Robust and Practical WiFi Human Sensing Using On-device Learning with a Domain Adaptive Model

Elahe Soltanaghaei
Carnegie Mellon University
esoltana@andrew.cmu.edu

Rahul Anand Sharma
Carnegie Mellon University
rahulans@andrew.cmu.edu

Zehao Wang
Carnegie Mellon University
zehaow@andrew.cmu.edu

Adarsh Chittilappilly
Carnegie Mellon University
achittil@andrew.cmu.edu

Anh Luong
Sandia National Laboratories
anh.luong.n@gmail.com

Eric Giler
Endeveo, Inc
eric.giler@endeveo.com

Katie Hall
Endeveo, Inc
katie.hall@endeveo.com

Steve Elias
Endeveo, Inc
eli@endeveo.com

Anthony Rowe
Carnegie Mellon University
agr@ece.cmu.edu

Abstract

The ubiquity of WiFi devices combined with the ability to cover large areas, pass through walls, and detect subtle motions makes WiFi signals an ideal medium for sensing occupancy. While extremely promising, existing WiFi sensing solutions have not been rigorously tested outside of lab environments and don't often consider real-world constraints associated with non-expert installers, cost-effective platforms and long-term changes in the environment. This paper presents M-WiFi, a user-in-the-loop self-tuning framework for WiFi-based human presence detection with on-device learning and domain adaption capabilities that operates entirely on an embedded platform. M-WiFi robustly detects human presence by separating human-specific disturbances on WiFi signals from those of static objects, moving furniture or even pets. The high-level features of human presence are captured in an initial generalized classification model which adapts over time to a new building by selectively asking users to annotate a small number of critical time periods. We evaluate M-WiFi in 7 different houses, for a total of 100 days, with a mixture of pets and including periods of sleep and stationary activities. We show that our domain adaptive model can detect the human presence with an average accuracy of 90% in a completely new house after only 3 days of self-tuning and rapidly reaches a steady-state performance of 98% in long-term operations.

CCS Concepts

• **Human-centered computing → Ubiquitous and mobile computing design and evaluation methods**.

Keywords

human sensing, CSI, WiFi, multipath propagation

ACM Reference Format:
Elahe Soltanaghaei, Rahul Anand Sharma, Zehao Wang, Adarsh Chittilappilly, Anh Luong, Eric Giler, Katie Hall, Steve Elias, and Anthony Rowe. 2020. Robust and Practical WiFi Human Sensing Using On-device Learning with a Domain Adaptive Model. In *The 7th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation (BuildSys '20), November 18–20, 2020, Virtual Event, Japan.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3408308.3427983

BuildSys '20, November 18–20, 2020, Virtual Event, Japan

ACM ISBN 978-1-4503-8061-4/20/11…$15.00
https://doi.org/10.1145/3408308.3427983

1 Introduction

Accurate, reliable, cost-effective, and easy-to-deploy human presence detection has long remained a missing component for home automation, heating, and cooling optimization, or elderly monitoring. For example, multiple studies indicate that improved occupancy sensor systems could be utilized to realize 30% energy savings in both residential and commercial buildings [21]. However, existing occupancy sensors such as CO2, passive infrared (PIR), ultrasonic, image, or sound sensors, suffer from multiple problems including high deployment and maintenance costs, complicated user interfaces, privacy concerns, or low accuracy that result in user frustration and uncomfortable living conditions. To address these challenges, significant progress has been made in device-free human sensing that utilizes the information collected from wireless signals without requiring occupants to wear or carry devices. These approaches characterize the disturbances caused by the human body on wireless signals and then use variations in signal measurements to detect the presence of people across wide areas or even through walls. Given the pervasive nature of WiFi in most buildings, these approaches have the potential to be extremely cost effective in terms of accuracy given a relatively low sensor density. Unfortunately, outside of lab environments, these approaches still fall short in practice for two main reasons:

- **Robustness and Generalization.** WiFi signal propagation is sensitive to many different factors such as the placement of objects and furniture in the environment, the direction and distance between wireless nodes and people within the space, or external factors like reflections from moving objects outside of the desired coverage area. For these reasons, it is challenging to design a robust and generalizable WiFi sensing system that can automatically adapt to a new building or previously unseen context. The most robust current systems rely on user motion in the Line-of-Sight (LoS) of wireless devices, which limits the range and fails to detect people without significant motion such as during periods of sleeping or sitting.
- **Realtime Operation on Embedded Platforms.** The current wireless sensing solutions are seldom tested in realistic setups, overlooking a number of important deployment challenges. First, the techniques proposed thus far require server-class computers and an abundance of labeled training data. Even split architectures with a cloud processing component are largely impractical due

to the high volume of wireless data that would need to be continuously streamed for real-time processing. Ideally, the entire pipeline of pre-processing, sensing and learning should happen at the edge. Second, most of the target applications for this type of sensing technology tend to be price sensitive further limiting processing, memory, and storage to embedded application-class targets.

