Article

Sensor Fusion for Occupancy Estimation: A Study Using Multiple Lecture Rooms in a Complex Building

Cédric Roussel *[ID], Klaus Böhm and Pascal Neis

i3mainz—Institute for Spatial Information and Surveying Technology,
Mainz University of Applied Sciences, 55128 Mainz, Germany
* Correspondence: cedric.roussel@hs-mainz.de

Abstract: This paper uses various machine learning methods which explore the combination of multiple sensors for quality improvement. It is known that a reliable occupancy estimation can help in many different cases and applications. For the containment of the SARS-CoV-2 virus, in particular, room occupancy is a major factor. The estimation can benefit visitor management systems in real time, but can also be predictive of room reservation strategies. By using different terminal and non-terminal sensors in different premises of varying sizes, this paper aims to estimate room occupancy. In the process, the proposed models are trained with different combinations of rooms in training and testing datasets to examine distinctions in the infrastructure of the considered building. The results indicate that the estimation benefits from a combination of different sensors. Additionally, it is found that a model should be trained with data from every room in a building and cannot be transferred to other rooms.

1. Introduction

Since spring 2020, the SARS-CoV-2 virus has made a global impact on humanity. There is a high risk of infection, especially in enclosed indoor environments. To achieve better containment of the virus, it is highly significant to manage visitors in buildings. This is necessary in real time, but also predictive. The most difficult part of this task is to estimate the real occupancy in each building room. For this, different sensors can be used. Every sensor has some limitations regarding resolution, costs, the ability of detection, privacy, scalability, and social acceptance [1]. The resolution of a sensor can be measured temporally (day, hour, minute, second), spatially (building, floor or zones [2,3], room), and in terms of occupancy (occupancy [4], count [5], identity, activity) [6]. Sensors can be categorized into terminal and non-terminal sensors [7]. Terminal sensors typically require an opponent. As an example, when utilizing Wi-Fi access points (APs) as occupancy sensors, a corresponding device, such as a smartphone or notebook, is necessary. With this method, the Hawthorne effect [8] must be respected. It describes the change in the natural behavior of a person during an experiment. Non-terminal sensors do not need any other sensors than the measuring sensor itself, for example when measuring room air quality. Many investigations used carbon dioxide (CO_2) concentrations [4,5,9], Bluetooth [2,3], or Wi-Fi [10] to estimate room occupancy. However, the privacy of subjects and the Hawthorne effect were not always considered. Especially when using terminal sensors, this has to be respected. Ref. [3] used the media access control (MAC) address which raises privacy concerns. Ref. [2] used a smartphone application on the subjects' smartphones. This can change the subjects' behavior, which does not respect the Hawthorne effect. Other researchers used multiple sensors, such as light, sound, motion, and temperature [11], or the ventilation state of a heating, ventilation, and air-conditioning (HVAC) system [9] to estimate room occupancy. Ref. [9] discarded Wi-Fi data as it is not accurate enough.

However, if every room contains one single AP, the same methodology applies, like their HVAC system. Camera-based room occupancy estimation has been used in many studies and showed good results [12–15], but comes along with significant privacy concerns. Very few studies combined sensors for room occupancy. Furthermore, almost all of them used one single or two rooms as training and test data [2,3,5,9]. However, buildings show significant differences in infrastructure in rooms. The trained models can probably not be applied to all rooms in the considered building. For this reason, our addressed research questions are:

a. How much can the occupancy estimation accuracy be improved using multiple sensors?
b. How can one single model be trained for all rooms in a building?
c. How does the quality of the estimation in a room behave when only using data from other rooms?

