



Article

Pattern Recognition in Older Adults' Activities of Daily Living

Gonçalo Augusto ^{1,†} , Rui Duarte ^{1,2,*} , Carlos Cunha ^{1,2,†} and Ana Matos ^{1,2,†}

¹ CISeD—Research Centre in Digital Services, Polytechnic of Viseu, 3504-510 Viseu, Portugal; pv20260@alunos.estgv.ipv.pt (G.A.); cacunha@estgv.ipv.pt (C.C.); amatos@estgv.ipv.pt (A.M.)

² Research Center in Digital Services, 3504-510 Viseu, Portugal

* Correspondence: pduarte@estgv.ipv.pt

† These authors contributed equally to this work.

Abstract: Monitoring daily activities and behaviors is essential for improving quality of life in elderly care, where early detection of behavioral anomalies can lead to timely interventions and enhanced well-being. However, monitoring systems often struggle with scalability, high rates of false positives and negatives, and lack of interpretability in understanding anomalies within collected data. Addressing these limitations requires an adaptable, accurate solution to detect patterns and reliably identify outliers in elderly behavior data. This work aims to design a scalable monitoring system that identifies patterns and anomalies in elderly activity data while prioritizing interpretability to make well-informed decisions. The proposed system employs pattern recognition to detect and analyze outliers in behavior analysis, incorporating a service worker generated with Crontab Guru for automated data gathering and organization. Validation is conducted through statistical measures such as accumulated values, percentiles, and probability analyses to minimize false detections and ensure reliable performance. Experimental results indicate the system achieves high accuracy, with an occupancy probability across compartments and fewer outliers detected. The system demonstrates effective scalability and robust anomaly detection. By combining pattern recognition with a focus on interpretability, the proposed system provides actionable insights into elderly activity patterns and behaviors. This approach enhances the well-being of older adults, offering caregivers reliable information to support timely interventions and improve overall quality of life.

Keywords: pattern recognition; anomaly detection; outlier detection; data organization; older adults monitoring



Citation: Augusto, G.; Duarte, R.; Cunha, C.; Matos, A. Pattern Recognition in Older Adults' Activities of Daily Living. *Future Internet* **2024**, *16*, 476. <https://doi.org/10.3390/fi16120476>

Academic Editor: Petros Patias

Received: 8 November 2024

Revised: 8 December 2024

Accepted: 9 December 2024

Published: 20 December 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Pattern recognition and outlier detection have become integral components of various fields, ranging from finance [1–3] to healthcare [4–6]. These concepts aim to extract meaningful information from data and identify patterns that may go unnoticed. The foundation for successful analysis lies in having a well-organized dataset. The ability to interpret and explain models is valuable for understanding decision-making processes and building trust and confidence in their outputs. Interpretability and explainability [7] enable stakeholders, including domain experts, policymakers, and end-users, to comprehend and validate the results and make informed decisions based on them. With this, pattern recognition and outlier detection are powerful tools for extracting knowledge from data. A well-organized dataset is the backbone of these techniques, providing the necessary information to train accurate models. Whether utilizing supervised or unsupervised learning approaches [8] these methods can reveal hidden patterns and detect outliers, leading to valuable insights and improved decision-making.

Interpretability and explainability are crucial aspects in the field of ML. Whereas ML models have shown remarkable performance in various domains, their complex nature often makes understanding the reasons behind their decisions challenging. Interpretability refers to comprehending and explaining how an ML model arrives at its predictions

or classifications. On the other hand, explainability focuses on providing clear and understandable explanations for those predictions. In the context of this work—on pattern recognition and outlier detection—interpretability and explainability play significant roles. When dealing with well-organized datasets, interpretability allows us to understand the underlying relationships and patterns captured by models. By examining the features and patterns the model identifies, valuable insights are obtained into the problem domain. In the case of recognizing patterns in a person's daily life activities, interpretability can help to understand which features contribute the most to specific actions or behaviors, such as the impact of room occupancy probabilities and timestamps on particular routines.

Explainability becomes particularly important when dealing with unsupervised learning techniques. In unsupervised learning, where no explicit labels or guidance exist, explainability allows understanding the hidden structures and patterns the model discovers autonomously. By explaining the identified patterns and anomalies, we can better understand unusual behavior or outliers in a person's daily routines. This understanding can provide valuable feedback or intervention when anomalies are detected.

In the context of this paper, each user provides an independent dataset where all the data of a user relates only to that user. To acquire individual data, a smartwatch is used, and a service worker is responsible for periodically fetching all the new data for each user and saving it in a .csv file within a single folder stored in the system. This service worker was created with the help of the Crontab Guru tool [9]. This tool helps to schedule and automate the execution of tasks, such as running a Python script on a Unix-like operating system. It simplifies the process of creating cron expressions, which define when and how frequently a task should run.

In conclusion, this paper presents a system that can organize the data to recognize the patterns in older adults' daily life activity and, consequently, detect potential outliers in their behavior. The following sections will explain how this study is organized.

2. Related Work

The detection of anomalies in behavior is a critical aspect of activities of daily living (ADLs) in older adults. It involves identifying unusual patterns in behavior, which could indicate a potential problem, such as a fall, a medical emergency, or a change in health condition. In recent years, several studies have focused on developing algorithms and methods to detect anomalies in behavior using various sensors and data analysis techniques.

In [10], an algorithm to detect falls in older adults using wearable sensors is presented. The algorithm uses a combination of ML techniques and signal processing to detect abnormal movements and postures that are characteristic of falls. The algorithm was evaluated using real-world data collected from older adults, and the results showed high accuracy in detecting falls.

In [11], an algorithm is developed to detect anomalies in eating behavior using a smartwatch. The algorithm uses data from the accelerometer and gyroscope sensors in the smartwatch to identify patterns of hand-to-mouth movements, which are indicative of eating behavior. The algorithm was evaluated using data collected from a group of volunteers, and the results showed a high accuracy in detecting eating behavior.

In addition to detecting falls and eating behavior, several studies have also focused on detecting anomalies in other ADLs, such as sleeping, walking, and medication adherence. For instance, ref. [12] proposed an algorithm to detect sleep disturbances in older adults using data from a wearable device. The algorithm uses ML techniques and signal processing to analyze data from the device's accelerometer and gyroscope sensors to detect abnormal sleep patterns. Similarly, ref. [13] proposed an algorithm to detect walking anomalies in two different groups of adults: one with cognitive impairment and the other composed of healthy adults. They use a virtual classifier algorithm that uses data from GPS sensors to identify walking speed changes, which indicate walking abnormalities. The results show a high level of accuracy in detecting walking anomalies.

Using both CNN and LSTM architectures, ref. [14,15] can predict a person’s behavior while representing actions with neural embedding.

Finally, ref. [16] proposed an algorithm to detect medication adherence in older adults using data from a smartwatch. The algorithm uses data from the smartwatch’s accelerometer and gyroscope sensors to detect hand-to-mouth movements that indicate taking medication. The algorithm was evaluated using data collected from a group of older adults, and the results showed a high level of accuracy in detecting medication adherence.

In conclusion, detecting anomalies in behavior is an essential aspect of monitoring the ADLs of older adults. It involves using sensors and data analysis techniques to identify unusual patterns in behavior that could indicate a potential problem. Several studies have focused on developing algorithms and methods to detect anomalies in behavior using various sensors and data analysis techniques, with promising results. These studies can potentially improve the quality of life of older adults and provide valuable insights into their health and well-being. Table 1 compares the approach taken in this work and those from the literature.

Table 1. Comparison of the approach presented in this work with the techniques from the referenced papers, highlighting the key differences and unique contributions.

Aspect	This Work	Benhammou et al. [10]	Fraga-Lamas et al. [11]	Tahir et al. [12]	Sprint et al. [13]	Bilbao-Jayo et al. [14]	Almeida and Azkune [15]	Amiribesheli and Bouchachia [16]
Scalability	Designed for diverse settings and user groups.	Limited focus on specific anomaly detection in elderly daily activities.	Broad IoT review; scalability not addressed.	Tailored for fall detection; not scalable to other use cases.	Limited to detecting group behavior in smart homes.	Focused on indoor systems, scalability not emphasized.	Scalability not a focus; behavior prediction centered on known patterns.	Designed for dementia monitoring, primarily for home settings.
User and Caregiver Needs	Explicitly addresses both older adults’ and caregivers’ requirements.	Focused solely on elderly activity anomalies.	High-level IoT applications; does not address user-specific needs.	Caregiver needs not addressed; focused on technical fall detection.	User needs inferred but not explicitly prioritized.	Focuses on technical behavior modeling, without caregiver considerations.	Caregiver needs not addressed; focuses on predictive behavior.	Includes caregivers’ concerns but lacks dual focus on specific user requirements.
Anomaly Detection	Efficient pattern recognition, minimizing false positives and negatives.	Anomaly detection integrated but less emphasis on reducing false alerts.	Anomaly detection not a primary focus.	Anomaly detection specific to falls; prone to misclassification in complex scenarios.	Behavioral transitions analyzed, but anomaly detection not a priority.	Minimal focus on anomaly detection beyond location-based patterns.	Does not explicitly prioritize anomaly detection in behavior.	Includes anomaly detection but with limited exploration of error reduction.
Focus Area	Comprehensive ADL monitoring system for older adults.	Anomaly detection in ADLs for elderly individuals.	IoT applications for defense and public safety.	Real-time fall detection using BAN systems.	Smart home data for behavior analysis.	Beacon-based indoor location for behavior modeling.	Predicting human behavior with neural networks.	Smart home IoT platform for monitoring dementia patients.
Data Preprocessing	Robust, including grouping, timestamp decomposition, and redundancy measures.	Relies on statistical and machine learning techniques for preprocessing.	High-level aggregation, less focus on specific preprocessing steps.	Sensor fusion and real-time filtering for falls.	Preprocessing involves temporal segmentation for transitions.	Timestamp grouping and RSSI values for localization.	Preprocessed sequential data for neural network inputs.	Timestamp processing and sensor integration emphasized.
Adaptability	Applicable to independent living, care homes, and mobility-impaired users.	Focused on general elderly care, less adaptable to specific settings.	General applications, lacks specificity for healthcare.	Limited adaptability to broader healthcare applications.	Specific to smart home data; less adaptable to outdoor or other settings.	Primarily indoor-focused; limited generalizability.	Restricted to predictive models, less adaptable across contexts.	Adaptable for dementia care but narrowly scoped to specific conditions.

The contribution of the present work lies in its ability to provide a scalable solution for monitoring the ADLs of older adults. While there have been several works in the past that have attempted to address the same issue, what sets our solution apart is its ability to cater to diverse users. One of the critical challenges previous works have faced is the need for more scalability of their solutions. This has made it challenging to apply their solutions in

different settings, such as in rural areas or for individuals with special needs. Our method is scalable due to its modular construction and the ability to process large datasets iteratively. The modular manageable steps are retrieving beacon information, assigning compartments, and calculating metrics. This allows for independent scaling of each part. The inherent parallelism in the loops over beacons, compartments, and data records enables efficient execution on multi-core processors, GPUs, or distributed systems (not implemented in this paper but referred to in the future work section). Redirecting and consolidating data minimize redundancy and memory usage, ensuring it can handle increasing input data. Another significant aspect of our approach is its emphasis on satisfying the needs of older adults and their caregivers. While previous works have focused on monitoring the ADLs of older adults, they have often overlooked the needs of their caregivers.

Efficient pattern recognition and behavior anomaly detection methods are also critical features of our solution. Previous works have often struggled with detecting anomalous behavior, which has resulted in a high number of false positives or negatives. This paper implements effectiveness using logical validation, metric calculations, and data handling. Comparison of strings with a threshold to group-related data minimizes noise and improves accuracy (which reduces the occurrence of false positives and false negatives). Its use of multiple statistical metrics—such as probabilities, averages, standard deviations, and percentiles—provides a holistic view of patterns, enhancing the reliability of its outputs. Early data validation prevents unnecessary computations, which ensures the results are meaningful and based on sufficient data. Focusing on specific compartment–time combinations allows for pattern recognition. This ensures that caregivers can act promptly in case of abnormal behavior, which can significantly improve the well-being of an older adult.

In summary, the present work contributes to monitoring the ADLs of older adults. Its scalability, ability to satisfy users' and caregivers' needs, and efficient pattern recognition and behavior anomaly detection methods make it a reliable and effective system.

3. Materials and Methods

3.1. System Architecture

The primary aim of this study is to develop a highly autonomous and scalable system designed to identify the daily life patterns of older adults. Additionally, the system is intended to detect deviations in their behavior throughout the day, thereby enabling the transmission of alerts to their family members or caregivers. The construction of a system capable of fulfilling these objectives is essential. The architectural framework of the project is depicted in Figure 1 and is divided into two stages.

