSI values for each of the beacons. To construct each data point of our training set, we used a windowing approach and for each beacon we calculated the average and the standard deviation of its RSSI over the window samples. This resulted in eight features per data point. As we will see in Sections 4.2.1 and 4.2.2 we used three different window sizes (5, 10 and 20 samples). The bigger the size of the window, the longer the time needed to construct one data point. In our experiment gathering 10 samples takes 1.25 seconds on average.

We partitioned the dataset into 80% training set and 20% test set and used 10-fold cross validation for hyperparameter tuning. Our dataset size varies with the window size. More specifically, the dataset size for Sector 1 was 1677, 835 and 414 data points for window sizes of 5, 10 and 20 samples respectively. The corresponding dataset sizes for Sector 2 were 1700, 866 and 430 data points. We have trained three different classifiers: Support Vector Machines with Radial Basis Function kernel (SVM), Logistic Regression (LR) and K-Nearest Neighbours (KNN). The number of neighbours for the KNN classifier differs among our experiments, since it was optimised for each experiment using cross-validation.

One performance metric for evaluating a classifier is its accuracy, which is defined as the number of correct predictions over the total number of predictions. However,

accuracy in general does not optimally define the performance of a classifier since it is negatively affected by class imbalance. Although the methodology we followed to create our dataset inherently minimises class imbalance (since we spent an equal amount of time in each of the areas while performing data collection and labelling), we decided to use a confusion matrix to better describe the performance of each classifier.

In our confusion matrix, **each row represents the instances in an actual class** and **each column represents the instances in a predicted class**. The diagonal elements represent the number of instances for which the predicted label is equal to the true label. Off-diagonal elements represent instances that are misclassified. The higher the diagonal values of the confusion matrix the better. We have chosen to show the confusion matrices normalised by the number of elements in each class. In case of class imbalance, this approach better illustrates which classes are being misclassified. To improve visualisation, we have colour-coded the matrices by assigning black to 1.0 (100%) and white to 0.0 (0%).

4.2.1. Experimental results for Sector 1. Tables 4 to 12 show the experimental results for the areas belonging to Sector 1, for different values of the window size and different classification algorithms. By inspecting the confusion matrices, we observe that the Logistic Regression (LR) classifier is exhibiting the worst performance. More specifically, the classification accuracy of the LR classifier for area A2 is consistently low, with values ranging from 0.4 to 0.57. Moreover, area A2 is mostly mistaken for area A3 or A5 by the LR classifier. This can be explained if we inspect the floorplan diagram in Figure 6. As we can see, beacons are located in the four corners of Sector 1 and each of them can be associated with a "closest" area (B1 with A1, B2 with A5, B3 with A3 and B4 with A4). However, area A2 is "between" beacons B2 and B3. This deployment setup affects the performance of the LR classifier. Looking at the other two classifiers, SVM and KNN, we see that their performance is considerably better. This can be explained by observing that KNN and SVM (with Radial Basis function kernel) are non linear classifiers, while Logistic Regression is a linear classifier. Consequently, when the data is non linearly separable the predictions of Logistic Regression are less accurate. Finally, we can observe that the performance of the classifiers improves when we increase the window size to 10. As we discussed in Section 4.2, a bigger window size results in averaging the RSSI values over a bigger time interval. However, further increasing the value of the window to 20, does not improve the classification results. Among all classifiers and experimental configurations the best classification results are given by the SVM classifier for a window size of 10, as depicted in Table 7.

