Enabling Ubiquitous Occupancy Detection in Smart Buildings: A WiFi FTM-based Approach

Wenpeng Wang*, Fateme Nikseresht*, Viswajith Govinda Rajan, Jiechao Gao, Bradford Campbell
* *The first two authors contributed equally to this work*
University of Virginia, Charlottesville, USA
{ww2cg, fn5an, gyx4bw, jg5ycn, bradjc}@virginia.edu

***Abstract*—Occupancy information is an essential primitive for a wide range of applications like building energy management, security, and behavior analysis. However, balancing accuracy and cost in occupancy sensing is a long-standing challenge. Traditional sensor-based occupancy sensing either fails in the coverage area or is inaccurate delivery of occupancy information in certain scenarios, e.g. detecting stationary occupants.**

We propose WiFine, a device-free solution for occupancy detection by leveraging WiFi Fine-Time Measurement (FTM) signals, with enough resolution to detect stationary and moving occupants on single-antenna WiFi devices. Compared to existing WiFi-based methods, WiFine demonstrates higher accuracy while using less sampling rate. WiFine can be adopted by any set of two or more WiFi IoT devices and turning them into occupancy sensors, enabling ubiquitous sensing without requiring new hardware. In real-world experiments, WiFine achieves 95.8% accuracy in different room setups and occupancy statuses with up to three participants, and eight hours of data, outperforming CSI-based approaches with higher accuracy and lower data rate.

***Index Terms*—WiFi, Fine Time Measurement (FTM), channel state information (CSI), occupancy detection, wireless sensing**

I. Introduction

Occupancy information is an important primitive for a wide range of applications in smart buildings. Energy management, smart lighting, and security monitoring require accurate and ubiquitous occupancy sensing [1]–[4]. Effective occupancy attribution can help reduce the wasted energy in existing air conditioning systems in buildings which consume up to 40% of total building energy in the US [5]. It also enhances users' comfort, well-being, and quality of life when working with newer Internet of Things (IoT) systems, e.g., turning off appliances when detected nobody is at home, or identifying an intruder when the home is supposed to be unoccupied.

Acquiring accurate occupancy information inside buildings is a challenging task. Radio frequency (RF)-based approaches using WiFi or BLE, however, can leverage existing wireless infrastructure inside buildings with minimum additional installation costs. WiFi-based approaches usually use received signal strength (RSS) or channel state information (CSI) for occupancy sensing [6]–[8]. CSI provides more fine-grained measurements with better resolution over RSS [8], but requires complex hardware such as multiple antennas to offset the noise, and most approaches are built on powerful access points (AP). This creates a ubiquity challenge, most residential homes have dozens of WiFi devices, but only one APs. The hardware cost also differs a lot between commodity WiFi devices ($3 for ESP32) and APs ($90 for Atheros QCA9558). As a result, a research question arises: Can we design an approach with similar level of performance as powerful APs, while also being deployable on distributed devices?

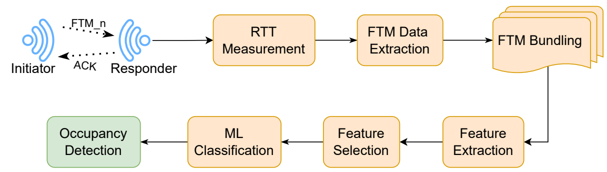


Fig. 1: WiFine utilizes commodity WiFi devices to monitor the occupancy status of an indoor environment. WiFine leverages multi-channel FTM estimates, extracts time-series features to be used by ML classifiers for occupancy predictions.

In this paper, we present WiFine, an occupancy sensing system leveraging the newly released WiFi Fine-Time Measurement (FTM) protocol to detect both stationary and moving occupants. WiFine requires no changes to the standard and can be implemented as a stand-alone, user-level application on simple single-antenna WiFi devices. The widespread adoption of FTM on multiple platforms [9]–[11] reflects the great potential to turn commodity IoT devices into occupancy sensors.

