

Research Article

Using Bluetooth Low Energy Advertisements for the Detection of People Temporal Proximity Patterns

Jose Garcia-Alonso,¹ Javier Berrocal ,¹ Alejandro Pérez-Vereda,² Jaime Galán-Jiménez,¹ Carlos Canal,² and Juan M. Murillo¹

¹Department of Computer and Telematic Systems Engineering, Escuela Politécnica, University of Extremadura, 10003 Cáceres, Spain

²ITIS Software, Universidad de Málaga, 29071 Malaga, Spain

Correspondence should be addressed to Javier Berrocal; jberroalm@unex.es

Received 22 August 2019; Revised 30 January 2020; Accepted 5 February 2020; Published 13 June 2020

Academic Editor: Carlos Tavares Calafate

Copyright © 2020 Jose Garcia-Alonso et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The pervasive presence of smartphones has emerged as one of the key elements for sensing people contextual information. Their sensors and communication capabilities can be used to gather a huge amount of data. Such capabilities have made it possible to compose profiles of people by relating different parameters such as time and location. This paper contributes in this sense by providing the basis for the composition of temporal proximity patterns—when and whom people share their time with each other. For this purpose, the Bluetooth Low Energy (BLE) advertisement protocol was used. The contribution of this work departs from that of those who use BLE technology focused on measuring the intensity of the signals to, for example, determine distances. In this field, a huge amount of work has been already done with very interesting results. Instead, in this work, BLE is used to emit and sense the presence of people. A set of algorithms are then used inside the smartphones to analyse the data gathered and to detect proximity patterns between people. This scenario avoids the difficulties that appear in other works—like those focused on people positioning—derived from the lack of precision of the sensors and the differences between BLE chipsets. Tests to evaluate the consumption, precision, and reliability of using this technology, together with the proposed algorithms, confirmed the feasibility of the approach. In addition, the proposal has proved very useful for the automatic construction of social networks based on physical closeness of people.

1. Introduction

The pervasive presence of smartphones, which include a wide range of sensors, has enabled the massive gathering of information from people. This information has been proved useful in many situations, as demonstrated by the interest of large corporations such as Google, Facebook, and Amazon in gathering as much information as possible from their users.

Similarly, the work presented in this paper is part of a research initiative of the authors about sensing contextual information from people. In previous works, the authors presented a model with which to build virtual profiles of people including their routines of movement and outdoor

positioning [1], as well as fine-grained indoor positioning [2].

The goal of the present work is to estimate whom users share their time with. This information is still missing in most approaches considering the development of virtual profiles of people, although it is of interest for social networks and other companies [3]. To do that, we propose an approach to detect proximity and accompaniment patterns between people based on the use of the Bluetooth Low Energy (BLE) protocol [4]. This proposal follows the People as a Service (PeaaS) paradigm [5], in which the increasing capabilities of smartphones are used to collect information, with the aim of using it to provide rich contextual services about their owners.

Dozens of datasets about people’s ge positioning and mobility are publicly available. They gather very different kinds of data, ranging from places of birth vs. places of residence or long-distance travelling to urban mobility, traffic events, cabs, running, or biking traces. Focusing on those more fine-grained, most of them contain information coming from GPS sensors (for instance, Geolife (Geolife. <https://www.microsoft.com/en-us/download/details.aspx?id=52367>) or Cabspotting (Cabspotting. <http://crawdad.org/~crawdad/epfl/mobility/20090224/cab/>)). There are only a few exceptions, such as the dataset from the roadway network of the city of Austin (TX) (Austin’s Bluetooth Travel Sensors–Individual Address Files (IAFs)), that contain information coming from BLE devices. We have not found any dataset specifically related to the proximity between individuals, although this information could be inferred to a certain extent from some of the sources considered. For instance, MDC (Mobility Data Challenge (MDC). <https://www.idiap.ch/dataset/mdc>) is a dataset collected in the Geneva Lake area between 2009 and 2011 containing GPS positions from the smartphones of a group of volunteers. These, and other private datasets, have been used for analysing different aspects of human mobility and interaction between people in different scenarios [6–9] (see [10] for a broad review in the field), including repeated patterns of movement [11].

In the literature, we can also find several approaches for detecting when two or more people are together [12–14]. They differ on how information about the proximity of users is collected. The pervasive presence of smartphones means that solutions based entirely on their capabilities have several advantages. One is the minimal cost of their deployment and maintenance, since users already have their own smartphones, and they take care of keeping them always in working condition. Other advantages are that they require no specific infrastructure, and the fact that they can work similarly both indoors and outdoors so that they are usable in almost any environment without previous instrumentation.

Although some previous studies have used Bluetooth detection for inferring connections or relationships between people [15–18], these works involve the use of a greater number of sensors, signals, or, even, ad hoc devices. This paper presents the results of an implementation of such an approach using only smartphones’ BLE sensing capabilities. It is important to notice that our goal is not to use BLE in order to obtain accurate measurements of the distance between people or to improve the positioning information of the users, but to determine when they are close enough to infer that they are together, and how long this accompaniment lasts. Hence, we consider proximity more as a boolean condition, than as a precise measurement of distance. Having a simple and efficient way of obtaining these patterns would allow mobile application developers to create new context-aware applications with a profound impact in several domains like social networking or eldercare.

Nevertheless, in order to assess the feasibility of the proposed system, we have analysed its most relevant features. In particular, in this paper, we have evaluated the

precision with which the proposed system assesses the proximity of other people, its reliability under different circumstances, and its consumption of the smartphone’s resources, especially the battery. Results show that the BLE capabilities of smartphones can be successfully used to gather information about the proximity between people.

Most of the previous related work was applied to the elderly care context [19], gathering information about indoor and outdoor positioning of the elderly person. The work proposed here can be used in a similar context. Indeed, knowing whom the elderly share their time with is useful in a wide range of situations. For instance, an alarm can be raised when they are alone or not close to their caregivers. On the contrary, the alarm can be avoided when they are accompanied by a known caregiver even if they are outside their safe area. Nevertheless, our proposal represents a more general solution for detecting temporal patterns of proximity between people, applicable to scenarios different from caregiving, such as social networks that have a similar setting. The knowledge obtained from these patterns can be enriched by combining them with GPS positioning. This would enable the determination of routine patterns of when, where, and with whom someone shares their time and use this information to forecast their future behaviour.

The rest of this paper is organized as follows. After this introduction, Section 2 presents the motivation of this work and describes the most relevant related works. Then, Section 3 presents the proposed system to use BLE to gather information about the temporal proximity between users. Section 4 describes the experimental validation of the approach. This involves monitoring its precision, consumption, and reliability. Finally, Section 5 discusses the results and draws some conclusions from the study.

2. Background

The main goal of gathering contextual information about a system’s users is for the system to take this context into account, to learn from it, and to proactively meet the users’ needs with as little intervention on their part as possible. These contextual data are used to infer more complex information and to create virtual profiles of the users [20, 21]. These profiles may then be used to control the system without direct human interaction [22, 23].

Currently, there are different approaches to create these virtual profiles (see, for instance, [24–26]). As mentioned above, the present authors have proposed the PeaaS computing model which relies on smartphones as the key element for gathering all the information needed to construct a virtual profile of their owners [5].

For the goal of adapting systems to the preferences of their users, the closer to complete these profiles are the better [27]. Having access to richer virtual profiles allows systems to react to more complex situations. Hence, virtual profiles must include a varied set of information. One of the core elements of most of the proposals for representing user context that can be found in the literature is location. Location coordinates can be used to infer more complex aspects of the virtual profiles such as where users are, which are

their favourite places, which are their movement patterns, or with whom they spend their time [28, 29]. This information is more relevant every day, especially for social networks and other social-related areas [3, 30]. Then, this information can be combined with other data provided by the smartphone and other external sensors to infer even more complex knowledge about the user.

Currently, the location of a smartphone can be estimated using its GPS sensor. However, GPS precision drastically decreases when the user is inside a building. Although other techniques, such as WiFi or 4G networks, could be used indoors, neither they have great precision so that they would have only limited use when a person should be located with accuracy [31], nor they are easily deployed in the wild as their better results are obtained by fingerprints [32]. As a consequence, they cannot be used to determine the proximity between users.

Techniques for determining indoor positioning have improved greatly over the last few years [33]. There are studies using RFID, which provides face-to-face interaction detection within 1-2 meters [13, 14], or beacons [34], which are BLE devices that broadcast information to determine fine-grained indoor positioning.

However, while this approach is useful in controlled, instrumented environments, it cannot be used with any guarantee in more general situations in which beacons are unavailable or not properly maintained. Additionally, they only provide information about the individual position of users and not about who are they with.

To address this issue, in this paper, we present an extension to our Beacon Management System [2]. The approach is to exploit users' smartphones as both emitting and receiving BLE beacons. These virtual beacons are therefore able to detect each other and estimate their relative distances. This information is then used to infer proximity between people.

This is not the first time, however, that BLE devices have been used to gather contextual information. The next section will summarize the most relevant works.

2.1. BLE for Context Awareness. Gathering user positioning information in different scenarios has been a topic of interest for researchers for a long time [35]. In particular, beacon technology has been specifically designed to improve indoor positioning. The use of this kind of device is therefore of great benefit for systems that require more precise positioning information. In [36], the use of beacons to improve previously existing location-based methods is advocated. The authors propose transforming the mobile devices of users into virtual beacons so that other users might be able to benefit from improved location information. This use of beacon technology improved previous results without any relevant detriment to the users whose devices act as beacons. A similar approach was followed in [37] where personal devices such as laptops or phones were converted into beacons to provide proximity location. In this work, the authors focused on the privacy aspect of having a personal device acting as a positioning beacon. However, they also

analysed the social interactions between coworkers thanks to the proximity information.