To answer these questions, we implemented different sensors in different rooms with a variation of occupancy and implement machine learning models. We highly respected privacy concerns and the Hawthorne effect to obtain realistic data. Therefore, we waived camera-based detections and only used sensors that have a low level of intrusiveness and gain anonym data.
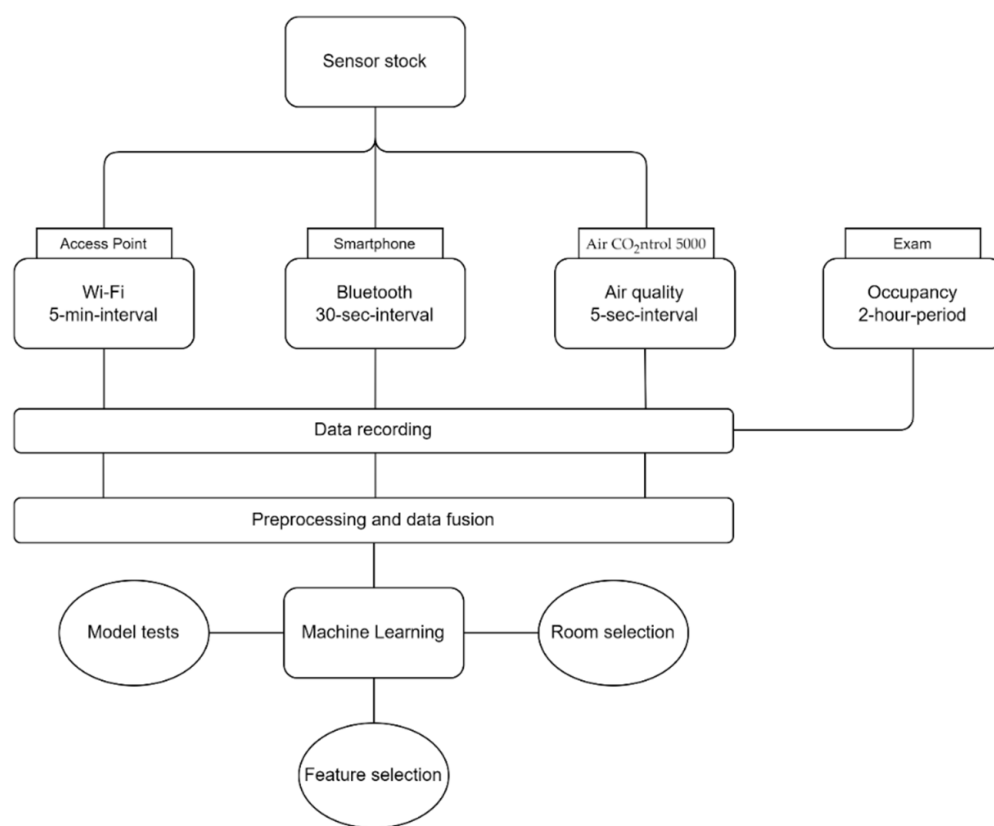
2. Materials and Methods

2.1. Methodology

To predict occupancy data, the first step was to define which sensors we will use to measure data. It is relevant how the sensors can be integrated into the infrastructure of the building and whether they can be acquired cost-effectively for the development of a prototype. In the present building of the Mainz University of Applied Sciences, Wi-Fi access points already exist in many rooms. There were also air quality sensors in selected rooms which were purchased at the beginning of the COVID-19 pandemic. The sensors measure CO_2 concentration, room temperature, and relative humidity. A sensor for the measurement of the number of Bluetooth devices was not available in the infrastructure. For this, we used a standard smartphone with a monitoring app, tailored to our application case.

To capture data, we used different premises. Different rooms provide different data and results due to their infrastructure, which cannot be disregarded in a prediction by a machine learning model. Using only one test room was not enough for a conclusion of the prediction model in the real world. Furthermore, we varied the number of people in a room. We used the two-week exam period in selected rooms for data collection. This period is suitable for data collection, as we know the real number of people in the room during the exams of two hours. Different rooms and sizes of occupancy favor data variations. In total, we collected data during 13 exams. If possible, due to room reservations, our sensors took measurements before and after the two-hour period. This was necessary to obtain data with no subject in the room. We placed the sensors as inconspicuously as possible in the room to respect the Hawthorne effect [8]. Obvious measurements with sensors would probably influence the natural behavior of the subjects. In addition, it was important that the sensors did not interfere with the students during the exam. All sensors used anonym data to protect the privacy of subjects.

After the collection phase, we processed and merged the data of all sensors into one dataset (sensor fusion). We applied various machine learning models, as well as a neural network, to predict the occupancy in rooms or to determine whether a room is occupied or vacant. For this, we used different combinations of rooms in training and testing. In addition to the prediction, we calculated the feature importance of each individual sensor. Finally, we presented and discussed our results in their quality.