WiFine takes FTM ranging estimates from multiple channels in the 2.4 GHz ISM band and detects occupant presence based on the reflection of the signal from the human body. The presence of the human body creates a multipath effect, resulting in deviations in FTM estimates. Since FTM estimates carry offsets from both the wireless channel and the environment, we collect FTM estimates across 10 WiFi channels to minimize channel effects while preserving the changes due to reflection from the human body. We then process the time-series data to extract environmental change and feed it into a classification model to identify the room occupancy status. An overview of WiFineis shown in Figure 1 We implement a prototype of WiFine on ESP32 chips and conduct real-world experiments, collecting over eight hours of occupancy and FTM data in four rooms with different layouts in an office building, with up to three occupants in both stationary and moving scenarios. We also implement a state-of-art CSI-based approach [12] as a comparison baseline. We trained five machine learning models to evaluate the performance of WiFine.

Results demonstrate that WiFine is able to achieve 95.8% accuracy in occupancy detection regardless of the subject's mobility, establishing a 5.4% improvement as well as 10X less required sampling rate over a similar CSI-based approach. This also showcases the potential of using FTM for wireless sensing on constrained single-antenna devices.

II. Background and Related Work

A. Fine Time Measurement (FTM)

The FTM protocol was standardized in IEEE 802.11-2016 as part of the 802.11mc amendment [13], enabling WiFi devices to perform association-less pairwise ranging with improved 16-bit timestamping. Figure 2 presents the process of FTM estimation, which starts with a device (initiator) scanning for nearby devices; if an FTM-enabled device (responder) is detected, the initiator transmits an initial FTM request. The responder then sends back an acknowledgment (ACK) packet to initiate the FTM process. Then, the responder would exchange a series of packets (FTM, ACK) with the initiator, called *burst*. The Round Trip Time (RTT) of flight is calculated based on the timestamps at which the burst packets are transmitted and received, with a correction for hardware delays and calibration delta. The RTT is calculated as follows:

$$RTT = \frac{1}{n}\sum_{k=1}^{n}((t_4(k) - t_1(k)) - (t_3(k) - t_2(k))) \quad (1)$$

where t_1 and t_2 represent the receive and transmit time of the initiator and the responder, t_3 and t_4 represent the time at which the ACK from the initiator is received by the responder.

B. Related Work

Given the RF signal's ubiquity in smart buildings, a wide variety of research is adopting this technology for occupancy sensing, especially using WiFi protocol.

Traditional wireless-based methods use RSSI measurements to detect occupancy [14]–[16]. CSI-based occupancy detection is becoming prevalent with increasing support from manufacturers. FreeDetector [7] uses the variation in CSI data caused by human presence and employs a greedy subcarrier selection algorithm to select the most representative subset of subcarriers to build the machine learning classifier. WiFree [8] proposed occupancy detection by measuring the shape similarity among adjacent time series CSI curves and proposed a crowd-counting classifier based on transfer kernel learning and information fusion. PeriFi [17] detects people with no movement by analyzing multipath reflections of WiFi signals. Rapid [18] detects human presence using CSI and acoustic information for robust person identification. FreeSense [19] performs a series of operations including principal component analysis, dynamic time wrapping, and discrete wavelet transform to capture the human influence on CSI data. Wi-Cal [12] proposes a crowd counting and localization scheme utilizing single-antenna ESP32 chips, extracting CSI features for both stationary and moving states of the occupants.

Another way of using RF for human sensing is to measure the respiration signal. A particular RF profile can be developed by analyzing the chest motion during breathing [20], [21]. However, they only work in a few meter range and require a dense device deployment in every room to scale.

III. System Design

In this paper, we propose WiFine, a new approach utilizing the FTM protocol for occupancy detection. WiFine captures the variance in FTM estimates caused by human presence over multiple channels and transforms the feature for classification. An overview of our system is shown in Figure 1. FTM data is collected over multiple channels in the 2.4 GHz ISM band instead of a single channel to avoid interference with ongoing wireless traffic. This is challenging because FTM was designed to only work on a single channel, which requires careful design of the communication between devices to switch channels and maintain FTM process. The time-series FTM data are then bundled together and pre-processed to extract and select useful features for the machine learning classifiers. We then apply 5 different classification algorithms to evaluate the FTM data performance in the occupancy sensing use case.