Performance analyses comparing different protocols have been made before. In [38], after comparing the performance of BLE with two other similar protocols, the authors concluded that BLE provides an inexpensive and power-efficient solution for small coverage areas. In fact, the authors in [12, 39, 40] derive mathematical models to study the performance of BLE's device discovery process, concluding that it can be drastically improved by implementing different strategies. However, none of them performed experiments using smartphones, although from their conclusions we derived and applied some good practices in the presented system to increase the battery life and to improve its performance.

The suitability of BLE for these types of scenarios is of course supported by many research works. In [41], the authors propose the use of contextual information, including indoor positioning mechanisms, to create a system that fosters social interactions among strangers. This proposal makes use of indoor location to improve the quality of the contextual information gathered. Moreover, in [42], the authors propose the use of BLE wearable tags in combination with an application running on commercial mobile devices to detect social interactions between users.

Similarly, in [43], the authors present four mobile apps that use BLE to provide contextual relevance and personalized experiences for the user. Again, this work's results highlight the importance of indoor positioning and how it can be improved by using beacons. As mentioned above, the focus of this work is not indoor positioning but an alternative use of the BLE technology to determine with whom the users expend time.

Another example of BLE usage can be found in [44]. Its authors propose the use of BLE devices to improve the recognition of the daily activities performed by users of ambient-assisted living systems. A combination of the user's smartphone and other wearable devices together with the information emitted from a set of physical beacons allowed the researchers to identify more daily activities than the ones obtained by the research community in previous works.

Similarly, in [45], the authors propose the use of traditional Bluetooth communications to estimate proximity networks between people. The main goal was to determine the social context of a given user based on the state of the proximity network around him at a given time.

Finally, the present authors previously proposed a Beacon Management System [2] designed to enrich user virtual profiles with indoor information in places where physical BLE beacons are deployed. The approach reported in this work is an extension of such proposal which converts each smartphone into a virtual beacon and uses the resulting information to acquire proximity patterns between users.

As can be seen, there is a wide range of related studies in the field, with some of them even using mobile devices as virtual beacons for indoor positioning and to measure the proximity between users. While similarly, in the present work, our focus is on two main issues that, as far as, the authors know, they are not covered in previous works. On the one hand, as will be shown in Section 3, our aim is to be

able to detect temporal patterns of proximity between users that are repeated on a (more or less) regular basis using Bluetooth technologies, so we can determine with whom and when the users spend time. On the other hand, in Section 4, we shall describe experiments that we conducted in order to investigate the limitations of the current state of the technology. The aim is to try to assess whether current smartphones acting as virtual BLE beacons can actually be used for our purposes, and how such use affects their consumption, a highly relevant question for resource-constrained devices such as smartphones.

3. Temporal Proximity Patterns

The work presented in this paper was initially developed to be part of an elderly care system [1]. The system comprises two mobile apps, one for monitoring cognitively impaired users/elders who still have some degree of autonomy and the other for their caregivers. The aim of this system is to detect the user’s everyday routines of location and movement and to alert their caregiver if a significant deviation is detected. More recently, we extended our system in order to improve the contextual information that the user’s app gathers by adding temporal proximity patterns obtained using BLE. This richer information would later be used to better evaluate the need to send an alarm to a caregiver.

Although originally designed for the elders and their caregivers, our proposal can be applied to gather temporal proximity patterns of any kind of user, since such patterns can be of interest in many other context-awareness scenarios. Therefore, the BLE system to compute the proximity patterns is presented here independently of the system in which such information will be used.

The following sections describe the detailed architecture of the system, how the proximity data are gathered and stored, the data model used to describe proximity patterns, and the algorithms that actually identify those patterns.

3.1. Architecture. Figure 1 details the architecture of the system. It consists of two kinds of components: a Proximity Patterns smartphone application and a nimBees server. The nimBees (<http://nimbees.com/>) platform is a commercial implementation of the PeaaS paradigm. For our purposes, it is mainly used for user registration, but it can also serve for indirect communication between users or for matching the received BLE packets with the registered users. It has a fixed, well-defined service entry point—the nimBees API—into which every mobile device involved in a given application must register in order to authenticate and provide information about itself and to retrieve information about the rest of the devices connected to that application.

Our proximity pattern application takes advantage of the existing nimBees implementation. However, it could be replaced by an alternative implementation.

The main component of the architecture is the Proximity Patterns smartphone application. This app converts users’ smartphones into both advertising and scanning devices as defined in the BLE protocol. Each smartphone will

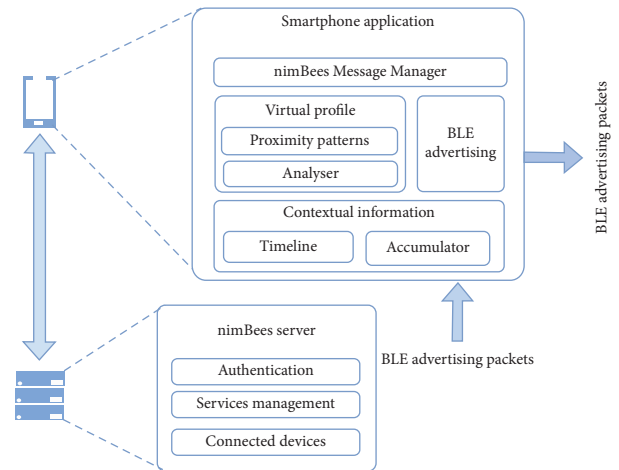


FIGURE 1: Architecture of the BLE proximity pattern system.

periodically emit an advertising packet, while it also listens for and scans packets from other BLE advertising devices. Thus, all the smartphones behave symmetrically and exchange information cooperating in a P2P architecture.

The Contextual Information component is responsible for capturing and recording the advertising packets received. It comprises two modules: the Accumulator and the Timeline. The Accumulator module scans the raw information included in the advertising packets, in particular, the identifier of the user sending the packet and an estimation of the distance to that user, derived from the strength of the received signal. The information gathered is stored in the Timeline component, which sequentially organizes the acquired information.

Figure 2 shows a plot of the information stored in the Timeline component corresponding to the different advertising packets that a smartphone received during a period of time. By combining these packets in the Timeline, the system is able to track the temporal course of the proximity of different users. The green line in the figure represents the proximity of a family member going for a walk with the elder. A similar proximity line would be stored in the Timeline of the family member’s smartphone. The dark blue line represents the proximity of a caregiver, and the light blue one refers to the proximity of another elder, both of whom spent some time close to the elder in a daycare facility.

The Virtual Profile component is divided into two modules: the Analyser and the Proximity Patterns store. The Analyser is responsible for periodically analysing the information stored in the Timeline in order to identify temporal proximity patterns. It implements an algorithm which evaluates the BLE packets of information stored in the Timeline in order to identify sequences of packets corresponding to a given user’s smartphone. This is indicative of its proximity to the detecting smartphone. When these sequences are repeated at different times, they will eventually lead to the definition of a temporal proximity pattern. All the patterns identified in this way are stored in the Proximity Patterns module, following a data model that will be described in Section 3.2. Locally storing the detected

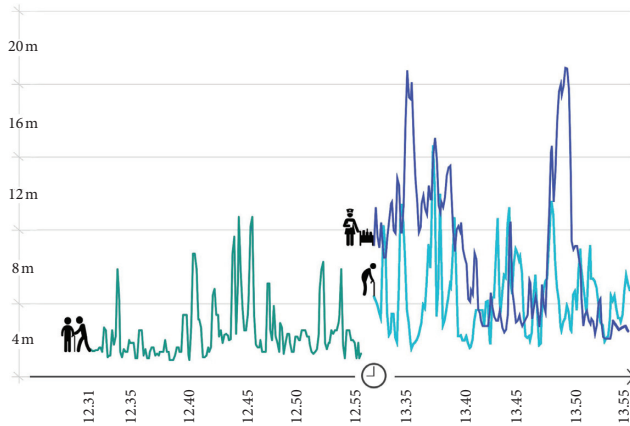


FIGURE 2: Graphical representation of the information stored in the Timeline component.

information improves its privacy, since different policies could be implemented allowing users to control their own data. At the same time, the BLE Advertising component is responsible for emitting BLE packets with the user identifier.

With this architecture, each smartphone is able to compute the temporal proximity patterns of its owner. These patterns can be therefore provided as a service to third parties. For example, an elder’s smartphone composes and stores her proximity patterns, and her caregiver’s smartphone could request to be alerted when any of these patterns change. These operations are handled by the nimBees Message Manager component of the architecture.

3.2. Data Model for Temporal Proximity Patterns. The architecture detailed above gathers information from any BLE advertising device. This allows the detection of temporal proximity patterns, regardless of the time shared between users or, even, if the detections involve devices that have no relation with the user. These nonrelated devices can be filtered using the nimBees platform. In addition, in order to handle small and sporadic interactions, we define a flexible conceptual data model that facilitates the definition of proximity patterns but that can accommodate small contextual changes. Figure 3 details part of the conceptual model defined to satisfy these requirements.

A proximity pattern is defined as a time window in which a user shares her time with some specific companions. Its representation therefore consists of the start and end time of the pattern, the frequency with which the pattern is repeated (e.g., daily, on weekends, and sporadically), its weight (which indicates the expected probability that the pattern actually takes place at the times indicated by its frequency), and a number of companions corresponding to the set of people who are with the user when the pattern takes place. Each companion has their own weight, indicating the probability of that person participating in the pattern.

A proximity pattern is connected to other patterns by a set of transitions. This allows information to be stored about sequences in which the different patterns occur, composing a picture of the sequencing of all the people the user spends time with throughout the day. Moreover, it allows one to

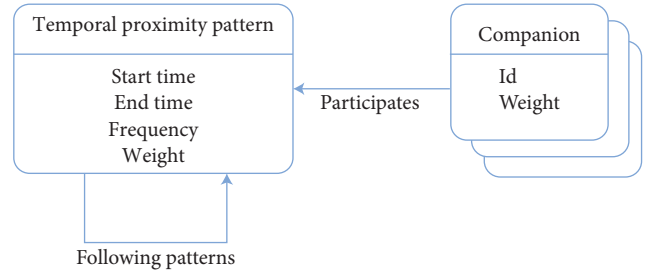


FIGURE 3: Conceptual data model for the definition of proximity patterns.

know with whom a person should be at a specific time and also to forecast which proximity patterns would occur.

