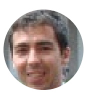
BlueSentinel: a first approach using iBeacon for an energy efficient occupancy detection system

CONFERENCE PAPER · NOVEMBER 2015

5 AUTHORS, INCLUDING:


Giorgio Conte
Politecnico di Milano

Massimo De Marchi
Politecnico di Milano


Alessandro Antonio Nacci
Politecnico di Milano

Vincenzo Rana
École Polytechnique Fédérale de Lausanne

BlueSentinel: a first approach using iBeacon for an energy efficient occupancy detection system

Giorgio Conte
Politecnico di Milano
P.za Leonardo Da Vinci
Milano, Italy
giorgio1.conte@mail.polimi.it

Massimo De Marchi
Politecnico di Milano
P.za Leonardo Da Vinci
Milano, Italy
massimo1.demarchi@mail.polimi.it

Alessandro A. Nacci
Politecnico di Milano
P.za Leonardo Da Vinci
Milano, Italy
alessandro.nacci@polimi.it

Vincenzo Rana
Politecnico di Milano
P.za Leonardo Da Vinci
Milano, Italy
vincenzo.rana@polimi.it

Donatella Sciuto
Politecnico di Milano
P.za Leonardo Da Vinci
Milano, Italy
donatella.sciuto@polimi.it

Abstract

In the last years, the concept of *smart buildings* has been proposed and proved to be an effective solution to tackle the problem of reducing the power consumption of complex (both residential and commercial) buildings, while providing the users with a very high level of comfort. In this context, knowing the exact position of users inside the buildings has been identified as a needed feature to optimize the behavior of the building itself. Recently, using the occupants mobile devices as sensors has been validated as an effective solution to have accurate occupancy detection systems, even if no energy efficient solution in therm of battery consumption has been found so far. On the contrary, with this work, we present *BLUE-SENTINEL*, an accurate and power efficient method to identify the occupants of each room of a smart building using mobile devices as source of information. The proposed approach faces the occupancy detection problem with a good accuracy by exploiting iBeacon, a very recent low-power technology proposed by Apple. In particular, since the iBeacon protocol is built upon Bluetooth Low Energy (BLE), it represents a very highly power-efficient solution. In addition to this, the iBeacon technology is characterized by a good level of compatibility and portability, supporting both iOS- and Android-based devices. The proposed approach has been validated in a real environment with a prototype system released as open source showing how this technology is suitable for the occupancy detection in a smart building.

Categories and Subject Descriptors

I.5 [**Pattern Recognition**]: Design Methodology—*Classifier design and evaluation*; C.3 [**Special-Purpose and application-based systems**]: Real-time and embedded systems

General Terms

Performance, Experimentation

BuildSys'14, November 5–6, 2014, Memphis, TN, USA.

ACM 978-1-4503-3143-2/14/11 ...$15.00
http://dx.doi.org/10.1145/2676061.2674078

Keywords

Positioning Systems, Indoor Localization, iBeacon, Bluetooth Low Energy, Classification Algorithms

1 Introduction

The term *smart buildings* is generally used to identify places where sensors and actuators are used to increment the power-efficiency of the buildings themselves and/or the comfort of their occupants [34]. Their spreading is continuously increasing over the years, as the effort of the researchers to better exploit this acquired intelligence. In this context, it is possible to find two different approaches (that can be however mixed together) to the definition of the policies/rules the *smart buildings* have to follow:

- the building manager could be in charge of setting all the policies that govern the building behavior [12];
- the users/occupants could directly interact with the *smart building* by specifying their personal rules, thus bringing a more fine-grained definition of the behavior of the building at the cost of a much more complex management system (e.g., conflicts resolution) [31].

In both cases, identifying the occupants of a building and retrieving their location within the building is fundamental [8]: in the first case, for instance, to let the building manager specify policies that involve the number of people in a room, while in the second case to select the set of rules that have to be activated for a particular room (the ones belonging to the occupants of the room itself). On the one hand, knowing the number of people in a room is especially important when considering that simply heating/cooling rooms only when they are not empty has a considerable impact on the overall energy consumption. In fact, as stated in [1], Heating, Ventilation and Air Conditioning (HVAC) systems contribute, in the US smart buildings, for 39.6% of the entire energy usage. On the other hand, both the energy efficiency and the level of comfort of a building could be greatly increased by exploiting information on who is in its room, in addition to the number of occupants of each room. In this way, in fact, the behavior of the building could be easily tailored to a specific target (represented by the occupants of a particular room), thus augmenting the flexibility of the whole building, while avoiding energy wastes.

The problem of retrieving the location of users in a building (including information about when they enter in, or exit from, a room) is known as the *occupancy detection problem* [16]. In this paper, we propose to solve it with *BLUE-SENTINEL*, which exploits the iBeacon technology recently released by Apple [6]. In particular, iBeacon has been originally developed to enable the definition of indoor proximity systems, thus it has not been specifically designed to solve the occupancy detection problem. We show how it is possible to adapt iBeacon to detect the number of users in a room, and how it can be used to gather information about their movements (thus identifying and tracking them) inside the building. We also compare the proposed solution with other approaches, described in Section 2, that can be found in the literature and that deal with the same problem.

A main advantage of the proposed solution over the state-of-the-art ones is the client-side power efficiency, since the iBeacon protocol is built upon Bluetooth Low Energy (BLE), a highly power efficient version of the Bluetooth standard protocol. The implementation we have used to gather the experimental results is based on iOS devices, but it could be easily ported (with a very limited effort) on other platforms, such as the Android-based ones (some free implementation are in fact available online, like the one implemented by *Radius Networks* [2]).

The paper is structured as follows. In Section 2 we present on overview of the methodologies and of the systems that can be found in literature and that are related with the occupancy detection problem. Then, a brief description of how the iBeacon protocol works is given in Section 3. The proposed approach and the implementation details are presented in Section 4, while Section 5 shows the experiments we have performed and the results we have obtained. Finally, we will highlight the current limitation of the proposed system; we will conclude then with some final remarks.

2 Related works

The occupancy detection is a well know problem in the academic literature. However, despite numerous works have been conducted to find a cheap, simple (mainly in terms of number and kind of hardware devices needed), power-efficient and reliable solution to this issue, the problem is still open and an *optimal* solution, satisfying the three constrains at the same time, still has to been found.