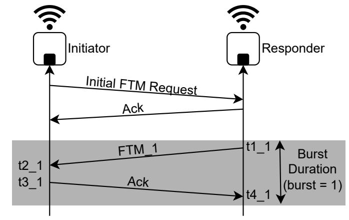


Fig. 2: Overview of WiFi FTM protocol.

A. Association-less Multi-channel FTM

In conventional WiFi protocol, devices have to go through a seven-step association process to establish communication [22]. The 802.11mc FTM protocol runs in an efficient association-less fashion. In order to preserve the efficient trend brought by FTM protocol, we leverage the vendor-specific information element inside the 802.11 action frames to transmit channel-switching commands and useful data between devices. The process begins with a normal FTM process in a particular channel. After the process is finished, the FTM initiator transmits the channel-switching command via a unicast message leveraging the vendor-specific information element and then switches to the new channel. The FTM responder parses this message and then switches to the designated channel. Then the initiator and responder perform another FTM process in the new channel. We design the multi-channel FTM scanning to scan through all available WiFi channels and return to the default channel after finished (channel 1 in this case).

B. Data Collection and Pre-processing

WiFine starts the occupancy detection process by collecting time series FTM measurements over c different channels in sequence, utilizing one initiator-responder pair. Before we process the data and extract useful features, we pre-process the data to combine FTM measurements over a round of multiple channels into an observation. This way, the same index in different observations is pointing at FTM data in the same channel, which ensures that the measurement variance due to different channels is not affecting the processing results.

C. Data Segmentation

Each observation of the processed time series FTM data is denoted as $d_i = [d_{i1}, d_{i2}, ..., d_{ic}]$ where i is the observation index, and c is the total number of channels used to measure the distance between the initiator and the responder.

Once the data is collected, WiFine aggregates and segments the time series FTM measurements into a given time window size. The FTM data, therefore, is D^w for each time window. Assuming the window to be 5, the data bundle is denoted as:

$$\begin{aligned} D^w &= [D_1^w, D_2^w, ..., D_c^w] \\ D^w &= f(d_w, d_{w+1}, .., d_{w+4}) \end{aligned} \tag{2}$$

where w is the index of the time window. These windows are sliding windows with each having four samples overlapping with the next one. Thus, if the processed data has i observations, the aggregated data will have $i-4$ bundles.

D. Feature Extraction

The next step is to extract features from the observation bundles. For that purpose, we performed an exploratory analysis where we measured the variance and standard deviation (std) of the observations over each channel. Figure 3 shows our exploratory analysis results with a sample of normalized FTM measurements in a single session, including both occupied and unoccupied scenarios. The FTM measurements are normalized based on the maximum and minimum FTM values measured during that session. As Figure 3 depicts, variance and std are very low when the room is unoccupied, and they are much higher when there is an occupant in the room. From these observations, we decided to extract three main summary statistic features of variance, std, and root means square (rms) from the bundles based on the monotonic behavior of measurements in occupied and unoccupied observations.

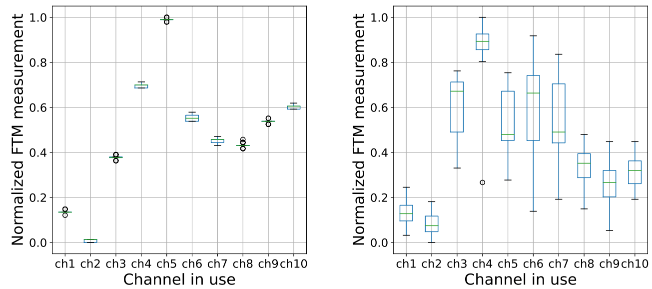
Figure 3 shows the effect of an occupant's presence in the room on the measurements' variance. The variance vector extracted from bundles ($variance^w$) can be denoted as $variance^w = [\sigma^2(D_1^w), \sigma^2(D_2^w), ..., \sigma^2(D_c^w)]$, where $\sigma^2(x)$ represents function of variance of any vector x. With the same justification, we extract the standard deviation vector calculated from the bundles (std^w), which are denoted as $std^w = [\sigma(D_1^w), \sigma(D_2^w), ..., \sigma(D_c^w)]$, where $\sigma(x)$ denotes the standard deviation function of any vector x. Similarly, we extract the root mean square vector (rms^w):