Figure 4 shows an excerpt of an example of proximity pattern. The figure gives details of the following patterns:

- (i) *P1.* Every morning, the user spends some time doing light physical activity. She is usually accompanied by the physical trainer and a close friend. The devices of these companions emit BLE advertising packets that include their ids, so they are recognized by the user’s smartphone. While other friends sometimes join her in the morning activity, their weight is not high enough for them to be represented in the pattern. The same would be the case for other BLE emitting devices that are sporadically detected during the time of the pattern. If a user is detected often enough during the pattern, her weight would increase to the point where it would be included.
- (ii) *P2.* This pattern captures the user’s lunchtime after the physical activity. On weekdays, due to the fact that she usually has lunch alone, the pattern includes no companions. Again, if a companion is detected often enough during this period, it would be added to the pattern.
- (iii) *P3.* At weekends, however, her daughters join her for lunch. Although the time period and location are similar to the previous pattern, the different companions lead the system to recognize them as two different patterns with different frequencies. The system consequently includes two possible transitions from P1 to each of these two (new) patterns that may occur after it.

As indicated before, this pattern structure is managed by means of different data processing algorithms which will be briefly described in the next section.

3.3. Identification of Temporal Proximity Patterns. The Analyser module of the smartphone application is responsible for identifying the phone owner’s temporal proximity patterns and their storage. A fine-grained description of the algorithms involved would require too much space. In [1], the authors present a process for the identification of movement patterns. Some of the algorithms for identifying

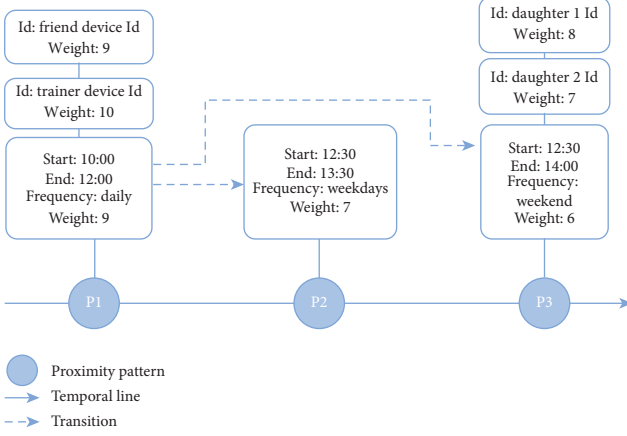


FIGURE 4: Excerpt of a proximity pattern timeline example.

repetitions and variability are the same, so they are not repeated here. However, for the sake of completeness, we shall present the main features of the algorithm used to detect proximity patterns.

This algorithm is executed once a day at a predetermined time, usually at night when the phone is plugged in for charging. It processes all the contextual information stored in the Timeline component of the architecture since the last time it was executed.

Let $c^t = (c_1^t, \dots, c_i^t, \dots, c_{n_c}^t)$ be a tuple with the values of contextual information gathered by the virtual or physical sensors in the smartphone at a certain moment, t . This information will include the location of the user, information from the accelerometer, and the distance to other users. It is dated with a timestamp (ts) and stored in a timeline $\mathcal{T} = \langle (c^{ts_1}, ts_1), \dots, (c^{ts_i}, ts_i), \dots, (c^{ts_n}, ts_n) \rangle$, with ts_i being the timestamp at which the data c^{ts_i} was gathered.

In order to reduce the smartphone's resource consumption, the distance between pairs of devices is measured by means of BLE technology. For this purpose, equation (1) is used generically [46]:

$$d = \exp[a * (\text{RSSI} - \text{txPower})], \quad (1)$$

where d is the estimated distance between the phone and the transmitter and a is a precalibrated exponential decay term. Therefore, the distance is estimated from the received signal strength denoted by RSSI (received signal strength indicator) and the calibrated signal strength at 1 meter of distance (denoted for simplicity by txPower (Transmitted Power)).

Let $\mathcal{U} = \{u_1, \dots, u_i, \dots, u_{n_u}\}$ be the set of n_u users with whom the smartphone's owner interacts. The detected distance $d_{u_i}^{ts_x}$ would therefore be the distance between the owner's device and the device of user u_i at time ts_x .

Let us assume that the day being analysed corresponds to the time interval $[ts_x, ts_y]$ between timestamps ts_x and ts_y . Then, the algorithm identifying the user's proximity patterns computes all the information gathered during that interval in an incremental process consisting of three steps that are explained as follows.

In the first step, the algorithm analyses all the BLE advertising packets stored in the Timeline component. During

this process, all the packets received from the same device are extracted and used to compute the proximity of that device to the user during the day being analysed. The result of this first step is a set of devices $\mathcal{U}_{ts_x}^{ts_y} = \{u_1, \dots, u_j, \dots, u_{n_u}\}$ that have been close to the user, each of them with the information about their distance to the user over time, i.e., $u_j = \langle (d_{u_j}^{ts_x}, ts_x), \dots, (d_{u_j}^{ts_y}, ts_y) \rangle$. This situation can be represented graphically as in the example shown in Figure 2.

The second step takes as input this information and splits it into fragments k_1, \dots, k_n , each of which captures a period of time during which the companions remained stable, i.e., periods during which the user was close to the same people. In order to discard the largest measurement errors, the distance to each companion is estimated using the trimmed mean of all the distances covered by that fragment (see equation (2)). This step of the algorithm includes different types of allowed standard deviations, σ_x , to let through changes in companions which may have been caused by measurement errors or which are irrelevant for the user's context, for example, if a companion is undetected for a couple of minutes because he left to make a phone call. In addition, when the difference between two consecutive distances is greater than the allowed deviation, σ_d , or when the difference between their timestamps is greater than the allowed value, σ_{ts} , a movement is identified and then a new fragment has to be computed (see equations (3) and (4)):

$$\overline{d_{u_i}^{kn}} = \frac{1}{n-2k} \sum_{i=k+1}^{n-k} d_{u_i}^{ts_i}, \quad (2)$$

subject to

$$|d_{ts_x} - d_{ts_{x+1}}| > \sigma_d, \quad \text{where each } d_i \in \mathcal{D}, \sigma_d \in \mathbb{R}, \quad (3)$$

$$|ts_x - ts_{x+1}| > \sigma_{ts}, \quad \text{where each } ts \in \mathcal{T}, \sigma_{ts} \in \mathbb{R}. \quad (4)$$

Finally, the third step takes the fragments obtained in the previous step and compares them with all the proximity patterns \mathcal{P} stored in the application for the same time of day.

Let $\mathcal{P} = \{p_1, \dots, p_j, \dots, p_{n_p}\}$ be the set of the n_p temporal proximity patterns, where each pattern comprises a timestamp interval $[ts_x, ts_y]$, its frequency fr and weight ω , and a tuple of contextual information, as shown in Figure 4:

$$p_j = \left([ts_x, ts_y], fr_j, \omega_j, (\overline{c_{1j}}, \dots, \overline{c_{ij}}, \dots, \overline{c_{n_{c_j}}}) \right). \quad (5)$$

Again, in the comparison between proximity patterns, a given standard deviation is allowed to account for possible measurement errors. If a pattern p is found to be equal to a fragment k within the allowed deviation, then this means that the phone owner is repeating an already established proximity pattern. To reflect this, the weight ω of the pattern is increased as also will be the weights of all the companions that are present in such pattern:

$$\text{if } \exists p \in \mathcal{P}, |p.d - \overline{d_{u_i}^k}| < \sigma_d, \quad \text{then } p.\omega \leftarrow p.\omega + 1. \quad (6)$$

For a specific occurrence, if there is a difference for some companions but it is within the allowed deviation, the weight of those companions would be updated accordingly. The weight of an absent companion would be reduced to the point at which that companion would be removed from the pattern if their weights were below some threshold. Similarly, the weight of a companion not included in the pattern would be increased to the point where they would be added to the pattern when their weight rose above that threshold.

If no equal pattern is found for a timeline fragment, then a new pattern will be created in the application with its weight and the weight of all the involved companions set to zero. If this pattern is found again in the future, its weight would be updated.

The weights of patterns that are stored in the application but not performed by the user will be reduced in accordance with their frequency. If a detected pattern is no longer followed by the user, then it will be removed over time.

Preliminary results of the extended elderly care system show that temporal proximity patterns help to enrich the contextual information that the application gathers from its users, providing a new source of information about their habits. This information can be combined with that coming from the analysis of patterns of location and movement already inferred by the application. All this knowledge is applied to better assess whether the elder is in a safe situation or, contrarily, an alarm should be raised. Nevertheless, there should be additional validation of these results, with information on proximity patterns collected from a number of real users.

4. Experiments and Results

In the previous section, we described the proposed system for determining the user's proximity patterns and how the data are registered and processed. None of this has any value, however, unless one can guarantee that the proximity data acquired are correct. In this section, we shall consider the feasibility of the system by evaluating the following aspects of the proposed solution:

- (i) *Proximity precision.* The determination of temporal proximity patterns depends on the reliability of the information about the distance between users. To ensure that the distances detected have enough precision to identify the proximity between people, we measured the real distances between smartphones acting as both emitting and receiving BLE devices and compared them with the detected distances.
- (ii) *Communications reliability.* The reliability of the communications is also important. In this sense, the system must take communication failures into account. To evaluate the system's reliability, we measured the ratio between the received and missed communications in different scenarios.
- (iii) *Battery consumption.* The proposed system is designed to run on the users' smartphones, and it is

well known that resource consumption, in particular battery use [47, 48], is a determining factor for the application success. We therefore evaluated the battery consumption with the smartphone functioning as both a virtual beacon emitting BLE advertising patterns and a sensing device registering the signals.