The main differences among the state-of-the-art occupancy detection systems reside in the different technologies they exploit to reach their goal. For instance, one of the first work addressing this issue is [32], which proposes **infrared sensors** as a suitable technology to develop an occupancy detection system. However, this solution requires both the users and the buildings to equip extra devices to make the system work. On the one hand, users must wear an active badge that broadcasts a unique identifier, while, on the other hand, a quite huge number of infrared sensors must be placed all around the target building. In addition to this, the accuracy of this solution in crowded rooms is definitely poor, mainly because of the high number of collisions. Also for this reason, the infrared technology has been basically abandoned and new solutions and protocols have been proposed, even though the underlying architecture has not experienced substantial modifications.

As an example, more recent approaches, such as FastSlam [18] and Landmarc [27], exploit the new **RFID** technology. Similarly to the previously described solution, these works require the displacement of several antennas in the space to be monitored, since the working range of each one of them is only around 6 meters. Furthermore, all the users have to be provided with passive receivers (the tags) so that they can be correctly identified when they pass close to one of the antennas. In fact, whenever a tag enters the range of an antenna, it receives the signal, it activates and it replies by sending back to the antenna a unique identifier.

A different technology is proposed in [33], which proposes a system based on the measure of times-of-flight of **ultrasound pulses**. The transmitters are generally carried (worn) by the users, while the receivers are placed at known positions all around the place to be monitored. Unluckily, ultrasounds are severely biased by human bodies (Dolphin [20] proposed an alternative way to build receivers able to achieve a better average accuracy). However, the weakest part of the whole system is that the exact position of each antenna must be known, otherwise it is impossible to accurately locate users. Conversely, the approach we propose in this paper does not require such kind of information, since it can learn the fingerprints of the target rooms during a training phase.

In [28], the **GSM** technology has been exploited to find a simple and cheap solution to the occupancy detection problem. The main advantages of this approach rely on the fact that GSM antennas are built by mobile telecommunication providers and almost all users are already equipped with a GSM device (usually their phone or smartphone). However, this approach also presents some relevant drawbacks, such as the very low reliability of the obtained results. The authors performed several test and showed that the accuracy of their solution can range from 57% to 97%, thus presenting a very high variance.

The best results, in terms of accuracy, for the indoor localization of a set of users, have been obtained with approaches exploiting **Wi-Fi networks**, such as Ariel [21], RedPin [9], [24], [8] and UMLI [26]. Currently, these systems are among the ones characterized by the highest levels of reliability. For instance, UMLI can achieve an accuracy of 99.84%, without requiring any training phase (the decision algorithm employed, based on well-known clustering techniques, is unsupervised). However, the power consumption (especially the client-side one) has not been explicitly taken into account in these approaches, even though the authors of Ariel underline that their algorithm aim at reducing the energy usage on the users devices.

Differently from the previously described approaches, PlaceLab [22] is an interesting work that exploits more than one technology to locate users in a region of interest. However, this approach mainly focuses on a context (e.g., open spaces such as university campuses) that is slightly different with respect to the one considered in our work.

For what concerns **Bluetooth**, the most interesting approaches exploiting this protocol are [29] and BIPS [7]. As reported by [23], Bluetooth allows communication to take place with a lower power consumption with respect to Wi-Fi, especially in applications with low data rate (usually small bursts), and thus it is generally preferred in all the contexts in which battery life is a major concern. In particular, [29] is a very interesting work in which the authors use a K-Nearest Neighbor Algorithm (KNN) algorithm to correctly predict the location of the users. The training data is not completely collected by means of physical experiments, but it is created with a Weibull distribution. In order to tune the distribution and to set the values of the system parameters, only few samples are gathered on the real infrastructure. Unfortunately, the accuracy of this approach is much worse than the one of the works exploiting Wi-Fi networks. The authors justify these results by reporting that the number of Bluetooth APs was much lower than the number of Wi-Fi APs.

In Table 1 we have summarized the main characteristics of the state-of-the-art approaches, also considering if a particular technological solution requires extra devices for the user (e.g., tags or receivers) or for the buildings (e.g., antennas or sensors). None of the existing works exploits the latest bluetooth technology, called Bluetooth Low Energy (BLE), which has been developed with the purpose of enabling low-energy data communications. Our aim is to explore this new technology in order to understand if and how it can be exploited to solve, with an acceptable accuracy and an intrinsic low power consumption, the occupancy detection problem, highlighting both the positive and negative aspects.

3 Apple iBeacon protocol

In the WorldWide Developers Conference (WWDC) 2013 event, Apple announced *iBeacon*, a technology that enables a device

Table 1. Main characteristics of the state-of-the-art approaches

Technology	Approaches	Extra device user-side	Extra devices building-side	High accuracy	High power efficiency
Infrared	[32]	Yes	Yes	No	N. A.
RFID	[18] [27]	Yes	Yes	No	N. A.
Ultrasound	[33] [20]	Yes	Yes	No	N. A.
GSM	[28]	No	No	No	Yes
Wi-Fi	[21] [9] [24] [8] [26]	No	No	Yes	No
Classic Bluetooth	[29] [7]	No	Yes	No	N. A.
BLE	Proposed Approach	No	Yes (few)	Yes	Yes

(called *beacon*) to send push notifications to nearby iOS devices [6] (even though the same technology can be used also with other kinds of devices, such as the Android-based ones). A potential application of this new protocol consists in the displacement of location-aware and context-aware wireless sensors that could point out users location. The iBeacon protocol works on Bluetooth Low Energy (BLE), a Bluetooth SIG standard introduced to provide low power consumption and cost while maintaining similar communication range of classical bluetooth. As stated in [17], BLE can benefit from the widespread use of the Bluetooth technology, because it could be easily integrated into classic Bluetooth architectures. Thus, BLE is expected to be used by billions of smartphones in the near future. BLE is optimized, in terms of energy consumption, for burst data communications, in other words, BLE behaves very well when the messages that have to be exchanged are short, such as identifiers or status messages [23].

Each BLE device that advertises information using the iBeacon protocol is identified by the following values, which combination represents a beacon region:

- **proximity UUID** (Universally Unique IDentifier): a 128 bit value that identifies a beacon region (that can be composed of many beacons);
- **major value**: a 16-bit unsigned integer that can be used to group related beacons that have the same proximity UUID;
- **minor value**: a 16-bit unsigned integer that differentiates beacons with the same proximity UUID and major value.

