Estimating indoor occupancy through low-cost BLE devices

Florenc Demrozi, *Member, IEEE*, Fabio Chiarani, Cristian Turetta, Philipp H. Kindt, *Member, IEEE*, and Graziano Pravadelli, *Senior Member, IEEE*

***Abstract*—Detecting the presence and estimating the number of subjects in an indoor environment has grown in importance recently. For example, the information if a room is unoccupied can be used for automatically switching off the light, air conditioning, and ventilation, thereby saving significant amounts of energy in public buildings. Most existing solutions rely on dedicated hardware installations, which involve presence sensors, video cameras, and carbon dioxide sensors. Unfortunately, such approaches are costly, subject to privacy concerns, have high computational requirements, and lack ubiquitousness. The work presented in this article addresses these limitations by proposing a low-cost system for occupancy detection. Our approach builds upon detecting variations in Bluetooth Low Energy (BLE) signals related to the presence of humans. The effectiveness of this approach is evaluated by performing comprehensive tests on 5 different datasets. We apply different pattern recognition models and compare our methodology with systems building upon IEEE 802.11 (WiFi). On average, in different environments, we can correctly classify the occupancy with an accuracy of 97.97%. When estimating the number of people in a room, on average, the estimated number of subjects differs from the actual one by 0.32 persons. We conclude that the performance of our system is comparable to existing ones based on WiFi, while leading to a significantly reduced cost and installation effort. Hence, our approach makes occupancy detection practical for real-world deployments.**

***Index Terms*—Bluetooth Low Energy (BLE), Indoor occupancy detection , Indoor occupancy counting, Pattern recognition.**

I. Introduction

The deployment of smart buildings has gained momentum in recent years, thanks to the wide spread of low-cost and accurate sensing and actuating devices, controlled by advanced artificial intelligence-based systems. This has promoted the development of several smart applications in many relevant scenarios, like, for example, health care [1], [2], assistance of elderly and people with special needs [3]–[5], human activity recognition [6], [7], heating, ventilation, and air-conditioning systems [8], recognition of the environmental status [9], and in general smart home solutions [10].

One of the key problems in these scenarios is detecting if a room is occupied. When this information is available in real-time, relevant decisions can be quickly and automatically taken. For example, air-conditioning/heating and lighting can be switched off once a room is empty. If the number of people in a room can be estimated, properties such as the ventilation can be controlled adaptively, thereby improving the air quality and further reducing the energy consumption. The occupancy information is also relevant in case of an emergency, when rescue actions should be directed towards rooms in which the presence of people has been detected. Hence, automatically detecting the presence of people and estimating their quantity has been studied frequently in recent times [11]–[14].

Related work: Technologies for occupancy detection and estimation of the number of people in a room can be broadly categorized into a) room installations, e.g., [13], and b) body-worn devices, e.g., [11]. Body-worn solutions rely on sensors that are worn on the body of each subject in a room. These devices emit wireless signals, which are analyzed by receivers installed in the environment. However, the assumption that every person always wears such a device when entering a certain room is unrealistic in practical scenarios. For this reason, recent research has focused on techniques that rely only on devices that are deployed within the room infrastructure [7], [15], without requiring any body-worn devices. Techniques that solely rely on room installations exploit analyzing properties such as, e.g., the CO_2 level, audio signals, video images, or radio signal propagation patterns. Another feasible way are indirect measures, such as the air quality, which can be assessed using electromagnetic fields [16]. Most of the published approaches rely on inferring occupancy information from the Channel State Information (CSI) of WiFi networks. Occupancy detection using wireless signals is a challenging problem, especially because of the multitude of activities and positions that people can occupy within an environment, leading to a large variety of different signal patterns. For example, Yang et al. [17] presented a real-time, device-free, and privacy-preserving WiFi-enabled IoT platform for occupancy sensing. This approach makes use of CSI information and can achieve a detection accuracy of 96.8%. Depatla et al. [18] propose a methodology to identify two distinctive patterns in the CSI data, which are related to people in a room: blocking the Line Of Sight (LOS) and scattering. Based on this, it is possible to estimate the total number of occupants, achieving an accuracy of 96%. Similarly, Zou et al. [13] presented an occupancy detection methodology based on CSI data. It measures the similarity between adjacent CSI time series and reaches an accuracy of 99.1%. Chen et al. [19] proposed an occupancy detection system, which relies on analyzing the changes in the statistical metrics of the power consumed by the building

This research work has been partially supported by the project Dipartimenti di Eccellenza 2018-2022 funded by the Italian Ministry of Education, Universities and Research (MIUR). (Corresponding author: Florenc Demrozi.)

F. Demrozi, F. Chiarani, C. Turetta, P. H. Kindt and G. Pravadelli are with the Computer Science Department, University of Verona, Italy (e-mail: name.surname@univr.it).

(i.e., the electricity for ventilation and lightning). Similarly, Akbar et al. [20] presented an occupancy detection system based on the electric power consumption. Here, machine learning algorithms, such as kNN and SVM were applied. This approach claims an average accuracy of 94%. Furthermore, BLE-based systems have been widely studied. Mateos et al. [21] presented a body-worn methodology that uses BLE beacons (with ≈ 10 Hz broadcast frequency) and smartphones to determine the level of occupancy in indoor and outdoor spaces, achieving an average accuracy of 95%. Chen et al. [22] presented a stochastic methodology that simultaneously uses BLE and WiFi data. It uses a frequency of 1 Hz and utilizes connection/traffic information to estimate the number of users and the environment status.

Limitations of existing approaches: Most of the previously known approaches require special hardware, which in most cases needs to have high computational power. In addition, dedicated devices have to be installed into the rooms under surveillance. In particular, commercially available WiFi devices, such as routers and laptops, typically do not provide access to CSI data. Only very few WiFi System on Chips (SoCs) natively provide access to the CSI data, and all of them are deprecated. For this reason, most recent approaches build upon Broadcom's BCM43 series, for which a firmware patching framework [23], [24] unlocks access to the CSI data. Hence, special WiFi receivers need to be installed into a building only for the purpose of gathering CSI data. Since WiFi radios are typically power-hungry, they need access to the electricity grid. In addition, the patched firmware prevents such devices from actively communicating over WiFi networks, since they can only act as observers that extract WiFi signals. Hence, such receivers also require access to a wired network for relaying the gathered CSI data to a server. In summary, occupancy detection systems based on CSI incur a significant installation effort and hence cost. Furthermore, they need to be planned and installed in advance, and cannot be flexibly used on the fly when the necessity of occupancy detection arises at short notice. Finally, the main limitation of existing BLE-based methodologies is that they are mostly integrated into body-worn systems interacting with an existing WiFi/BLE infrastructure. In addition, their sampling frequency is usually lower than 10 Hz, which negatively impacts their accuracy.