Figure 1 visualizes the methodology in a flowchart, beginning with the selection of sensors and the following data recording. We further explain this step in Section 2.2. After that, we process and merge the data in Section 2.3. In the final step, we apply machine learning methods with different combinations of features and rooms in Section 2.4. We show and discuss the results in Section 3.
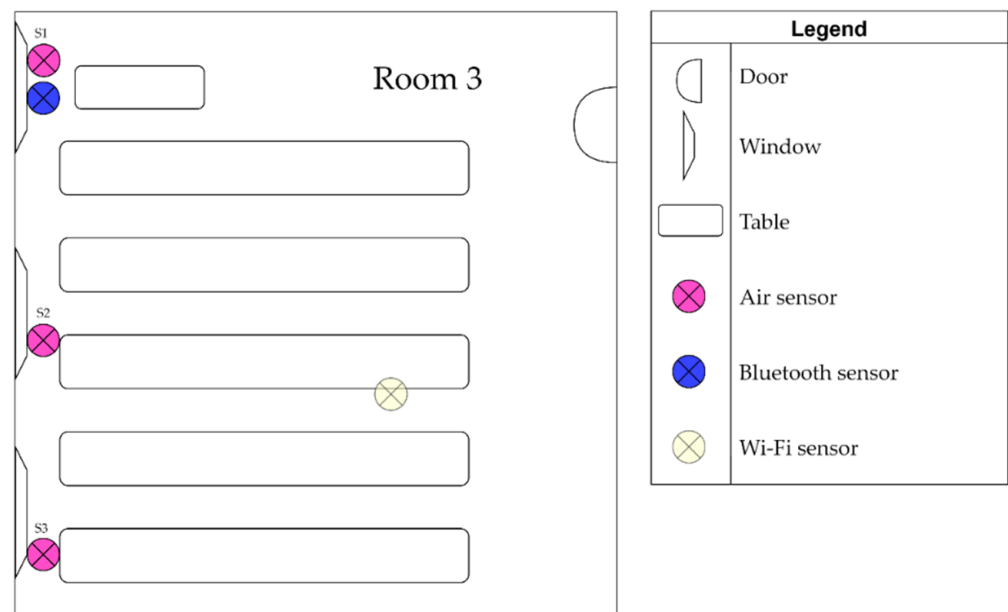
Figure 1. Methodology flow chart.

2.2. Data Recording

The given infrastructure provides data on the access points of the Wi-Fi. Every five minutes, the access points—with the Aruba7030 Mobility Controller [16]—recorded how many devices were logged into the network via the corresponding AP. We did not store further identification such as a MAC address. Additionally, there were existing sensors in selected rooms to measure the air quality. The sensors logged the carbon dioxide concentration, room temperature, and relative humidity locally every five seconds in a CSV file. In order to detect deviations from the measured values among the devices, we placed three sensors at different locations in each room. However, we could not place the sensors in the middle of the room as they needed a power source. Open windows and doors influenced the measured values. In our case, in every room, at least one window was open throughout the exam. At least one sensor, but not all, was expected to detect this. This respects real-world conditions. During the pandemic, windows and doors were rarely closed throughout a whole exam or lecture. The infrastructure of the respected building did not include a ventilation and air conditioning system, like in [9]. There was no further ventilation in the rooms.

To detect Bluetooth devices located in the room, we developed a tailored smartphone application using the Flutter framework [17]. The application was—theoretically—usable on different platforms. However, we only used Android smartphones for the test scenario. The application scanned nearby Bluetooth devices for 30 s at a time. We stored detected devices with received signal strength indication (RSSI) and again without MAC address. The RSSI represented a ratio indicating the received quality of the signal in decibels. We stored the detected device information on a Raspberry pi 4 after every 30 s. The Raspberry pi and the smartphone were connected via Wi-Fi. We discarded the option of using the Raspberry pi as a measuring sensor for Bluetooth devices due to its inflexibility, as it was tied to a power source. The smartphone was more flexible in positioning in the room.

As an example, Figure 2 shows the setup of all sensors in room 3. The access point was located on the ceiling of the room. We placed the smartphone for measuring the Bluetooth

devices at the supervisor's desk. We distributed the air quality sensors inside the room as best as possible. In this example, they were all located at the windowsill, as necessary power sources in this room were only located on this side. In general, we placed all sensors on the sides so that students would not feel disturbed during the exam. Furthermore, the sensors remained as unobtrusive as possible, so they did not influence people's behavior. In total, we collected data in five different rooms, each with a different number of people (Table 1).

Figure 2. Recording structure in room 3.

Table 1. Overview of recordings and room properties.

Date	Time	Identifier	Room Size [m^2]	People in Exam
4 July 2022	09:47–12:32	Room 1	407.75	49
6 July 2022	07:25–10:25	Room 1	407.75	58
6 July 2022	10:13–12:58	Room 2	59.61	6
7 July 2022	11:35–14:21	Room 3	59.01	16
8 July 2022	08:25–12:30	Room 4	90.23	15
8 July 2022	12:30–17:01	Room 4	90.23	25
11 July 2022	08:33–11:24	Room 3	59.01	10
12 July 2022	08:28–11:44	Room 5	78.40	12
12 July 2022	11:52–14:49	Room 2	59.61	13
13 July 2022	08:40–11:50	Room 5	78.40	22
14 July 2022	09:29–12:05	Room 5	78.40	18
14 July 2022	12:05–14:19	Room 5	78.40	8
15 July 2022	08:42–11:27	Room 1	407.75	45