Both the major and the minor values are optional, while the proximity UUID is mandatory. This structure is quite flexible, since a region could be associated with only the UUID, a combination of the UUID and the major value or with all the three parameters. As an example, it is possible to identify all the beacons within the same building by assigning to them the same proximity UUID, then using the major value to identify all the beacons within the same room and finally the minor value to distinguish each beacon.

Since the release of iOS 7, Apple has provided iBeacon functionalities in a framework called CoreLocation [4]. This framework embeds both beacons location and geolocation functionalities and has the purpose to offer application developers a simple and effective way to gather users location. Applications can use region monitoring to be notified when users enter (or exit) in (from) a beacon region (in addition, they can also evaluate the relative distance among the beacons). In iOS, regions associated with an application are continuously tracked, including when the application itself is not running; if a region boundary is crossed, the application is activated in background to handle the event.

A possible application of iBeacon is a store where users receive a welcome (plus, eventually, a set of special offers) whenever they enter the store. In this kind of applications, the accuracy in determining the location of the users is not critical, and even an approximation of several meters can be acceptable. Another application is the advertising of products: users could be updated with information related to the goods that are close to them while they visit a store. This is a fine-grained application, but it has to work (in real-time) only when the application seeking for products information is active, i.e. when the user is using the application in order to get more information about the products.

Unluckily, iBeacon technology has not been optimized by Apple for fine-grained applications when the iOS device is in standby mode, and a way to keep the monitoring active, with a high temporal resolution, when the device is in standby is basically missing. However, we can imagine a scenario in which the user is walking in a building, going to his office and passing by several rooms. When he arrives at his office the light needs to be turned on, with a maximum delay of 2-3 seconds, otherwise the automation will not be considered working and perceived as useful. Achieving this kind of behavior is not so easy, within the standards of the iBeacon technology. For what concern the accuracy, for instance, the system has to go continuously track the users, but with the iBeacon implementation currently available on iOS, beacons can be received by user devices only when the application is active. When the application is in background, it is woken up only to handle enter/exit region events, namely we can know only when the user device receives a region for the first time, or when it does not receive signal from a region for more than 30 seconds. When one of these two events is detected, the operating system runs the application for 4-5 seconds, during which it is possible to sense the beacons and obtain information about the power of the signal received from them. After this short period, the application is killed by the operating system and it will be reactivated only when the application exits from the previously detected region or when it enters in a new one. Between the two activations, the application cannot obtain information on the signal strength of the nearby beacons, thus leading to a very coarse-grained tracking of the users. This implies that it is not possible to make an accurate occupancy detection system exploiting the information coming from different beacons with different strength levels (indicating the distances, ranging from few centimeters to tens of meters, with respect to the source beacons).

4 Proposed approach

In this section we describe how the iBeacon protocol can be exploited to realize a cheap, simple, power efficient and reliable occupancy detection system. We then propose our solution, called *BLUE-SENTINEL* and we describe the overall hardware architecture, as well as the iOS applications we have developed. The source code of the entire system is available at `http://www.necst.it/energybox`.

4.1 Modification to the iBeacon protocol

In order to overcome the main limitations of iBeacon, described in the previous section, and to exploit all its potential for the solution of the occupancy detection problem, we need to introduce some modifications to the iBeacon protocol.

The main idea behind our implementation of the iBeacon protocol is to change the way the beacons advertise the region associated with them. In particular, instead of correlate each one of them with only a single region (similarly to what happens in the currently existing commercial solutions), we propose to let the beacons advertise more than one region, in a cyclic sequence, as depicted in Figure 1 and 2. Thanks to this, we are able to create *false* events that force the operating system to wake up the application more frequently (whenever we want). In particular, every time a beacon changes the region advertised, the device will receive a notification as the ones received when it enters in a new region, thus the associated application will be woken up and it will be able to obtain and process the data gathered from the beacons (related to their relative distance with respect to the user device). This, as shown in the following sections, allows us to realize an occupancy detection system characterized by high levels of accuracy and power efficiency, while guaranteeing the required responsive behavior. As shown in Section 4.2.2, in order to have a real system implementing this modified version of the iBeacon protocol, we have developed some custom iBeacon antennas using an Arduino board.

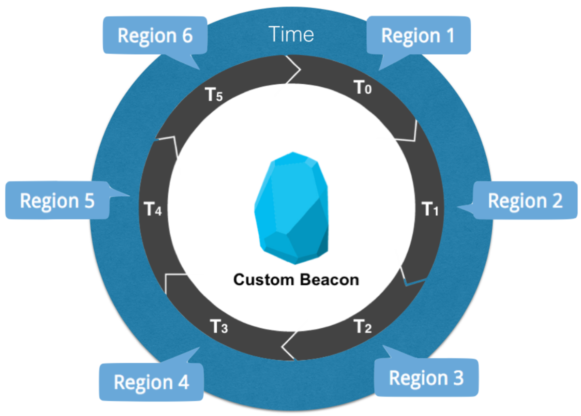


Figure 1. The same *BLUE-SENTINEL* beacon can advertise more than one region in a cyclic sequence

4.2 BLUE-SENTINEL system architecture

The *BLUE-SENTINEL* system has been implemented as a standard client-server architecture, shown in Figure 3, in which the information gathered from the sensors displaced on the buildings is collected and stored on a web application that offers services through RESTful interfaces. At run-time, the *BLUE-SENTINEL* system works as follow: whenever a new region is detected by the device, the *BLUE-SENTINEL* application starts collecting data from all the discovered beacons, storing all the useful information, such as the strength of the signals received. This data is sent, through an http request, to the *BLUE-SENTINEL* web application, which is in charge of processing it and elaborate the desired results.

4.2.1 BLUE-SENTINEL core

Figure 4 shows the modules that compose the core of the *BLUE-SENTINEL* system, which aims at handling raw data coming from the users smartphones and at producing as output the locations of the occupants within the smart building. This information can be then forwarded to other services, such as a Building Management System (BMS [35]). The *BLUE-SENTINEL* core manages all the users of the system, keeps track of the regions that are associated with the beacons that advertise them and collects data sent by occupants devices. It also provides some functionalities that are useful to automatically configure the iOS applications, which will be described in Section 4.2.4. The BlueSentinel core is based on Flask [5], which is a python micro-framework built to support RESTful applications.