- (iv) *Feasibility of the algorithms.* The defined algorithms identify temporal proximity patterns, but we have to validate that those patterns are correct. To that end, we evaluated the detected patterns along with false-positive and false-negative rates.

4.1. Methodology. To perform the experiments, we developed an application that transforms the smartphone on which it is run into a BLE advertising device at the same time as registering the reception of BLE advertising packets broadcasted by other devices. This application was designed in a similar way to the one used in our previous work to measure the resource consumption of a GPS sensor [49]. In the present case, in order to implement the reception and transmission of advertising packets, the application makes use of the Android Beacon Library [50]. This library was selected because it is widely used [51] and provides support for both broadcasting and listening for BLE advertising packets.

For the transmission of advertising packets, the most important parameters that can be configured with this library are *Transmit Power* (controlling the distance to which the packets are broadcasted) and *Advertising Interval* (frequency at which the packets are sent). In order to increase packet visibility while also considering power consumption, the Transmission Power was configured to achieve the maximum distance since, as indicated by Montanari et al. [12], its impact on battery lifetime is minimal and it does not affect the accuracy. Likewise, the Advertising Interval was configured to send packets at a frequency of 10 Hz for two reasons. The first was that this or lower frequencies have least impact on packet collision (and hence on communication reliability) when the number of packet-transmitting devices increases [12], and the second was that the transmission of advertising packets at this frequency does not involve any major battery consumption (as will be detailed below).

In the case of listening for BLE advertising packets, the most important parameters to configure are the *Scan Window* (the time that the application spends listening for new advertising packets) and the *Scan Interval* (the time the application waits before starting a new scan). Scanning for new packets has a major impact on battery life, but, at the same time, it has to be configured to last long enough to ensure that at least one advertisement is captured at the desired frequency. Therefore, a balance has to be found between minimizing battery consumption and increasing the number of packets captured. To this end, we again followed the recommendations of Montanari et al. [12], by defining a Scan Interval of 100 ms and a Scan Window of 20 ms.

In order to be able to evaluate the above aspects, information on the advertisements sent and received and on the battery consumption during the execution of these operations is stored in a log file. Among other data, this file contains the number of packets received and their UUIDs, the distance detected to the transmitting device, and the battery consumption in microampere-hours (μAh). The functionalities provided by the Android Beacon Library are used to get the information related to the advertisements, and the *BatteryManager* class provided by the Android OS is exploited to obtain the information related to the battery. Knowing in advance the distance between the two devices (transmitter and receiver) allows processing these logs to give each operation's average consumption, the precision of the distance detected, and the communication reliability in different scenarios.

For the sake of the validity of the experiment, it is also necessary to ensure that the smartphones involved are not subject to interference from external agents or OS routines. For this reason, all the operations are executed with the device at rest, with no other applications running at the same time, and they are encapsulated in an Android Service with an associated WakeLock [52] so that the OS neither stops nor relegates to the background execution of each operation.

Additionally, measuring the consumption pattern of an operation from a single execution does not always provide real data since there may be contamination from the consumption of the screen, the app's interface, or other sensors of the smartphone. In order to get more accurate average measurements of the consumption, the test application was developed to execute each operation in a loop. In this way, we were able to launch an execution, put the device to rest, acquire a large set of measurements, and then halt the execution. In particular, each test was run for more than 20 minutes so that each emitting device transmitted, at least, 12 000 advertisements and each sensing device executed 10 000 packet scans.

All the experiments were carried out on a set of Motorola Moto E2 devices (Android OS 5.0.2). In addition, in order to evaluate the performance of BLE using different devices, the most important experiments were also replicated on the following smartphones: Honor 8, Honor 9, Moto G, Huawei P9, and LG G5. The experiments were performed with the screen off, no other application running, and the mobile at rest except for the outdoor experiment which was done with the device in motion. To allow the reproducibility of the experiments detailed here, the source code of the described application and the results obtained in each experiment are included as additional material to this paper.

Once all the experiments were carried out, a multivariate analysis was performed to statistically validate the obtained results.

Finally, the feasibility of the proposed algorithms was validated using students. A set of students carried a mobile device with the proposed application running and performed different routines accompanied by others. These experiments allowed us to first define the standard deviation thresholds to be used in the detection of proximity patterns

and second evaluate the correctness of the algorithms in a more realistic environment.

The following sections describe in detail the experiments run and their results.

4.2. Proximity Precision. In order to determine the feasibility of using smartphones to estimate the proximity of other smartphones emitting BLE packets, we performed a set of experiments.

First, we measured the precision of the readings of distance with the devices separated by 1 meter. The objective was to analyse the results so as to be able to detect situations in which two people are next to each other.

Figure 5(a) shows the results obtained when the devices in the experiment were the only ones present so as to avoid external interferences. This plot shows when the devices were placed in direct line of sight, and when they were in the same flat but separated by a brick and mortar wall. As can be seen, when there was a direct line of sight, the results were much more consistent, with less dispersion, than when a wall separated the two devices.

The experiment also measured how the precision might be affected by interference from other emitting devices. Figure 5(b) shows the results of the experiment when performed in the presence of more BLE emitting devices (three and five). This allowed us to assess the feasibility of the proposed system in a more social environment in which multiple users might be present, each of them emitting their own BLE advertising packets. It is clear from the figure that the presence of additional transmitting devices has a far smaller effect on the accuracy of the measurements than the presence of an obstacle like a wall between the devices.

Table 1 summarizes the results of these experiments. It can be observed that the proximity accuracy is reasonable when there are no obstacles between devices and also that the standard deviation in the sensed values is small even when there are more transmitting devices. Results are worse, however, when there is a wall between the devices.

Similar experiments were performed with the devices 3 meters apart in order to analyse this technology's ability to detect situations in which two people are in close proximity. Table 2 summarizes the results of these experiments. Note that because of page restrictions, and in order to improve the readability of the document, the charts with the detected distances are not shown. As can be seen, when there were no obstacles between the devices, the precision and deviation of the measurements are sufficient for the detection of proximity patterns even in the presence of other transmitting devices. However, the precision falls when a wall or a floor was separating the two devices.

In order to evaluate situations in which two people are not in close proximity, similar experiments were performed with the devices distanced by 5 meters. These results are summarized in Table 3. As can be seen, when there is a wall between the devices or they are at different floors, results are much more disperse. Again, the presence of additional transmitting devices affects the precision of the measurements far less than the presence of obstacles.

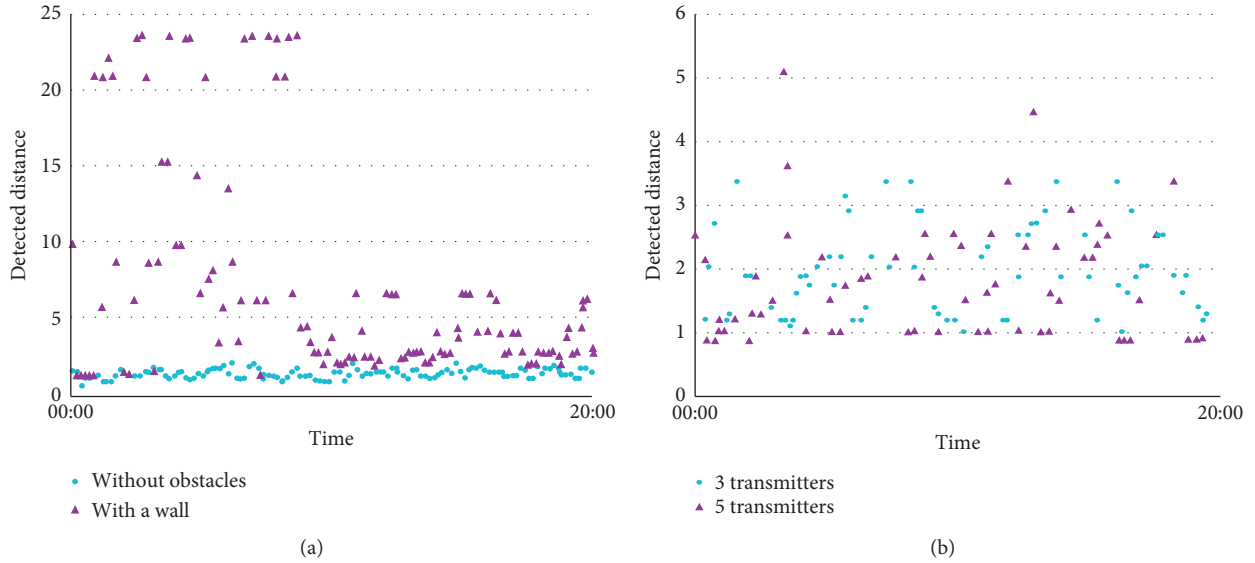


FIGURE 5: BLE devices separated by 1 m. (a) With one transmitter. (b) With several transmitters.

TABLE 1: Average detected distance between BLE devices situated at 1 m.

Situation	Average detected distance (m)	Deviation (σ)
Without obstacles and one transmitter	1.14	0.55
With a wall and one transmitter	6.28	5.33
Without obstacles with three transmitters	1.95	0.69
Without obstacles with five transmitters	1.80	0.91

TABLE 2: Average detected distance between BLE devices situated at 3 m.

Situation	Average detected distance (m)	Deviation (σ)
Without obstacles and one transmitter	2.73	0.99
With a wall and one transmitter	10.27	2.95
Different floor and one transmitter	12.37	4.52
Without obstacles with three transmitters	5.12	2.09
Without obstacles with five transmitters	3.57	1.96

TABLE 3: Average detected distance between BLE devices situated at 5 m.