Contributions of this paper: To overcome the previous limitations, this paper presents an easily accessible occupancy detection and people counting platform based on a mobile Android application and BLE devices (e.g., using the low-cost nRF52832 SoC).

The proposed system, which relies on a few BLE devices, can be flexibly deployed in a room, e.g., by gluing them onto the walls. Such devices can run from a battery for multiple months without recharging. Hence, they do not need any access to the electricity grid, and also do not need to be connected to the wired network. An Android smartphone, a Single Board Computer (SBC) or a PC, which can send the occupancy results to a server via WiFi, acts as a BLE receiver. Our detection system analyzes the received BLE signals using pattern recognition techniques. It is driven by the insight that occupancy causes variations in the radio signal propagation patterns, which can be observed in the Received Signal Strength Indicator (RSSI). The main characteristics of the proposed approach are the following ones.

- Low-cost: Each node costs only a few USD (e.g., the nRF52832 SoC costs ≈ 2 USD), which is only a fraction of the cost of a WiFi AP (i.e., ≈100 USD). The receiver can be a standard Android smartphone, some of them are available below 100 USD;
- Non invasive: Users do not have to carry any devices on their body and cameras are not used;
- Compatible: Our approach works properly with any BLE sender/receiver that provides the ability of measuring the RSSI with a frequency of at least 45Hz;
- Ubiquitous and flexible: Being a mobile, "pocket-size" system, our approach is suitable for environments where there is no existing infrastructure;
- Accurate: The performance of the proposed approach is comparable to that of existing CSI-based systems.

Overall, the main contributions of this work are as follows.

- We propose a BLE-based occupancy detection system, which comes with considerably lower cost and installation effort than existing approaches;
- We evaluate its performance by using multiple real-world measurements and by applying a workflow exploiting several pattern recognition algorithms (i.e., regression and classification algorithms), fed by both feature representations and raw measurement data;
- We experimentally compare its performance by using state-of-the art CSI-based information. Our results suggest that the much simpler RSSI signal of BLE is sufficient for both accurately detecting the occupancy of a room, as well as for estimating the number of people inside of it.

Our own previous work: This paper is an extension of our conference paper [25]. Besides many details, we have added an evaluation that compares our BLE-based approach against a method that exploits state-of-the-art CSI-based information.

Organization of the paper: The rest of the paper is organized as follows. Section II provides the necessary background. Section III details the proposed methodology. Section IV discusses experimental results. Finally, concluding remarks are reported in Section V.

II. Preliminaries

This section introduces the necessary background, on which our proposed approach is built upon.

A. Received signal strength indicator

In radio communication technologies, the Received Signal Strength Indicator (RSSI) indicates the signal power measured by the receiver. In BLE, the RSSI is an integer value that indicates the received power in dBm. The RSSI is often exploited for fingerprinting [26], [27] in localization applications, where a certain signature of RSSI values is used to identify a known location. In this paper, we exploit different RSSI propagation patterns to identify whether a room is occupied and to estimate the number of people inside of it. The analysis of the gathered data through dedicated pattern recognition techniques makes it possible to identify patterns that correspond to the number of people in the environment and, in some cases, to the activity that these people are carrying out (e.g., walking, laying down, or sitting) [17].

B. Channel state information

WiFi networks following the 802.11ac standard use Orthogonal Frequency Division Multiplexing (OFDM) for data transmission. Here, every channel utilizes a bandwidth of 20, 40, 80 or 160 MHz [28]. Each channel is subdivided into 64 (for 20 MHz channels) to 512 (for 160 MHz channels) subcarriers. Each subcarrier uses its own, distinct frequency within the channel, and data is transmitted simultaneously on all of these subcarriers in parallel. Let X be the signal emitted on a certain subcarrier, and Y the corresponding signal that has arrived at the receiver. Then, typically the following relation is assumed [29].

$$Y = H \cdot X + N \tag{1}$$

Here, N represents noise, whereas H is the *channel state information (CSI)*. When a signal is transmitted over the wireless link, it is attenuated and/or undergoes a phase change. Both effects are quantified by the CSI H. Hence, H for a single WiFi frame is a vector that contains a complex number for every subcarrier, describing how the amplitude and phase of the signal has changed. The presence of human bodies and their movements have a major impact on H. Therefore, it is possible to infer information on the room occupancy from CSI signals.

In our experiments, we use a bcm43455c0 WiFi radio with a modified firmware based on the Nexmon firmware patching framework [23], [24]. We use 20 MHz channels with 64 subcarriers. Hence, for every received WiFi frame, we obtain 64 amplitude and phase values, of which 56 are related to actual data transmissions [28]. In this work, we use both the phase and amplitude information sent by multiple different access points in a room for inferring whether the room is occupied. Towards this, we use the classifiers described in Section III-C, and compare the results to those obtained from a system that builds on BLE and RSSI using the same classifiers.

C. Regression and classification algorithms

Regression and classification are artificial intelligence-based techniques, which apply a supervised learning algorithm on labeled training data. After having learned from the labeled data, an unknown input can be classified, or a property that has some dependency on the input data can be predicted [30].