4.2.2. Experimental results for Sector 2. Tables 13 to 21 summarise the experimental results for the areas belonging to Sector 2, for different values of the window size and different classification algorithms. We can observe again

	A1	A2	A3	A4	A5
A1	0.87	0	0	0.07	0.06
A2	0.01	0.71	0.12	0	0.15
A3	0.08	0.11	0.77	0.05	0
A4	0.11	0.02	0	0.88	0
A5	0.06	0.15	0	0.01	0.77

TABLE 4. AREAS 1-5, SVM, WINDOW = 5

	A1	A2	A3	A4	A5
A1	0.93	0	0	0.07	0
A2	0.07	0.73	0.2	0	0
A3	0	0.05	0.95	0	0
A4	0.06	0	0.06	0.88	0
A5	0.16	0.11	0	0	0.74

TABLE 10. AREAS 1-5, SVM, WINDOW = 20

	A1	A2	A3	A4	A5
A1	0.93	0.01	0	0.03	0.03
A2	0.01	0.74	0.12	0	0.12
A3	0.05	0.24	0.64	0.06	0.02
A4	0.18	0.02	0	0.81	0
A5	0.07	0.18	0	0	0.75

TABLE 5. AREAS 1-5, KNN(k=11), WINDOW = 5

	A1	A2	A3	A4	A5
A1	0.93	0	0	0.07	0
A2	0.07	0.93	0	0	0
A3	0.05	0.11	0.84	0	0
A4	0.06	0	0.12	0.81	0
A5	0.16	0.11	0	0	0.74

TABLE 11. AREAS 1-5, KNN(k=3), WINDOW = 20

	A1	A2	A3	A4	A5
A1	0.73	0.01	0	0.19	0.07
A2	0.07	0.4	0.26	0	0.26
A3	0	0.06	0.79	0.14	0.02
A4	0.05	0	0	0.93	0.02
A5	0.14	0.04	0	0.01	0.8

TABLE 6. AREAS 1-5, LR, WINDOW = 5

	A1	A2	A3	A4	A5
A1	0.86	0	0	0.14	0
A2	0.07	0.47	0.4	0	0.07
A3	0.05	0.11	0.84	0	0
A4	0.06	0	0.06	0.88	0
A5	0.21	0	0	0	0.79

TABLE 12. AREAS 1-5, LR, WINDOW = 20

	A1	A2	A3	A4	A5
A1	0.94	0.03	0	0.03	0
A2	0	0.93	0.04	0	0.04
A3	0	0	1	0	0
A4	0.14	0	0	0.86	0
A5	0.09	0.09	0	0	0.83

TABLE 7. AREAS 1-5, SVM, WINDOW = 10

	A1	A2	A3	A4	A5
A1	0.87	0.06	0	0.06	0
A2	0	0.82	0.11	0	0.07
A3	0	0.08	0.92	0	0
A4	0.2	0.03	0	0.77	0
A5	0.06	0.13	0	0.02	0.79

TABLE 8. AREAS 1-5, KNN(k=7), WINDOW = 10

	A1	A2	A3	A4	A5
A1	0.81	0	0	0.16	0.03
A2	0	0.54	0.36	0	0.11
A3	0	0	1	0	0
A4	0.09	0	0	0.91	0
A5	0.17	0.04	0	0	0.79

TABLE 9. AREAS 1-5, LR, WINDOW = 10

performance. The classification accuracy of the LR classifier for area A10 is consistently low, with values ranging from 0.43 to 0.56. Moreover, area A10 is mostly mistaken for area A6 or A9 by the LR classifier. Again, this can be explained if we inspect the floorplan diagram in Figure 6. As we can see, beacons are located in the four corners of Sector 2 and each of them can be associated with a "closest" area (B5 with A6, B6 with A7, B7 with A8 and B8 with A9). However, area A10 is "between" beacons B5 and B8. This deployment setup affects the performance of the LR classifier. Looking at the other two classifiers, SVM and KNN, we see that their performance is again considerably better. As we discussed in Section 4.2.1, this behaviour is related to the fact that these classifiers can better handle non-linearly separable data. Finally, we also observe that the performance of the classifiers improves when we increase the window size to 10. However, further increasing the value of the window to 20, does not improve the classification results. The best classification results among all classifiers and experimental configurations are given by the SVM classifier for a window size of 10, as depicted in Table 16.