Situation	Average detected distance (m)	Deviation (σ)
Without obstacles and one transmitter	6.44	2.96
With a wall and one transmitter	31.73	9.36
Different floor and one transmitter	21.81	6.23
Without obstacles with three transmitters	5.26	1.50
Without obstacles with five transmitters	16.13	8.53

The different experiments have been analysed to obtain information useful to detect proximity patterns. Figure 6 shows a summary of the obtained results. This graph details the mean and the standard deviation of the detected measurements for each distance for which tests have been performed. It can be clearly seen that the mean of the measurements is greater than the real distance whenever there is an obstacle between the devices and that, usually, the deviation is also greater. Therefore, the combination of both parameters can be used to identify proximity patterns in which there are no obstacles between users.

Finally, a completely different experiment was performed to complete the analysis. The proposed application is meant to detect the proximity patterns of users both when they are still and when they are moving. All the previously detailed experiments were performed with the devices still, so in this last, experiment our aim was to analyse the behaviour when the devices are being carried by users walking by the side of each other. In some environments, such as elder care, these accompaniments may be required for performing outdoor activities. This experiment also lasted 20 minutes and while the distance between devices was not

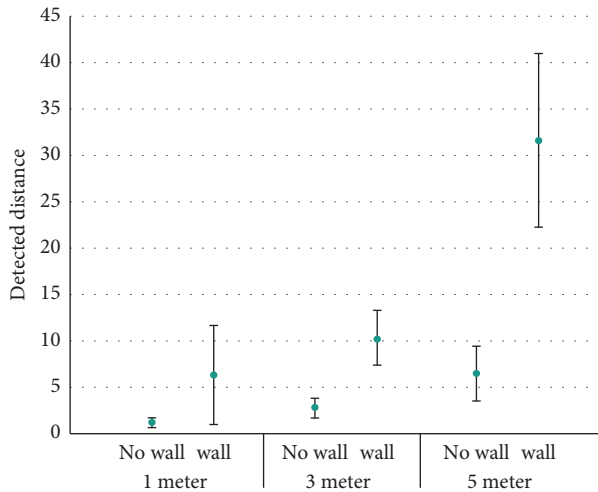


FIGURE 6: Average detected distance and deviation per test.

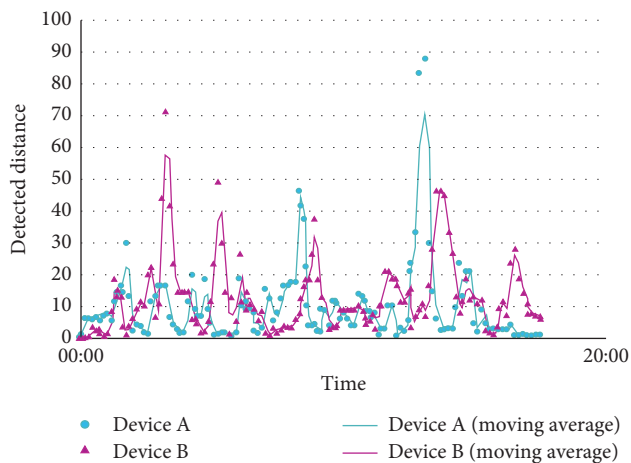


FIGURE 7: BLE devices in movement.

fixed since it would change as the users walked together, at no point in the experiment was it greater than 3 meters.

In Figure 7, the blue dots represent the distance between the devices as sensed from Device A and the pink triangles represent the same distance as sensed from Device B. As can be seen in the figure, the dispersion of the obtained measurements is greater than the ones obtained with the devices stationary. Table 4 summarizes these results.

However, it is interesting to note that most of the greatest deviations in the measured distances between devices are replicated in the two devices at the same time. Examples of this situation are clearly seen in Figure 7, especially around minutes 09:30 and 15:00 of the experiment. Since these deviations were measured by the two devices at the same time, one can assume that they have an external cause, either from an interference of some type or from some property of the place the devices were at that moment.

As a conclusion of the experiments conducted, the results confirm the feasibility of using smartphones to detect proximity patterns. Although the results indicate that one cannot trust the precision of the measured distances when

TABLE 4: Average detected distance between devices in movement.

Situation	Average detected distance (m)	Deviation (σ)
Device A	9.80	12.89
Device B	11.45	11.36

there are obstacles between the two devices, it still allows to determine the closest group of people in ordinary conditions. Moreover, the results show that it is possible to reduce the number of false-positive proximity patterns that are detected when a person is close to another but not really sharing their space because they are on different floors or have a wall between them.

4.3. Communication Reliability. To measure the reliability of using smartphones' BLE capacities in order to communicate the device's presence, we took into account all the communications that took place during the experiments previously described. Figure 8 shows the average number of BLE advertising packets detected in each of the experiments.

Combining this information with the values of the Advertising Interval, Scan Window, and Scan Interval that were used in the experiments detailed above, one can obtain a clear image of the reliability of the communications in the different scenarios.

As can be seen in the figure, the reliability of the communications is unaffected by the presence of obstacles between the devices. However, the number of packets detected is seriously affected by the presence of other BLE transmitting devices. This must be taken into account to prevent false negatives when detecting proximity patterns in the presence of multiple emitting devices. This problem can be mitigated by choosing the appropriate values for the Advertising Interval, Scan Window, and Scan Interval.

From the experiments carried out in this work, and also taking into account the recommendations highlighted by other researchers (such as Montanari et al. [12] and Julien et al. [53]), some guidelines can be identified to reduce the false negatives:

- (i) The Transmission Power controls the distance to which the packets are broadcasted. Initially, this parameter can be configured to its maximum power, since its impacts on battery life are not critical. Nevertheless, if one only wants to detect the patterns of people that are in close proximity, it can be reduced in order to also decrease the number of collisions and of false negatives.
- (ii) The Advertising Time is dependent on the size of the data to be transmitted. Therefore, in order to reduce collisions and false negatives, this size should be as small as possible, decreasing the Advertising Time to the minimum.
- (iii) The Advertising Interval, which is the frequency at which the advertisement is sent, should be configured to 10 Hz or lower in order to decrease the number of collisions. The lower the frequency, the lower the number of packages sent and of collisions

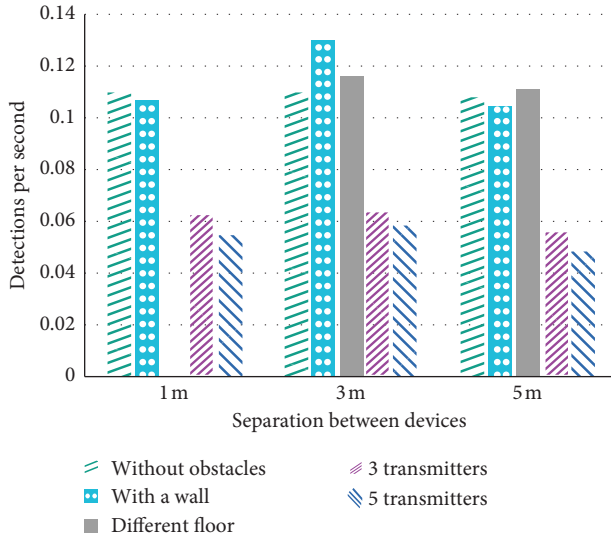


FIGURE 8: Detections per second in the different tests.

but also more difficult is for the receiver device to capture the advertisement.

- (iv) The Scan Window should be long enough to capture the advertisements. If the frequency of the Advertising Interval is low, then a longer Scan Windows may be required in order to detect the proximity among devices. Also, longer Scanning Windows provide better accuracy and higher reliability. Nevertheless, the scanning is a battery consuming operation, so it should be as short as possible in order to reduce the battery consumption.
- (v) The frequency of the Scan Interval should be low in order to decrease the battery consumption. Nevertheless, this value depends on the Advertising Interval. Usually, it is recommended to have a Scan Window that assures the reception of the advertisements (because it is a little bit longer than one Advertising Interval) and Scan Interval with a low frequency in order to reduce the battery consumption.

For detecting temporal proximity patterns, the granularity of the detection does not have to be high, since it is not important to detect the exact second in which two people are no longer together. Therefore, the frequency of the Scan Interval can be even lower. Seamlessly, it is not crucial if some advertisements are lost, since the temporal proximity pattern can be detected. Although the defined configuration reduces the detections per second when there is a higher number of devices, that configuration is good enough to detect the temporal proximity patterns.

4.4. Battery Consumption. Table 5 lists the average battery consumption per second of the different operations used during the experiments.

The first row of the table presents the average consumption of the operating system with the mobile device at rest. We calculated this value in order to eliminate the

background consumption of the operating system from the consumption of the rest of the operations. For the devices used in the experiments, the average battery consumption was $2.45 \mu\text{Ah}$ per second, with a standard deviation of 2.02.

The second row presents the average battery consumption of the devices when transmitting BLE advertising packets ten times per second. In this value, the consumption of the operating system has already been subtracted. As can be seen, this consumption is even lower than the OS idle consumption. This result was to be expected since one of the main benefits of BLE is precisely the low battery consumption, as demonstrated by most of the physical beacons available on the market which present battery lives of up to several years. Additionally, most of the consumption deviation in this operation is attributable to variations in the OS consumption.

Finally, the third row presents the average battery consumption of the devices when listening for BLE advertising packets at the frequency detailed in the description of the experiments. This consumption is an order of magnitude higher than the transmission case. This result was also expected since it coincides with previous studies in the literature [12]. The deviation is also greater for this operation, but not markedly.

To summarize, the battery consumption obtained in the experiments performed, both transmitting and receiving BLE advertising packets, is comparable with the results of similar studies [12, 38, 40]. Moreover, the results confirm the feasibility in resource consumption terms of using smartphones in the dual role of emitting beacons and scanning devices to detect the proximity of companions. The accumulated consumption of performing these operations over time is below the average consumption of social apps as reported by M2AppInsight [54].