The first stage corresponds to the configuration of devices to gather data on older adults' daily activities. This involves beacons emitting periodic Bluetooth Low-Energy (BLE) signals, which are then assigned to several rooms within the household. Concurrently, two applications for data collection and configuration are developed. Firstly, an Android mobile application facilitates the setup of the physical system and maps beacon representations within the application interface. Subsequently, user association with the beacons is achieved. A smartwatch widget application compatible with Garmin OS is developed to enable this. Both systems, the mobile application and the smartwatch widget, encompass distinct functionalities that allow the integration of components for data storage. These two developed applications are out of this paper's scope. The data collection process encompasses acquiring beacon locations through the mobile application and identifying users through the smartwatch widget application, facilitating the capture of vital biometric data to monitor the user's physiological signs. This phase culminates in the generation of data stored within a MongoDB database.

Subsequently, these accumulated data undergo processing to derive insights into usage patterns. The mobile application configures the environment by associating all household beacons with the individual, ensuring scalability by accommodating unlimited environments. Conversely, the smartwatch widget application captures periodic beacon signals, stores them, and records the user's heart rate. The stored data are then transmitted to an

API, bridging the applications with the database. The database assumes responsibility for storing all pertinent data gathered by the smartwatch throughout the user’s daily activities, which subsequently serves as the basis for pattern recognition and outlier detection.

The second stage is the focus of this paper. This phase encompasses data processing, analysis, and decision-making. The project is designed to operate autonomously, devoid of any requirement for human intervention in decision-making processes. So, as more data are obtained, the assembled algorithm becomes more accurate and prepared to make decisions and not create false positives concerning the detection of outliers.

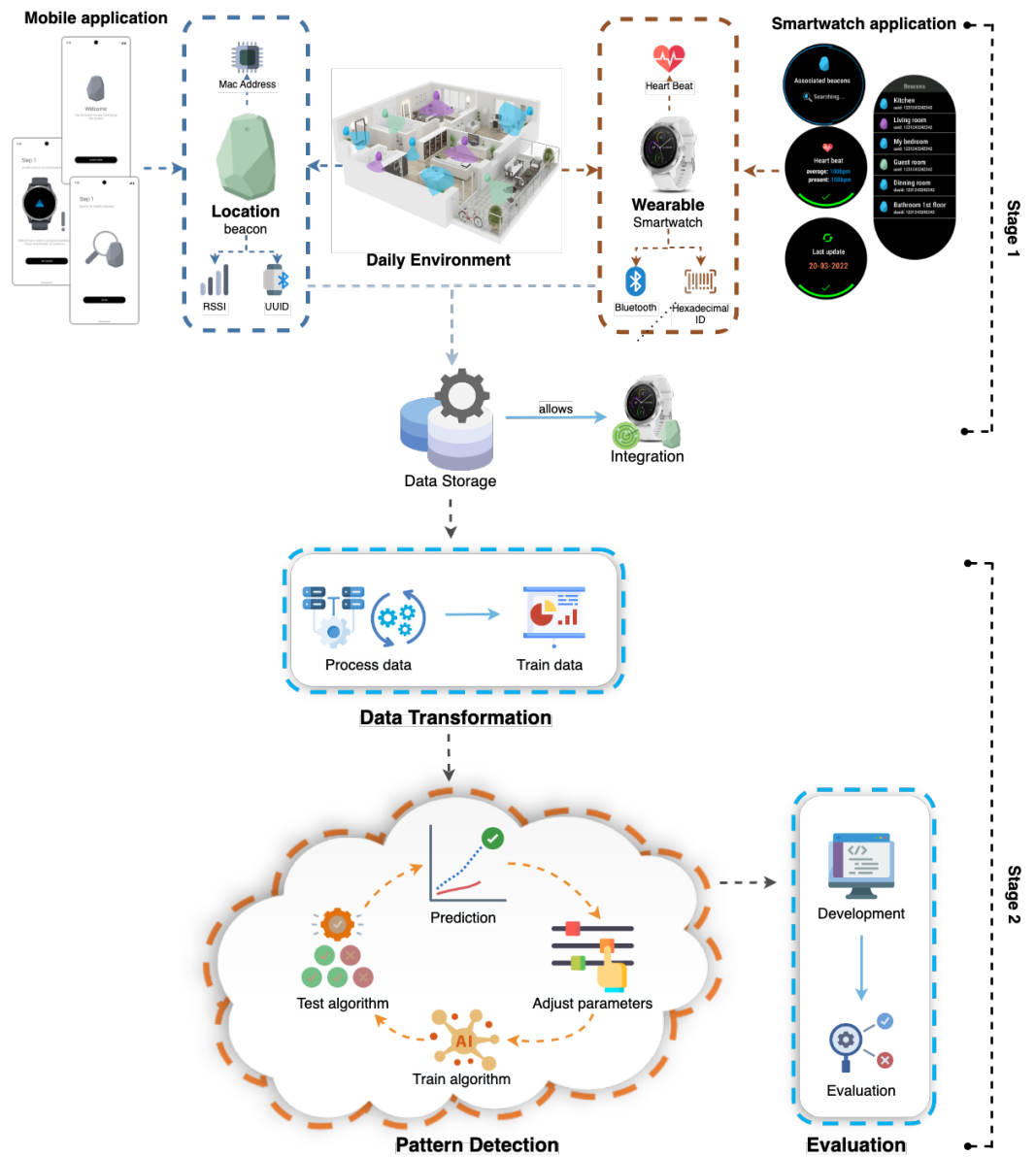


Figure 1. High-Level view of the proposed architecture.

3.2. Observed Context

The observed context focuses on the user (observed subject) and the dwelling. The user in the experimental scenario had a university-level education and was already retired during the analysis period, which took place between the ages of 68 and 70. At the social level, the user lived alone and did not receive regular visits from family members or others. The lifestyle was moderately active, limited to essential activities such as grocery shopping.

The house is in a rural area, surrounded by a green area, with other houses nearby. In this house, the monitoring scenario was configured and equipped with a set of BLE beacons

from Estimote [17]. A smartwatch was used to monitor the presence of the inhabitant in the different compartments. Each room to be monitored was equipped with one or more BLE beacons.

3.3. Data Gathering

This work combines BLE beacons with pattern detection algorithms to analyze individual patterns and detect anomalies in daily activities. Beacons are used as IoT devices, emitting periodic BLE signals at 2.4 GHz, containing Universal Unique Identifiers (UUIDs). Garmin smartwatches are also employed, with functionalities such as external communication, BLE device scanning, and Beats Per Minute (BPM). The smartwatch sends collected data to an API and gathers beacon signals via BLE scanning. BPM data are included as additional information to help identify moments of concern, though it is not used as a standalone indicator. This allows for an individual setup of each user through the smartwatch. To ensure the system is both scalable and user-friendly, it is designed to require minimal human intervention, limited to installing and configuring devices in users' homes. A mobile application facilitates the connection between monitoring daily activities and setting up the devices (beacons and smartwatches).

The solution determines the user's proximity to a beacon using the Received Signal Strength Indicator (RSSI) and transmission power (T_x) values. RSSI measures the signal strength received by the smartwatch, ranging from -50 (strong) to -100 (weak), while T_x represents the beacon's transmission power, typically between -40 dBm and $+4$ dBm. Higher T_x values increase signal range but consume more battery power, reducing durability. These factors define beacon placement to ensure reliable and efficient operation.

To minimize interference, beacons are installed in the center of ceilings, avoiding corners where signal strength might weaken due to walls or obstacles. They are spaced approximately every 5 m, providing reliable coverage for about 25 m^2 areas each, as presented in Figure 2. Although beacons can transmit signals over longer distances, this configuration prioritized accuracy and consistency.

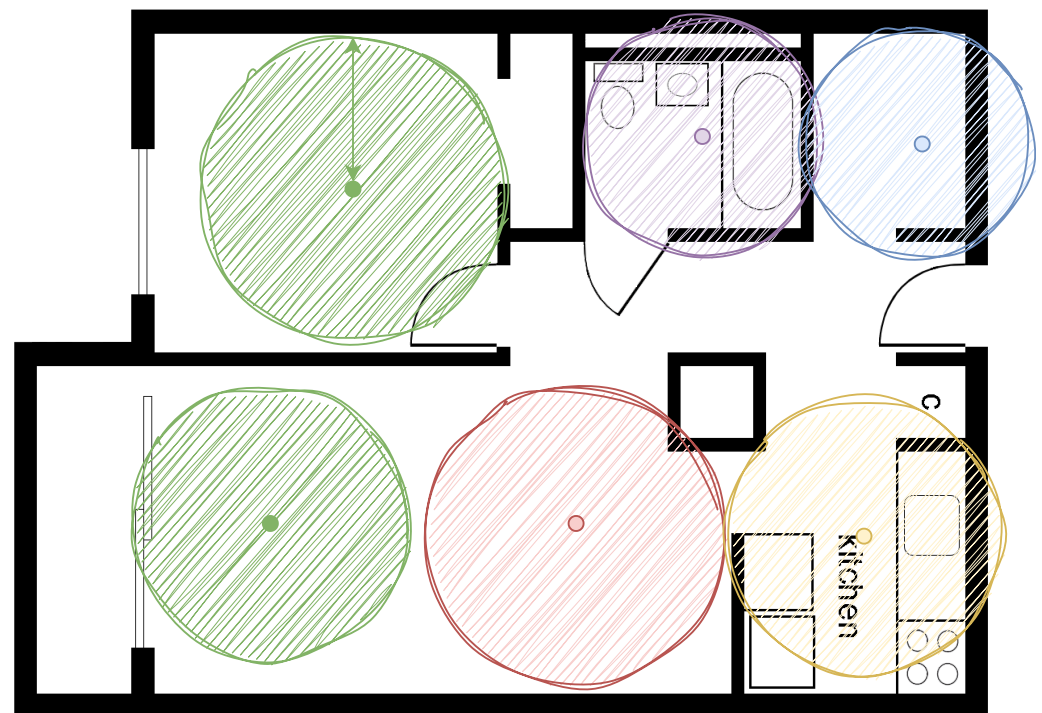


Figure 2. Configuration and placement of BLE beacons in each room of a house. Six beacons were used to setup the system in a house of five compartments.

In this process, BLE beacons function as devices that periodically transmit data. The smartwatch application operates in the background, capturing and storing the received data in the format depicted in Figure 3. Additionally, the application records the BPM obtained from the smartwatch, as shown in Figure 4. Subsequently, the data are temporarily stored within the smartwatch's storage until it establishes a Bluetooth connection with a mobile phone. Once connected, the data are transmitted via Wi-Fi to communicate with an Application Programming Interface (API). The application manages these data, removing them upon successful communication with the API.

Figure 3 shows that data are recorded every five minutes, including the date of the reading, the RSSI value, and the UUID of the associated beacon. This information is organized into a list and prepared for transmission to the API. Figure 4 shows how the reading of BPM data is stored locally, where the date and heart rate values are stored in each data point.

```
[{uuid=>2ba202aa6c226b2efcdff4043a88f420, date=>2023-04-26 00:00:49, rssi=>-65},
{uuid=>1a3ac94b8282494d94f39b65a13cb5a, date=>2023-04-26 00:00:49, rssi=>-93},
{uuid=>2ba202aa6c226b2efcdff4043a88f420, date=>2023-04-26 00:05:49, rssi=>-66}]
```

Figure 3. Format of storing temporary data.

```
[{date=>2023-04-26 00:00:49, heart_rate=>67},
{date=>2023-04-26 00:05:49, heart_rate=>72}]
```

Figure 4. Format of storing temporary BPM.

In conclusion, this comprehensive process of collecting, storing, and transmitting data allows for the acquisition of valuable insights. By leveraging the resources of BLE beacons and a smartwatch, this solution paves the way for future analyses related to collected data regarding monitoring activities of daily living in older adults.

3.4. Data Integration

A seamless data flow occurs when the smartwatch establishes a Bluetooth connection with its associated mobile phone to perform updates. All the newly collected data from the smartwatch are transmitted to the designated API and are stored in a structured manner within a database (MongoDB). The format in which these data are stored aligns with the representations shown in Figures 5 and 6.

```

v heart_rate: Array
  v 0: Object
    date: "2023-05-02 00:00:49"
    heart_rate: 69
  v 1: Object
    date: "2023-05-02 00:05:49"
    heart_rate: 66

```

Figure 5. Format of storing the BPM in daily living.

Figure 6 visually represents the specific structure employed to store the data in the database about nearby beacon information during the day. It highlights the arrangement of different fields, such as date, time, and RSSI, and these metadata enrich the understanding of the collected information. Bearing in mind that the beacon data records always have a date and time and a UUID, this guarantees the uniqueness of the records, mainly since the time value contains the hours, minutes, and seconds, and the UUID is a single value.

```

  ✓ near_beacons: Array
    ✓ 0: Object
      date: "2023-05-02 00:00:49"
      uuid: "2ba202aa6c226b2efcdff4043a88f420"
      rssi: -65
    ✓ 1: Object
      date: "2023-05-02 00:00:49"
      uuid: "1a3ac94b8282494d94f39b65a13cb5a"
      rssi: -93
    ✓ 2: Object
      date: "2023-05-02 00:05:49"
      uuid: "1a3ac94b8282494d94f39b65a13cb5a"
      rssi: -89
    ✓ 3: Object
      date: "2023-05-02 00:05:49"
      uuid: "2ba202aa6c226b2efcdff4043a88f420"
      rssi: -69

```

Figure 6. Format of storing the data about near beacons in daily living.