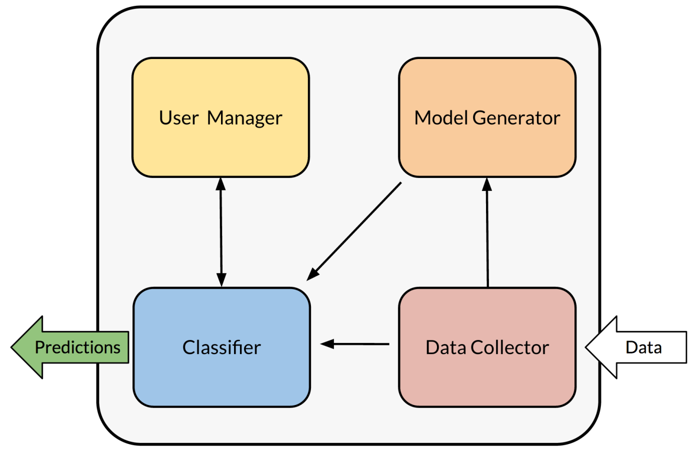


Figure 4. Modules composing the core of the *BLUE-SENTINEL* system

4.2.2 BLUE-SENTINEL arduino-based beacons

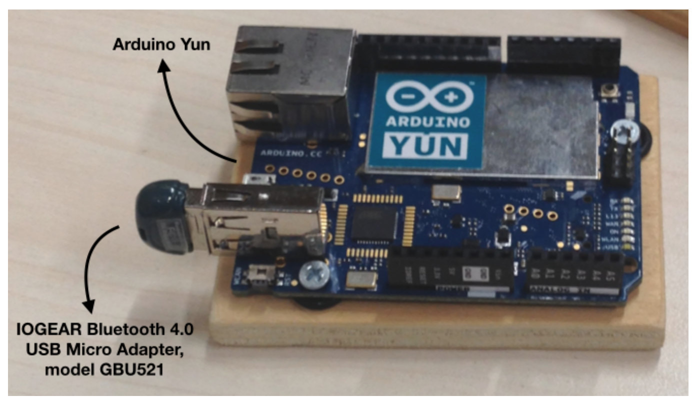


Figure 5. A photo representing the hardware platform used to implement the custom iBeacon antenna

The *BLUE-SENTINEL* beacons implement the modifications to the iBeacon protocol we have proposed and described in Section 4.1. As shown in Figure 5, the hardware of these custom beacons is based on an Arduino Yun plus a USB BLE dongle (the IOGEAR Bluetooth 4.0 USB Micro Adapter, model GBU521). The Arduino Yun consists of two main components, connected by means of a bridge:

- an ATmega32u4, a low-power microcontroller that supports the normal Arduino environment;
- an Atheros processor supporting Linino, a Linux distribution based on OpenWRT.

The USB dongle is connected to Linino, while the custom code we have written to implement the modifications described in Section 4.1 is executed by the ATmega32u4. This code communicates to the Linux distribution all the commands that have to be executed in order to advertise the different regions, as specified by the *BLUE-SENTINEL* protocol.

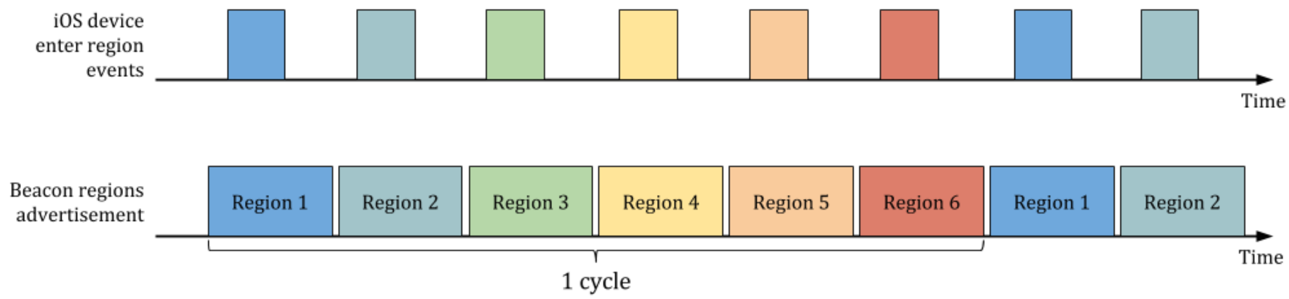


Figure 2. The behavior of the *BLUE-SENTINEL* beacon and the corresponding state of the *BLUE-SENTINEL* iOS application

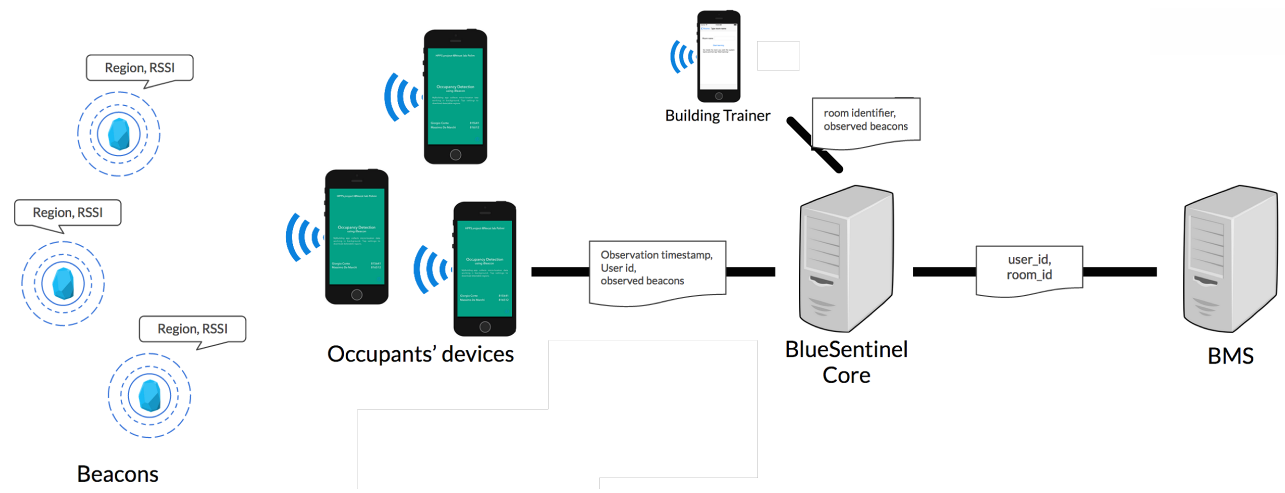


Figure 3. *BLUE-SENTINEL* architecture (including the interface with an external service such as a Building Management System)