$$\begin{aligned} \gamma(x) &= \sqrt{\frac{\Sigma(x_i)^2}{len(x)}} \\ rms^w &= [\gamma(D_1^w), \gamma(D_2^w), ..., \gamma(D_c^w)] \end{aligned} \tag{3}$$

where $\gamma(x)$ denotes the rms function calculated over any x vector, and x represents the bundles.

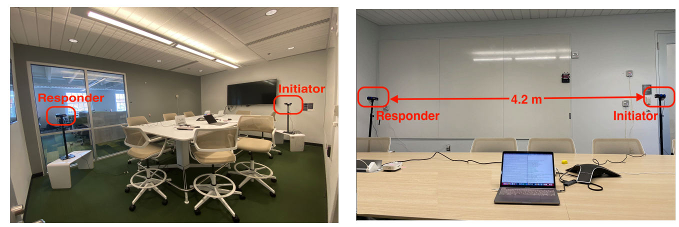
E. Classification

The extracted features then are concatenated and form our training and testing dataset. To build the dataset, each feature vector we explained in Section III-D is concatenated vertically along the time windows (w) order and horizontally along the channel numbers order (c). As a result, we extract $3 * 10$ features in total from each dataset. After the datasets are created, we select the five most popular ML classification algorithms for binary classification tasks to train models and evaluate their performance. Those algorithms are random forest (RForest) , k-nearest neighbors (KNN) , XGBoost (XGB) ,



(a) Unoccupied room (b) Occupied room

Fig. 3: An example of distribution in FTM data over ten channels. The standard deviation and variation are minimal for an unoccupied room and significantly increased when occupied. Human presence significantly affects FTM estimates which allows us to use FTM for human detection.



(a) (b)

Fig. 4: (a) An example of system deployment in a conference room. (b) The comparison experiment in another room.

logistic regression (LR) , and support vector machine (SVM). To have a more optimal training process, we select the most contributing features in the above-mentioned five machine learning models using the meta-transformer *SelectFromModel* in *Scikit-Learn* [23] as the feature selection function setting the threshold to 0.05.

IV. Implementation

We implement a prototype of WiFine using a pair of commodity ESP32S2 boards on 2.4 GHz band with a single PCB antenna. The association-less channel-switching mechanism is implemented using the ESPNOW protocol, described in [24]. The FTM data is collected from channels 1 to 10, CPU frequency is set to 240 MHz, and the tick rate for FreeRTOS is set to 1000 Hz to ensure accurate measurements. All collected data is processed locally on a laptop with Ubuntu 20.04.

V. Evaluation

A. Experimental Setup and Data Collection

In our main experiment, we collect FTM data in four rooms of different sizes ($255cm \times 410cm$, $365cm \times 420cm$, $365cm \times 490cm$, and $365cm \times 530cm$) and layouts in an office building. In each room, we place two ESP32S2 boards in two opposite corners of the room to ensure we cover as much area as possible in the room. Figure 4a shows an example setup in one of the conference rooms ($365cm \times 490cm$). For the main experiment, we collect data in 27 sessions with zero to three occupants sitting at different locations in the room to approximate normal scenarios, and each session

Layout	Room1				Room2	Room3		Room4	
Sessions	2	7	4	2	4	1	2	2	3
Occupants	0	1	2	3	1	0	1	1	2

TABLE I: The attributes of 27 sessions in different layouts with various numbers of occupants.