2.3. Preprocessing

Before we examined the data with machine learning methods, we had to process it and combine the sensors into one dataset. The Bluetooth data, in particular, could not be further analyzed without filtering. When scanning nearby devices, it was not possible to set a radius to scan. Accordingly, the smartphone could also detect devices that were located outside the room. We used the stored RSSI to detect and remove these devices. However, it was difficult to find a threshold for filtering out devices based on the RSSI

alone. For this reason, we estimated the metric distances between the Bluetooth devices and the smartphone with the measured RSSI values, using Formula (1) [18]:

$$\text{Distance} = 10^{\left(\frac{\text{Measured Power-Instant RSSI}}{10 \cdot N}\right)} \text{ with } N \in [2; 4]. \tag{1}$$

We could only calculate the distances with uncertainties since not all variables of the formula were clearly determined. The measured power is the calibrated RSSI at a distance of one meter. The formula is normally used in the development of an indoor navigation to estimate the distance to beacons. This is because only one beacon model exists with a known consistent measured power. In the case of this project, we detected all surrounding Bluetooth devices. These send signals of different powers, which are unknown and differ from each other. To determine an estimated value for the measured power, we averaged the ten closest measurements to the smartphone in each room. The largest RSSI should not be used alone. A person with a Bluetooth device could walk past the sensor very closely. The measurement could then be closer than one meter. The constant N is arbitrary and represents the individual building. Each building was different in its infrastructure and affected the detection of Bluetooth devices. N was determined in the interval $2 \leq N \leq 4$ via a test series in order to represent the distances as realistically as possible. A higher value generally represents the attenuation of distances. Ref. [19] already used the formula in the same building and determined the value two for N. However, Ref. [19] only used one beacon type with known measured power. To attenuate the uncertainties in the measured data, we increase the value to three. A higher value for N would attenuate the measured data too much and subsequently not classify any Bluetooth devices as *outside the room*. With an estimated distance from the smartphone to the Bluetooth devices, we could eliminate devices that were too far away. We used the diagonal of the room as a threshold since we always positioned the smartphone in a corner of the room. Then, we grouped the remaining Bluetooth devices according to their timestamp in the interval of 30 s. For the other sensors, we used this interval as well to merge the data in the last step.

We only knew the number of persons during the time of the examination. The premises of the Mainz University of Applied Sciences were open all day during the exam period. This allowed students to enter the room before the start of the exam. This could lead to wrong occupancy data. In order to find an approximation of the number of people before and after an exam, we set the occupancy value to zero half an hour before. Then, we linearly increased the value every 30 s until the start of the exam with the known occupancy number. At the end of the exam, we linearly decreased the value for the following five minutes (Formula (3)). For this, we established two tailored formulas to linearly estimate the real occupancy 30 min before (2) and five minutes after an exam (3):
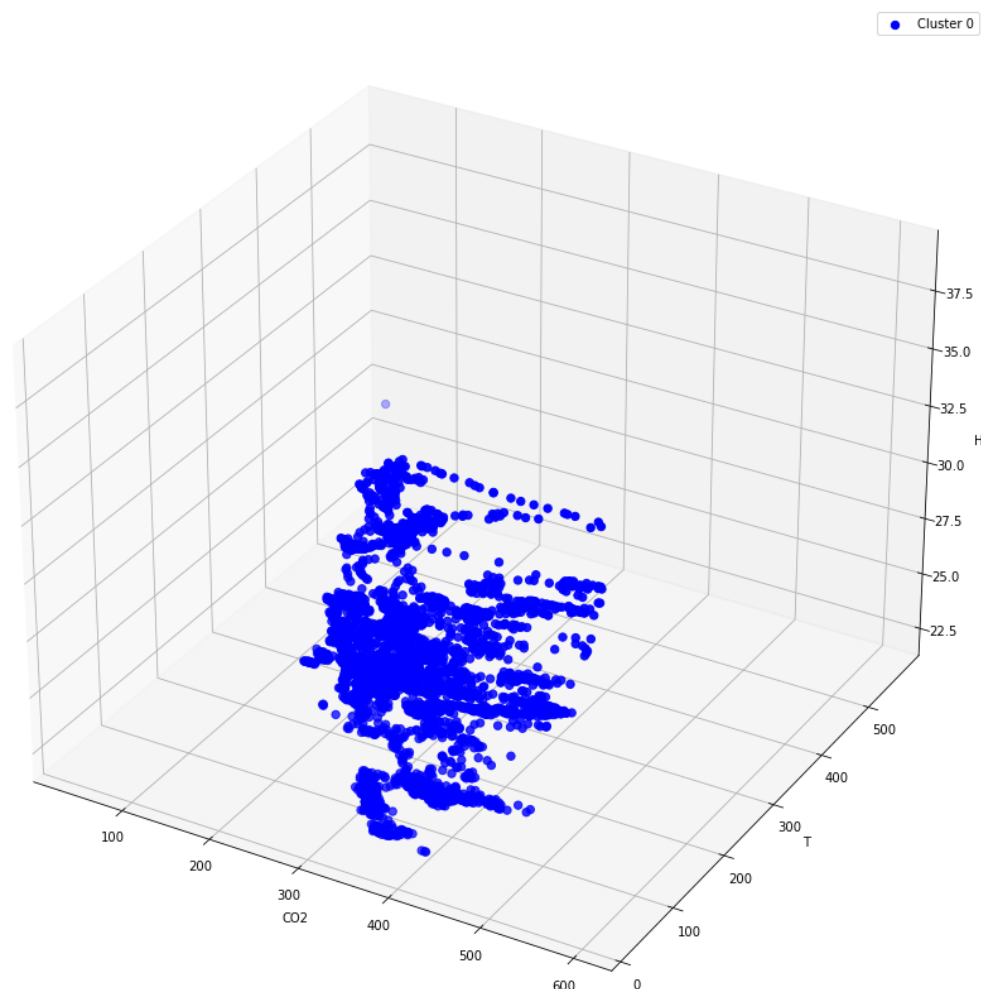
$$O_1(t)_{\in N} \approx \frac{a}{30} \cdot t + a, \ t \in \{ 0.5k \mid -60 \leq k \leq 0, \ k \in \mathbb{Z}\} \tag{2}$$