Classification aims to decide which choice among a set of classes explains best a certain, previously unknown input. It is the most common operation in machine learning and depends strongly on the data representation (i.e., raw data, manually or automatically extracted features). The quality of this classification is commonly assessed by the *precision* P, *specificity* S, *recall* R, also called *sensitivity*, and overall *accuracy* A. These metrics are defined as follows [31]:

$$\begin{array}{ll} P = \frac{tp}{tp+fp} & S = \frac{tn}{fp+tn} \\ R = \frac{tp}{tp+fn} & A = \frac{tp+tn}{p+n} \end{array} \tag{2}$$

Here, tp represents the number of true positives, fp the number of false positives, n the total number of negatives, and p the total number of positives. Precision and recall quantify how the classifier can avoid false positives and correctly classify all of the samples that belong to a specific class, while specificity quantifies the ability to classify true negatives correctly. Finally, accuracy is the number of precisely predicted data samples out of all the data samples.

While the goal of classification is determining to which class a certain observation belongs, regression attempts to predict a value based on an observation of the input data. A set of independent variables that form the input are called "predictors" or "features" [30]. The accuracy of the regression model is measured in terms of Root Mean Square Error (RMSE) and Mean Absolute Error (MAE), calculated as given by the following equation [31]:

$$\begin{array}{l} RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(|y_i - x_i|)^2} \\ MAE = (\frac{1}{n})\sum_{i=1}^{n}|y_i - x_i| \end{array} \tag{3}$$

Here, x_i identifies the actual outcome of the input data, y_i the estimated outcome, and n the number of samples under consideration.

III. Methodology

The methodology we propose for occupancy detection and people counting consists of the following three phases (cf. Figure 1).

1) Offline RSSI data collection for the training of the pattern recognition models. The RSSI is related to the communication between multiple BLE senders (e.g., smartphones, BLE beacons or SBCs) and a single BLE receiver (e.g., smartphone or SBC);
2) Training of multiple pattern recognition models by using the data obtained from the previous phase;
3) Online evaluation of the pattern recognition models for predicting the occupancy in real-time.

The methodology makes use of a communication architecture exploiting a central receiver (tipically a smartphone or a SBC) that collects RSSI data from several BLE devices located in the environment. The received data are then forwarded to a database on a server. The goal is to study the changes in the radio signal propagation patterns generated between the senders and the receiver to estimate the environment status (i.e., environment occupancy, number of people, distance between emitters and receiver). For this purpose, we identify the most appropriate recognition/regression model among multiple choices.

A. Experimental setup

Initially, the BLE devices establish a synchronized connection to the receiver. The receiver periodically forwards data

Timestamp	$MacAddr_1$	$MacAddr_2$	$MacAddr_3$	$MacAddr_4$	...	$MacAddr_n$	Occupancy	Nr. Occupants
26/08/2020 09:56:45.005	-51	-65	-80	-100	...	-35	true	2
26/08/2020 09:56:45.010	-41	-55	-70	-90	...	-45	true	4
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
26/08/2020 10:00:00.000	-37	-49	-65	-70	...	-35	false	0
Distance (cm)	25	500	100	300	...	600		

TABLE I
DATASET OF RSSI MEASUREMENTS

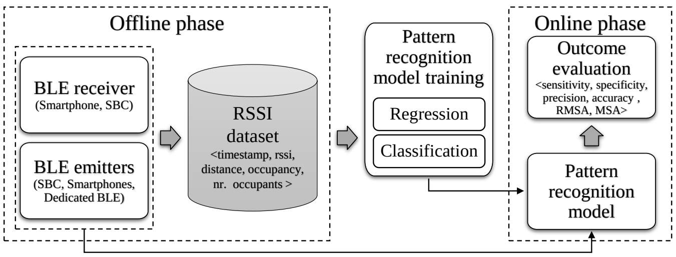


Fig. 1. Schematic view of the proposed methodology.

to a server for offline data processing and pattern recognition model training. Though the approach will work with a large variety of BLE devices, in our experiments, we used Nordic Thingy™ 52 (nRF6936) nodes as transmitters. The data transmission frequency (i.e., the BLE connection interval) between these nodes and the BLE receiver is fixed. In our experiments, we evaluated different data transmission frequencies in the range between 5 Hz and 200 Hz. Our approach supports arbitrary transmitter positions in the environment, with the only restriction that the location cannot vary between the offline training and the online prediction phase. We use an Android smartphone as the receiving device. Hence, our methodology supports simple ad-hoc installations without requiring any fixed infrastructure, since all devices are battery powered and do not need access to a wired network. Considering the requirements of future deployments, Android has been chosen due to its compatibility with several pattern recognition libraries (e.g., Keras, Tensorflow, or Weka).Though we have carried out the classification and regression on a computer in our experiments, the resulting system supports deployments in which only the training phase is executed on a server, whereas the detection can be done directly on the smartphone. This eliminates need for the smartphone maintaining an internet connection.

B. Offline data collection

During the offline data collection phase, the receiver receives the data collected by the BLE sensors distributed in the environment at a fixed sampling frequency. Each received packet is timestamped and associated to its corresponding RSSI measurement. For saving power, we accumulate the samples received by the phone and periodically transmit them to the server in a batch. Finally, we manually label this data with the number of persons present in the environment at each timestamp. We also manually enter the distance between each emitter and the receiver.

The resulting dataset for training the classifiers, as stored on the server, is exemplified in Table I. Each column MAC_{Addr_i}, $i = 1, 2, .., n$, refers to the RSSI measurements for the emitter with MAC address i. The column *occupancy* is *false* when the number of people in the environment is 0, and *true* otherwise. The Column *Nr. Occupants* contains the number of people inside the observed environment at the corresponding timestamp. The number of people inside the environment is provided by an external, human observer who updates the number of occupants every time they leave or enter the environment. These labels are used for supervised learning. The *Distance* row contains the distance between the emitter i and the receiver. RSSI measurements represent the input (i.e., *predictors*) for the pattern recognition models used during the online evaluation. Similarly, the occupancy state and the number of people in the environment are the target outcome (i.e., *outcome variables*) that we attempt to predict. In other words, these are the labels for supervised learning. Finally, to mitigate the effects of a potential "overfitting" of the model, we eliminate any duplicates present in the dataset, maintaining only the first occurrence.

C. Pattern recognition model training

In this step, the collected data are used to train a pattern recognition model for the following scenarios:

1) Recognition of the occupancy status of the environment;
2) Estimation of the number of people inside the environment.