	A6	A7	A8	A9	A10
A6	0.88	0.03	0.02	0	0.07
A7	0.09	0.8	0.08	0.02	0.02
A8	0.04	0.05	0.82	0.05	0.04
A9	0.01	0.01	0.03	0.86	0.08
A10	0.14	0	0.02	0.14	0.7

TABLE 13. AREAS 6-10, SVM, WINDOW = 5

a similar pattern in results, as the one in Section 4.2.1. More specifically, the confusion matrices indicate that the Logistic Regression (LR) classifier is exhibiting the worst

	A6	A7	A8	A9	A10
A6	0.87	0.04	0.02	0	0.07
A7	0.09	0.85	0.05	0.02	0
A8	0.05	0.05	0.8	0.04	0.05
A9	0	0	0.07	0.82	0.11
A10	0.13	0.02	0.02	0.13	0.71

TABLE 14. AREAS 6-10, KNN($\kappa=6$), WINDOW = 5

	A6	A7	A8	A9	A10
A6	0.94	0.06	0	0	0
A7	0	1	0	0	0
A8	0.11	0.06	0.67	0.11	0.06
A9	0	0	0.14	0.86	0
A10	0.12	0	0	0.19	0.69

TABLE 20. AREAS 6-10, KNN($\kappa=6$), WINDOW = 20

	A6	A7	A8	A9	A10
A6	0.87	0.07	0.01	0.01	0.04
A7	0.12	0.7	0.15	0.02	0.02
A8	0	0.11	0.75	0.11	0.04
A9	0	0.01	0.11	0.79	0.08
A10	0.32	0	0.02	0.24	0.43

TABLE 15. AREAS 6-10, LR, WINDOW = 5

	A6	A7	A8	A9	A10
A6	0.88	0.12	0	0	0
A7	0	1	0	0	0
A8	0.06	0.11	0.67	0.11	0.06
A9	0	0	0.05	0.91	0.05
A10	0.19	0	0.06	0.19	0.56

TABLE 21. AREAS 6-10, LR, WINDOW = 20

	A6	A7	A8	A9	A10
A6	0.78	0.11	0.02	0	0.09
A7	0	0.93	0.03	0	0.03
A8	0	0	0.96	0	0.04
A9	0.03	0	0.06	0.88	0.03
A10	0.03	0	0.03	0.08	0.87

TABLE 16. AREAS 6-10, SVM, WINDOW = 10

	A6	A7	A8	A9	A10
A6	0.8	0.16	0.02	0	0.02
A7	0.07	0.93	0	0	0
A8	0	0	0.96	0	0.04
A9	0	0	0.03	0.94	0.03
A10	0.05	0	0.08	0.05	0.82

TABLE 17. AREAS 6-10, KNN($\kappa=10$), WINDOW = 10

	A6	A7	A8	A9	A10
A6	0.76	0.13	0	0	0.11
A7	0.03	0.87	0.07	0	0.03
A8	0	0	0.96	0.04	0
A9	0.03	0	0.06	0.88	0.03
A10	0.28	0	0.03	0.21	0.49