3.5. Data Pre-Processing

It was necessary to organize the collected data to identify patterns in the collected data. Data pre-processing consisted of segmenting the dataset to group the data by compartment, removing days with missing records, and temporarily grouping the records.

The collected data are ordered chronologically. To carry out the individual analysis of the movement patterns in each compartment, it was necessary to organize the data, separating the records referring to each data point. The division of data allowed the study of compartment-by-compartment patterns to start without the complexity of the remaining records.

3.6. Incomplete Data Removal

To address incomplete periods, we calculated a minimum threshold of records necessary to represent a full day. By comparing the total daily records—given that data are recorded every 5 min, resulting in up to 288 measurements per day—we established that only days with a total occurrence exceeding half of this value would be considered for further analysis. This criterion guaranteed the availability of numerous recordings to represent an entire day adequately. Consequently, periods with fewer recordings were excluded from the analysis, enhancing accuracy and mitigating the impact of data irregularities.

3.7. Grouped Data

Initially, we selected one-hour intervals for data segmentation. The pattern becomes clearer when data are precisely grouped into hourly periods. Although we explored other periods, shorter intervals exhibited significant noise, while longer intervals yielded overly abstract data with limited utility for the intended study. Consequently, we segmented the data by each hour of the day to facilitate pattern assessment within a reasonable time frame. We could observe a complete cycle of participants' daily activities by evaluating data across twenty-four hours.

We adopted a method involving grouping data into equal periods based on the intended configuration to achieve the desired granularity. Initially, we grouped data by hour for each day and then grouped these data by hour across all days. This approach allowed for a comprehensive view of the system, enabling recognition of daily activity patterns. Specifically, at any given moment, the closest beacon device with the highest RSSI value indicates a 100% probability of the individual being in that compartment. Subsequently, the data were grouped to capture the activity pattern for each hour across all days.

3.8. Data Formatting

We established a system where a service worker periodically retrieves new data for each individual. This setup ensures the existence of a local version of the data, which is utilized for pattern recognition, alongside its presence in the database. Consequently, data

security is enhanced as the data are stored in two locations, providing redundancy in case of system failure.

The data collected from BLE beacons are organized as time series data. Each dataset record denotes the time of day the individual spends within a compartment. It is crucial to accurately represent the timestamp feature as a numerical variable to specify the time interval between other records. Therefore, to conduct behavioral analysis based on individuals' positions throughout the day, we decomposed the timestamp into multiple features, including date–time, the UUID of the beacon corresponding to a specific compartment, and RSSI. Each record represents the reading captured at that moment, indicating which beacon the smartwatch and the mobile phone detected every five minutes.

3.9. Pattern Recognition

The identification of patterns was carried out considering the distribution of all recorded data, the result of the calculation of the 25th and 75th percentiles, and the measure of the occupancy probability.

The patterns consist of observing characteristics that, as a general rule, tend to repeat themselves. The data were loaded into a temporal data structure (time series) and subsequently analyzed based on their graphical representation. Two files will be created and used during the pattern recognition process: the train and the pattern files. These files will be essential for the follow-up of the work that will explain further on how they will be used. In this way, the training file will be the file that will have all the data related to the respective person's daily life activity since the first day and has the format that can be seen in Figure 7 (which contains approximately 26,000 records). The pattern file will be the file that will have the data organized by hours and compartments, and the calculation of the occupancy probability, standard deviation, and 25th and 75th percentiles for each are presented in Figure 8 (containing approximately 11,000 records).

```
date, uuid, rssi
2023-04-20 00:00:49,2ba202aa6c226b2efcdff4043a88f420,-65
2023-04-20 00:00:49,1a3ac94b8282494d94f39b65a13cb5a,-93
2023-04-20 00:05:49,2ba202aa6c226b2efcdff4043a88f420,-66
2023-04-20 00:05:49,1a3ac94b8282494d94f39b65a13cb5a,-59
2023-04-20 00:10:49,2ba202aa6c226b2efcdff4043a88f420,-69
2023-04-20 00:10:49,1a3ac94b8282494d94f39b65a13cb5a,-93
2023-04-20 00:15:49,2ba202aa6c226b2efcdff4043a88f420,-65
```

Figure 7. The train file.

```
date, location, probability, std, p25, p75
00:00,2ba202aa6c226b2efcdff4043a88f420,0.8888,0.09621,0.7712,0.9791
00:00,1a3ac94b8282494d94f39b65a13cb5a,0.0625,0.0721,0.0,0.1458
00:00,2b202aa8282494efcdff4004f2acae20,0.04861,0.0429,0.0,0.0833
00:00,3a7020d6dca6683702b2d004f2acae15,0.0,0.0,0.0,0.0
00:00,None | Outside, 0.0,0.0,0.0,0.0
01:00,2ba202aa6c226b2efcdff4043a88f420,1.0,0.0,1.0,1.0
01:00,1a3ac94b8282494d94f39b65a13cb5a,0.0,0.0,0.0,0.0
01:00,2b202aa8282494efcdff4004f2acae20,0.0,0.0,0.0,0.0
01:00,3a7020d6dca6683702b2d004f2acae15,0.0,0.0,0.0,0.0
01:00,None | Outside, 0.0,0.0,0.0,0.0
02:00,2ba202aa6c226b2efcdff4043a88f420,0.9653,0.0429,0.9167,1.0
02:00,1a3ac94b8282494d94f39b65a13cb5a,0.0347,0.0429,0.0,0.0833
02:00,2b202aa8282494efcdff4004f2acae20,0.0,0.0,0.0,0.0
```

Figure 8. The pattern file.

Algorithm 1 orchestrates the calculation and pattern recognition processes and is tasked with conducting the necessary computations to analyze the patterns of each individual. To accomplish this, the data must be retrieved from the train file and stored in a structure (represented by $\mathcal{D} \leftarrow \text{ReadData}(\text{"train.csv"})$). After that, the algorithm accesses the database to look for all the beacons $b_i \in \mathcal{B}$ (six beacons were used, as presented in Figure 2),

that is, all the compartments $c_i \in \mathcal{C}$ (five compartments) associated with the respective person $p_i \in \mathcal{P}$, and stores them in a structure. Next, it is crucial to understand if there is more than one beacon inside the same compartment. If this situation arises, the algorithm combines the data associated with $b_i \in \mathcal{C}$, treating them as unified entities. For example, if the living room is relatively large, approximately 35–40 m², then there are at least two beacons in this room. When configuring these two beacons, the location field will read “living room 1” (b_1) and “living room 2” (b_2). To this end, we have applied the Ratcliff/Obershelp pattern-matching algorithm [18]. It returns the equivalence percentage between two strings ($Eq(s_1, s_2)$, where s_1 and s_2 represent the UUID strings associated with b_1 and b_2 , and Eq represents the value of the equivalence from 0 to 1, where 1 is a complete match for two given strings). In these contexts, the precision of strings is determined by its size. If the equivalence is greater than 85%, there is at least one case where these beacons’ data must be concatenated since they correspond to the same compartment. The minimum threshold for equivalence was set at 85%, determined through comparisons among sets of compartments. It was the most accurate value through various comparisons, as shown in Table 2.

Algorithm 1: Pattern Recognition

```

Data:  $\mathcal{D}$                                  $\triangleleft$  Set of all recorded data  $\mathcal{D} = \{d_i\}, i = 1 \dots 26,000$ 
Data:  $\mathcal{C}$                                  $\triangleleft$  Set of Compartments  $\mathcal{C} = \{c_i\}, i = 1 \dots 5$ 
Data:  $\mathcal{B}$                                  $\triangleleft$  Set of Beacons  $\mathcal{B} = \{b_i\}, i = 1 \dots 6$ 
Result:  $\mathcal{P}$                                 $\triangleleft \mathcal{P} = \{p_i\}, i = 1 \dots 10,800$  (pattern.csv file)

1 begin
2    $\mathcal{D} \leftarrow \text{ReadData}(\text{"train.csv"})$ 
3    $\mathcal{B} \leftarrow \text{RerieveBeaconInfo}()$ 
4   foreach ( $b_i \in \mathcal{B}$ ) do
5      $\text{Comp}(b_i) \leftarrow \text{GetCompartment}(\text{UUID}(b_i))$ 
6      $s_i \leftarrow \text{Comp}(b_i)$                                  $\triangleleft S_i$  string associated to  $b_i$ 
7   end
8   foreach ( $(b_i, b_j) \in \mathcal{B}$ ) do
9     if  $Eq(s_i, s_j) > 0.85$  then
10    |    $\text{Concatenate}(s_i, s_j)$ 
11  end
12  foreach ( $c_i \in \mathcal{C}$ ) do
13    |    $m_b \leftarrow \text{GetSmaller}(\text{UUID}(b_j))$                                  $\triangleleft b_j \in c_i$ 
14    |    $\text{RedirectData}(b_j, m_b)$                                  $\triangleleft$  redirect  $b_j$  to  $m_b$ 
15  end
16  foreach ( $d_i \in \mathcal{D}$ ) do
17    |    $\text{earliestD} \leftarrow \text{Min}(d_i)$ 
18    |    $\text{latestD} \leftarrow \text{Max}(d_i)$ 
19    |   if  $\text{numDays}(\text{earliestD}, \text{latestD}) < 7$  then
20    |   |    $\text{out} \leftarrow \text{"InsuficientData"}$ 
21    |   else
22    |   |    $t_i \leftarrow (\text{earliestD}, \text{latestD})$ 
23    |   |   foreach ( $t_i \ \&\& \ c_j \in \mathcal{C}$ ) do
24    |   |   |    $\text{prb} \leftarrow \text{Prob}(c_j, t_i)$ 
25    |   |   |    $\text{avg} \leftarrow \text{Avg}(\text{RSSI}_{b_i}, c_j, t_i)$ 
26    |   |   |    $\text{std} \leftarrow \text{StdDev}(\text{RSSI}_{b_i}, c_j, t_i)$ 
27    |   |   |    $\text{perc25} \leftarrow \text{Perc}(\text{RSSI}_{b_i}, c_j, t_i, 25)$ 
28    |   |   |    $\text{perc75} \leftarrow \text{Perc}(\text{RSSI}_{b_i}, c_j, t_i, 75)$ 
29    |   |   |    $\mathcal{P} \leftarrow \text{Store}(\text{prb}, \text{std}, \text{perc25}, \text{perc75})$ 
30    |   |   end
31  end
32 end

```

The algorithm stores the correspondence of beacons in a separate structure upon recognizing that they are located in the same compartment. It then selects the “master” beacon, m_b , for that compartment based on the alphabetical order of each beacon’s UUID. The beacon with the smaller UUID is designated as the master beacon. Consequently, all other beacons within the same compartment are configured to point to the “master” beacon. For instance, if the living room has three associated beacons (b_1 , b_2 , and b_3), b_1 is chosen as the master beacon. This means that b_1 points to itself, while b_2 and b_3 point to b_1 . Thus, when data from b_2 or b_3 are detected, the algorithm interprets them as originating from b_1 , enabling a more accurate calculation of the user’s location.

Table 2. Equivalence percentage between two strings.

Strings Equivalence		
String 1	String 2	Result (%)
Toilet 1	Toilet 2	87.5
Living room 1	Living room 2	92.3
Bedroom 1	Bedroom 2	88.8
Kitchen 1	Kitchen 2	88.9
Bathroom 1	Bathroom 2	90.0

The system then examines the structure containing chronologically ordered data corresponding to the older person’s daily activities. It searches for the earliest and latest dates within this structure. Subsequently, it verifies if the difference in days between the latest and earliest dates exceeds seven days. If not, the algorithm cannot recognize a pattern, as the minimum required duration for pattern recognition is seven days. In such cases, the algorithm initiates a reorganization of all data. Consequently, the structure is reconstructed, where each date (including date and hour) encompasses all information related to BLE beacons and their RSSI values recorded by the smartwatch at a specific time. This approach facilitates the assessment of the volume of data obtained daily, allowing for improved organization based on date discrepancies. Furthermore, all available beacon data are included for each date to ensure comprehensive data organization.

After organizing the data, it is necessary to verify which days do not meet the minimum criteria set by the algorithm, as detailed in Section 3.6. Then, the system initiates a new re-verification process if at least seven consecutive days of relatively stable data are available for pattern recognition. This precaution is essential because the elimination of certain days may lead to situations where there are insufficient days to meet this requirement.

Therefore, the algorithm ranges from the first date in the structure to the last, and every five minutes of every day, it seeks to see if there is information from beacons that can be read. If so, it increases the beacon’s probability field where the RSSI value is higher; i.e., the algorithm will look for all the beacon information that the smartwatch reads and acquire the information in which the RSSI value is higher. This means that the respective person was closer to that beacon than others at that moment, increasing the probability value of that moment (which corresponds to the hour of minutes). If no data are received from any beacon at a given moment, the smartwatch cannot detect any associated BLE beacons. Possible reasons include sensor disconnection or the person being outside the home. To address this, a new location named “None | Outside” is introduced to account for these occurrences and distinguish them from other events.