4.2.3 BLUE-SENTINEL algorithms

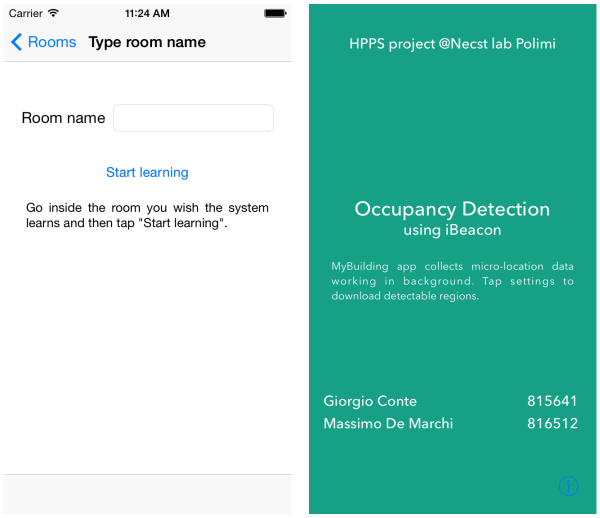
In the previous sections, we have described how it is possible to adapt the iBeacon protocol to our needs, in order to solve the occupancy detection problem while satisfying all the requirements (simplicity, power efficiency, accuracy, etc.), and how it is possible to implement all the proposed modifications in order to build the proposed *BLUE-SENTINEL* system, able to gather users location data.

Once these data have been collected, it is necessary to process it to extract useful information about the number, the identities and the locations of the occupants of the smart building. This task could be formalized as the problem of mapping each user/occupant to the room he/she is currently in, starting from the information sent to the *BLUE-SENTINEL* system by his/her smartphone about his/her relative distances with respect to the *BLUE-SENTINEL* beacons. We propose to solve this problem by means of a two-step application:

- first of all, we perform an offline phase (during the setup of the smart building) in which we collect both the label of each room to be monitored and the level of strength of all the *BLUE-SENTINEL* beacons in the range for each room. At the end of this phase, we build a model that will be used as input for the next step;
- finally, we compute at run-time the location of each user by exploiting the previously described model and the data gathered from his/her mobile device. We have decided to face this task by means of a classification algorithm, that tries to map each user to the best fitting room.

The choice of adopting a classification-based approach leads to several advantages. First of all, it considerably simplifies the problem of displacing the *BLUE-SENTINEL* beacons in the smart building. This is because it is not necessary to know the exact position of each one of them, since the *BLUE-SENTINEL* system automatically exploits their relative position. Thus, the displacement problem is directly reduced to the (considerably simpler) problem of finding a good coverage of the rooms to be monitored. On the other hand, the proposed classification-based approach is extremely flexible and independent from the structure of the building itself, thus it can be very easily adapted to different environments. Finally, the state of the art presents a huge amount of classification algorithms (instance-based, rules, trees, etc.) that are highly optimized for this kind of problems and that have already been proven to provide very good and reliable solutions, such as RedPin [9], which exploits an instance-based classification approach and is characterized by an accuracy close to 90%.

Among the different classification algorithms that can be found in the literature, we have decided to implement the following ones: k-Nearest Neighbors (KNN) [15] and trees [30]. On the one hand, KNN usually works very well with numerical data, while, on the other hand, trees are very efficient in terms of memory usage and computational resources needed, which are two fundamental aspects to be taken into account for the scalability and energy efficiency of the *BLUE-SENTINEL* system. The main difference between these two solutions is the following: while KNN requires a quite high computational effort for the second phase of our flow (the run-time mapping of the users to the rooms), since a complete scan of the entire dataset is needed, the solution based on trees re-



(a) Training application (b) Monitoring application

Figure 6. Figure (a) shows the application used during the training phase, while Figure (b) depicts the application that has to be installed on all the users smartphone devices in order to collect *BLUE-SENTINEL* beacons signals. While the training application requires the interaction with the person in charge of the setup of the smart building, the monitoring application has been designed to work in background

quires a higher computational effort for the offline construction of the model, but then it performs much better for what concerns the run-time mapping.

4.2.4 BLUE-SENTINEL iOS applications

The *BLUE-SENTINEL* system includes two different mobile applications:

- the first one, shown in Figure 6(a), is used during the offline training phase, in order to gather information about the displacement of the *BLUE-SENTINEL* beacons and the locations of the rooms to be monitored;
- the second one, shown in Figure 6(b), is the mobile application installed on all the devices of the users/occupants of the smart building, which is exploited to sense the *BLUE-SENTINEL* beacons and to transfer the collected data.

The first application is usually installed only on the smartphone of the person responsible for the setup and the offline analysis of the smart building (required to build the model). This application allows the user to specify a new room (characterized by an alphanumerical value), then it starts collecting data coming from the *BLUE-SENTINEL* beacons until it is manually stopped (usually few seconds are more than enough) and finally it sends the gathered data to the *BLUE-SENTINEL* server, which is responsible for the generation of the fingerprint of the room.

Conversely, the second application asks each user to login to the *BLUE-SENTINEL* system (in order to be able to identify them) and then it starts collecting data every time a *BLUE-SENTINEL* beacon advertises a new region identifier. Obviously, this application is kept in background and, once started by the user, it does not requires any further interaction.

5 Experimental results

In order to validate the proposed approach, we tested the precision of the algorithms described in Section 4.2.3 and the energy consumption of the iOS application that works as a location sensor inside the building. Knowing the accuracy of the classification algorithms is important to understand if the data gathered from the iBeacon antennas are useful to understand the occupancy of a building, while knowing the energy consumption of the iOS application tells us if the proposed approach is feasible (having a system that completely drains the occupant smartphone battery in a short time is something that must be avoided).