The overall workflow for this phase is shown in Figure 2. The resulting model is capable of predicting the occupancy status and the people count online, based on RSSI data from live measurements. Initially, the collected RSSI data are segmented into time-windows of 1 s. Subsequently, from each segment, we extract a comprehensive set of features, as given by Table II, using the software library presented in [32].

We have chosen time windows of 1 second, because we assume that the occupancy status cannot change at shorter time scales. For each such window, we compute 159 features per BLE transmitter. At the end of the segmentation and feature extraction phase, we have obtained two datasets: one that consists of raw RSSI data, which we call the *raw dataset* in the following, and RSSI feature data obtained as described above, which we call the *feature dataset*. Both of them are used in the analysis reported below. Before this analysis, we "clean"/"normalize" the data by removing outliers. We thereby use the *robust scalar* method, which works as given by the next equation:

$$D_{nor} = \left\{ x_{nor} : \forall x \in D,\ x_{nor} = \frac{x - Q_2^D(x)}{Q_3^D(x) - Q_1^D(x)} \right\} \quad (4)$$

Here, Q_1^D, Q_2^D, Q_3^D are the first, second (aka median), and third quartiles of the dataset D, x is a sample of D, and D_{nor}

Time Domain	Frequency Domain
maximum, minimum, mean, standard deviation, root mean square, range, median, skewness, kurtosis, time-weighted variance, interquartile range, empirical cumulative density function, percentiles (10, 25, 75, and 90), sum of values above or below percentile (10, 25, 75, and 90), mean amplitude deviation, mean power deviation, autoregression.	fast fourier transform (FFT) coefficients, discrete fourier transform (DFT), discrete wavelet transform (DWT), first dominant frequency, ratio between the power at the dominant frequency and the total power, ratio between the power at frequencies higher than 3.5 Hz and the total power, wavelet entropy values, energy band.

TABLE II
MOST IMPORTANT TIME AND FREQUENCY DOMAIN FEATURES USED IN OUR ANALYSIS.

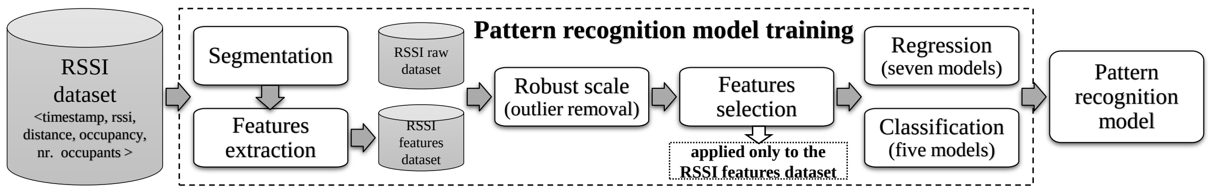


Fig. 2. Pattern recognition model training workflow.

is the dataset D after applying the robust scale outlier removal technique.

Feature selection: Not all extracted features contribute to a high detection accuracy. A higher number of features does not always imply a greater accuracy of the pattern recognition model; similarly, a smaller number of features does not always lead to a reduction of the accuracy. Therefore, we identified those features that actually contribute most to the quality of the classification. For this purpose, we applied feature selection techniques known from the literature [33]. The main benefits derived from the removal of unnecessary/misleading features (e.g., features with very low variance, duplicates of existing features, or high noise) are: i) reduction of overfitting; ii) reduction of noise-related errors; iii) improvement of the accuracy; iv) reduction of the training time [34]. The proposed methodology makes use of a tree-based feature selection technique, applied only to the RSSI feature dataset. Tree-based estimators, by definition, internally create an ordering of the features representing the training dataset, which makes them very suitable to be used by feature selection methods. More details on tree-based feature selection can be found in [33].

After feature extraction, the training phase takes place. It creates the models for occupancy detection and occupancy counting. More information on the training phase are reported in Section III-D.

1) Occupancy detection: Occupancy detection represents a binary classification problem, where *false* means the environment is empty, and *true* that there is at least one person in the environment. In particular, five different classification models (i.e., k-Nearest Neighbor (kNN), Weighted kNN (WkNN), Linear Discriminant Analysis (LDA), Quadratic LDA (QLDA), and Support Vector Machine (SVM)) are tested.

The aim of such a scenario is to estimate the occupancy status over a time window of 1 second by using only the RSSI feature dataset, since the used classifiers do not handle raw time series data. The results of this classification problem are evaluated by using the quality metrics we have introduced in Equation 2.

2) Occupancy counting: Unlike occupancy detection, which is a binary classification problem, the occupancy counting scenario is an estimation problem. Here, we aim to identify the number of people being present in the environment. Multivariate regression analysis (i.e., Gradient Boosting, Random Forest, Linear, Ridge, RANSAC, Bayesian, and TheilSen) on both the raw and feature dataset predict one variable (viz., the Nr. of occupants), based on multiple input variables (raw data samples and features) [30]. We study two different cases. In the first case, we only consider raw data. Here, the model returns an estimation of the number of occupants every 5 ms (when a sampling rate of 200Hz is used). In the second case, we only consider feature data, obtaining an estimation of the number of occupants every second (i.e., once per window length used for data segmentation). The results of this prediction are evaluated by using the quality metrics we have introduced in Equation 3.

D. *Training and evaluation*

To evaluate the quality of our pattern recognition models, we partition the dataset into *training* and *testing* data. In particular, for each considered scenario, the raw RSSI or feature dataset is initially split (according to the hold-out procedure [34]) into a training dataset (75% of all measurements) and a testing dataset (25% of all measurements). On the training dataset, we run the pattern recognition model training phase, as described above in Section III-C. We use a k-fold cross validation procedure with $k = 3, 5, 10$ [34]. Finally, we use the testing dataset to further examine the used models. In the end, the procedure returns the above-mentioned evaluation metrics, viz., RMSE and MAE for regression, and sensitivity, specificity, precision, and accuracy for classification.

IV. EXPERIMENTAL RESULTS

For evaluating the proposed methodology, an extensive set of experiments and analyses have been conducted. The goal is to evaluate which of the trained models leads to the best classification and prediction in the scenarios we consider.