Therefore, these data must be divided by the days the pattern is studied. This is because, for example, if we have twelve days of data, and at 1 am we have the average room value of 11.4, this indicates that each average day compared to twelve days is very close to 100%. With this, the algorithm has to divide this value by the total number of days so that it has the probability of 0 to 1 that the older adult is inside a room at 1 am. This division is made for all data so that it has consistency in the data and can be worked on. Thus, everything is prepared to determine the probability of a person being in a

compartment (c_i) at a time (t_j) (given by $Prob(c_i, t_j)$). Briefly, the algorithm will travel each hour and every five minutes and calculates the probability through the average and the number of days, the standard deviation, and the 25th and 75th percentiles to have data to analyze the data.

Finally, the algorithm writes what is in the structure for the pattern file, organized in ascending hours, and the probability of each compartment in descending order, visible in Figure 8. A complete list of the functions defined in Algorithm 1 is presented in Appendix A.

3.10. Outlier Detection

The Interquartile Range (IQR) rule [19] and the Z-score method [20] emerged as the most suitable techniques for this study. These methods offer robust statistical insights into data deviations from the norm. It is important to note that each technique may require specific adaptations depending on the context in which it is applied. Furthermore, it is worth mentioning that both methods are recommended for data conforming to a normal distribution.

For the outlier detection process, several works have been carried out [21–23]. A formula incorporating the Interquartile Range (IQR) with a constant value of 2.2 is suggested to be used as the optimal value. Consequently, the upper and lower limits of the IQR are given by Equations (1) and (2)

$$Upper = Q3 + (2.2 * (Q3 - Q1)) \quad (1)$$

$$Lower = Q1 - (2.2 * (Q3 - Q1)) \quad (2)$$

where *Upper* and *Lower* represent the upper and lower limits of the IQR, and $Q1$, $Q2$, and $Q3$ represent the first, second, and third quartiles, respectively.

Based on these two formulas, if a specific value is lower than the result of Equation (2) or if it is higher than the result of Equation (1), then it means that this value is an outlier.

The Z-score method is widely used in many other statistics-related operations [24,25]. By default, any Z-score of 3 or more is considered an outlier since it falls outside the range of approximately 99.7% of the data. The Z-score is given by Equation (3).

$$z = \frac{x - \mu}{\sigma} \quad (3)$$

The algorithm accesses the database to look for all the beacons—that is, all the compartments associated with the respective person—and stores them in a structure. This way, it does precisely the same step explained in Section 3.9, associating all the beacons to just one in the same compartment.

If new data exist, the algorithm proceeds to another verification phase: determining whether the difference between the last date of these new data and the last date in the file name are more significant than one hour. The algorithm can only detect outliers in the behavior if there is at least one hour since the data are organized by hours.

The algorithm processes the new day's data by organizing them based on the hours and minutes associated with each compartment. For each time interval (every hour and every five minutes), it computes the probability using the mean and the total number of days as described. Additionally, it calculates the standard deviation and the 25th and 75th percentiles, ensuring the availability of detailed statistics for further data analysis. This process mirrors the approach explained in Section 3.9, providing a consistent framework for analyzing patterns and distributions across time and compartments. After this organization, a base to detect possible outliers is created. For that, the algorithm will look for all the data saved in the pattern file to compare the data of each hour with the new data, keeping them in a new structure.

To detect possible outliers in the new data, a 95% confidence level is considered. Thus, with the number of days of the pattern, the number of degrees of freedom that will be used to calculate the t-score is calculated. So, in the structure that contains the indicator data related to pattern recognition, they will be necessary at the beginning of each verification, as they will be traversed hour by hour in chronological order. Therefore, the algorithm will fetch all the information related to each compartment, namely the probability of being in the respective compartment at a specific moment and the 25th and 75th percentiles, and depending on the time of day, it will fetch the data calculated with the received data in this process. Each compartment’s probability of obtaining the new data will compare whether this value is below Equation (2) or higher than Equation (1), taking the default values. This is the first step to take during verification.

The algorithm realizes that with the set of new data, it indicates possible outliers at a specific time. Still, it could be a false positive, and that is why the detection of outliers is divided into three levels: “low”, “medium”, and “high”. The detected outliers will be stored later in a new structure, divided into the levels mentioned above. Therefore, depending on the resulting value, using the Z-score method with Equation (3) will characterize the level. As mentioned in the previous section, any value calculated above three utilizing this method is considered an outlier. When the algorithm was still in the testing phase, it was found that this value was unsuitable for the problem in question. So, what was thought was that, in addition to this calculated result, there was an associated constant relative to the level of confidence that we wanted to impose on this detection. The result generated by calculating this method would be multiplied by a constant, where this constant is the previously calculated t-score. Adding this constant to the result increased the accuracy of the calculation by about 65%, making this approach much better than setting a static value. Examples of this are presented in Table 3.

Table 3. The comparison between Equation (3) includes the constant, and without it, elucidates the impact of the constant on the results.

Without Constant	With Constant	Real Level
High	Low	Low
Medium	Low	Low
High	Medium	Medium
None	Low	Low

Furthermore, this solution involves how many days there are for pattern recognition since the t-score depends on the number of degrees of freedom, thus implying dependence on the number of days of the pattern. In this way, determining which level is being treated will depend on the result generated by Equation (3) multiplied by the constant, and this result will determine the level.

Still, in detecting the outlier level, there is an auxiliary variable: the person’s BPM over a given hour. Garmin does not provide specific accuracy specifications in measuring heart rate. However, it was reported that the accuracy of each watch would depend on multiple factors, but more specifically on the smartwatch used for testing. The typical error is ± 7 BPM [26]. Therefore, according to [27], the normal BPM range in older people is between 60 and 100 BPM. This interval will be used so that if the average of BPM values within the respective hour is outside this interval (taking into account the error associated with the measurement), an outlier level will be added to the previously detected one. With this, the algorithm organizes the outliers so that it is possible to understand when the outlier appeared at their respective time and which contributed to and harmed these new data. That is when a pattern is recognized in this study; as a rule, a compartment always stands out at each hour. This means that at the respective time, the probability of the person being in that compartment is greater than the others. So, if, for example, there is a time of day when the person is not in that compartment, the algorithm will identify that

the compartment affected in calculating this outlier is the division that is usually ahead of the rest. The beneficiary is the one who, at that time, has more data than everyone else. Furthermore, there may be the possibility that at a particular time, there is not a well-defined pattern, which means that if any compartment is one of the highest among the others and that at a certain time of a specific day, the probability of it having been at the same time but on a different day is much higher, and the algorithm detects this and identifies that there was a compartment in which its probability was increased above the standard. Still, it is one of those with the highest chance.

This detection is only possible when the data are not continuous when detecting the pattern, that is, when there is a standard deviation during the calculation of the probability, because if the standard deviation of a probability at a given time is zero, then Equation (3) cannot be used as division by zero. The algorithm receives a probability value, converts it into the corresponding five-minute intervals within each hour, and assesses whether the deviation between them exceeds specific threshold values. This determination enables the algorithm to discern the outlier level of the new data. The probability of being in each division at a particular time is calculated, yielding a value ranging from 0 to 1. This value is then translated into the corresponding number of five-minute intervals within an hour, allowing for the establishment of threshold values for each calculated probability. For instance, if A is assigned a probability of 0.6 in a given hour within a compartment, this value translates to approximately 35–40 min of presence within that hour. Suppose a maximum limit of 15 min is set for detecting outliers in new data. In that case, any subsequent data indicating a presence of 28 min or less within an hour suggest behavioral changes. Depending on the interval, the algorithm classifies the levels of outliers being detected. Furthermore, as described earlier, this process utilizes an auxiliary system that averages the BPM readings for each hour. This information is leveraged to identify individuals who may benefit or be adversely affected by the observed behavioral changes.

After this detection and characterization of each outlier, the algorithm will check, within the structure that stores the levels of outliers and their respective information and whether an outlier detected at a given time had already been seen in the past hour. So, for the algorithm to carry out this analysis, it is necessary to analyze the information on who benefited and who was harmed in each detected outlier and, in turn, increase the level if so justified. With this, the algorithm can have a temporal precession of each detected outlier. After calculating and analyzing these data, if outliers are detected, the family member or caregiver with whom the patient has associated the detailed information on how each outlier was seen is emailed.

4. Results

The research objectives described in this paper depend on the behavioral normality of the person. We analyze the lower and higher limits, exposed by the 25th and 75th percentiles. Finally, we evaluate the occupation probability of each compartment, indicating the likelihood of the person being present at a specific time.

4.1. Accumulated Data

Observing the accumulated data makes it feasible to visualize the distribution without excluding any records. These records are grouped solely by sixty-minute periods. All recorded data are depicted in graphs corresponding to the respective compartments. The charts in the following figures will display recorded values ranging from 0 to 12. Here, 12 corresponds to the number of five-minute intervals within one hour. Consequently, every hour is divided into 12 intervals of five minutes each.

In the kitchen (Figure 9), periods of heightened presence, denoted by a notably increased frequency of data occurrences, occur between 10 am and 1 pm, as well as between 7 pm and 9 pm. This pattern suggests a pronounced presence in the kitchen during meal times. However, between 2 pm and 4 pm, a reduction in presence is observed, attributed to

the recognized tendency of individuals to relocate to the living room or embark on outdoor activities such as walking.

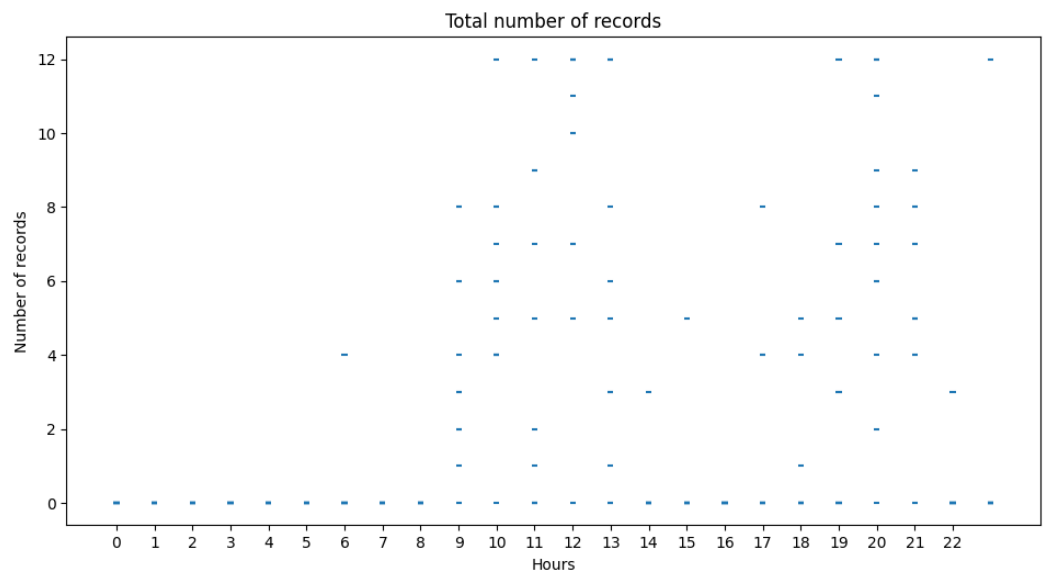


Figure 9. Total occurrences per hour in the kitchen compartment.

The bathroom is a traditional short attendance compartment, as shown in Figure 10. Nevertheless, it is possible to identify some recurring periods of use, namely the early morning (probably related to morning hygiene) between 8 am and 9 am, and 10 pm and 11 pm before moving to the bedroom.

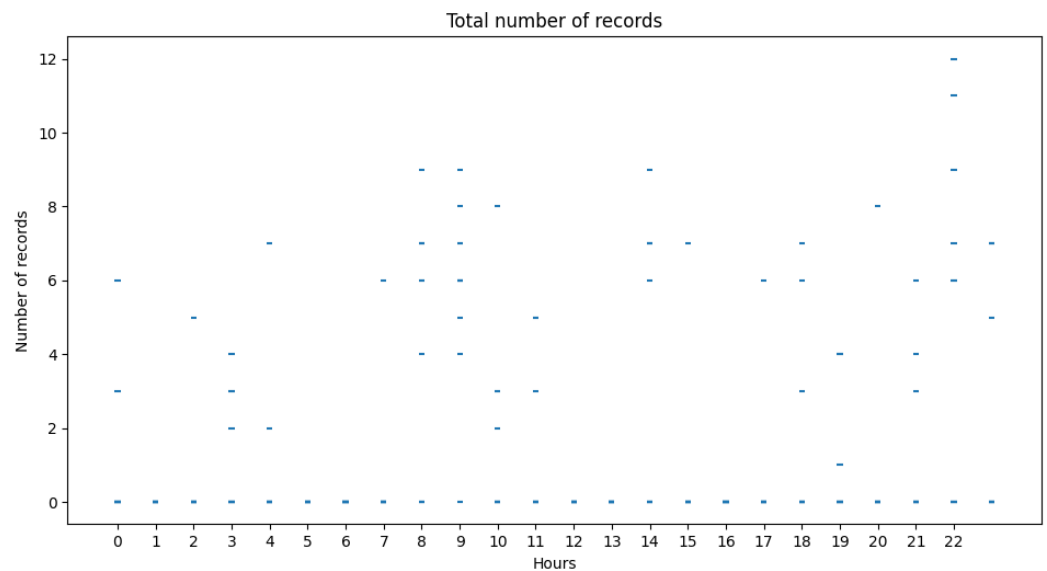


Figure 10. Total occurrences per hour in the bathroom compartment.