To address these challenges, we present M-WiFi, a robust and embedded WiFi sensing system with on-device learning and domain adaption capabilities. M-WiFi uses a set of low-cost plug-in embedded modules that contain WiFi radios with multiple antennas that periodically communicate and monitor the Channel State Information (CSI). To detect the presence of people, M-WiFi first characterizes the multipath environment of a building, which includes static wireless reflections. Then, it monitors these wireless paths and their changes due to human presence. The key innovation of M-WiFi is its capability in separating human specific signatures from environment-dependent features. M-WiFi does this by initially using a generalized (baseline) classification model that captures the high-level wireless disturbances caused by the human body learned from a large multi-house dataset. This model is then adapted to a new house by learning its multipath profile using a small amount of training data from the users. We present a user-in-the-loop self-tuning framework that combines transfer learning with domain adaptive models optimized to both retrain and execute on an embedded target.

Compared to existing approaches that tend to operate holistically on the aggregate channel response, M-WiFi enhances the spatial coverage and sensing sensitivity by disentangling multipath signals and using each unique path as an independent sensor. The intuition is that the amount of fading and variation that each wireless reflection suffers from the human presence will be different, resulting in independently fading channels. This is achieved using a *multipath profile* of the physical space and combining signal measurements across time, space (multiple antennas at the WiFi transceivers), and frequency (e.g. OFDM subcarriers). This additional information increases the chance of capturing fluctuations indicative of human presence even in dynamic environments with moving furniture and pets.

After the system has been deployed in a new building, M-WiFi asks the user to label a small amount of data for domain adaptation. One of the major challenges with training a machine learning model at the edge is being able to select the ideal regions of data that provide the most information while minimizing potentially frustrating user touch-points. In addition, unlike the data used to train the baseline model, we have to assume that user input is error prone. To strike a balance between user convenience and classification performance, M-WiFi uses an unsupervised segmentation algorithm to detect the potential occupancy changes. These selected time periods are sent to the user for verification in the first few days of deployment, which are then used to adapt the model to the new environment. In our current system, this is a text report generated at the end of each day, but it could easily be collected with a more user-friendly mobile app or dashboard. It is worth noting that unlike fingerprint-based approaches, our method does not require scenario-specific tuning for every location or activity. Instead, we tune a pre-trained model with partially annotated data from the daily activities of occupants.

To realize a real-time and economical implementation, we design M-WiFi such that it performs the entire multipath profile extraction, tuning, and testing at the edge on an embedded application class platform. We use a wrapper model that evaluates the model performance with a certain subset of features until reaching an optimal feature vector. Since our technology solution uses proprietary software algorithms that interface to off-the-shelf embedded devices, it can be seamlessly upgraded to WiFi and IoT devices that are already deployed in residences. We implemented M-WiFi using TPLink N600 OpenWRT platforms [1] that collects Channel State Information using an Atheros WiFi chipset [24] attached to 2 antennas. The system collects data, extracts features, classifies occupancy and retrains entirely locally on the embedded platform. We conduct extensive experiments in 7 different houses for a combined 100 days of data and total of 25 different experiments with a mixture of pets and during periods of activity and sleep. We also evaluate the sensitivity of the proposed system to furniture movements, or the presence of moving objects and pets. In summary, our adaptive model achieves an average accuracy of 90% in a new house after 3 days of model tuning. Even with accidentally miss-labeled information, we see that the system is still able to improve accuracy and eventually converge on our expected level of performance albeit with a penalty in terms of training time. We show that as M-WiFi continues domain adaptation, the models rapidly adjust to the new house and approximate the steady-state performance of 98% in long-term operations.
The main contributions of this paper include:

- A self-tuning and self-calibration framework to adapt a generalized model to a new environment on embedded edge computing platforms by minimizing user-involvement in the domain adaptation process.
- Robust human presence detection by directly sensing the human body through wireless multipath signals. M-WiFi can detect stationary occupancies such as long periods of sleeping, and can differentiate human from pets or other moving objects.
- Extensive evaluation experiments across multiple houses while the occupants have their daily routine over the course of 100 days. The annotated datasets will be released publicly for the use of the research community.