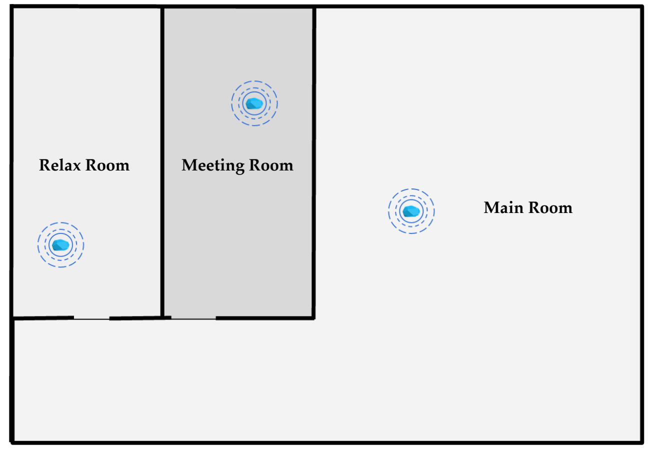


Figure 7. Schema representing the rooms we have monitored in our case study and the locations of the *BLUE-SENTINEL* beacons within them

5.1 Algorithms experimental evaluation

This section describes the experiments we have performed in order to validate the proposed approach and to evaluate its quality. First of all, we describe the experimental setup we have built in our laboratory. Then, we report the results we have obtained by employing the two algorithms introduced in Section 4. All the results on the classification algorithms we present in this section have been computed by using the data mining tool Weka [19].

5.1.1 Experimental setup

The case study we have built for our experiments consists of three different rooms (even though the proposed approach can be easily extended to an arbitrary number of rooms) and three *BLUE-SENTINEL* beacons: one in the main room, one in the meeting room and one in the relax room. In this way we wanted to tested the accuracy of the system with a minimum number of antennas (one per room). It must be noticed that using only three beacons in order to show the validity of the proposed approach is fair since achieving a good accuracy results even when a very limited number of beacons is employed is a lower bound case (increasing the number of antennas, it will be possible to increase the system accuracy). In order to simulate the presence of huge number of people, we entered the three different rooms many times in order to collect a big number of statistical samples. As shown in Figure 7, each beacon has been placed in the center of the room in order to maximize the coverage of the room itself.

We have conducted a large number of experiments in order to check if people inside (or outside) a room are correctly recognized by the *BLUE-SENTINEL* system and if the proposed approach can be effectively used to determine the number and the identities of the people inside a particular room.

5.1.2 K-Nearest Neighbors classification

This section reports the results we have obtained by employing the k-Nearest Neighbors algorithm with k equal to 5. As shown in Table 2, the accuracy we have been able to achieve is around 83%. We have computed this value by using a stratified ten-fold cross validation, because of the number of samples collected (around 1000

Correctly Classified Instances	1033 (83.71%)
Incorrectly Classified Instances	201 (16.29%)
Total Number of instances	1234
Kappa statistic	0.6448
Mean absolute error	0.1803
Root mean squared error	0.3622
Relative absolute error	39.28 %
Root relative squared error	75.60 %

(a)

True Positive Rate	83.7%
False Positive Rate	19.3%
Precision	83.7%
Recall	83.7%
F-Measure	83.7%
ROC Area	89.3%

(b)

Outside	**Inside**	⬅ Classified as
339	101	Outside
100	694	Inside

(c)

Table 2. K-Nearest Neighbors classification results: (a) summary; (b) accuracy by class; (c) confusion matrix

Correctly Classified Instances	1038 (84.11%)
Incorrectly Classified Instances	196 (15.88%)
Total Number of instances	1234
Kappa statistic	0.6492
Mean absolute error	0.234
Root mean squared error	0.3632
Relative absolute error	50.99%
Root relative squared error	75.82%

(a)

True Positive Rate	84.1%
False Positive Rate	20%
Precision	84%
Recall	84.1%
F-Measure	84%
ROC Area	85.1%

(b)

Outside	**Inside**	⬅ Classified as
329	111	Outside
85	709	Inside

(c)

Table 3. Tree-based results: (a) summary; (b) accuracy by class; (c) confusion matrix

samples). The classifier seems to perform well even if it is pretty simple, leading to a good precision, recall, F-measure and ROC area (Table 2.b). The first two metrics evaluate the impact of false positives and false negatives, while the F-measure takes in account both precision and recall. The higher is this metric, the better is the quality of the classifier. Similarly, the receiver operating characteristic (ROC) characterizes the tradeoff between hit rate and false-alarm rate in this noisy environment [36]. Furthermore, as it is possible to see from Table 2.c, the confusion matrix obtained with the KNN is quite balanced: the number of false positives and false negatives is very similar. However, in the considered application field, false positives are less problematic than false negatives. In fact, if the system is not able to detect that a user is inside a room while he/she actually is, the user experience can be dramatically negatively affected. On the other hand, if the system detects a user in a room while he/she is not inside that room, this is typically transparent to the user, thus it usually leads to almost negligible glitches in the behavior of the smart building, making it possible to consider it as an acceptable error.

5.1.3 Tree-based classification

In this section we present the results we have obtained by using classification trees. The specific algorithm we have employed is C4.5 [30], mainly because it works well with numerical data. As shown in Table 3.a, the accuracy of the C4.5 algorithm is very similar to the one obtained with the KNN algorithm. In particular, the accuracy, evaluated with a stratified ten-fold cross validation, is around 84%.

Analyzing more in detail the cross validation results (Table 3.b-c), it is possible to notice that C4.5 performs like KNN, having a good precision, recall, F-measure and ROC area. Unluckily, even though the accuracy of the tree-based classification algorithm is slightly higher of the KNN one, the confusion matrix shows a slightly worse behavior. In fact, the recall metric, which is related to false negatives, is characterized by a higher value. This could theoretically represents a problem, since, as previously stated, false negatives have a bigger impact on the degradation of the user experience.

However, as stated in the previous sections, the tree-based classification algorithm performs much better during the run-time phase, thus it has to be preferred every time the cost of the offline training phase does not affect the overall quality of the system (which usually happens in most cases, where the physical layout of the smart building changes quit rarely). From the point of view of timing performance, the time required to the tree-based classification algorithm to produce the results is in the order of few milliseconds, while the ones of the KNN algorithm are in the order of tens of milliseconds. Thus, even though both the approaches meet the quasi real-time performance typically required by this kind of applications, the tree-based algorithm represents the fastest solution, being around one order of magnitude better than KNN.