The third compartment analyzed was the bedroom. This space is usually used to rest and sleep. From Figure 11, it is possible to identify the occupation in the evening periods. Between 10 pm and 9 am, there is a strong movement, as well as at 7 pm.

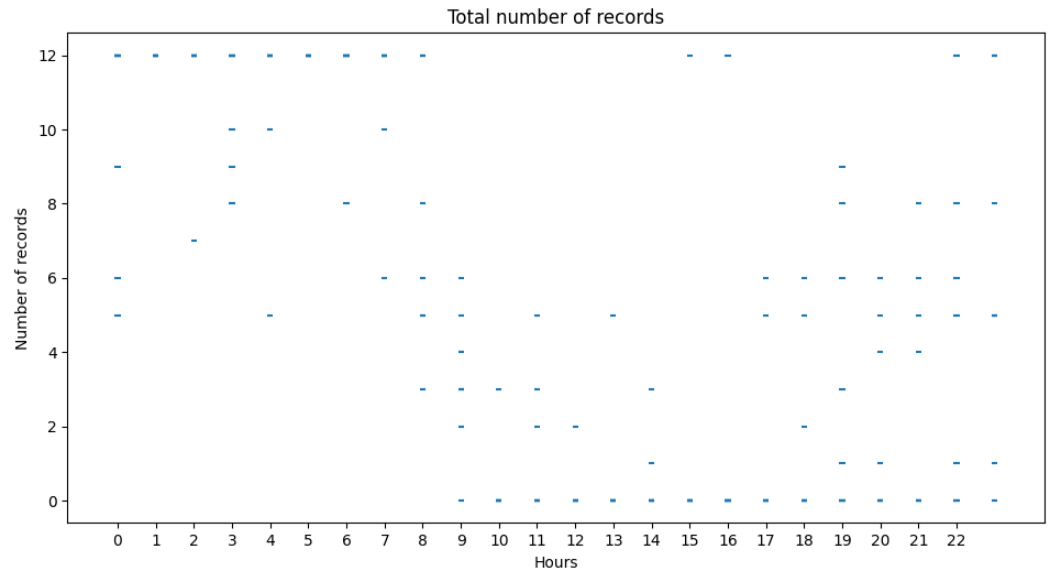


Figure 11. Total occurrences per hour in the bedroom compartment.

The living room compartment is visible in Figure 12, where the presence is detected in the periods between 2 pm to 4 pm and 9 pm to 11 pm. The inhabitants confirmed the routine of moving from the kitchen to the living room after dinner and staying there until they moved to the bedroom to sleep. The room is an open space that is midway to each compartment, which justifies the existence of some movements during the day.

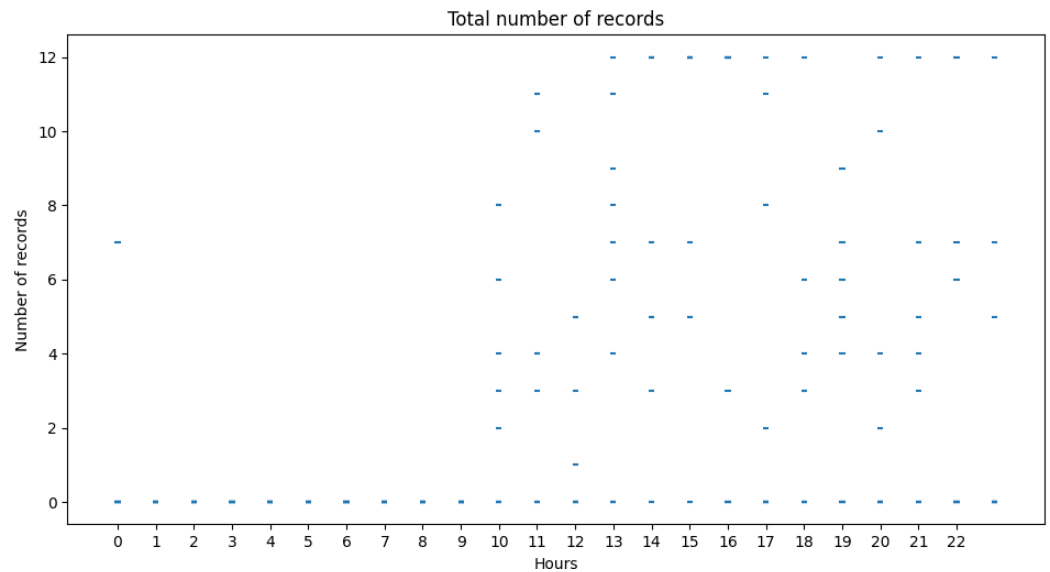


Figure 12. Total occurrences per hour in the living room compartment.

The last area analyzed is related to situations where no data are available or the person was away from home. From Figure 13, it can be seen that between 3 pm and 6 pm, it is frequent for the person to leave the house. The person himself confirmed this information.

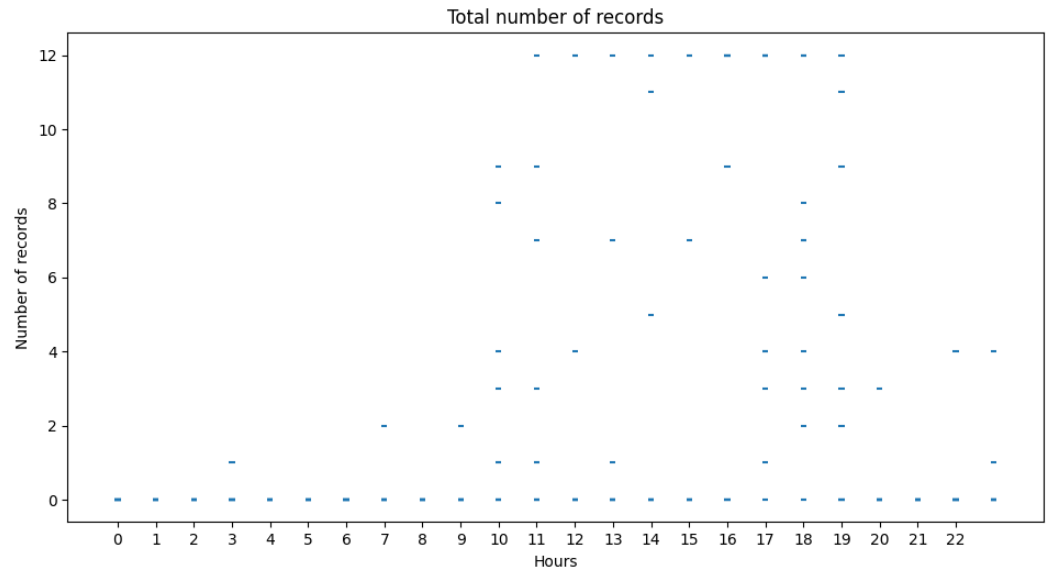


Figure 13. Total occurrences per hour in the none or outside scenarios.

4.2. Time Distribution of Attendances

The analysis of attendance totals does not allow for assessing their distribution (by hour) during the period. In turn, outliers in the training dataset can compromise the identification of patterns by analyzing the maximum and minimum number of presences. A percentile analysis can eliminate noise in the data caused by outliers generated by non-routine events.

After assessing the data distribution by viewing all the daily records, the 25th and 75th percentiles were analyzed. In the kitchen compartment (Figure 14), we can identify periods of movement, namely between 9 am and 10 am, 12 pm and 1 pm, and 8 pm and 9 pm. As expected in the bathroom, no movements are identified during most of the day, as shown in Figure 15. The 75th percentile reveals a more significant presence in the early morning between 8 am and 9 am, in the early afternoon after lunch around 2 pm, and at the end of the day between 10 pm and 11 pm.

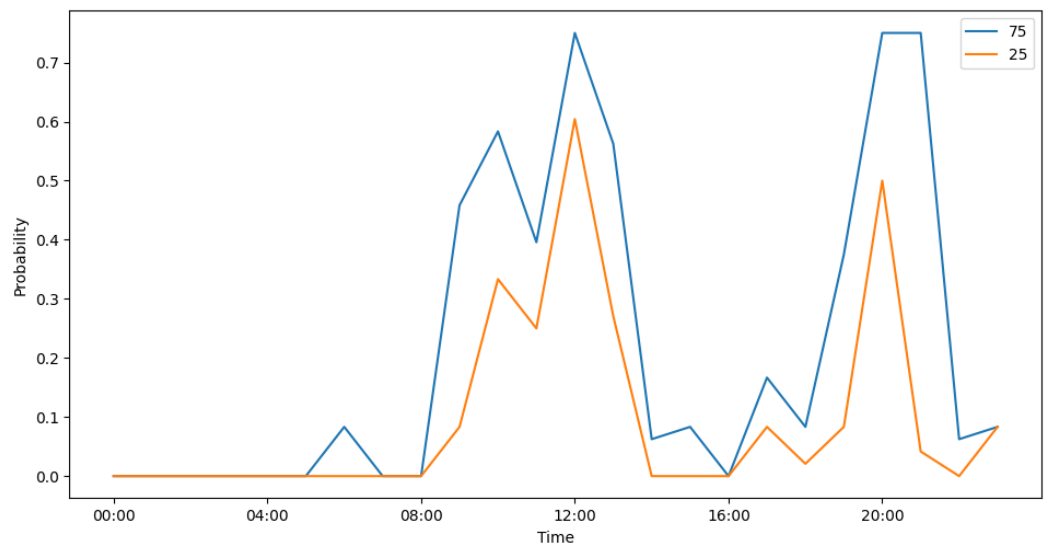


Figure 14. The 25th and 75th percentile values per hour in the kitchen compartment.

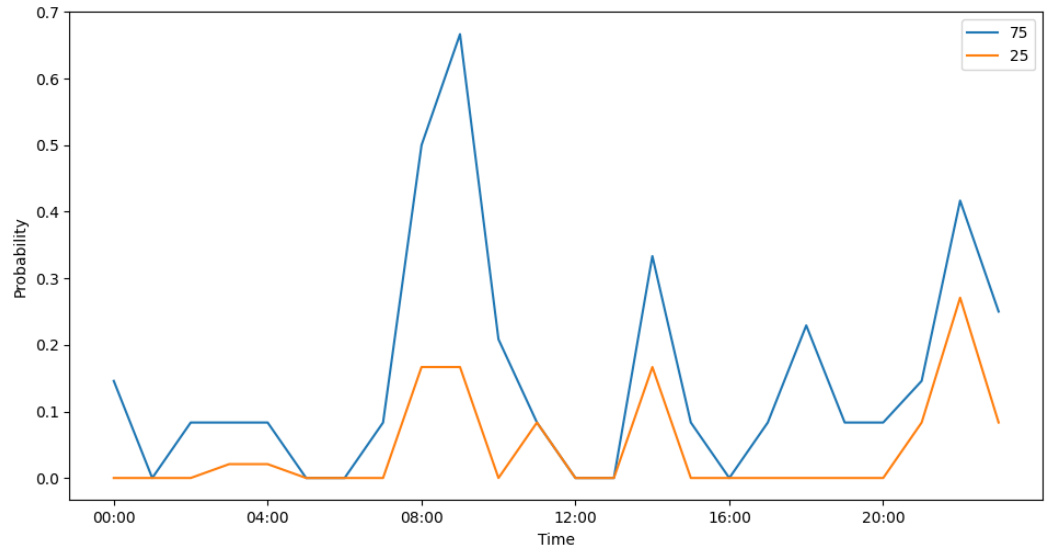


Figure 15. The 25th and 75th percentile values per hour in the bathroom compartment.

As for the bedroom compartment, Figure 16 shows that the 25th and 75th percentiles in certain parts of the day are very constant and almost always equivalent throughout the day. Thus, analyzing the percentiles, it can be concluded that they reveal a more significant presence between 12 am and 9 am, and also, from 11 pm onward, this presence begins to increase. This is because the person usually sleeps during this period.