TABLE 18. AREAS 6-10, LR, WINDOW = 10

	A6	A7	A8	A9	A10
A6	0.94	0.06	0	0	0
A7	0	1	0	0	0
A8	0	0	0.83	0.06	0.11
A9	0	0	0.09	0.82	0.09
A10	0.06	0	0.06	0.06	0.81

TABLE 19. AREAS 6-10, SVM, WINDOW = 20

4.3. Energy consumption

To evaluate our approach in terms of energy consumption, we have also logged the battery level of the mobile phone during the data gathering phase. The results for each experiment are shown in Figures 7 and 8. We used three different values for the beacon transmission frequency, namely 2 Hz, 4 Hz and 8 Hz. As we can observe, a higher transmission frequency results in higher energy consumption, as indicated by the faster discharge of the phone's battery. This is expected, since the mobile phone receives and processes a higher number of packets per unit time as the transmission frequency of the beacons increases. More specifically, for the Sector 1 experiment, the highest transmission frequency results in the battery level dropping by 8% at the end of the experiment, while the corresponding number for the lowest transmission frequency is 5%. Although this is a 37.5% difference, the total remaining battery power after 40 minutes is over 90% for both cases, which highlights the energy efficiency of BLE. For the Sector 2 experiment (Figure 8), the battery level drops by 9% for the highest frequency and by 6% for the lowest frequency. This is a 33% difference, but again the battery level after 40 minutes is again over 90%. The similarity between the energy consumption for the two experiments is an important factor, since it allows us to predict how long a battery operated phone can operate while our system is in use.

5. Conclusions and Future Work

We have proposed an approach for inferring occupancy based on BLE beacons installed inside a building and communicating with a mobile application. Our approach does not require any localisation processing to take place on the mobile phone, as a server running in a remote control center is responsible for processing the data sent by the mobile phone and calculating the building occupancy using a classifier. This is achieved by conducting an initial data gathering phase for the system during which the data gathered are

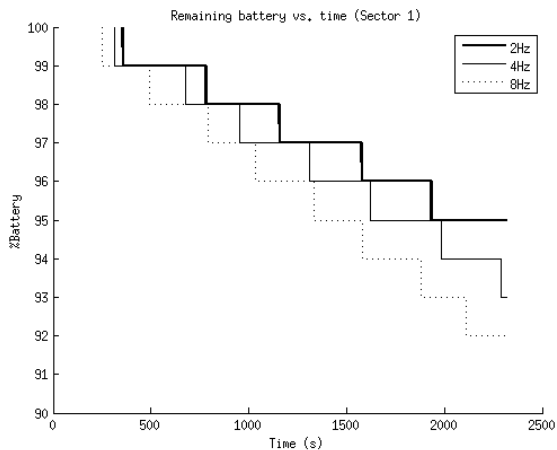


Figure 7. Mobile phone battery level Vs. Time

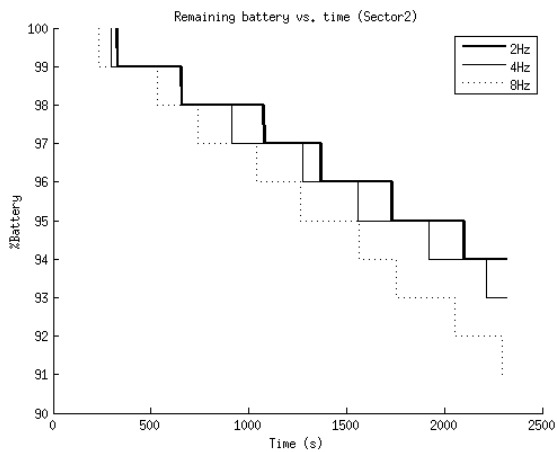


Figure 8. Mobile phone battery level Vs. Time

used to train the classifier. We have evaluated our approach in a real-world experiment involving a computer laboratory space. Our experimental results show that the combination of machine learning techniques with BLE is a promising approach for estimating a building's occupancy.

In future work we will study the performance of our system for different beacon deployment densities. We also plan on further investigating the effect of beacon transmission frequency on the occupancy estimation accuracy, as well as the effect of various walking speeds and patterns during the system's training phase. Finally, we will extend the range of machine learning algorithms that we employ and will study the use of neural networks and deep learning. We believe that such methods are worth investigating in view of the importance of efficient emergency management.