	Accuracy	F1-score	Precision	Recall
RForest	98.06	97.83	98.54	97.12
KNN	93.87	92.72	**99.18**	87.05
XGB	97.10	96.70	98.51	94.96
LR	98.06	97.83	98.54	97.12
SVM	**98.39**	**98.19**	98.55	**97.84**

TABLE II: WiFine performance comparison over ML models where five classifiers were trained with extracted features to predict the occupancy of the rooms.

consists of 20 minutes of measurements. To minimize energy consumption while maintaining reasonable resolution, we collect one sample every 20 seconds, meaning the distance between consecutive channels' samples is two seconds. For this purpose, we set up the burst period to 200 ms, the number of bursts to 8 bursts, and the number of frames to 32 frames. We chose 130 cm for the device's height from the ground, based on the height of the furniture in the room. We tested the raw FTM measurements by deploying the devices at the height of 90 cm and 130 cm above ground and realized that 130 cm gives us more accurate measurements when the room is empty. Therefore, we decided to set our devices to that height in all of our experiments. In addition, the high resolution of FTM data allows us to use fewer device links to capture the change in the wireless environment.

The division of the 27 sessions is given in Table I, where for each session, depending on the number of occupants (n_o), we allocate $20/(n_o+1)$ minutes to each occupancy state. This means for $20/(n_o+1)$ minutes, the room is unoccupied. Then the first occupant enters the room for a $20/(n_o+1)$ minutes, and this continues until the last $20/(n_o+1)$ minutes when all the occupants are in the room.

B. WiFine's Performance

As the data is time series, we treat each data session as a separate dataset. We train the classifiers using 21 of the datasets, which are randomly selected and then evaluate the models with the rest of the datasets. As we explained in Section III, we leverage five different ML classification algorithms for our binary classification task, which are RForest, KNN, XGB, LR, and SVM. As we can see in Table II, the SVM model performs best with 98.4% accuracy and F1-score among other algorithms in this case study. Figure 6a also displays that the SVM model confusion matrix is representative enough when only 5 samples are misclassified among 310 test samples. After SVM, RForest has an accuracy of 98.1%, and as Figure 6b shows, only 6 samples are misclassified here.

C. Comparison with CSI-based Approach

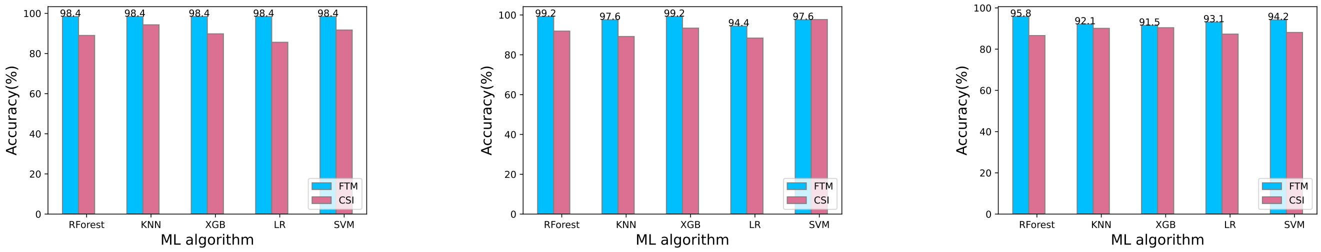
To evaluate the performance of our proposed system for both moving and stationary occupants and to compare it with similar works in this area, we replicate a recent CSI-based occupancy sensing study, Wi-Cal [12]. Wi-Cal proposes a crowd counting and localization scheme utilizing ESP32 chips [11]. Compared to Wi-Cal, which just targets moving occupants, our system targets detecting both stationary and moving occupants. Wi-Cal is also built upon the single antenna ESP32S2 platforms, which makes its replication more viable. Therefore, we ran an evaluation experiment to compare WiFine with Wi-Cal in an identical controlled environmental setup to test its occupancy detection performance in both moving and stationary occupants' presence scenarios.

The environmental setup for this evaluation experiment is shown in Figure 4b where the distance between the devices is 4.2 m, and the devices' height from the ground is 130 cm, which is similar to our main study. The only difference between this setup and the main experiment setup is that we deploy the devices on one side of a big conference room with 20 occupants capacity where there is no furniture to be able to collect data in both moving and stationary scenarios.

For this evaluation, we collect two hours of CSI and FTM data. Each of the CSI and FTM datasets includes balanced data of the unoccupied room, occupied room with one and two moving occupants, and one and two stationary occupants.