$$O_2(t)_{\in N} \approx \frac{a}{-5} \cdot t + a, \ t \in \{ 0.5k \mid 0 \leq k \leq 10, \ k \in \mathbb{Z}\}. \tag{3}$$

Parameter *a* represents the known number of people in the room during the exam. Parameter *t* is the time variable for 30 min before the start and five minutes after the end of the exam in increments of 0.5. We rounded the result to an integer to obtain realistic occupancy data 30 min before (O_1) and five minutes after (O_2) the exam. The estimated approximation only partially reflects reality, but we considered it a better choice than setting the number of people before and after the exam to zero or excluding them from the dataset.

We took the readings from the air quality sensors from the CSV file on the SD cards. During a measurement, external influencing factors may briefly distort the measured values. Ref. [20] obtained a maximum value of 3800 ppm in their measured values. Based on this value, we considered all measured values with a CO_2 concentration above 4000 ppm as unrealistic and eliminate them from the dataset. Subsequently, we average the data to a 30 s interval, reducing further fluctuation. To check for systematic measurement differences

between the three sensors, we performed a cluster analysis using the K-means algorithm. Figure 3 visualizes all data points with CO_2 concentration, room temperature, and relative humidity. We only found one cluster. This shows that we can rule out systematic deviations among the devices.

Figure 3. Cluster analysis of the air sensor data.

We took the number of logged-in devices in the corresponding access points from the present FROST server—an open-source implementation of the Open Geospatial Consortium Sensor Things API [21].

Finally, we merged the data from all sensors into a CSV file. We added missing values due to measurement errors via an imputer using the K-nearest-neighbor method.

2.4. Machine Learning Approach

With the dataset processed, we examined the data using various machine learning methods. In the first step, we reduced the three air quality measuring sensors to one measured value. For this, we used the largest values for carbon dioxide concentration, room temperature, and relative humidity of the sensors among each other. In many rooms, some open windows and doors altered the measurement of one sensor. We hypothesized that the largest values were most likely to simulate the number of people in rooms. For a comparison of results, we further experimented using the mean of the three sensor readings.

Before applying machine learning models, we checked for correlations. Then, we divided the data into training and testing. For the neural network, we further divided the test into test and validation. We defined the ratio of the split as 80:20 for the training and testing dataset and 50:50 for the testing and validation dataset. It was particularly

important that time series data were not mixed for the split. In a further analysis, we defined one room in the entire dataset as the test data and all other rooms as training data for the model. This showed the difference in rooms of the infrastructure. It could point out if a model has to use data in the training of the room, in which the prediction will be used. If the quality of the model is sufficient without the training data of the respective room, buildings with a large number of rooms would benefit. Otherwise, data from all rooms must be collected.

After splitting and scaling the data, we used various machine learning methods (linear regression; K-nearest-neighbor; zero-inflated regression with linear regression; and decision trees) to predict the exact occupancy number. We further used the classifiers (logistic regression; decision trees; support vector machine; Naive Bayes) to determine if a room was simply occupied or vacant. For this purpose, we also trained the neural network on the training data and validate and optimize it with the validation data. The test data served as a final statement on the quality of the model. We used different methods and approaches to find a model with the best possible accuracy. We show and discuss all results of the models in the next chapter.

Moreover, we investigated the feature importance of the sensor values in the dataset using an ordinary least squares (OLS) regression. This is also possible using a neural network, but it is unsuitable given the time required.