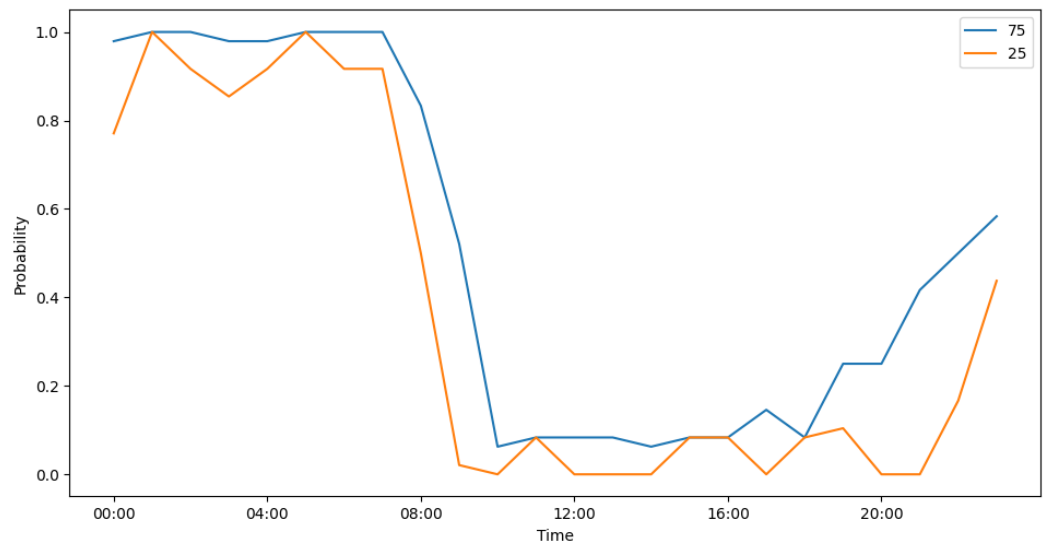


Figure 16. The 25th and 75th percentile values per hour in the bedroom compartment.

In the living room compartment, the presence of users between 2 pm and 4 pm is once again validated through Figure 17. During this period, the number of records is constant. Furthermore, between 10 am and 11 am, and 10 pm and 11 pm, there is also a significant presence in this compartment.

In the none or outside area, by viewing the graph in Figure 18, it can be seen that between 4 pm and 6 pm, there is a greater possibility of the person being away from home. In addition, other time intervals may have more significant potential, but this may have to do with sensor failures when collecting the data sent by the beacons in each compartment or the person being out of the house.

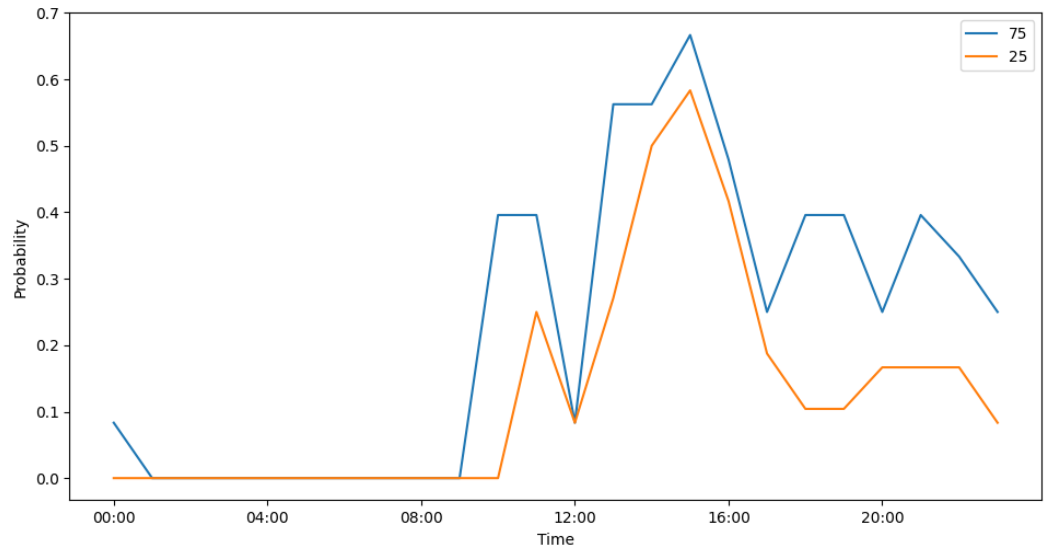


Figure 17. The 25th and 75th percentile values per hour in the living room compartment.

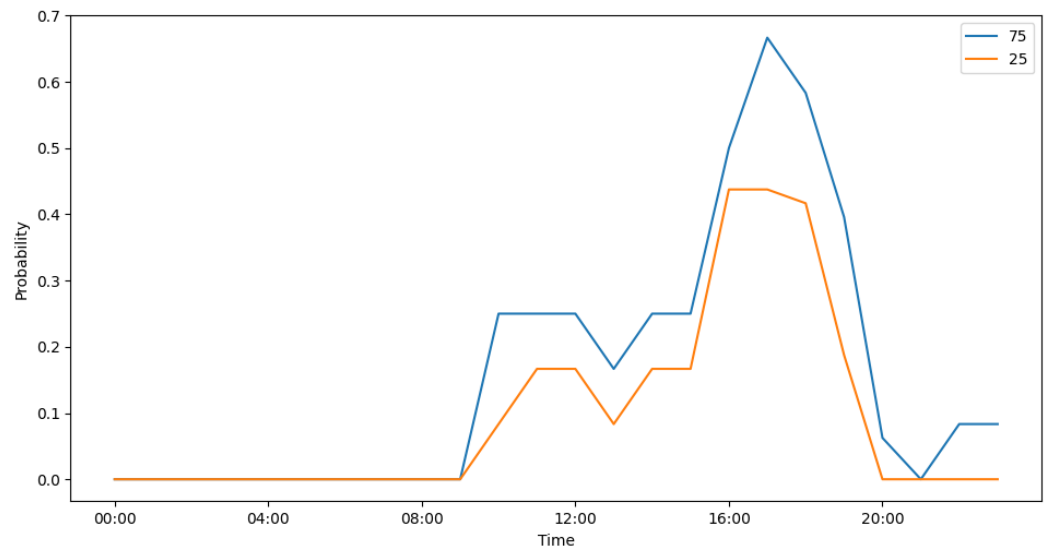


Figure 18. The 25th and 75th percentile values per hour in none or outside.

4.3. Occupation Probability

Calculating the probability of an inhabitant’s presence in a specific compartment was an additional mechanism to study movement patterns and occupancy within the living environment. This approach provided valuable insights into the likelihood of finding an individual in a particular compartment during a given hour. To assess the compartment’s occupancy probability hour by hour, the average for each compartment is calculated based on the occurrences, and the probability is calculated by dividing the average number of occurrences for each compartment by the number of days, hour by hour. The calculation of compartment occupancy probabilities, as outlined in Section 3.9, contributed to the broader analysis of daily living patterns and behavior. It complemented detecting outliers and patterns identified through other mechanisms, providing a comprehensive view of the inhabitants’ activities and movements within the live environment.

After calculating the probability, hour by hour, the similarity between the occurrence chart (Figure 9) and the probability chart (Figure 19) was again verified. According to the probability graph in the kitchen compartment, it is more likely that the person is in this compartment due to breakfast, lunch, and dinner. It can be seen that the intervals are between 10 am and 11 am, 12 pm and 1 pm, and 8 pm and 9 pm, respectively.

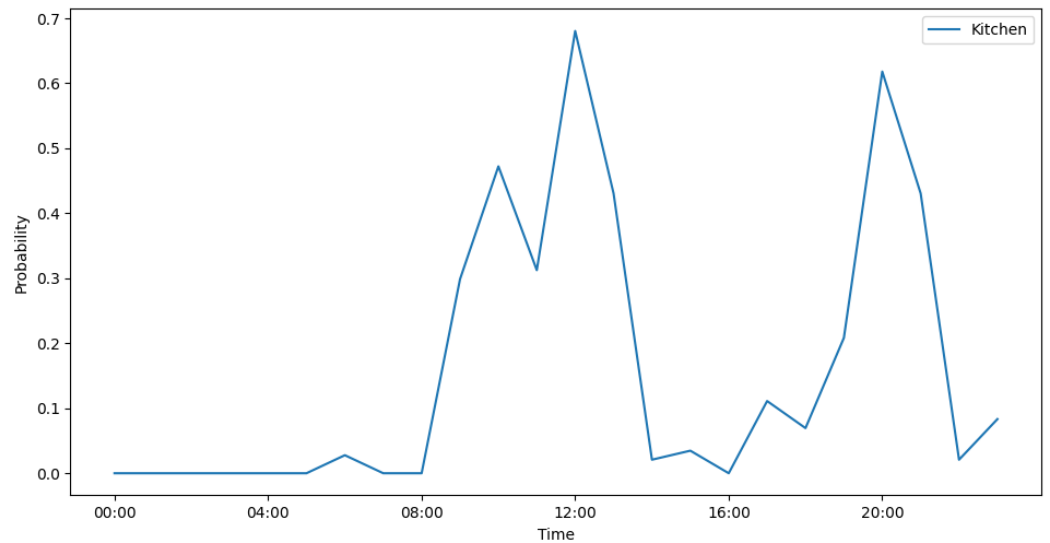


Figure 19. Probability of occupancy in the kitchen compartment.

The bathroom compartment does not have very high values about the probability of presence, essentially justified by the purpose of this compartment and the type of short-term activities that take place here, verified through Figure 20. Even so, the highest values observed between 9 am and 10 am and 10 pm and 11 pm agree with the patterns previously identified in Figures 10 and 15.

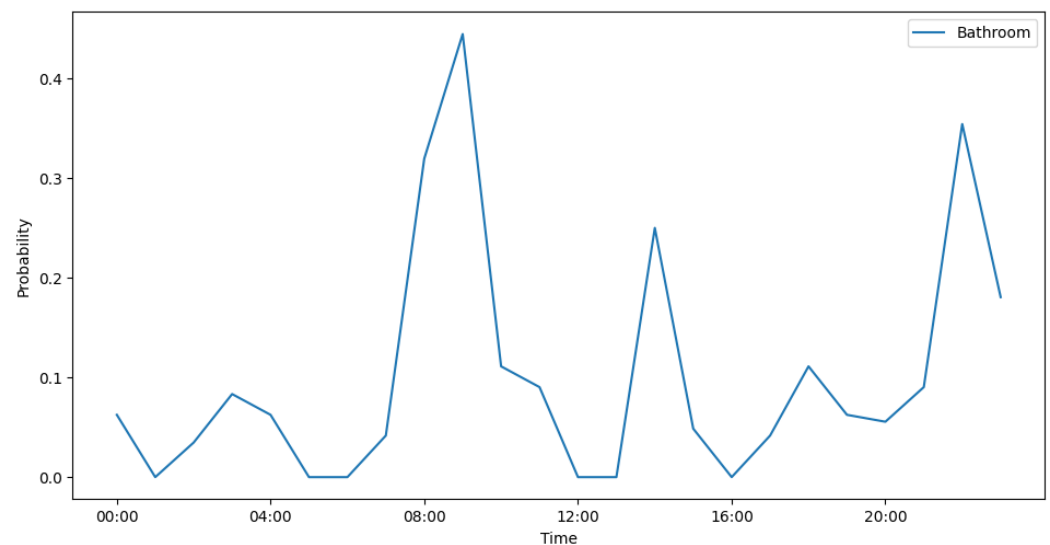


Figure 20. Probability of occupancy in the bathroom compartment.

In the bedroom compartment, the probability graph of Figure 21 shows a very well-defined probability of being in the bedroom between 12 and 9 am. In addition, from 11 pm onward, the likelihood of being in the room increases. This is because the person is sleeping during this time interval.

In the living room compartment, confirmation of presence around mid-afternoon is visible when the probability value is calculated, reaching high values between 3 pm and 4 pm, as shown in Figure 22. This result is excellent compared to the values obtained with the same analysis in the none or outside area (Figure 23).

As for the none or outside area, the probability graph in Figure 23 shows that it is expected or has a reasonable probability from 4 pm to 6 pm. Other times when the probability of this area may be relevant have to do with the fact that the sensor, sometimes, cannot read the data from the beacons of each compartment.

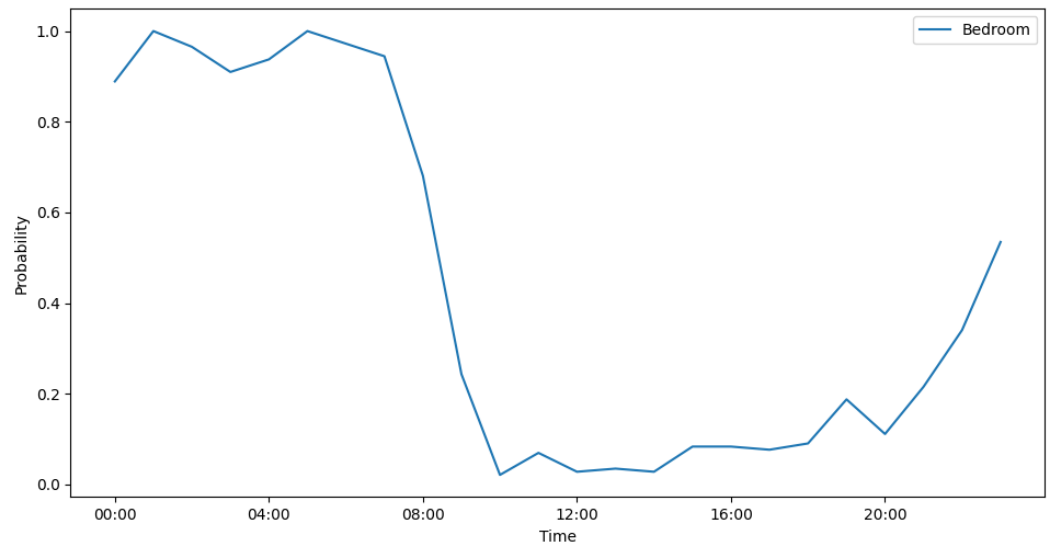


Figure 21. Probability of occupancy in the bedroom compartment.