2 Background and Related Work

2.1 Human Presence Detection

Among the traditional technologies for human presence sensing, PIR [2, 19, 20], CO2 [27] and cameras [26] have been identified as commercial occupancy sensors. However, none of these sensors accurately measure occupancy. They suffer from high deployment and maintenance costs, are limited to LoS scenarios and easily triggered by pets or non-human motions. In addition, cameras raise privacy concerns specially for residential buildings. Even more advanced techniques such as wide-band ultrasonic methods [16] or Radar-based techniques [23] still struggle to cover residential spaces with many smaller sized rooms. In addition, they are often using specialized bulky hardware that are economically impractical as presence sensor. This paper targets the limitations of previous work by leveraging the pervasive WiFi infrastructure and the wireless multipath reflections.

Another research line focuses on sensor fusion such as combination of electrical energy demand, water usage, and number of wireless devices connected to the WiFi network [4]. However, these techniques are not always reliable specially for the elderly as they

may not wear or carry devices all the time, or large families with multiple wireless devices that may be left behind. However, we believe that these techniques are excellent complimentary technologies to M-WiFi and can be opportunistically used for training upon availability to perhaps further minimize or reduce user data labeling.

2.2 Sensing using WiFi CSI.

Channel State Information (CSI) measured from MIMO-OFDM WiFi packets has been widely used for different sensing applications. CSI captures the key information about how wireless signals propagate between transceivers, subject to small-scale multipath fading. This makes it extremely sensitive to changes in the environment including human motion and activities. We can categorize this previous work into two main approaches: threshold-based, and learning-based sensing. Threshold-based methods [28] reduce the measurement vector to a single metric that can detect human motion. These metrics are either defined based on temporal signal variations or signal correlation in the frequency domain[14]. However, these solutions are limited to human motion and cannot detect human presence specially in during periods of stationary activities.

With the advancements in supervised and deep learning techniques, more recent works focus on learning-based approaches [25]. These systems apply a data-driven mechanism by defining intrinsic features of the CSI in the presence of human and use a large CSI database for training from the empty room and human motion. Unfortunately, the CSI is inherently dependent to the physical space and the static multipath propagations, so it is challenging to obtain stable features that are immune to environment dynamics [6]. Additionally, all of these techniques are tested in lab environments [12, 29] using powerful server-class computers, which are far from cost-effective for most real-world systems. These limitations motivated us to develop a user-in-the-loop self-tuning framework that takes the CSI dependency to the physical space and limited hardware resources into account.

2.3 Sensing Domain Adaption

In the machine learning community, domain adaptation is a well-known solution for reducing the difference between source and target feature distributions, thus improving generalization performance. In the context of WiFi sensing, domain adaption provides consistent high accuracy and robustness to environmental dynamics. Recent WiFi-based sensing systems leverage domain adaption for environment independent gesture recognition [30], activity detection [7, 31], or people counting [3]. However, existing methods usually assume cloud access for retraining and adapting the models and would struggle to operate on resource constrained platforms. Occupancy sensors, specially for HVAC control, must be low-cost, user transparent, and extremely accurate to encourage user adoption, especially in residential applications where consumers may reject automated technology solutions that result in uncomfortable living conditions. In this paper, we mainly focus on minimizing user involvement in domain adaption while providing real-time and on-device computation. We borrow techniques from computer vision and object detection [5] to enable WiFi domain adaption in edge computing environments by only adapting shallow model knowledge to the new environment. Nevertheless, our proposed framework is compatible with different transfer learning algorithms and can be extended to other techniques.

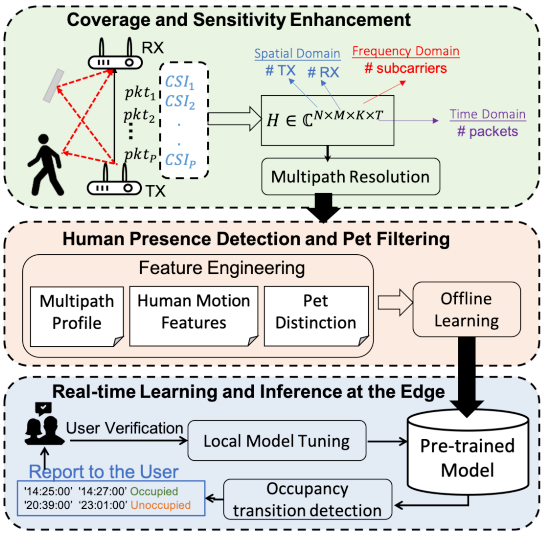


Figure 1: System overview.