A. *Characteristics of the analyzed datasets*

Table III shows the characteristics of the datasets we collected in three different environments (i.e., a university classroom, a home living-room, and an industrial laboratory) by adopting the setup described in Section III-A. Columns 1 to

Dataset ID	# of senders	Use case scenario	Environment size l(m) x w(m) x h(m)	Min # of people	Max # of people	Min distance (cm)	Max distance (cm)	# of samples	Size (Mb)
1	6(BLE)	baseline	8.8x8.6x3.2=75.6m^2	0	0	25	500	892k	261
2	5(BLE)	detection	8.8x8.6x3.2=75.6m^2	0	6	100	600	494k	344
3	5(BLE)	counting	8.8x8.6x3.2=75.6m^2	0	6	100	600	1483k	181
4	4(BLE)/5(AP)	detection/counting	6.6x4x2.75=26.4m^2	0	2	200	330	1500k	1000
5	4(BLE)/12(AP)	detection/counting	18x6x6=108m^2	0	5	800	800	7000k	2000

on average, each access point (AP) has 4 interfaces for the 2.4GHz and 4 interfaces for the 5GHz band.

TABLE III
DATASETS CHARACTERISTICS (DATASETS 1 TO 3 USE ONLY RSSI MEASUREMENTS. DATASETS 4 AND 5 USE BOTH RSSI AND CSI DATA).

4 contain the dataset ID, the number of radio transmitters distributed in the environment, i.e., BLE devices or WiFi access points (APs), the considered scenario, and the dimensions of the environment. Columns 5 to 8 show the lowest and highest number of people inside the environment during the considered period of time, and the minimal and maximal distance between the transmitters and the receiver. Finally, Columns 9 and 10 show the number of samples and the size of the dataset.

1) Experimental environments: We next describe the 3 environments we used in our experiments. In all of them, the position of the emitters and receivers were fixed. The users participating in the experiment did not know the positions of the devices and could not touch or modify them.

Datasets 1, 2 and 3: The experiments corresponding to these datasets were carried out in a university classroom (8.8 m x 8.6 m x 3.2 m) with 15 working stations. They involved 6 subjects: 1 female (29 years, 1.58 m height) and 5 males (25-29 years, 1.75-1.95 m height). We used 1 smartphone as a receiver and 5 or 6 BLE beacons as senders, as specified in Table III.

Datasets 4 and 5: The purpose of these datasets is to compare RSSI data of a BLE piconet to CSI data of an IEEE 802.11ac network (WiFi). In particular, to extract CSI data, we used a Raspberry Pi 4/B+ that interacts with the nearest APs of the environment. For CSI data extraction, we used the Nexmon firmware patching framework [23], [24]. We developed a custom software in C++ for recording and storing this data.

The experiments for Dataset 4 were carried out in a home living room (6.6 m x 4 m x 2.75m). They involved 2 subjects: 1 female (54 years, 1.66 m height) and 1 male (26 years, 1.80 m height).

The tests for Dataset 5 were carried out in an Industrial Computer Engineering (ICE) laboratory (18 m x 6 m x 6 m), containing one production line, several electronic machinery, and various devices, such as sensors and actuators, that communicate using different communication protocols. Our tests for Dataset 5 involved 1 female (27 years, 1.80m) and 4 males (25-29 years, 1.75-1.95 m height).

Overall, all the datasets together contain 4 hours of collected data and require 5GB of storage.

B. Occupancy detection

In this section, we evaluate the proposed occupancy detection technique. To adequately mimic as many of the different situations that occur during normal use of a room, we have carried out the following experiments. Subjects entered and left the environment in an undefined, random order, with the only constraint that each of them must stay in the environment for at least one minute. Besides, they have carried out the following different activities i) all were standing still, ii) all were in motion simultaneously, iii) all were sitting simultaneously, iv) some where moving, while some were sitting, and v) in Dataset 5, everyone of them took a position in one of the working stations.

The achieved results are reported in Table IV. The results were computed by processing Datasets 1, 2, and 3 as if they belonged to a single contiguous dataset. Data contained in these 3 datasets are made up of 68% non-empty environment instances (i.e., the room was occupied) and of 32% empty environment instances (i.e., they represent an empty room).

We evaluated 5 different classification models by using the RSSI feature datasets, as depicted in Figure 2. The outcome represents the environment status: *false* when the environment is empty, *true* if at least one person is in the environment. In Table IV, Columns 2 to 5 show the results in terms of specificity (S), recall (R), precision (P), and comprehensive accuracy (A). Overall, the SVM model with a linear kernel

Model	Specificity (S)	Recall (R)	Precision (P)	Accuracy (A)
kNN	98.53%	99.07%	99.07%	99.07%
WkNN	98.17%	98.97%	98.97%	98.97%
LDA	99.83%	99.70%	99.70%	99.70%
QLDA	99.78%	99.77%	99.77%	99.77%
SVM	99.91%	99.92%	99.91%	99.92%

TABLE IV
OCCUPANCY DETECTION RESULTS (DATASET 1, 2 AND 3).

achieved the most noticeable results, i.e., 99.92% recall (empty environment), 99.91% specificity (not-empty environment), 99.91% precision, and 99.92% accuracy. Compared to all other models we considered, the SVM model requires higher computational capabilities; however, the Keras library [35] provides a Quasi-SVM model implementation for Android-based mobile devices, with sufficiently low computational complexity to be run on a smartphone.

By examining the classification outcome in more detail, we observed that the incorrectly classified samples are related to the situation in which people inside the environment are all seated, regardless of their number.

It is worth mentioning that we have presented the results for a detection in *real-time*, with a detection delay of around 1 s. A longer allowed delay of e.g., 30 s, would be sufficient for most applications, e.g., for controlling the ventilation and air conditioning. This would also allow for sampling windows of 30 s, for which we expect a much higher detection accuracy, also when all subjects are sitting.

C. Occupancy counting

Occupancy counting is realized using dedicated regression algorithms, as already described. As for occupancy detection, we study the quality of this scenario using Datasets 1, 2, and 3 with 7 different regression models. Table V presents the results obtained by using both the raw and feature datasets. The outcome is an estimation of the number of persons that are staying in the environment.