3. Results and Discussion

Initially, we used the largest measured value for carbon dioxide concentration, room temperature, and relative humidity of the sensors. However, the first results showed that the quality of the model increases slightly when we use the mean value from the three air sensors. Accordingly, the following results refer to the use of the mean values.

Figure 4 shows the correlations of the measured values with each other and with the target variable. In particular, the Bluetooth devices and the number of logged-in devices in the access point strongly correlate (0.88 and 0.8) with the real number of people in the room. These correlations are much higher than [15], who show their highest correlation with acoustic sensors (0.48). However, Ref. [15] reached higher values for CO_2 (0.36), relative humidity (0.32) and temperature (0.12). The differences result due to the different infrastructure of the building with office rooms. Office rooms are smaller than lecture rooms. This is why [15] have better results for environmental features, as they increase faster and higher because of smaller room sizes.
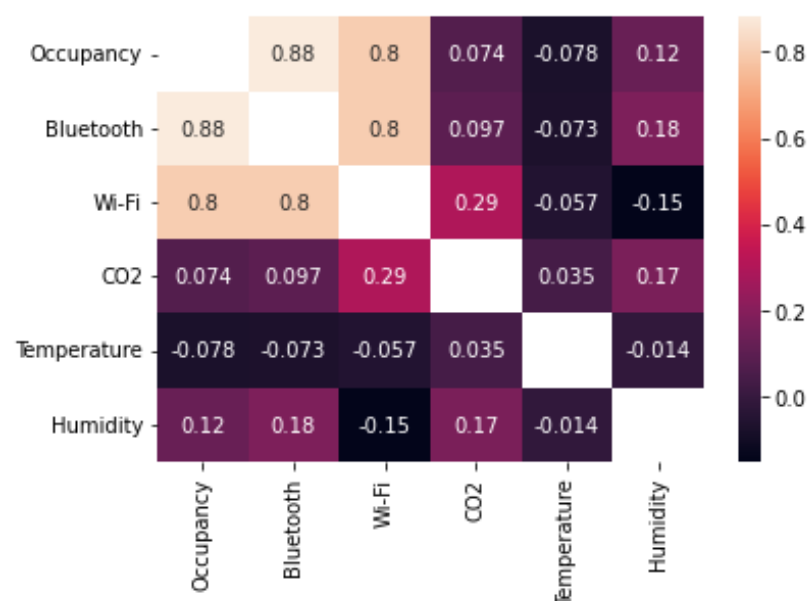

Figure 4. Correlations of features.

We further investigated the importance of the attributes to model the number of people by calculating the feature importance. Accordingly, we obtained the following values from the OLS regression with an accuracy of 0.70 (Table 2):

Table 2. Feature importance of sensors using OLS regression.

Attribute	Feature Importance	Standard Error	*p*-Value
Bluetooth	10.82	0.336	0.000
Wi-Fi	1.59	0.361	0.000
Carbon dioxide	−1.03	0.287	0.000
Temperature	−0.28	0.259	0.283
Relative humidity	0.47	0.286	0.098

The first three attributes show highly significant values and can be interpreted. The Bluetooth data show the greatest influence, followed by Wi-Fi and CO_2. We used all attributes for the next algorithms. We performed initial model tests with a data split of 80:20. The linear regression achieves an accuracy of 0.65 in training and 0.76 in testing with a RMSE of 7.9. To interpret and compare the RMSE to other results, we used Formula (4) [5], which takes the mean number of subjects N_{ave} in the dataset into account:

$$ CV = \frac{RMSE}{N_{ave}} = \frac{\sqrt{\frac{\sum (N_{est} - N_{real})^2}{n}}}{N_{ave}}. \tag{4} $$

For the above-mentioned result, the value is 52.67%, which is in the result range of [5] with 40–60%. **That is to say, we achieved the same quality for a model for multiple rooms**. When we only used two attributes, Bluetooth and Wi-Fi, the accuracy increases to 0.71 and 0.84, respectively, and the RMSE decreases to 6.2 (CV = 41.3%). The coefficients of the linear regression are also of similar magnitude with 6.82 and 5.24. The KNN algorithm achieves an accuracy of 0.98 in training and 0.77 in testing with an RMSE of 7.6. The difference in accuracy shows poorer generalization. The zero-inflated regression does not show better results with an accuracy of 0.69 and 0.67, respectively, with an RMSE of 9.1. **When we only use one feature for training, no model achieves sufficient accuracy.** Especially when using only one air quality feature, the RMSE significantly increases up to 17.4. This clearly shows the advantage of combining at least two different sensors.

Next, we test the linear regression for different data splits. We define the data of one room as the test set and train the regression with all other rooms. Table 3 shows the results for all individual rooms.