3 System Overview

M-WiFi consists of three main components:

(1) **Features for increased coverage and sensitivity.** One common limitation for existing RF-based sensing techniques is coverage: the size of the area that can be covered with a given number of nodes. The main challenge is that the properties of the received signal are dominated by objects in the Fresnel zone of the line of sight path, resulting in a linear sensing region despite the omni-directional nature of the antennas. Our proposed solution will enhance the sensing coverage by taking advantage of multipath reflections that are common in indoor environments. We define a pseudo super-resolution algorithm that captures the signature of wireless multipath profile for a new space.

(2) **Human presence detection and pet filtering.** By combing multipath profile and signal measurements in three dimensions of time, space, and frequency, M-WiFi is capable of identifying the signature of a human body in the room whether moving or stationary, both in LoS and non-LoS. In addition, we differentiate between pets and humans by capturing differences in body types and motion patterns in our training data set. We empirically combined handcrafted features that create unique signatures for humans and pets.

(3) **Learning at the edge.** In addition to the human body, geometry of the physical space is a key factor in defining wireless multipath propagation. Therefore, to generalize a human sensing system across different physical environments, we at least need to capture the multipath profile of the new environment to calibrate the model. However, this necessitates (1) learning at the edge due to large volume of data (2) annotated data from the new space. To address these requirements, we exploit transfer learning techniques that allows us to adapt a general model to a new domain, in this case a specific house.

In the next sections, we elaborate on each of these components and explain how M-WiFi addresses the required robustness, generalization, and real-time operation of a practical human sensing system.

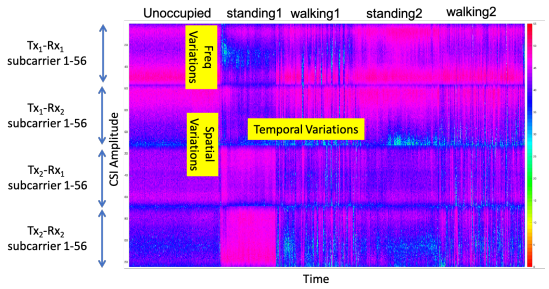


Figure 2: CSI variations under different occupancy scenarios

4 Enhancing Coverage and Sensitivity

The goal of M-WiFi is to use the wireless mesh created by multiple WiFi nodes to directly sense the human body in the home. The key innovation of M-WiFi is the ability to resolve and analyze wireless multipath signals between each transmitter and receiver, increasing both the spatial coverage and sensing sensitivity. Complex indoor environments cause wireless signals to propagate along multiple paths, reflecting off of walls, furniture, and people, as shown in Figure 1. Each of these paths reveals information about a different part of the physical environment. M-WiFi resolves these multipath signals and analyzes them individually, using each path as an independent sensor.

The basic approach is to use the Channel State Information (CSI) provided by modern commercial WiFi cards to deduce key features for the dominant multipaths such as their power, Angle-of-Arrival (AoA), and Time of Flight (ToF). In the following sections, we first explain the required pre-processing on CSI amplitude and phase and then elaborate on how to extract the wireless multipath profile. Next, we define the feature set which allows us to provide more accurate estimation of the stationary occupancy and also differentiate humans from pets.

4.1 Pre-processing

The CSI provided by commercial WiFi cards includes the introduced signal attenuation and phase shift due to the wireless channel between a WiFi transmitter and receiver. The CSI collected in our WiFi devices includes 56 frequency sub-carriers for 2 transmitting and 2 receiving antennas, which includes the received signal from all paths.

Figure 2 shows an example of the CSI amplitude variations over time (across the x axis) and over multiple frequency and antennas (across the y axis). The occupancy state of the house changes over time as labeled in the figure. As we can see, the CSI amplitude can easily capture the large human movements due to significant temporal variations of multiple paths. However, when the home is unoccupied or occupied by a stationary occupant (e.g. sleeping), there is nearly no temporal variations. The presence of people is only detectable if they are in the Fresnel zone of the WiFi link, in which the human body creates either a new wireless reflections or it blocks some of the existing reflections. For example, the first standing scenario in figure 2 happens in the LoS of the WiFi link, thus creating a new pattern compared to empty room, in frequency and spatial domain.

To address this problem, we leverage CSI phase information because it is dramatically more sensitive to small movements. However phase can not be used on its own since it is prone to arbitrary errors caused by Packet Detection Delay (PDD) and Sampling Frequency Offset (SFO). To address this issue, we calculate the relative phase between antennas instead of using the absolute values. The intuition is that the radio front-end in WiFi nodes is shared between phased antenna arrays, so the offsets are constant across the antennas. By calculating the relative phase, we still have the same level of information but free of arbitrary offsets. Now the CSI relative phase and amplitude can be used to calculate the spatial multipath components.