Our results suggest that the Random Forest regression model achieves the best results on the feature dataset. In particular, given a set of features based on RSSI measurements, the proposed occupancy counting system can estimate the number of people in the environment with an RMSE of 0.4 and an MAE of 0.3. In other words, in almost all cases, the Random Forest estimator can correctly identify the number of the environment occupants, with an error of at most ± 1 person.

Regression model	Raw data RMSE	Raw data MAE	Features RMSE	Features MAE
Gradient boosting	0.9	0.6	0.4	0.3
Random forest	0.7	0.4	0.4	0.3
Linear	1.4	1.0	1.3	1.0
Ridge	1.4	1.0	2.2	4.1
RANSAC	1.8	1.3	3.1	3.1
Bayesian	1.4	1.0	1.9	1.9
TheilSen	1.9	1.2	2.0	1.8

TABLE V
OCCUPANCY COUNTING RESULTS (DATASETS 1, 2 AND 3).

When using only the raw dataset, we achieved an RMSE of 0.7 and a MAE of 0.4. Hence, the computation of the features is justified by the increased regression quality. As for the occupancy detection scenarios, the estimation error is amplified when all people inside the environment are sitting.

It is worth noting that unlike many other existing works, in which a limited number of environmental situations are studied, our goal is to account for the most realistic environmental situations, of which a group of sitting people is definitely part.

D. RSSI vs. CSI

As already described, the results presented so far concern BLE networks and are computed by using RSSI data. In this section, we evaluate the proposed methodology by comparing the performance obtained when using CSI data on WiFi instead of RSSI data. For this purpose, we created Datasets 4 and 5, which contain both RSSI and CSI data for the same environments.

Table VI shows the results we obtained for Datasets 4 and 5 for occupancy detection. Column 1 lists the 5 considered classifiers. Columns 2 to 5 (RSSI-based) and 6-9 (CSI-based) show the achieved results when using Dataset 4, while Columns 10 to 13 (RSSI-based) and 14-17 (CSI-based) show the results for Dataset 5. Similarly, Table VII depicts the obtained results for Datasets 4 and 5 for occupancy counting. We can summarize the achieved results by comparing CSI vs. RSSI as follows:

- As expected, the results for Dataset 4 are much better than those for Dataset 5. This is mainly due to the environment size and shape. In particular, various electrical machines were present and potentially interfering in the environment of Dataset 5. In addition, the large height of the environment has negatively affected the results;
- Running the classifiers on the feature dataset achieved more reliable results than when using the raw dataset;
- In our experiments, on average the SVM classifier achieved the most reliable results concerning the occupancy detection problem. The Random forest regression algorithm achieved the best results concerning the occupancy counting problem;
- The more complex and comprehensive CSI data lead to an only marginally higher detection accuracy than our proposed BLE-based system, as can be seen in Tables VI and VII. Hence, there is little benefit in setting up a more complex and expensive WiFi-based detection system. It is worth mentioning that in the studied environments, the number of information sources for CSI was larger than that for BLE (i.e., four vs. five in Dataset 4 and four vs. twelve in Dataset 5).

The overall accuracy of the proposed methodology, among all the 3 tested environments and taking into account only the most accurate classifiers, is 97.97% for occupancy detection and a RMSE of 0.32 for occupancy counting in the mean of all the 5 tested datasets.

E. Sampling frequency and energy consumption

The sampling frequency used for the results presented above was 200 Hz for BLE and 45 Hz for the WiFi APs. Concerning BLE devices, a lower sampling frequency would also reduce the energy consumption of the BLE transmitters and also of the receiver. Since the battery lifetime also affects the usability of our approach, in this section we study the energy consumption in detail. We first discuss the energy characteristics of the hardware we have used. We evaluated the average RAM and memory demand, as well as the energy consumption for the transmitters (i.e., Nordic Thingy 52 with a 64 MHz Cortex M4 MCU, 512 Kb Flash, 64 Kb RAM, and a Battery of 1440 mAh) and the receiver (Oneplus 6 with a Qualcomm SDM845 Snapdragon 845, featuring an Octa-core 4x2.8 GHz Kryo 385 Gold & 4x1.7 GHz Kryo 385 Silver CPU, 128 GB memory, 8 GB RAM, with Android 10, OxygenOS 10.3.7, and a 3300 mAh Battery).

Energy consumption of the transmitters: When operating using a sampling frequency of 200 Hz, the BLE devices can efficiently operate for more than seven days without recharging their battery. A lower sampling frequency would further extend their battery lifetime significantly [36]. We therefore reduced the sampling frequency in all considered scenarios and studied the resulting detection accuracy. We found that the detection quality degrades quickly for sampling frequencies below 45 Hz. For frequencies above this, there is no further gain in the accuracy. We therefore recommend a sampling frequency of 45 Hz.

Energy consumption of the receiver: The energy consumption of the smartphone is only marginally affected by the energy needed for BLE communication [37]. For our smartphone, we found the following values: 54 mAh concerning battery consumption, 122 Mb/h of memory, and 59 Mb/h of RAM.

Model	S	R	P	A	S	R	P	A	S	R	P	A	S	R	P	A
kNN	100%	100%	100%	100%	100%	98%	99%	98%	90%	91%	91%	91%	98%	99%	98%	98%
WkNN	100%	100%	100%	100%	97%	98%	98%	98%	90%	91%	91%	91%	97%	99%	98%	99%
LDA	99%	99%	99%	99%	100%	100%	100%	100%	89%	89%	89%	89%	99%	99%	99%	99%
QLDA	98%	99%	99%	99%	99%	98%	98%	98%	53%	58%	57%	57%	96%	96%	96%	96%
SVM	99%	99%	99%	99%	98%	99%	99%	99%	94%	94%	94%	94%	99%	99%	99%	99%
	Dataset 4 RSSI-based				Dataset 4 CSI-based				Dataset 5 RSSI-based				Dataset 5 CSI-based			

TABLE VI
OCCUPANCY DETECTION RESULTS (DATASETS 4 AND 5) - BLE VS. CSI.