Table 3. Linear regression results using room data split.

Test Room	Training	Testing	RMSE	CV [%]
Room 1	0.70	−2.10	31.4	82.6
Room 2	0.68	0.67	2.7	38.6
Room 3	0.71	−1.82	8.5	77.3
Room 4	0.78	−0.05	10.4	74.3
Room 5	0.85	−2.81	13.6	123.6

In four cases, the test accuracy is negative. This means that an estimation via the mean value provides better results than the model. Only room 2 shows usable results. This points out that the different rooms with their infrastructure show strong differences in the data. **A model should therefore always include data from the corresponding room in the training data**. It was not possible to set up a model for each room separately since the data basis was not sufficient. Refs [15] show that at least 20,000 data points stabilize the state of estimation. In our prototype, we used less than 5000.

As a further test, we implemented classifiers to determine whether a room was simply occupied or not. The exact number of people was not of interest. We adjusted the target variable to the values zero and one. The logistic regression shows the best result with an accuracy of 0.85 and 0.91 in the test. Other classifiers show slight differences between training and test. For this reason, we tested a voting classifier with logistic regression, decision trees, and Naive Bayes with the weights [4;1;1]. We proceeded with soft voting, where all probabilities were added and the highest probability determined the result. The result was an accuracy of 0.89 and 0.90 in the test, therefore showing good generalization. Last, we trained a neural network with an input layer and two hidden layers with the rectified linear unit activation function. We built the output with one neuron and Softmax activation function. Usually, the dimension of the output layer is equal to the number of classes present. In binary classification, a neuron with the Softmax activation function can be used to keep the complexity of the model low. We trained the model for 200 epochs and continuously decreased the learning rate by the factor $e^{-0.1}$ after 180 epochs via a callback. The accuracies of training and validation show good generalization. On the test data, the model gave an accuracy of 0.97. The neural network thus shows the best results in the case of classification of whether a room is occupied or not.

4. Conclusions and Future Work

The actual number of occupants in a room plays a crucial role in visitor management. For this, we used different sensors to capture training and test data. We used Wi-Fi, Bluetooth, carbon dioxide concentration, room temperature, and relative humidity. After processing the data through different necessary steps, we applied various machine learning models. With respect to our research questions, we made three major findings:

1. Wherever applicable, due to infrastructure, multiple sensors should be used for data gathering. The quality of estimation always benefits from combining different sensors, compared to models with only one sensor. However, using all sensors might not be the best solution. Through test cases, the best combination of different sensors should be determined. In the case of our study, we improved the RMSE from 17.4 to 6.2, combining different features compared to only using one feature.
2. It is possible to train a single model for all rooms in a building. However, the model must be trained with data from all rooms in the building, which may lead to higher costs in bigger buildings with more rooms. This leads to our final finding.
3. When defining training data for the model, the dataset should contain data from every room. A trained model from certain rooms shows no convincing results when tested in a new unknown room. This shows the complex differences in infrastructure inside a building. By only testing their model on one or two rooms, almost all studies did not respect this factor. For smaller buildings with fewer rooms, the effort would be manageable. For bigger buildings, sensors should be integrated into infrastructure and the data readings should be as automatic as possible to minimize effort.

This paper showed the relevance of using different sensors and multiple rooms during the data recording. With the knowledge of the benefit of different sensors, machine learning models can be improved. If a model/prototype will be transferred to a whole building, the impact of the infrastructure must be respected. Our finding clearly helps to avoid quality problems when implementing machine learning for occupancy estimation not in one or two rooms, but in a building with multiple different premises.

Further study should implement more sensors such as light, acoustics, or motion. The knowledge of open windows and doors should be included. For this, the outdoor air quality can be modeled and used as another feature input. In a new experiment, air conditioning should be documented during different seasons to analyze the impact of atmospheric air. We believe that these further studies are worth being tested to gage a better understanding of influencing factors on occupancy estimation. After new tests with different sensors and a better understanding of the impact of natural ventilation, new

state-of-the-art machine learning models should be implemented and tuned to optimize the accuracy.