References

- [1] N. Dimakis, A. Filippoupolitis, and E. Gelenbe, "Distributed building evacuation simulator for smart emergency management," *The Computer Journal*, vol. 53, no. 9, pp. 1384–1400, 2010.
- [2] S. Timotheou and G. Loukas, "Autonomous networked robots for the establishment of wireless communication in uncertain emergency response scenarios," in *Proceedings of the 2009 ACM symposium on Applied Computing*. ACM, 2009, pp. 1171–1175.
- [3] A. Filippoupolitis, G. Gorbil, and E. Gelenbe, "Spatial computers for emergency support," *The Computer Journal*, vol. 56, no. 12, pp. 1399–1416, 2012.
- [4] F. Santoso and S. J. Redmond, "Indoor location-aware medical systems for smart homecare and telehealth monitoring: state-of-the-art," *Physiological measurement*, vol. 36, no. 10, p. R53, 2015.
- [5] K. Sugino, S. Katayama, Y. Niwa, S. Shiramatsu, T. Ozono, and T. Shintani, "A bluetooth-based device-free motion detector for a remote elder care support system," in *Advanced Applied Informatics (IIAI-AAI), 2015 IIAI 4th International Congress on*. IEEE, 2015, pp. 91–96.
- [6] A. Fujihara and T. Yanagizawa, "Proposing an extended ibeacon system for indoor route guidance," in *Intelligent Networking and Collaborative Systems (INCOS), 2015 International Conference on*. IEEE, 2015, pp. 31–37.
- [7] M. Alam, N. Pathak, and N. Roy, "Mobeacon: an ibeacon-assisted smartphone-based real time activity recognition framework," in *Proceedings of the 12th international conference on mobile and ubiquitous systems: computing, networking and services*, 2015.
- [8] A. Corna, L. Fontana, A. Nacci, and D. Sciuto, "Occupancy detection via ibeacon on android devices for smart building management," in *Proceedings of the 2015 Design, Automation & Test in Europe Conference & Exhibition*. EDA Consortium, 2015, pp. 629–632.
- [9] M. Choi, W.-K. Park, and I. Lee, "Smart office energy management system using bluetooth low energy based beacons and a mobile app," in *Consumer Electronics (ICCE), 2015 IEEE International Conference on*. IEEE, 2015, pp. 501–502.
- [10] G. Conte, M. De Marchi, A. A. Nacci, V. Rana, and D. Sciuto, "Bluesentinel: a first approach using ibeacon for an energy efficient occupancy detection system," in *BuildSys@ SenSys, 2014*, pp. 11–19.
- [11] X.-Y. Lin, T.-W. Ho, C.-C. Fang, Z.-S. Yen, B.-J. Yang, and F. Lai, "A mobile indoor positioning system based on ibeacon technology," in *Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE*. IEEE, 2015, pp. 4970–4973.
- [12] S. Kajioka, T. Mori, T. Uchiya, I. Takumi, and H. Matsuo, "Experiment of indoor position presumption based on rssi of bluetooth le beacon," in *Consumer Electronics (GCCE), 2014 IEEE 3rd Global Conference on*. IEEE, 2014, pp. 337–339.
- [13] F. Palumbo, P. Barsocchi, S. Chessa, and J. C. Augusto, "A stigmergic approach to indoor localization using bluetooth low energy beacons," in *Advanced Video and Signal Based Surveillance (AVSS), 2015 12th IEEE International Conference on*. IEEE, 2015, pp. 1–6.
- [14] R. Faragher and R. Harle, "Location fingerprinting with bluetooth low energy beacons," *Selected Areas in Communications, IEEE Journal on*, vol. 33, no. 11, pp. 2418–2428, 2015.
- [15] G. Ionescu, C. M. de la Osa, and M. Deriaz, "Improving distance estimation in object localisation with bluetooth low energy," *SENSORCOMM*, vol. 2014, pp. 45–50, 2014.
- [16] G. G. Anagnostopoulos and M. Deriaz, "Accuracy enhancements in indoor localization with the weighted average technique," *SENSORCOMM*, vol. 2014, pp. 112–116, 2014.