Regression	Raw Data		Features		Raw Data		Features		Raw data		Features		Raw data		Features	
Model	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
Gradient boosting	0.19	0.09	0.13	0.05	0.14	0.05	0.11	0.04	1.22	0.92	1.02	0.62	0.31	0.22	0.29	0.18
Random forest	0.18	0.05	0.16	0.04	0.08	0.01	0.04	0.01	0.91	0.47	0.91	0.52	0.83	0.37	0.32	0.27
Linear	0.46	0.36	0.33	0.22	0.81	0.50	0.94	0.30	1.62	1.34	1.31	1.21	0.92	0.64	1.02	0.54
Ridge	0.46	0.36	0.34	0.22	0.59	0.35	0.60	0.40	1.68	1.37	1.71	1.34	1.44	0.99	1.28	0.68
RANSAC	0.45	0.37	0.35	0.25	0.59	0.35	0.61	0.41	1.80	1.45	1.75	1.38	1.44	0.99	1.36	0.72
Bayesian	0.46	0.37	3.40	2.30	0.59	0.35	9.21	4.31	1.65	1.34	7.10	4.72	1.75	1.23	1.68	1.12
TheilSen	47.0	1.37	14.7	29.9	46.1	3.64	1.12	0.33	1.85	1.47	6.55	7.21	18.7	11.1	23.8	16.2
	Dataset 4 RSSI-based				Dataset 4 CSI-based				Dataset 5 RSSI-based				Dataset 5 CSI-based			

TABLE VII
OCCUPANCY COUTING RESULTS (DATASETS 5 AND 6) - BLE VS. CSI.

V. CONCLUDING REMARKS AND FUTURE WORK

Occupancy detection and occupancy counting provide important information for smart cities and smart building environments in several scenarios. However, existing solutions have many limitations, which are mainly related to high economic cost, low accessibility, high computational requirements, difficulties of installation, and lack of ubiquitousness. This paper presented a pattern recognition-based methodology that uses low-cost BLE communication technology for occupancy detection and counting. It can be retrofitted into any environment with negligible installation effort. Different regression and classification algorithms were used, achieving promising results in different environments. In particular, occupancy can be detected, taking into account only the best classifier, with an average accuracy of 97.97% over all datasets. The number of people in a room can be estimated with a average RSME/MAE of 0.32/0.28 people. Moreover, we showed that our methodology working on BLE RSSI data achieves practically the same accuracy as WiFi/CSI-based approaches. At the same time, it comes with a much lower cost and installation effort.

The datasets we used in our experiments have been created with a relatively limited number of persons. This is due to regulations in response to the ongoing SARS-CoV-2 pandemic, which prevent us from placing more persons into the same environment. While detecting the occupancy is expected to work even more reliably when a larger number of persons is present in a room, the occupancy counting method needs to be evaluated further when the SARS-CoV-2 restrictions have been eased.

Our objective for future research is to reduce the number of senders while maintaining the same performance. At the same time, it is desirable to reduce the dependence of the methodology on the sender positions and the explicit knowledge of their distances from each other. This might be done e.g., by estimating their distance automatically.

Some smartphone models, such as the Honor 7 or Samsung Galaxy S5 and S7, fail to keep up with the required sampling rates. The limitations of such devices are mainly related to the operating system's version. To reduce the power consumption, some custom OS and HW versions do not allow the dedicated application to extract RSSI values at frequencies higher than 40 Hz. On the contrary, devices that do not present such limitations are, for example, all Oneplus devices and the Samsung Galaxy S8. In future research, we attempt to address this by further reducing the sampling rate requirements, making it compatible with literally all smartphone models.

Finally, our current system carries out the online classification on a server. However, our detection algorithms are lightweight enough to be run on a smartphone. Hence, the server is only needed for the learning phases, while the online detection can be done on the smartphone in the future.

REFERENCES

[1] G. Acampora, D. J. Cook *et al.*, "A survey on ambient intelligence in healthcare," *Proceedings of the IEEE*, vol. 101, no. 12, pp. 2470–2494, 2013.

[2] B. Tan, Q. Chen *et al.*, "Exploiting wifi channel state information for residential healthcare informatics," *IEEE Communications Magazine*, vol. 56, no. 5, pp. 130–137, 2018.

[3] F. Demrozi, R. Bacchin *et al.*, "Towards a wearable system for predicting the freezing of gait in people affected by parkinson's disease," *IEEE journal of biomedical and health informatics*, 2019.

[4] P. Rashidi and A. Mihailidis, "A survey on ambient-assisted living tools for older adults," *IEEE journal of biomedical and health informatics*, vol. 17, no. 3, pp. 579–590, 2013.

[5] F. Demrozi, V. Bragoi *et al.*, "An indoor localization system to detect areas causing the freezing of gait in parkinsonians," in *ACM/IEEE DATE*, 2019, pp. 952–955.

[6] F. Demrozi, G. Pravadelli *et al.*, "Human activity recognition using inertial, physiological and environmental sensors: A comprehensive survey," *IEEE Access*, pp. 1–1, 2020.

[7] Z. Chen, C. Jiang *et al.*, "Building occupancy estimation and detection: A review," *Energy and Buildings*, 2018.

[8] G. A. Ehlers and J. Beaudet, "System and method of controlling an hvac system," Oct. 31 2006, uS Patent 7,130,719.

[9] I. M. Pires, R. Santos *et al.*, "Recognition of activities of daily living based on environmental analyses using audio fingerprinting techniques: A systematic review," *Sensors*, vol. 18, no. 1, p. 160, 2018.

[10] D. J. Cook, M. Youngblood *et al.*, "Mavhome: An agent-based smart home," in *IEEE PerCom 2003*, 2003, pp. 521–524.

[11] J. Yang, A. Pantazaras *et al.*, "Comparison of different occupancy counting methods for single system-single zone applications," *Energy and Buildings*, vol. 172, pp. 221–234, 2018.

[12] S. Golestan, S. Kazemian *et al.*, "Data-driven models for building occupancy estimation," in *ACM Int. Conf. on Future Energy Systems*, 2018, pp. 277–281.

[13] H. Zou, Y. Zhou *et al.*, "Device-free occupancy detection and crowd counting in smart buildings with wifi-enabled iot," *Energy and Buildings*, vol. 174, pp. 309–322, 2018.