5.2 iOS app energy consumption evaluation

While developing the *BLUE-SENTINEL* system, we had as a major concern the evaluation of the battery consumption of the iOS applications needed as an occupancy location sensors; this data is crucial to understand the feasibility of the system, since nowadays the energy consumption is the most remarkable constraint during the development of mobile devices applications [25, 10]. Thanks to our previous experiences on this kind of devices [13, 14], we have been able to identify as a possible source of inefficient behaviors in terms of power consumption, the Internet transmission of the data from the device to the server. For this reason, we performed three different experiments: a first one to test the **baseline energy consumption** (when the app is not installed), a second one to test the **overall energy consumption** (normal app behavior) and a third one to test the *HTTP over WiFi communication energy consumption* (app with the HTTP communications disabled). The measurements have been performed by using Apple Instrument [3]. This tool does not provide a quantitative information about the exact amount of power usage, but it shows some profiles of the energy consumption by using a metric that ranges from 0 to 20[1]. Thus, our analysis is based on the comparisons that can be performed on different working contexts of the app. In Figure 8, we reported some screenshots from the Apple Instrument tool showing the profile of energy consumption registered in the three previously described experiments. Figure 8 clearly shows that the main impact on the battery consumption is related to the HTTP requests. In the following sections, we propose some solutions that we envision to completely solve this issue (thus enhancing the battery life while using the proposed application).

5.3 Results discussion and Limitations

The results obtained are a good starting point for further and more detailed investigations. In fact, from the **classification algorithms** side, the accuracy registered with the simplest possible implementation is not far from the most accurate WiFi based solutions available in the state of the art (RedPin [9] and Ariel [21], see Section 2). It is also important to remark that the algorithms we are using are in their basic version, so there is room for optimizations that could further improve the obtained results. We have noticed, in fact, that we collect missing values caused by the cyclic behavior of our beacons. When a beacon changes the region advertised, client-side we can not range the beacon and we are missing

[1] To the best of our knowledge, the documentation about how this value is computed is not publicly available.

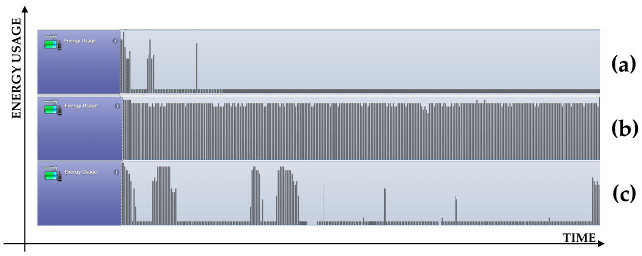


Figure 8. Apple Instrument screenshots. (a) Baseline energy consumption: the app is not installed. (b) Monitoring and updating the server: the app is working and it sends data to the server through http requests. (c) Monitoring without updating server: the app is working, but the http requests have been disabled.

values that create noises and worsen the performance of our model. These missing values are part of the so called Not Missing At Random (NMAR) data. This kind of missing data requires particular attention and needs to be managed with a domain specific policy. Finding, proving and evaluating a good policy could, in our opinion, remarkably improve the performance and the accuracy of the system. Another way to get better results is to evaluate other data mining techniques that are more robust against outliers. In fact, in the data we have collected during our tests we have observed several points that significantly differ from the others. This could have created some erroneous localizations. We think that data mining algorithms such as random forest [11] could obtain better performance in the localization of users, basically since it is more robust to outliers thanks to its structure: it builds a set of tree models and the localization is based on a majority voting policy.

From the **energy consumption** point of view, we have discovered that the HTTP protocol over WiFi, even if simple to use, is not the best choice when a continuos communication is required, since it is characterized by a very low energy efficiency. In order to reduce the impact on battery life, we are currently working on a news solution that does not rely on the HTTP protocol; since we have noticed from our tests that bluetooth communications are extremely power efficient when a continuos messaging is required, we are designing a new version of the system where the data is transmitted by the devices back to the beacons antennas through BLE; the antennas are then in charge of sending them back to the servers where the classification algorithms are executed.

Another factor that have to be considered is the possible interferences interference of multiple Bluetooth devices with the BLUE-SENTINEL system and how much the environmental conditions can affect the system reliability. A possible solution to this kind of problems is to have an adaptive system able to change its behavior according to the environmental condition: for instance, since any BLE receiver can be also used as an antenna, it is possible to imagine a self-configurable system, where each receiver senses every night the other receivers signal strength in order to recalibrate the entire network.

All these improvements are currently under investigation, but they are out of the scope of this paper, which mainly aims at demonstrating the feasibility of implementing an iBeacon-based occupancy detection system.

6 Concluding remarks

With this paper, we have shown how it is possible to exploit BLE, which is an intrinsically low-power communication protocol, to solve the occupancy detection problem. We have proposed a modification of the iBeacon protocol, based on BLE, in order to adapt it to the target problem and we have described *BLUE-SENTINEL*, a BLE-based system which face the occupancy detection problem with a very simple and scalable approach. In fact, the proposed solution does not require extra devices for the users (we assume that almost all the users of a smart building are already provided, or will be in the very near future, with a mobile smartphone device) and a very limited number of devices (the beacons) that have to be displaced in the building to be monitored. As shown in our experiments, even a single beacon per room could be sufficient to achieve good accuracy results, as opposed to the state-of-the-art-approaches, such as [18] and Landmarc [27], that require a quite high number of antennas.

The power efficiency of the proposed approach is guaranteed by the underlying BLE communication protocol, which has been designed with the main purpose of reducing the power consumption of this kind of applications, where small messages (usually identifier) are sent in burst mode with a quite low frequency (in the order of tens or hundreds of milliseconds). On the other hand, the accuracy achieved with the proposed *BLUE-SENTINEL* occupancy detection system almost is comparable to the ones of the other existing approaches, such as Ariel [21] and RedPin [9].

Finally, the fact that Apple has released, during the last World-Wide Developers Conference (June 2, 2014), a new API, called HomeKit, aiming at providing a set of tools to communicate with smart buildings, further increases the importance of finding new ways, as we have done in this paper, to adopt commercial and widespread technologies to solve relevant issues such as the occupancy detection problem.