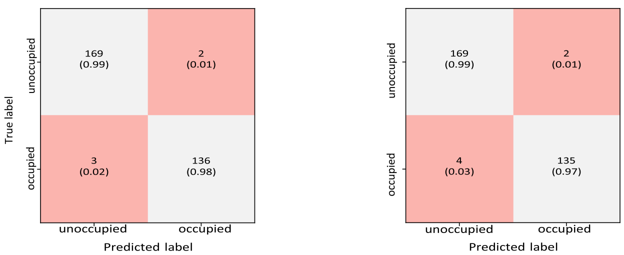
For the CSI data processing, we follow the processing flow of Wi-Cal, which first calculates the amplitude values across all the subcarriers and then smooths the amplitude signal by applying the Hample filter to eliminate the spike noises. Then it applies the Savitzky-Golay filter to remove the overall white noise of the signal. After smoothing the signal, Wi-Cal bundles every 6 secs of data with a 3 secs overlap between the two bundles in a sequence. After that, it extracts the dynamic and static features from the bundles. However, the only difference here with Wi-Cal is in the ML classifiers we use because we try to maintain consistency between our framework and the evaluation experiment. Thus, the RForest and XGB algorithms are two of the four classifiers that are also used in Wi-Cal. We also should mention that as Wi-Cal used only 13 subcarriers' data in its system, we also follow that path and select subcarriers with an identical distance on both sides from subcarrier 1 to 52 (1, 5, 9,..., 49). All the processing and feature extraction steps for the collected FTM data are exactly the same as what we explained in Section III.

Figure 5, compares WiFine and Wi-Cal performance where the models are trained only based on moving occupants in Figure 5a, trained only based on stationary occupants in Figure 5b, and trained based on all the data in Figure 5c without considering the occupants' status in the room. Our main goal for building two different models of moving and stationary cases was to investigate the pros and cons of WiFine and Wi-Cal compared to each other. As we can see, WiFine outperforms Wi-Cal in the moving case study by 4.1%, where the best model accuracy is 94.3% in KNN for Wi-Cal and 98.4% for WiFine. In addition, WiFine outperforms Wi-Cal in a stationery case study by 5.8% where the Wi-cal accuracy is 93.4% based on XGB, and WiFine accuracy is 99.2%. However, real-world scenarios will not resemble controlled environments. Therefore, we also build the models without considering the occupants' status to finalize our comparison,



(a) Moving Occupants (b) Stationary Occupants (c) Both

Fig. 5: WiFine's occupancy detection accuracy compared to a CSI-based approach [12] across moving, stationary, and the combination of those scenarios. We see 4.1% improvement in the moving case, 5.8% improvement in the stationary case, and 5.4% improvement overall, achieving 95.8% accuracy without considering the occupants' status for WiFine over the CSI-based approach.



(a) SVM (b) RForest

Fig. 6: (a) and (b) shows the confusion matrix for SVM and RForest. Both models achieve high accuracy with only 5 and 6 wrong classifications among 310 samples, respectively.

and as Figure 5c shows, WiFine outperforms Wi-Cal here as well, achieving a 95.8% of accuracy in overall cases, which is a 5.4% improvement over Wi-Cal.

VI. Conclusion

We present WiFine, a device-free occupancy detection system using commodity simple WiFi devices, leveraging WiFi FTM protocol over multiple channels on the 2.4 GHz ISM band. The real-world experiments demonstrate the feasibility of using FTM for occupancy detection with both stationary and moving occupants. We also compare our work with a similar CSI-based approach, and WiFine achieves a higher accuracy using 10X lower data rate. This study highlights the great potential for FTM to be used in more wireless sensing tasks. We hope the related research community embraces this new observation and works towards leveraging existing infrastructure for a more advanced IoT future.

VII. Acknowledgement

We thank the anonymous reviewers for their valuable insights and helpful feedback on improving this paper. We also thank Commonwealth Cyber Initiative (CCI) and NobleReach Foundation for their suggestions.

References

[1] J. Yang, M. Santamouris, and S. E. Lee, "Review of occupancy sensing systems and occupancy modeling methodologies for the application in institutional buildings," *Energy and Buildings*, 2016.