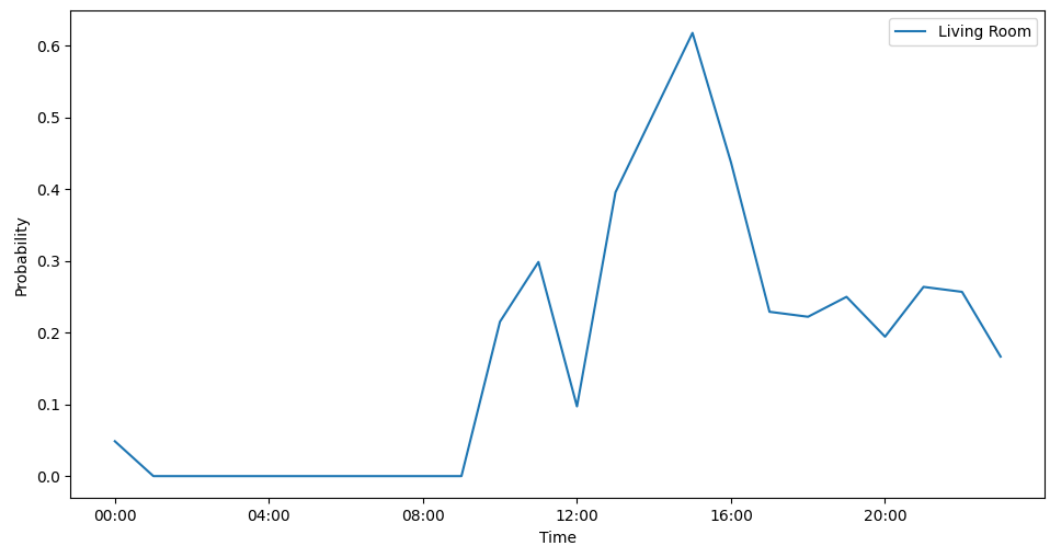


Figure 22. Probability of occupancy in the living room compartment.

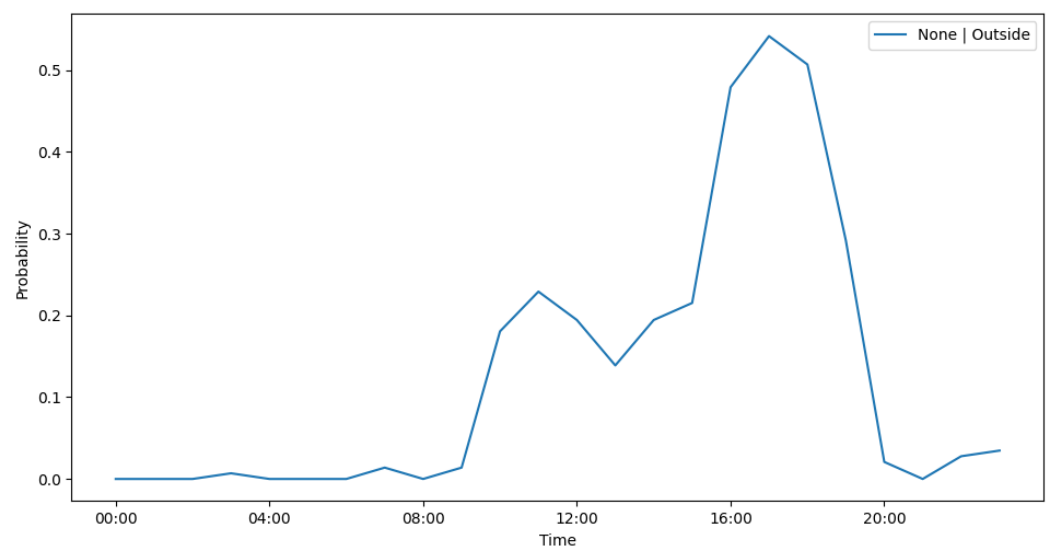


Figure 23. Probability of occupancy in none or outside.

Observing the probability of space occupation across all compartments, as depicted in Figure 24, allows for easy discernment of movement patterns. It is evident that the individual under study typically occupies the bedroom for sleeping between 11 pm and 8–9 am. During this period, the person will likely visit the bathroom at night. Subsequently, the person arises from bed, exits the bedroom, and heads to the bathroom for morning activities between 9 am and 10 am. Following this, between 10 am and 11 am, it is expected to find the individual in the kitchen for breakfast. Between 11 am and 12 pm, the person is typically between the living room and the kitchen. A high probability of being in the kitchen is observed between 12 pm and 1 pm, coinciding with the customary lunch hour. Post-lunch, individuals typically remain in the living room, with occasional bathroom visits, between 1 pm and 4 pm. Subsequently, between 4 pm and 7 pm, it is customary for the person to be outside the house for a walk. The individual is expected to be in the kitchen for dinner between 7 pm and 9 pm. Following dinner, the person may spend time between the living room and the bathroom until approximately 11 pm.

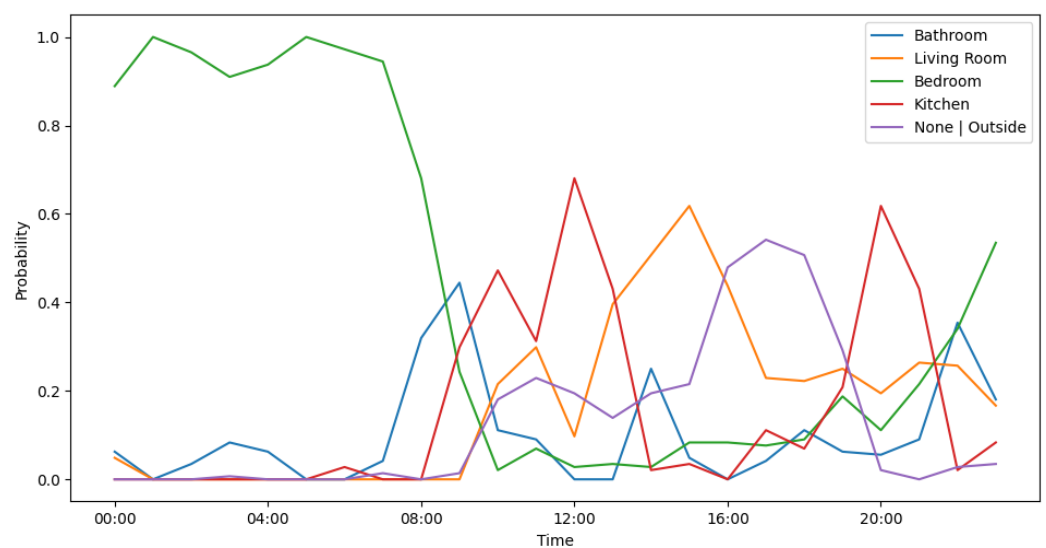


Figure 24. Occupancy probability in all compartments.

4.4. Outlier Detection

During the experimentation phase, the developed model was submitted to an analysis to identify outliers in the newly received data. These outliers were categorized into three levels, reflecting the degree of deviation from standard patterns. The objective was to continually improve the model’s performance, minimize false positives, and enhance its accuracy in responding to the evolving data.

Figure 25 provides a graphical representation of the outcomes to evaluate the model’s overall performance. This graph depicts the number of outliers identified at each level during the experimentation phase and the conclusion of the work. This iterative process aimed to fine-tune the model’s capabilities, enhancing its ability to detect and classify outliers accurately.

Therefore, in Figure 26, there is an example of an email generated when the algorithm detects an outlier when receiving new data collected by the person’s smartwatch. With the visualization of Figure 26, it is noticed that most of the outliers created were of the “low” level. This is because, during those hours, data that deviated from the standard were detected but were not critical or even negligible. Also, notice that the email has two types of messages.

The first message corresponds to the fact that there is a compartment with a significant occupancy probability at a given time. Still, its probability has been reduced this time because of another compartment. That is when usually the person’s pattern is to be in a particular compartment. Still, that day, there was another person’s behavior, spending

more time in another compartment than usual. For example, this first type of message can be visible with the phrase *At 03:00 AM, the usual location has been changed from the “Bedroom” to the “Bathroom”*. This means that the presence in the “Bedroom” has become a pattern at that hour of the day; moreover, the “Bathroom” is also trending on that specific day.

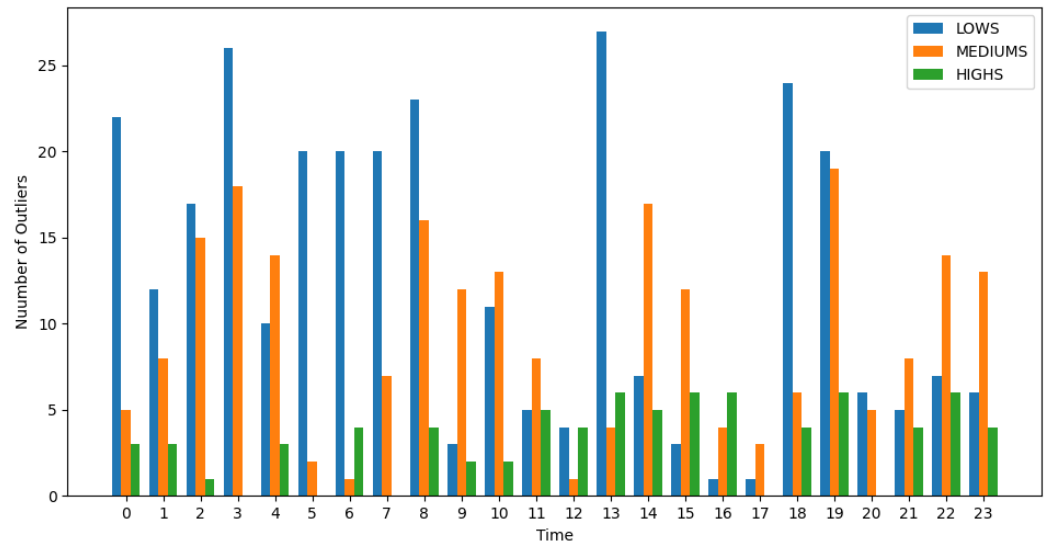


Figure 25. Number of outliers detected during the experimentation phase.

Outliers were detected in your patient today

LOW:

- At 03:00 AM, the usual location has been changed from the 'Bedroom' to the 'Bathroom'
- At 08:00 AM, the usual location has been changed from the 'Bedroom' to the 'Living Room'
- At 10:00 AM, the usual location has been changed from the 'Kitchen' to the 'Living Room'
- At 12:00 PM, the usual location has been increased by 'Kitchen' at the expense of 'None | Outside'
- At 02:00 PM, the usual location has been increased by 'Living Room' at the expense of 'Bathroom'

MEDIUM:

- At 03:00 PM, the usual location has been changed from the 'Living Room' to the 'None | Outside'

HIGH:

- At 11:00 AM, the usual location has been changed from the 'Kitchen' to the 'None | Outside'

This is an automatic email, please do not reply

Figure 26. Example of emails sent if outliers are detected.

The second message corresponds to the fact that a compartment is well defined but not significantly so. So, when there is a very high probability of that same division, it can generate a message indicating that the likelihood of that compartment has been increased even more. This second type of message will generally always appear at the “low” level of outliers, as it is rare for there to be a situation in which there is a time when there is not at least one well-defined compartment at that time, which at least makes it possible to analyze which the probability of frequenting that space is excellent. For example, in the generated email, we can see a message of this type: *“At 12:00 PM, the usual location has been increased by ‘Kitchen’ to the ‘Living Room’”*.

In this way, the email manages to be quite descriptive of what it detected, helping whoever is reading what is happening that day.

5. Conclusions

Pattern recognition was successfully achieved in all compartments through a series of analyses. The identification of movement patterns was confirmed through the different analysis, accumulated data, percentiles, probability, and the outlier detection mechanism. Even though the dataset may contain irregularities in the records, the model demonstrated the ability to adapt to the identified patterns. The model's ability to adapt and recognize the identified patterns highlights its robustness and reliability.