As mentioned above, battery consumption is considered a determining factor for an application success. However, there are other resources whose excessive consumption could have a negative impact on the solution. According to [49], battery and network traffic are the most constrained resources in modern mobile devices. The battery consumption of the proposed solutions has already been deemed feasible. Additionally, there is no network traffic generated by the proposed solution since BLE communications are not considered network traffic and all the computations regarding proximity patterns are performed in the device itself. Excessive consumption of other resources like CPU has a direct impact on the battery consumption of the solution. Therefore, as battery consumption is considered appropriate, we can argue that the proposed solution has not an excessive impact on such resources.

4.5. Statistical Analysis. In order to validate statistically the performance analysis described in the previous sections, a multivariate analysis was carried out using the following set of independent variables: the real distance between BLE devices (RealDistance) and the number of transmitting devices (NumTransmitters) as continuous variables and three categorical variables designed to compare the influence

TABLE 5: Consumption per second of the operations for the reception and transmission of BLE packets.

Name	Frequency (Hz)	Battery (μAh)	Deviation (σ)
Operating system	—	2.45	2.02
Transmitting a packet	10	1.32	2.47
Listening for a packet	8.33	14.23	4.38

of potential obstacles between devices—the existence of an unobstructed line of sight between devices (NoObstacles), of obstacles such as a wall (Wall), and when the devices are located on different floors of a building (Floor). For these last three, NoObstacles was taken as the reference variable.

Two dependent variables were considered: the relative root mean square error (RRMSE) and the battery consumption (BatteryConsumption). The former is one of the commonest metrics used to measure the accuracy of continuous variables. In our case, the aim was to evaluate the accuracy of the measurements by comparing the real distances with the distances obtained by the BLE devices. It is defined as follows:

$$\text{RRMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n \left(\frac{x_i - \hat{x}_i}{x_i} \right)^2}, \quad (7)$$

where the terms x_i and \hat{x}_i are the real and the estimated values of the distance between BLE devices, respectively. With the second dependent variable, BatteryConsumption, the goal was to determine the impact of each independent variable on the battery consumption of BLE devices when they are receiving beacons. In sum therefore, two multivariate analyses were performed, one for each dependent variable.

Table 6 presents the ordinary least squares (OLS) regression results for RRMSE, with the coefficient of determination being $R^2 = 0.987$. Statistical significance was found for all the variables considered in the analysis (< 0.05 in all cases). The unstandardized coefficients given in column B of the table show that there exist directly proportional relationships between the value of RRMSE and both of the continuous variables (RealDistance and NumTransmitters). In particular, an increase of one unit in the (real) distance between BLE devices during a test is statistically significantly associated with an increase of 0.729 in the value of RRMSE (0.092 in the case of adding one more transmitter). This means that the error between the real and the detected values increases both with distance and with the addition of new transmitters to the test. The case is similar when the categorical variables Wall and Floor are compared with the reference situation in which there is a clear line of sight between devices (NoObstacles). A wall located between the two devices (Wall) is associated with an increase of more than 7 in the RRMSE compared with the case where there is no obstacle. The fact of the devices being located on different

TABLE 6: OLS regression results with RRMSE as dependent variable.

Model	Variable type	B	Std. error	Beta	t	Sig.
1 (constant)		0.669	0.206		3.247	0.001
RealDistance	Continuous	0.729	0.065	0.165	11.148	0.000
NumTransmitters	Continuous	0.092	0.043	0.054	2.121	0.003
Wall	Categorical	7.536	0.154	0.594	49.068	0.000
Floor	Categorical	3.508	0.204	0.341	13.480	0.000

floors (Floor) has somewhat less impact on the potential error, but is still the cause of a significant increase in the value of RRMSE compared with the case of no obstacles.

The independent variables hardly affect the BatteryConsumption, however, confirming the expected behaviour. Since the number of transmitted advertising packets and the Scan Intervals are independent of the environmental conditions, the battery consumption is always the same (within possible normal deviations). The detailed results of the multivariate analysis for the dependent variable BatteryConsumption are included as part of the additional material of this paper as Supplementary Table 10.

This statistical analysis not only validates the performance study of the system’s precision and battery consumption, but can also be used to improve the algorithm for the identification of temporal proximity patterns. The first step of the algorithm, in which the BLE advertising packets are analysed, can be updated to take into account the results of this statistical analysis. In particular, the algorithm can use this information to enrich the data model it generates by taking into account the statistical probability of the different situations in which a companion may be detected, e.g., without obstacles, on the other side of a wall, and on a different floor. We are also working on using the detected situation to modify at run-time the different beacon transmission and reception parameters in order to increase the accuracy. This enriched data model can then be used to better assess a user’s temporal proximity patterns and thus provide a more detailed virtual profile.

4.6. Feasibility of the Algorithms. Once checked that BLE technology is reliable enough for the identification of temporal proximity patterns, we evaluated the feasibility of the algorithms defined in Section 3. The goals of this validation were as follows:

- (i) To evaluate the adequacy of the data model
- (ii) To establish the standard deviation thresholds to use in the algorithms
- (iii) To evaluate their correctness by determining the ratio of correct detection of temporal proximity patterns, their false-positive (FP) ratio, and their false-negative (FN) ratio

4.6.1. Experimental Setup. In the first phase of testing the algorithms, we decided to evaluate them with students simulating different accompaniment patterns. This first

phase allowed us to establish and validate the thresholds used in the algorithms, in particular for the standard deviation in the measurements. The values selected were as follows:

- (i) The maximum average distance ($\overline{d_{u_i}^{kr}}$) of a companion is 8 meters with an average deviation of 3 meters. We selected these values to identify situations in which companions are 5 meters away (Table 3), which is usually the maximum distance when they are in the same room. In addition, these values allow us to discard those situations in which companions are located in another room or floor (see Tables 1–3).
- (ii) The maximum difference between two consecutive distances (σ_d) is twice the first detected distance. This value allows us to cover variations in the measurements due to interference.
- (iii) The maximum difference between the timestamps of two distances (σ_{ts}) is 5 minutes, preventing us to detect a new pattern when companions are missing for a short interval.
- (iv) The minimum timestamp interval for considering a pattern ($ts_y - ts_x$) is 15 minutes in order to not detect accompaniment patterns when, for instance, a user has a small talk with other users.

To perform the experiment, twelve students were selected. These students were organized into two groups of six people. In each group, one student was selected as user. The device of this user would be the one detecting the temporal proximity patterns to be analysed in the experiment. The other users were companions.

Before beginning the experiment, a workshop was conducted to explain the situations they had to reproduce. These situations were similar to the ones in the test previously shown (i.e., one user and one companion, one user and three companions, and one user and five companions). For evaluating each situation, different characteristics were reproduced:

- (i) A distance between users and companions of 1, 3, and 5 meters
- (ii) Interruption between detections of 4 and 20 minutes
- (iii) Companions in the same room or in different rooms

Every test lasted 60 minutes. Throughout the experiment, students had to record on paper all the actions they performed. Special attention was paid to the times at which users and companions were together so that the identified patterns could be compared with the real life. The comparison of the information gathered by the system with the information manually recorded by the students was used to determine the correct detection of patterns and its FP and FN rates.

4.6.2. Results of the Experiments. Tables 7–9 detail, for each group, the number of patterns identified, and FP and FN

TABLE 7: Patterns identified with one user and one companion.

Group	1 meter			3 meters			5 meters		
	Id	FP	FN	Id	FP	FN	Id	FP	FN
1	3	0	0	2	0	1	2	0	1
2	3	0	0	3	0	0	2	0	1

TABLE 8: Patterns identified with one user and three companions.

Group	1 meter			3 meters			5 meters		
	Id	FP	FN	Id	FP	FN	Id	FP	FN
1	3	0	0	2	0	1	2	0	1
2	3	0	0	3	0	0	3	1	1

TABLE 9: Patterns identified with one user and five companions.

Group	1 meter			3 meters			5 meters		
	Id	FP	FN	Id	FP	FN	Id	FP	FN
1	3	0	0	3	1	1	0	0	3
2	2	0	1	4	1	0	0	0	3

values. Note that for every distance and number of companions, three different patterns should be identified (one pattern showing the user and companions together for a period of time that includes the interruption of four minutes and the other two patterns showing the user and companions together before and after the interruption of 20 minutes (which are considered different patterns) and none when the companions were in a different room).

In order to quantitatively validate the approach, we began by evaluating the pattern detection algorithms (FP and FN rates). As can be seen in Tables 7–9, when companions were one meter away, on average, the algorithms correctly identified the accompaniment patterns. We got 6% of false-negative and 0% of false-positive patterns (i.e., the algorithms correctly identified 17 out of the 18 possible patterns). When the distance increases to three meters, the accuracy is slightly reduced. Concretely, 17 patterns were detected, among which 12% were false positives (2 patterns). Therefore, 83% of the possible patterns were correctly identified. In addition, we obtained 17% of false-negative patterns. With a distance of five meters between student-users and student-companions, the accuracy is reduced but the results are still promising, identifying nine patterns with 11% of false positives and 56% of false negatives.

In the analysis of these results, we identified that the main reason for detecting FP and FN was the identification of outlier values. That is, some measurements wrongly detected because of interference, problems during the transmission, etc. Most of the outliers were removed with the trimmed mean applied in equation (2), but not all of them. This caused that, in some cases, some patterns were not correctly identified because the outlier values were prevented to obtain the minimum interval of 15 minutes, leading to the false negatives. In other tests, the outlier values led to detect two patterns, instead of one, obtaining the false positives. To reduce the number of outliers and to increase the algorithm

accuracy, we are currently evaluating different data cleaning techniques. Nevertheless, the constant monitoring of the user, together with the weight associated with every pattern, could reduce the number of FP and FN.

No patterns were identified when the companions were in a different (but adjacent) room. The algorithms correctly discarded the received measurements because of the increase in the estimated distance and its standard deviation.

Each group also performed a test in motion. However, the smartphones of the student-users did not identify these patterns due to the high deviation of the obtained distances and the increase in the measurements. We are currently working on how to improve the contextual information used to detect these patterns.