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

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

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

References

1. Ahmad, J.; Larijani, H.; Emmanuel, R.; Mannion, M.; Javed, A. Occupancy detection in non-residential buildings—A survey and novel privacy preserved occupancy monitoring solution. *Appl. Comput. Inform.* **2020**, *17*, 279–295. [CrossRef]
2. Filippoupolitis, A.; Oliff, W.; Loukas, G. Bluetooth Low Energy based Occupancy Detection for Emergency Management. In Proceedings of the 15th International Conference on Ubiquitous Computing and Communications and 8th International Symposium on Cyberspace and Security, Los Alamitos, CA, USA, 14–16 December 2016. [CrossRef]
3. Tekler, Z.D.; Low, R.; Gunay, B.; Andersen, R.K.; Blessing, L. A Scalable Bluetooth Low Energy Approach to Identify Occupancy Patterns and Profiles in Office Spaces. *Build. Environ.* **2020**, *171*, 106681. [CrossRef]
4. Guillaume, A.-A. Estimating Occupancy Using Indoor Carbon Dioxide Concentrations Only in an Office Building: A Method and Qualitative Assessment. In Proceedings of the 11th REHVA World Congress "Energy Efficient, Smart and Healthy Buildings", Prague, Czech Republic, 16–19 June 2013.
5. Alam, A.G.; Rahman, H.; Kim, J.-K.; Han, H. Uncertainties in neural network model based on carbon dioxide concentration for occupancy estimation. *J. Mech. Sci. Technol.* **2016**, *31*, 2573–2580. [CrossRef]
6. Melfi, R.; Rosenblum, B.; Nordman, B.; Christensen, K. Measuring Building Occupancy Using Existing Network Infrastructure. International Green Computing Conference and Workshops, Orlando, FL, USA. 25–28 July. [CrossRef]
7. Lee, S.; Ha, K.N.; Lee, K.C. A pyroelectric infrared sensor-based indoor location-aware system for the smart home. *IEEE Trans. Consum. Electron.* **2006**, *52*, 1311–1317. [CrossRef]
8. Diaper, G. The Hawthorne Effect: A fresh examination. *Educ. Stud.* **1990**, *16*, 261–267. [CrossRef]
9. Wolf, S.; Cali, D.; Krogstie, J.; Madsen, H. Carbon dioxide-based occupancy estimation using stochastic differential equations. *Appl. Energy* **2019**, *236*, 32–41. [CrossRef]
10. Simma, K.C.J.; Mammoli, A.; Bogus, S.M. Real-Time Occupancy Estimation Using WiFi Network to Optimize HVAC Operation. *Procedia Comput. Sci.* **2019**, *155*, 495–502. [CrossRef]
11. Yang, Z.; Li, N.; Becerik-Gerber, B.; Orosz, M. A Multi-Sensor Based Occupancy Estimation Model for Supporting Demand Driven HVAC Operations. In Proceedings of the 2012 Symposium on Simulation for Architecture and Urban Design, Orlando, FL, USA, 26–30 March 2012; pp. 49–56.
12. Benezeth, Y.; Laurent, H.; Rosenberger, C. Towards a sensor for detecting human presence and characterizing activity. *Energy Build.* **2011**, *43*, 305–314. [CrossRef]
13. Munoz-Salinas, R.; Medina-Carnicer, R.; Madrid-Cuevas, F.J.; Carmona-Poyato, A. People detection and tracking with multiple stereo cameras using particle filters. *J. Vis. Commun. Image Representat.* **2009**, *20*, 339–350. [CrossRef]
14. Wang, F.; Feng, Q.; Chen, Z.; Zhao, Q.; Cheng, Z.; Zou, J.; Zhang, Y.; Mai, J.; Reeve, H. Predictive control of indoor environment using occupant number detected by video data and CO_2 concentration. *Energy Build.* **2017**, *145*, 155–162. [CrossRef]
15. Zhang, R.; Lam, K.P.; Chiou, Y.-S.; Dong, B. Information-theoretic environment features selection for occupancy detection in open office spaces. *Build. Simul.* **2012**, *5*, 179–188. [CrossRef]
16. HPE—Hewlett Packard Enterprise Development LP. Aruba 7000 Series Mobility Controllers. 2022. Available online: https://www.arubanetworks.com/assets/ds/DS_7000Series.pdf (accessed on 20 July 2022).
17. Google LLC. Flutter. Available online: https://flutter.dev/ (accessed on 17 August 2022).
18. Iotbymukund. How to Calculate Distance from the RSSI value of the BLE Beacon. 2016. Available online: https://iotandelectronics.wordpress.com/2016/10/07/howto-calculate-distance-from-the-rssi-value-of-the-ble-beacon/ (accessed on 19 July 2022).

19. Roussel, C.; Ruthmann, S.; Klauer, T.; Czommer, R. Practical Indoor Navigation for Smartphones Based on a Metrological Investigation. *AGIT J. Appl. Geoinform.* **2021**, *7*, 26–35. [CrossRef]

20. Teleszewski, T.; Gładyszewska-Fiedoruk, K. The concentration of carbon dioxide in conference rooms: A simplified model and experimental verification. *Int. J. Environ. Sci. Technol.* **2019**, *16*, 8031–8040. [CrossRef]

21. OGC. OGC SensorThings API. Available online: https://www.ogc.org/standards/sensorthings (accessed on 17 August 2022).