[2] W. Wang, J. Su, Z. Hicks, and B. Campbell, "The standby energy of smart devices: Problems, progress, & potential," in *IoTDI*. IEEE, 2020.

[3] H. Saha and A. R. Florita, "Occupancy sensing in buildings: A review of data analytics approaches," *Energy and Buildings*, 2019.

[4] T. Wu, M. Aldeer, T. Chowdhury, A. Haynes, F. Nikseresht, M. P. Varnosfaderani, J. Gao, A. Heydarian, B. Campbell, and J. Ortiz, "The smart building privacy challenge," in *BuildSys*, 2021, pp. 238–239.

[5] "U.s. department of energy," https://www.energy.gov/, 2019.

[6] L. Banin, O. Bar-Shalom, N. Dvorecki, and Y. Amizur, "Scalable wi-fi client self-positioning using cooperative ftm-sensors," *IEEE Transactions on Instrumentation and Measurement*, 2018.

[7] H. Zou, Y. Zhou, and J. Yang, "Freedetector: Device-free occupancy detection with commodity wifi," in *SECON Workshops*. IEEE, 2017.

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

[9] "Google wifi," https://store.google.com/us/product/google_wifi.

[10] "Android," https://developer.android.com/guide/topics/connectivity/.

[11] Espressif, "ESP32," https://www.espressif.com/.

[12] H. Choi, M. Fujimoto, T. Matsui, S. Misaki, and K. Yasumoto, "Wi-cal: Wifi sensing and machine learning based device-free crowd counting and localization," *IEEE Access*, vol. 10, pp. 24 395–24 410, 2022.

[13] IEEE, "Draft standard for information technology—telecommunications and information exchange between systems local and metropolitan area networks—specific requirements," 2019.

[14] M. J. Akhlaghinia, A. Lotfi, and N. Sherkat, "Occupancy monitoring in intelligent environment through integrated wireless localizing aggents," in *2009 IEEE Symposium on Intelligent Agents*, 2009.

[15] G. Deak, K. Curran, J. Condell, and D. Deak, "Detection of multi-occupancy using device-free passive localisation," *IET Wireless Sensor Systems*, vol. 4, no. 3, pp. 130–137, 2014.

[16] M. F. R. M. Billah, N. Saoda, and B. Gao, Jiechao Campbell, "Ble can see: a reinforcement learning approach for rf-based indoor occupancy detection," in *IPSN*, 2021.

[17] E. Soltanaghaei, A. Kalyanaraman, and K. Whitehouse, "Peripheral wifi vision: Exploiting multipath reflections for more sensitive human sensing," in *Proceedings of the 4th International on Workshop on Physical Analytics*, 2017, pp. 13–18.

[18] Y. Chen, W. Dong, Y. Gao, X. Liu, and T. Gu, "Rapid: A multimodal and device-free approach using noise estimation for robust person identification," *Ubicomp*, 2017.

[19] T. Xin, B. Guo, Z. Wang, P. Wang, J. C. K. Lam, V. Li, and Z. Yu, "Freesense: a robust approach for indoor human detection using wi-fi signals," *Ubicomp*, 2018.

[20] C. Wu, Z. Yang, Z. Zhou, X. Liu, Y. Liu, and J. Cao, "Non-invasive detection of moving and stationary human with wifi," *IEEE Journal on Selected Areas in Communications*, 2015.

[21] N. Patwari, J. Wilson, S. Ananthanarayanan, S. K. Kasera, and D. R. Westenskow, "Monitoring breathing via signal strength in wireless networks," *IEEE Transactions on Mobile Computing*, 2013.

[22] IEEE, "802.11n Standard," http://standards.ieee.org/findstds/standard/802.11n-2009.html, 2009.

[23] "scikit-learn," https://github.com/scikit-learn/scikit-learn, Accessed in 2023.

[24] W. Wang, Z. Liu, J. Gao, N. Saoda, and B. Campbell, "Ubitrack: enabling scalable & low-cost device localization with onboard wifi," in *Buildsys*, 2021, pp. 11–20.