Finally, the completeness of the data model was evaluated. To that end, we checked whether the patterns were correctly identified and described and whether any FP or FN was due to the data model. The result was a confirmation of its feasibility and of its expressiveness for defining proximity patterns. Nevertheless, as previously mentioned, in order to identify certain patterns (in particular when users are moving), some additional contextual information (e.g., GPS geopositioning) may be required.

5. Discussion and Conclusions

Many software applications can benefit from having a specific behaviour depending on the preferences and the context of their users. This adaptation is achieved by using the contextual information that can be gathered from the users. The rise of BLE technology has provided a way to detect proximity between devices, information of which can be used to enrich users' contextual information.

In this paper, we have proposed a system that uses BLE to determine temporal proximity patterns. The proposed solution takes advantage of the BLE capabilities of modern smartphones and transforms these devices into virtual beacons that constantly emit advertising packets. These packets are received by other smartphones and processed to compute the user's proximity patterns. To assess the feasibility of the proposal, we performed a set of experiments with the system to measure its precision, its battery consumption, the reliability of the communications, and the accuracy of the algorithms for the identification of proximity patterns. As far as the authors know, there is no other work in the literature that detects this kind of proximity patterns between users without requiring additional infrastructure. As discussed in Section 2, several works use the BLE capabilities of mobile devices to improve the user positioning or to detect proximity between users. However, no prior work systematically extracts the proximity patterns of the users, so they can be added to their profile for later use. The results obtained from the experiments demonstrate that BLE technology can indeed be used to detect and monitor the users' temporal proximity patterns.

5.1. Implications. Although the results showed that BLE is not accurate in the measurements of distance, it is precise

enough to detect people's proximity patterns, especially if the potential deviations are taken into account during pattern detection. In particular, the proximity detection algorithm does not depend on the accuracy of the measurements, as it only uses these measurements to rank other users by proximity, and once a companion has been detected and added to a proximity pattern, the distance is no longer relevant for the system.

It is also important to emphasize that the greatest inaccuracies obtained during the experiments were found when there were obstacles between the devices, as it is said in [29]. This circumstance is far from being negative for our proposal, since this lack of precision when there is a wall between the devices, or when they are on different floors, prevents the system from detecting people that, while close in distance, do not really share time with the user as companions. Moreover, as concluded in [55], other communication standards, such as WiFi or other devices emitting within the same range, can affect the detected distance. Additional contextual information would be required to identify these situations. Nonetheless, in many scenarios, such as elder caring, the subjects are not in a social environment (so that there would be fewer interferences with the devices of other people) and it is more important to determine their proximity with their companions—and the continuity of that proximity—than the exact distance between them.

Regarding the reliability of the communications, the results showed that BLE is reliable enough to detect temporal proximity patterns between users. Even in the worst-case scenarios where other emitting devices caused interferences, the devices were still able to detect one another with sufficient frequency to compute the proximity patterns. In any case, the reliability can be improved by modifying the values of the application's advertising interval, scan window, and scan interval. Such changes, however, would involve a trade-off with worsened battery consumption and would therefore need to be implemented with caution.

The battery consumption in the devices in no way reached alarming levels. Indeed, the results showed battery consumption to be lower than the levels of currently accepted social network applications and therefore should not affect user acceptance.

The algorithms for the identification of proximity patterns can identify most of the situations in which two people share their time together. False positives and false negatives mainly resulted from interferences preventing us to clearly detect whether there is an obstacle between the smartphones or not. Again, more contextual information would be required to identify these situations.

Finally, the results not only demonstrated the feasibility of the system but also served to fine-tune the thresholds and accepted deviations used in the proximity pattern detection algorithm. Appropriate calibration of these values allowed us to improve the algorithm with respect to the false-positive and false-negative rates in the pattern detection. By adjusting the accepted deviation when splitting the pattern timeline into fragments with the information obtained during the experiments, we reduced the number of false-positive

patterns detected that were caused by measurement error. Similarly, by adjusting the thresholds used in the comparison of the stored patterns and timeline fragments with the results obtained in the experiments, we reduced the number of false negatives of pattern detections that were caused by measurement errors.

5.2. Limitations and Future Work. Now that we have verified the feasibility of using smartphones BLE capabilities to detect the temporal proximity patterns of their users, and we are preparing additional validation experiments. The objective of these new experiments is to obtain more precise information about the temporal proximity patterns of users in noncontrolled environments.

We are also currently working on an evolution of the PeaaS model. Our aim is to develop the capacity of smartphones to define and manage their owners' virtual profiles, enabling new applications to be designed. In the present work, the focus was on improving the quality of the contextual information stored in the virtual profiles by adding temporal proximity patterns. For the future, we are working on a new concept termed Situational Context.

The Situational Context is a form of analysing the conditions that exist at a particular time and place in order to predict, at run-time, the expected behaviour of IoT systems [56]. Considering environments in which there are different entities, each of which has a virtual profile, the Situational Context can be defined as the composition of the virtual profiles of all the entities involved in a particular situation. The result of composing the virtual profiles is the combined history of the entities ordered in a single timeline. The Situational Context thus provides a higher level of automation of smart things with people.

5.3. Concluding Remarks. The results presented in this work allow one to use BLE to compute users' temporal proximity patterns. One can then use these patterns to enrich the contextual information stored in each user profile. The proposed solution can be applied in both indoor and outdoor situations without the need to deploy any kind of infrastructure in the areas where it will be used.

This is a small, but relevant step in our goal of bringing technology closer to nontechnical users. The richer the contextual information gathered from the user, the farther one can advance in seamlessly integrating IoT systems into the users' daily lives.

Data Availability

The source code of the presented applications and the data produced by the performed experiments are included within the supplementary information files.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by 4IE+ project (0499_4IE_PLUS_4_E) funded by the Interreg V-A España-Portugal (POCTEP) 2014–2020 Program, by the Spanish Ministry of Science, Innovation and Universities (RTI2018-094591-B-I00 (MCIU/AEI/FEDER, UE) and PGC2018-094905-B-I00), by the Department of Economy and Infrastructure of the Government of Extremadura (GR18112, IB18030), and by the European Regional Development Fund Andalucía 2014–2020 (UMA18-FEDERJA-180).

Supplementary Materials

BatteryConsumptionOLSregression.pdf contains the results of the multivariate analysis for the dependent variable BatteryConsumption, as mentioned in Section 4.5. Logs folder contains spreadsheets with the information obtained in all the experiments detailed in Section 4, so results can be validated. BLEApp folder contains the source code of the Android Application used to perform the experiments detailed in Section 4, so the experiments can be replicated. (*Supplementary Materials*)

References

- [1] J. Berrocal, J. Garcia-Alonso, J. M. Murillo, and C. Canal, "Rich contextual information for monitoring the elderly in an early stage of cognitive impairment," *Pervasive and Mobile Computing*, vol. 34, pp. 106–125, 2017.
- [2] A. B. Rodriguez, A. R. Tena, J. Garcia-Alonso, J. Berrocal, R. Flores Rosco, and J. M. Murillo, "Using beacons for creating comprehensive virtual profiles," in *Ubiquitous Computing and Ambient Intelligence*, vol. 10, pp. 295–306, Springer, Berlin, Germany, 2016.
- [3] H. Byun and C.-K. Kim, "When friends move: a deep learning-based approach for friendship prediction in mobility network," in *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services*, ACM, New York, NY, USA, pp. 520–521, 2019.
- [4] Bluetooth Special Interest Group, "Bluetooth 4.0. Core specification," <https://www.bluetooth.com/specifications/adopted-specifications>.
- [5] J. Guillen, J. Miranda, J. Berrocal et al., "People as a service: a mobile-centric model for providing collective sociological profiles," *IEEE Software*, vol. 31, no. 2, pp. 48–53, 2014.
- [6] M. C. González, C. A. Hidalgo, and A.-L. Barabási, "Understanding individual human mobility patterns," *Nature*, vol. 453, no. 7196, pp. 779–782, 2008.
- [7] A. Noulas, S. Scellato, R. Lambiotte, M. Pontil, and C. Mascolo, "A tale of many cities: universal patterns in human urban mobility," *PLoS One*, vol. 7, no. 5, 2012.
- [8] P.-U. Tournoux, J. Leguay, F. Benbadis, V. Conan, M. D. De Amorim, and J. Whitbeck, "The accordion phenomenon: analysis, characterization, and impact on dtm routing," in *Proceedings of the 28th Conference on Computer Communications*, IEEE, Rio de Janeiro, Brazil, pp. 1116–1124, April 2009.
- [9] M.-O. Killijian, R. Pasqua, M. Roy, G. Trédan, and C. Zanon, "Souk: spatial observation of human kinetics," *Computer Networks*, vol. 111, pp. 109–119, 2016.