The accumulated data were crucial in establishing movement patterns within the compartments. By analyzing the cumulative data over time, we could discern consistent trends and understand how activities unfolded within each area. Percentiles provided further insight into the distribution of values within the dataset. This allowed for a more nuanced understanding of the variations in movement patterns and helped identify activities that deviated from the norm. The probability analysis significantly determined the likelihood of specific events or actions occurring within the compartments. By assigning probabilities to various scenarios, we could quantify the occurrence of particular behaviors and validate the identified patterns. Furthermore, the outlier detection mechanism was another validation tool for the identified movement patterns. The mechanism ensured that any irregularities or unexpected behaviors were accounted for and adequately addressed by detecting and flagging anomalies or outliers. Identifying outliers and studying patterns are distinct processes, but they complement each other in ensuring the correct interpretation of the information. We can validate the identified patterns and comprehensively understand the data by analyzing both aspects.

Limitations of the method occur when the number of beacons increases and in the data processing step. The complexity of Algorithm 1 is primarily determined by the pairwise comparison of beacons, which scales quadratically as $O(b^2)$, and the data processing step, which scales as $O(n \times c \times r)$, where b is the number of beacons, n is the number of data points, c is the number of compartments, and r is the number of RSSI values per compartment. Future work may involve optimizing these two components. To reduce the pairwise beacon comparison complexity, beacons can be grouped into clusters based on logical proximity, and only beacons within the same cluster can be compared. Another possibility is to use hashing techniques like Locally Sensitive Hashing [28] to approximate string similarity (s_i, s_j) instead of exact comparisons. To reduce the complexity of the data processing step, parallel processing can be used to process independent operations on different compartments (c_i) on time intervals (t_i) .

In conclusion, the combination of pattern recognition and outlier detection mechanisms enabled the correct interpretation of the information within the dataset. Various analyses identified and validated movement patterns, including accumulated data, percentiles, probability assessments, and outlier detection. This comprehensive approach ensured the accuracy and reliability of the results and provided valuable insights into the daily activities and behaviors of the individuals under study.

Author Contributions: Conceptualization, R.D.; methodology, R.D.; software, G.A.; validation, C.C. and A.M.; formal analysis, A.M.; investigation, G.A.; resources, R.D.; data curation, G.A.; writing—original draft preparation, G.A.; writing—review and editing, R.D., C.C., and A.M.; visualization, G.A.; supervision, R.D.; project administration, R.D.; funding acquisition, R.D. All authors have read and agreed to the published version of the manuscript.

Funding: This work is funded by National Funds through the FCT—Foundation for Science and Technology, I.P., within the scope of the project Ref. UIDB/05583/2020. Furthermore, we thank the Research Centre in Digital Services (CISED) and the Instituto Politécnico de Viseu for their support. Moreover, the authors greatly thank IPV/CGD—Instituto Politécnico de Viseu/ Caixa Geral de Depósitos Bank, within the scope of the projects PIDI/CISED/2022/007 and PIDI/CISED/2023/011.

Data Availability Statement: The datasets presented in this article are not readily available because the data are part of an ongoing study or due to technical limitations. Requests to access the datasets should be directed to Rui Pedro Duarte.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

This appendix relates to the definition of the functions that appear in Algorithm 1.

- ReadData(file_name):

$$\text{ReadData} : \mathbb{S} \rightarrow \mathcal{D}$$

Reads the data from a file specified by *file_name* and returns the set of recorded data \mathcal{D} .

- RetrieveBeaconInfo():

$$\text{RetrieveBeaconInfo} : \emptyset \rightarrow \mathcal{B}$$

Retrieves the set of all beacon information, returning \mathcal{B} .

- UUID(b_i):

$$\text{UUID} : b_i \rightarrow \mathbb{S}$$

- GetCompartment(UUID(b_i)):

$$\text{GetCompartment} : \text{UUID} \rightarrow \mathcal{C}$$

Maps the unique identifier (UUID) of a beacon b_i to its associated compartment $c_i \in \mathcal{C}$.

- Eq(s_i, s_j):

$$\text{Eq} : \mathbb{S} \times \mathbb{S} \rightarrow [0, 1]$$

Calculates the percentage of equivalence between two compartment strings s_i and s_j .

- Concatenate(s_i, s_j):

$$\text{Concatenate} : \mathbb{S} \times \mathbb{S} \rightarrow \mathbb{S}$$

Combines the strings s_i and s_j if they represent the same compartment.

- GetSmaller(UUID(b_j)):

$$\text{GetSmaller} : \{\text{UUID}(b_j)\} \rightarrow \text{UUID}(b_m)$$

Selects the smallest UUID among the beacons b_j within a compartment c_i , designating it as the “master” beacon.

- RedirectData(b_j, m):

$$\text{RedirectData} : \mathcal{B} \times \text{UUID} \rightarrow \mathcal{B}$$

Redirects the data from beacon b_j to the “master” beacon m .

- Min(d_i):

$$\text{Min} : \mathcal{D} \rightarrow t_{\text{earliest}}$$

Identifies the earliest timestamp in the data d_i .

- Max(d_i):

$$\text{Max} : \mathcal{D} \rightarrow t_{\text{latest}}$$

Identifies the latest timestamp in the data d_i .

- numDays($t_{\text{earliest}}, t_{\text{latest}}$):

$$\text{numDays} : \mathbb{T} \times \mathbb{T} \rightarrow \mathbb{N}$$

Computes the total number of days between t_{earliest} and t_{latest} .

- Prob(c_j, t_i):

$$\text{Prob} : \mathcal{C} \times \mathbb{T} \rightarrow [0, 1]$$

Calculates the probability of occupancy for a compartment c_j at a given time t_i .

- Avg(RSSI $_{b_i, c_j, t_i}$):

$$\text{Avg} : \mathbb{R} \times \mathcal{C} \times \mathbb{T} \rightarrow \mathbb{R}$$

Computes the average Received Signal Strength Indicator (RSSI) for beacon b_i in compartment c_j at time t_i .

- $\text{StdDev}(\text{RSSI}_{b_i, c_j, t_i})$:

$$\text{StdDev} : \mathbb{R} \times \mathcal{C} \times \mathbb{T} \rightarrow \mathbb{R}$$

Computes the standard deviation of RSSI values for beacon b_i in compartment c_j at time t_i .

- $\text{Perc}(\text{RSSI}_{b_i, c_j, t_i, p})$:

$$\text{Perc} : \mathbb{R} \times \mathcal{C} \times \mathbb{T} \times \mathbb{N} \rightarrow \mathbb{R}$$

Calculates the p th percentile of RSSI values for beacon b_i in compartment c_j at time t_i .

- $\text{Store}(\text{prb}, \text{std}, \text{perc25}, \text{perc75})$:

$$\text{Store} : [0, 1] \times \mathbb{R} \times \mathbb{R} \times \mathbb{R} \rightarrow \mathcal{P}$$

Stores the calculated probabilities and statistical metrics in the \mathcal{P} structure (e.g., the *pattern.csv* file).

References

1. Wang, H.; Bah, M.J.; Hammad, M. Progress in Outlier Detection Techniques: A Survey. *IEEE Access* **2019**, *7*, 107964–108000. [CrossRef]
2. Huang, D.; Mu, D.; Yang, L.; Cai, X. CoDetect: Financial Fraud Detection with Anomaly Feature Detection. *IEEE Access* **2018**, *6*, 19161–19174. [CrossRef]
3. Ferdousi, Z.; Maeda, A. Unsupervised Outlier Detection in Time Series Data. In Proceedings of the 22nd International Conference on Data Engineering Workshops (ICDEW'06), Atlanta, GA, USA, 3–7 April 2006; p. x121. [CrossRef]
4. Fahim, M.; Sillitti, A. Anomaly Detection, Analysis and Prediction Techniques in IoT Environment: A Systematic Literature Review. *IEEE Access* **2019**, *7*, 81664–81681. [CrossRef]
5. Christy, A.; Gandhi, G.M.; Vaithyasubramanian, S. Cluster Based Outlier Detection Algorithm for Healthcare Data. *Procedia Comput. Sci.* **2015**, *50*, 209–215. [CrossRef]
6. Gebremeskel, G.B.; Yi, C.; He, Z.; Haile, D. Combined data mining techniques based patient data outlier detection for healthcare safety. *Int. J. Intell. Comput. Cybern.* **2016**, *9*, 42–68. [CrossRef]
7. Linardatos, P.; Papastefanopoulos, V.; Kotsiantis, S. Explainable AI: A Review of Machine Learning Interpretability Methods. *Entropy* **2021**, *23*, 18. [CrossRef]
8. Alloghani, M.; Al-Jumeily, D.; Mustafina, J.; Hussain, A.; Aljaaf, A.J. A systematic review on supervised and unsupervised machine learning algorithms for data science. In *Supervised and Unsupervised Learning for Data Science*; Springer: Cham, Switzerland, 2020; pp. 3–21.
9. Crontab Guru. Available online: <https://crontab.guru> (accessed on 3 April 2024).
10. Benhammou, Y.; Ouhbi, B.; Qaraai, Y.; Boulmalf, M.; Khaloufi, H. A Hybrid Approach for Anomaly Detection in Elderly People's Daily Life Activities. *J. Ambient. Intell. Humaniz. Comput.* **2021**, *12*, 10749–10763.
11. Fraga-Lamas, P.; Fernández-Caramés, T.M.; Suárez-Albela, M.; Castedo, L. A review on Internet of Things for defense and public safety. *Sensors* **2016**, *16*, 10. [CrossRef]
12. Tahir, A.; Taylor, W.; Taha, A.; Usman, M.; Shah, S.A.; Imran, M.A.; Abbasi, Q.H. IoT Based Fall Detection System for Elderly Healthcare. In *Internet of Things for Human-Centered Design: Application to Elderly Healthcare*; Springer: Singapore, 2022; pp. 209–232. [CrossRef]
13. Sprint, G.; Cook, D.J.; Fritz, R. Behavioral differences between subject groups identified using smart homes and change point detection. *IEEE J. Biomed. Health Inform.* **2020**, *25*, 559–567. [CrossRef] [PubMed]
14. Bilbao-Jayo, A.; Almeida, A.; Sergi, I.; Montanaro, T.; Fasano, L.; Emaldi, M.; Patrono, L. Behavior modeling for a beacon-based indoor location system. *Sensors* **2021**, *21*, 4839. [CrossRef] [PubMed]
15. Almeida, A.; Azkune, G. Predicting human behaviour with recurrent neural networks. *Appl. Sci.* **2018**, *8*, 305. [CrossRef]
16. Amiribesheli, M.; Bouchachia, H. A tailored smart home for dementia care. *J. Ambient Intell. Human Comput.* **2018**, *9*, 1755–1782. [CrossRef]
17. Estimote UWB Beacons. Available online: <https://estimote.com/> (accessed on 3 April 2024).
18. Ratcliff, J.W.; Metzener, D. Gestalt: An introduction to the Ratcliff/Obershelp pattern matching algorithm. *Dr. Dobbs J.* **1988**, *7*, 46.
19. Vinutha, H.P.; Poornima, B.; Sagar, B.M. Detection of Outliers Using Interquartile Range Technique from Intrusion Dataset. In *Information and Decision Sciences*; Satapathy, S.C., Tavares, J.M.R., Bhateja, V., Mohanty, J.R., Eds.; Springer: Singapore, 2018; pp. 511–518.
20. Seo, S. A Review and Comparison of Methods for Detecting Outliers in Univariate Data Sets. Ph.D. Thesis, University of Pittsburgh, Pittsburgh, PA, USA, 2006.
21. Hoaglin, D.C.; Iglewicz, B. Fine-tuning some resistant rules for outlier labeling. *J. Am. Stat. Assoc.* **1987**, *82*, 1147–1149. [CrossRef]
22. Hoaglin, D.C.; Iglewicz, B.; Tukey, J.W. Performance of some resistant rules for outlier labeling. *J. Am. Stat. Assoc.* **1986**, *81*, 991–999. [CrossRef]

23. Tukey, J.W.; *Exploratory Data Analysis*; Addison-Wesley: Reading, MA, USA, 1977; Volume 2.
24. DeVore, G.R. Computing the Z score and centiles for cross-sectional analysis: A practical approach. *J. Ultrasound Med.* **2017**, *36*, 459–473. [[CrossRef](#)] [[PubMed](#)]
25. Pradhan, R. Z-score estimation for Indian banking sector. *Int. J. Trade Econ. Financ.* **2014**, *5*, 516. [[CrossRef](#)]
26. GADGET.FITNESS. Garmin Vivoactive 4 Review. 2020. Available online: <https://gadget.fitness/garmin-vivoactive-4/> (accessed on 8 December 2024).
27. Nunan, D.; Sandercock, G.R.H.; Brodie, D.A. A Quantitative Systematic Review of Normal Values for Short-Term Heart Rate Variability in Healthy Adults. *Pacing Clin. Electrophysiol.* **2010**, *33*, 1407–1417. [[CrossRef](#)] [[PubMed](#)]
28. Jafari, O.; Maurya, P.; Nagarkar, P.; Islam, K.M.; Crushev, C. A survey on locality sensitive hashing algorithms and their applications. *arXiv* **2021**, arXiv:2102.08942. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.