7 Acknowledgements

This work was partially supported by the Joint Open Lab "S-Cube" - Telecom Italia S.p.A. - Innovation division, Italy.

8 References

[1] Buildings energy data book. Technical report, US Department of Energy, August 2012.

[2] Android beacon library. 2014.

[3] Apple instrument. 2014.

[4] Core location framework reference. 2014.

[5] Flask microframework. 2014.

[6] ibeacon for developers. 2014.

[7] G. Anastasi, R. Bandelloni, M. Conti, F. Delmastro, E. Gregori, and G. Mainetto. Experimenting an indoor bluetooth-based positioning service. In *ICDCS Workshops*, pages 480–483, 2003.

[8] B. Balaji, J. Xu, A. Nwokafor, R. Gupta, and Y. Agarwal. Sentinel: occupancy based hvac actuation using existing wifi infrastructure within commercial buildings. In *SenSys*, page 17, 2013.

[9] P. Bolliger. Redpin - adaptive, zero-configuration indoor localization through user collaboration. In *MELT*, pages 55–60, 2008.

[10] A. Bonetto, M. Ferroni, D. Matteo, A. Nacci, M. Mazzucchelli, D. Sciuto, and M. D. Santambrogio. Mpower: towards an adaptive power management system for mobile devices. In *Proceedings of the 2012 IEEE 15th International Conference on Computational Science and Engineering*, pages 318–325. IEEE Computer Society, 2012.

[11] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[12] S. Dawson-Haggerty, A. Krioukov, J. Taneja, S. Karandikar, G. Fierro, N. Kitaev, and D. Culler. Boss: building operating system services. In *Proceedings of the 10th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2013.

[13] M. Ferroni, A. Cazzola, D. Matteo, A. A. Nacci, D. Sciuto, and M. D. Santambrogio. Mpower: gain back your android battery life! In *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication*, pages 171–174. ACM, 2013.

[14] T. S. S. Ferroni, Cazzola. On power and energy consumption modeling for smart mobile devices. In *Proceedings of The 12th IEEE International Conference on Embedded and Ubiquitous Computing*. IEEE Computer Society, 2014.

[15] E. Fix and J. L. Hodges Jr. Discriminatory analysis-nonparametric discrimination: consistency properties. Technical report, DTIC Document, 1951.

[16] S. Ghai, L. Thanayankizil, D. Seetharam, and D. Chakraborty. Occupancy detection in commercial buildings using opportunistic context sources. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2012 IEEE International Conference on*, pages 463–466, March 2012.

[17] C. Gomez, J. Oller, and J. Paradells. Overview and evaluation of bluetooth low energy: An emerging low-power wireless technology. *Sensors*, 12(9):11734–11753, 2012.

[18] D. Hähnel, W. Burgard, D. Fox, K. P. Fishkin, and M. Philipose. Mapping and localization with rfid technology. In *ICRA*, pages 1015–1020, 2004.

[19] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18, 2009.

[20] M. Hazas and A. Ward. A novel broadband ultrasonic location system. In *Ubicomp*, pages 264–280, 2002.

[21] Y. Jiang, X. Pan, K. Li, Q. Lv, R. P. Dick, M. Hannigan, and L. Shang. Ariel: automatic wi-fi based room fingerprinting for indoor localization. In *UbiComp*, pages 441–450, 2012.

[22] A. LaMarca, Y. Chawathe, S. Consolvo, J. Hightower, I. E. Smith, J. Scott, T. Sohn, J. Howard, J. Hughes, F. Potter, J. Tabert, P. Powledge, G. Borriello, and B. N. Schilit. Place lab: Device positioning using radio beacons in the wild. In *Pervasive*, pages 116–133, 2005.

[23] J.-S. Lee, Y.-W. Su, and C.-C. Shen. A Comparative Study of Wireless Protocols: Bluetooth, UWB, ZigBee, and Wi-Fi. In *Industrial Electronics Society, 2007. IECON 2007. 33rd Annual Conference of the IEEE*, pages 46–51, Nov. 2007.

[24] R. Melfi, B. Rosenblum, B. Nordman, and K. Christensen. Measuring building occupancy using existing network infrastructure. In *IGCC*, pages 1–8, 2011.

[25] A. Nacci, F. Trovò, F. Maggi, M. Ferroni, A. Cazzola, D. Sciuto, and M. D. Santambrogio. Adaptive and flexible smartphone power modeling. *Mobile Networks and Applications*, 18(5):600–609, 2013.

[26] N. T. Nguyen, R. Zheng, and Z. Han. Umli: An unsupervised mobile locations extraction approach with incomplete data. In *WCNC*, pages 2119–2124, 2013.

[27] L. M. Ni, Y. Liu, Y. C. Lau, and A. P. Patil. Landmarc: Indoor location sensing using active rfid. *Wireless Networks*, 10(6):701–710, 2004.

[28] V. Otsason, A. Varshavsky, A. LaMarca, and E. de Lara. Accurate gsm indoor localization. In *Ubicomp*, pages 141–158, 2005.

[29] L. Pei, R. Chen, J. Liu, T. Tenhunen, H. Kuusniemi, and Y. Chen. Inquiry-based bluetooth indoor positioning via rssi probability distributions. In *Proceedings of the 2010 Second International Conference on Advances in Satellite and Space Communications*, SPACOMM '10, pages 151–156, Washington, DC, USA, 2010. IEEE Computer Society.

[30] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.

[31] B. Ur, E. McManus, M. P. Y. Ho, and M. L. Littman. Practical trigger-action programming in the smart home. *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, 2014.

[32] R. Want, A. Hopper, V. Falcao, and J. Gibbons. The active badge location system. *ACM Trans. Inf. Syst.*, 10(1):91–102, 1992.

[33] A. Ward, A. Jones, and A. Hopper. A new location technique for the active office. *IEEE Personal Commun.*, 4(5):42–47, 1997.

[34] T. Weng and Y. Agarwal. From buildings to smart buildingssensing and actuation to improve energy efficiency. *IEEE Design & Test of Computers*, 29(4):36–44, 2012.

[35] T. Weng, A. Nwokafor, and Y. Agarwal. Buildingdepot 2.0: An integrated management system for building analysis and control. In *Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings*, pages 1–8. ACM, 2013.

[36] I. H. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2005.