- [10] C. Chen, J. Ma, Y. Susilo, Y. Liu, and M. Wang, "The promises of big data and small data for travel behavior (aka human mobility) analysis," *Transportation Research Part C: Emerging Technologies*, vol. 68, pp. 285–299, 2016.
- [11] Y. Yuan and M. Raubal, "Extracting dynamic urban mobility patterns from mobile phone data," in *Proceedings of the International Conference on Geographic Information Science*, Springer, Columbus, OH, USA, pp. 354–367, September 2012.
- [12] A. Montanari, S. Nawaz, C. Mascolo, and K. Sailer, "A study of bluetooth low energy performance for human proximity detection in the workplace," in *Proceedings of the International Conference on Pervasive Computing and Communications (PerCom)*, IEEE, Big Island, HI, USA, 2017.
- [13] C. Cattuto, W. Van den Broeck, A. Barrat, V. Colizza, J.-F. Pinton, and A. Vespignani, "Dynamics of person-to-person interactions from distributed rfid sensor networks," *PLoS One*, vol. 5, no. 7, Article ID e11596, 2010.
- [14] S. Jamil, A. Basalamah, L. Ahmed, and M. Youssef, "Hybrid participatory sensing for analyzing group dynamics in the largest annual religious gathering," in *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, ACM, Osaka, Japan, pp. 547–558, 2015.
- [15] N. Eagle and A. Pentland, "Social serendipity: mobilizing social software," *IEEE Pervasive Computing*, vol. 4, no. 2, pp. 28–34, 2005.
- [16] N. A. Eagle and A. S. Pentland, "Reality mining: sensing complex social systems," *Personal and Ubiquitous Computing*, vol. 10, no. 4, pp. 255–268, 2006.
- [17] X. Hu, T. H. S. Chu, V. C.-M. Leung, E. C.-H. Ngai, P. Kruchten, and H. C. B. Chan, "A survey on mobile social networks: applications, platforms, system architectures, and future research directions," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 3, pp. 1557–1581, 2015.
- [18] F. Naya, H. Noma, O. Ren, and K. Kogure, "Bluetooth-based indoor proximity sensing for nursing context awareness," in *Proceedings of the 2005 Ninth IEEE International Symposium on Wearable Computers*, IEEE, Osaka, Japan, pp. 212–213, October 2005.
- [19] J. García-Alonso, J. Berrocal, D. M. Juan Manuel Murillo, C. Fonseca, and M. Lopes, "Situational-context for virtually modeling the elderly," in *Proceedings of the 9th International Symposium on Ambient Intelligence, ISAmI 2018*, P. Novais and J. J. Jung, Eds., pp. 298–305, Toledo, Spain, June 2018.
- [20] P. Bellavista, A. Corradi, M. Fanelli, and L. Foschini, "A survey of context data distribution for mobile ubiquitous systems," *ACM Computing Surveys*, vol. 44, no. 4, pp. 1–45, 2012.
- [21] M. Raskino, J. Fenn, and A. Linden, *Extracting Value from the Massively Connected World of 2015*, Gartner Research, Stamford, CT, USA, 2005.
- [22] L. Atzori, A. Iera, and G. Morabito, "From "smart objects" to "social objects": the next evolutionary step of the internet of things," *IEEE Communications Magazine*, vol. 52, no. 1, pp. 97–105, 2014.
- [23] J. S. Silva, P. Zhang, T. Pering, F. Boavida, T. Hara, and N. C. Liebau, "People-centric internet of things," *IEEE Communications Magazine*, vol. 55, no. 2, pp. 18–19, 2017.
- [24] H.-S. Park, K. Oh, and S.-B. Cho, "Bayesian network-based high-level context recognition for mobile context sharing in cyber-physical system," *International Journal of Distributed Sensor Networks*, vol. 7, no. 1, Article ID 650387, 2011.
- [25] T. M. Gronli, G. Ghinea, and M. Younas, "Context-aware and automatic configuration of mobile devices in cloud-enabled ubiquitous computing," *Personal Ubiquitous Computing*, vol. 18, no. 4, pp. 883–894, 2014.
- [26] P. Makris, D. N. Skoutas, and C. Skianis, "A survey on context-aware mobile and wireless networking: on networking and computing environments' integration," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 1, pp. 362–386, 2013.
- [27] M. A. Kabir, J. Han, J. Yu, and A. Colman, "User-centric social context information management: an ontology-based approach and platform," *Personal and Ubiquitous Computing*, vol. 18, no. 5, pp. 1061–1083, 2014.
- [28] L. Šikšnys, J. R. Thomsen, S. Šaltenis, and L. Y. Man, "Private and flexible proximity detection in mobile social networks," in *Proceedings of the 2010 Eleventh International Conference on Mobile Data Management*, IEEE, Kansas City, MI, USA, pp. 75–84, 2010.
- [29] P. Utsch and T. Liebig, "Monitoring microscopic pedestrian mobility using bluetooth," in *Proceedings of the 2012 8th International Conference on Intelligent Environments (IE)*, IEEE, Guanajuato, Mexico, pp. 173–177, June 2012.
- [30] S. Han, Y. Park, and H. Kim, "Extending bluetooth LE protocol for mutual discovery in massive and dynamic encounters," *IEEE Transactions on Mobile Computing*, vol. 18, no. 10, pp. 2344–2357, 2019.
- [31] A. Kashevnik and M. Shchekotov, "Comparative analysis of indoor positioning systems based on communications supported by smartphones," in *Proceedings of the 12th Conference of FRUCT Association*, Tampa, FL, USA, 2014.
- [32] M. K. Kjergaard, M. Wirz, D. Roggen, and G. Tröster, "Mobile sensing of pedestrian flocks in indoor environments using wifi signals," in *Proceedings of the 2012 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, IEEE, Lugano, Switzerland, pp. 95–102, 2012.
- [33] A. K. M. Mahtab Hossain and W.-S. Soh, "A survey of calibration-free indoor positioning systems," *Computer Communications*, vol. 66, pp. 1–13, 2015.
- [34] S. S. Chawathe, "Beacon placement for indoor localization using bluetooth," in *Proceedings of the International IEEE Conference on Intelligent Transportation Systems*, IEEE, Washington, DC, USA, pp. 980–985, 2008.
- [35] J. Hightower and G. Borriello, "Location systems for ubiquitous computing," *Computer*, vol. 34, no. 8, pp. 57–66, 2001.
- [36] J. Zhu, K. Zeng, K. H. Kim, and P. Mohapatra, "Improving crowd-sourced wi-fi localization systems using bluetooth beacons," in *Proceedings of the 2012 9th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, pp. 290–298, Seoul, Republic of Korea, 2012.
- [37] A. Zidek, S. Tailor, and R. H. Bellrock, "Anonymous proximity beacons from personal devices," in *Proceedings of the 2018 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, Athens, Greece, March 2018.
- [38] K. Mikhaylov, N. Plevritakis, and J. Tervonen, "Performance analysis and comparison of bluetooth low energy with IEEE 802.15.4 and SimpliCI," *Journal of Sensor and Actuator Networks*, vol. 2, no. 3, pp. 589–613, 2013.
- [39] L. Jia, C. Chen, and Y. Ma, "Modeling and performance analysis of device discovery in bluetooth low energy networks," in *Proceedings of the Global Communications Conference (GLOBECOM)*, IEEE, Anaheim, CA, USA, pp. 1538–1543, December 2012.
- [40] L. Jia, C. Chen, Y. Ma, and Y. Xu, "Energy analysis of device discovery for bluetooth low energy," in *Proceedings of the 2013 IEEE 78th Vehicular Technology Conference (VTC Fall)*, IEEE, Las Vegas, NV, USA, September 2013.

- [41] A. Abouzied and J. Chen, "Commonities: a context-aware nudge towards social interaction," in *Proceedings of the Companion Publication of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing, CSCW Companion*, New York, NY, USA, 2014.
- [42] M. Girolami, F. Mavilia, F. Delmastro, and E. Distefano, "Detecting social interactions through commercial mobile devices," in *Proceedings of the 2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, Athens, Greece, March 2018.
- [43] E. R. Sykes, P. Stephen, and S. Nardi, "Context-aware mobile apps using iBeacons: towards smarter interactions," in *Proceedings of the 25th Annual International Conference on Computer Science and Software Engineering, CASCON '15*, pp. 120–129, Riverton, NJ, USA, 2015.
- [44] D. De, P. Bharti, S. K. Das, and S. Chellappan, "Multimodal wearable sensing for fine-grained activity recognition in healthcare," *IEEE Internet Computing*, vol. 19, no. 5, pp. 26–35, 2015.
- [45] T. M. T. Do and D. Gatica-Perez, "Human interaction discovery in smartphone proximity networks," *Personal and Ubiquitous Computing*, vol. 17, no. 3, pp. 413–431, 2013.
- [46] P. Martin, B.-J. Ho, N. Grupen, S. Muñoz, and M. Srivastava, "An iBeacon primer for indoor localization: demo abstract," in *Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings*, New York, NY, USA, November 2014.
- [47] H. Qian and D. Andresen, "Extending mobile device's battery life by offloading computation to cloud," in *Proceedings of the 2015 2nd ACM International Conference on Mobile Software Engineering and Systems, MOBILESoft 2015*, A. Abadi, D. Danny, and Y. Dubinsky, Eds., Florence, Italy, May 2015.
- [48] A. Merlo, M. Migliardi, and L. Caviglione, "A survey on energy-aware security mechanisms," *Pervasive and Mobile Computing*, vol. 24, pp. 77–90, 2015.
- [49] J. Berrocal, J. Garcia-Alonso, and C. Vicente-Chicote, "Early analysis of resource consumption patterns in mobile applications," *Pervasive and Mobile Computing*, vol. 35, pp. 32–50, 2017.
- [50] Radius Networks, Android Beacon Library, 2019, <https://altbeacon.github.io/android-beacon-library/>.
- [51] Radius Networks, Android Beacon Library Apps, 2019, <https://altbeacon.github.io/android-beacon-library/apps.html>.
- [52] O. S. Android, "Wakelock," 2019, <http://developer.android.com/reference/android/os/PowerManager.WakeLock.html>.
- [53] C. Julien, C. Liu, and A. L. Murphy, "Practical continuous neighbor discovery for bluetooth low energy," in *Proceedings of the 16th ACM/IEEE International Conference on Information Processing in Sensor Networks*, P. Zhang, P. Dutta, and G. Xing, Eds., , Pittsburgh, PA, USA, April 2017.
- [54] M2AppInsight, "M2AppInsight quarterly app performance report: global android app category averages," 2015, <http://m2appinsight.com/m2appinsight-quarterly-app-performance-report-global-android/-app-category-averages/>.
- [55] 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+'14)*, pp. 201–210, Tampa, FL, USA, September 2014.
- [56] J. Berrocal, J. Garcia-Alonso, C. Canal, and J. M. Murillo, "Situational-context: a unified view of everything involved at a particular situation," in *Proceedings of the Web Engineering—16th International Conference, ICWE 2016*, pp. 476–483, Lugano, Switzerland, June 2016.