[14] C. Gentner, D. Günther *et al.*, "Identifying the BLE advertising channel for reliable distance estimation on smartphones," *CoRR*, vol. abs/2006.09099, 2020. [Online]. Available: https://arxiv.org/abs/2006.09099

[15] D. Trivedi and V. Badarla, "Occupancy detection systems for indoor environments: A survey of approaches and methods," *Indoor and Built Environment*, p. 1420326X19875621, 2019.

[16] D. Hasenfratz, S. Sturzenegger *et al.*, "Spatially resolved monitoring of radio-frequency electromagnetic fields," in *Proceedings of First International Workshop on Sensing and Big Data Mining*, 2013, pp. 1–6.

[17] J. Yang, H. Zou *et al.*, "Device-free occupant activity sensing using wifi-enabled iot devices for smart homes," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3991–4002, 2018.

[18] S. Depatla, A. Muralidharan *et al.*, "Occupancy estimation using only wifi power measurements," *IEEE J. on Selected Areas in Communications*, vol. 33, no. 7, pp. 1381–1393, 2015.

[19] D. Chen, S. Barker *et al.*, "Non-intrusive occupancy monitoring using smart meters," in *Workshop on Embedded Systems for Energy-Efficient Buildings*, 2013, pp. 1–8.

[20] A. Akbar, M. Nati *et al.*, "Contextual occupancy detection for smart office by pattern recognition of electricity consumption data," in *IEEE ICC*, 2015, pp. 561–566.

[21] M. Mateos Sánchez, R. Berjón Gallinas *et al.*, "A tool to calculate the level of occupancy in indoor and outdoor spaces using ble and open data to be published in real-time," *Sensors*, vol. 20, no. 14, p. 3916, 2020.

[22] J. Chen, H. Chen *et al.*, "Collecting building occupancy data of high resolution based on wifi and ble network," *Automation in Construction*, vol. 102, pp. 183–194, 2019.

[23] M. Schulz, D. Wegemer *et al.* (2017) Nexmon: The c-based firmware patching framework. [Online]. Available: https://nexmon.org

[24] F. Gringoli, M. Schulz *et al.*, "Free your CSI: A channel state information extraction platform for modern Wi-Fi chipsets," in *International Workshop on Wireless Network Testbeds, Experimental Evaluation& Characterization (WiNTECH)*, 2019, p. 21–28.

[25] F. Demrozi, F. Chiarani *et al.*, "A low-cost ble-based distance estimation, occupancy detection and counting system," in *ACM/IEEE DATE*, 2021.

[26] A. Khalajmehrabadi, N. Gatsis *et al.*, "Modern wlan fingerprinting indoor positioning methods and deployment challenges," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1974–2002, 2017.

[27] F. Demrozi, K. Costa *et al.*, "A graph-based approach for mobile localization exploiting real and virtual landmarks," in *IFIP/IEEE VLSI-SOC*. IEEE, 2018, pp. 249–254.

[28] Rohde & Schwarz, "802.11ac technologyintroductionwhite paper," March 2012, cdn.rohde-schwarz.com/pws/dl_downloads/dl_application/application_notes/1ma192/1MA192_7e_80211ac_technology.pdf.

[29] Z. Wang, J. Jiang *et al.*, "A survey on CSI-based human behavior recognition in through-the-wall scenario," *IEEE Access*, vol. 7, pp. 78 772–78 793, 2019.

[30] C. M. Bishop *et al.*, *Neural networks for pattern recognition*. Oxford university press, 1995.

[31] D. M. Powers, "Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation," –, 2011.

[32] M. Barandas, D. Folgado *et al.*, "Tsfel: Time series feature extraction library," *SoftwareX*, vol. 11, p. 100456, 2020.

[33] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Computers & Electrical Engineering*, vol. 40, no. 1, pp. 16–28, 2014.

[34] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.

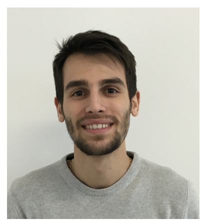
[35] F. Chollet *et al.*, "keras," 2015.

[36] P. H. Kindt, D. Yunge *et al.*, "Energy modeling for the bluetooth low energy protocol," *ACM Trans. Embed. Comput. Syst. (TECS)*, vol. 19, no. 2, Mar. 2020.

[37] P. Kindt, T. Chakraborty *et al.*, "How reliable is smartphone-based electronic contact tracing for covid-19?" *arXiv preprint arXiv:2005.05625*, 2020. [Online]. Available: https://arxiv.org/abs/2005.05625

Florenc Demrozi, Ph.D in computer science, IEEE member, received the B.S. and M.E. degrees in Computer Science and Engineering from the University of Verona, Italy, respectively in 2014 and 2016, and the Ph.D. degree in Computer Science in 2020. He is currently a Postdoctoral researcher and Temporary Professor at the Department of Computer Science, University of Verona, Italy, working on Ambient Intelligence (AmI), Ambient Assisted Living (AAL) and Internet of Things (IoT).

Fabio Chiarani, received the B.S. degrees in Computer Science from the University of Verona, Italy, in 2020. He is currently a M.E. student in Computer Science and Engineering from the University of Verona, Italy. His main interests focus on Edge Computing and IoT.



Cristian Turetta, received the B.S. and M.E. degrees in Computer Science and Engineering from the University of Verona, Italy, respectively in 2017 and 2020. He is currently a research fellow at the Department of Computer Science, University of Verona, Italy working on Ambient Assisted Living (AAL), Internet of Things (IoT) and IoT Security.

Philipp H. Kindt, received his Ph.D. in Electrical Engineering from Technical University of Munich (TUM) in 2019. His research interests are wireless communication, mobile computing and the IoT.

Graziano Pravadelli, PhD in computer science, IEEE senior member, IFIP 10.5 WG member, is full professor of information processing systems at the Computer Science Department of the University of Verona (Italy) since 2018. In 2007 he cofounded EDALab s.r.l., an SME working on the design of IoT-based monitoring systems. His main interests focus on system-level modeling, simulation and semi-formal verification of embedded systems, as well as on their application to develop IoT-based virtual coaching platforms for people with special needs. In the previous contexts, he collaborated in several national and European projects and he published more than 120 papers in international conferences and journals.