4.2 Extracting Wireless Multipath Profile

M-WiFi's approach to resolve multipath profile builds on well-established noise-subspace methods [15] and joint-estimation techniques [8, 18] to fuse data from multiple WiFi antennas and subcarriers. The basic idea is to resolve multipath components using phase shifts across the antenna arrays, which is a function of AoA (θ), or the expected phase shift across subcarriers due to ToF (τ). So, for a measurement matrix X across the antennas and subcarriers,

$$X(t) = [x_{1,1}(t),..x_{1,K},x_{2,1},...,x_{M,K}(t)]^T = a(\theta,\tau)s(t)+N(t) \quad (1)$$

the expected phase shift is denoted by $a(\theta)$

$$a(\theta,\tau) = [1..\Omega^{K-1}(\tau),\Phi(\theta),...,\Omega^{K-1}(\tau)\Phi(\theta),..., \Phi^{M-1}(\theta),...,\Omega^{K-1}\Phi^{M-1}(\theta)] \quad (2)$$

where M is the number of antennas, K is the number of frequency subcarriers, $s(t)$ is the received signal vector at the first antenna and $N(t)$ is the noise vector. To extract the AoA and ToF of multipath signals, the conventional approach is to perform eigenvalue decomposition on the covariance matrix XX^H to extract the noise and signal subspace and then find the steering vectors that are orthogonal to the noise subspace. However, this method is computationally heavy as it involves large matrix calculations and exhaustive search over the entire solution space. On the other hand, for a practical and real-time human sensing system, the entire inference and learning process has to operate at the edge given the volume of generated CSI data. So, it should be compatible with limited resources and hardware constraints.

To address the trade-off between high fidelity features and computational overhead, we define a pseudo super-resolution algorithm that captures the essence of multipath profile without going through the entire exhaustive search process. The basic idea is to define intermediate features that capture the signatures of multipath signals, but it does not require to be the final AoA, ToF, or Doppler estimates of each path. We define algorithm 1, which calculates the covariance of the measurement matrix and then reduces the feature space to the top eigenvalues of the covariance matrix on three dimensions of frequency (for ToF estimates), space (for AoA and AoD estimates), and time (for Doppler estimates). The number of selected eigenvalues correspond to the number of dominant paths in the environment. To further reduce the computational cost of this algorithm, we use the Implicitly Restarted Arnoldi method [9], which is best suited for sparse matrices. It iteratively calculates the top eigenvalues without requiring large memory or computationally heavy vectorization.

5 Human Detection and Pet Filtering

A recently proposed technique [10, 17, 25] has been demonstrated to identify multipath reflections from a moving person based on its phase incoherence with other signals. However, this technique can only detect large movements such as walking. Other recent studies

Algorithm 1 Pseudo super-resolution algorithm

Input measurement matrix X(N,M,K,T)
Output Feature set F

1: **for** i=1:2 **do** ▷ transmitting antennas
2: Flatten the first three dimensions of $X_{temp}[(M\times K)\times T]$, $X_{temp} = flat(X(i,,,),2)$
3: $corr_{sf} = X_{temp}X_{temp}^{H}$ ▷ Corr across freq and space
4: $corr_t = X_{temp}^{T}X_{temp}^{TH}$ ▷ Corr across time
5: Add top P eigenvalues of Corr to feature set
6: $F \leftarrow eig(corr_{sf}, P)$
7: $F \leftarrow eig(corr_t, P)$
8: return feature set F

have shown that WiFi signals can be used to detect human respiration [13] and even heartbeat [11] by filtering for disturbances at specific frequencies. However, these approaches are only effective if the people are completely still and in the LoS path. M-WiFi combines these techniques and extracts three main groups of features to also detect intermediate levels of activity (sitting or standing) that are not detected by either of the other two: (1) Doppler shift caused by a moving (e.g. walking) person, (2) Attenuation and reflections caused by a stationary (e.g. sitting or standing) person, (3) Frequency-specific disturbances caused by a completely still (e.g sleeping) person.

In addition, to differentiate pets from humans, M-WiFi trains on differences derived from body type such as size and breathing rate. Cats and most dogs will create only a small fraction of the RF disturbance that the human body creates, due to being approximately 1/5th the height and body mass, or less. However, a moving animal will have a similar effect as a stationary person. To differentiate these two, we develop new fusion techniques to measure features over both frequency and space, and are scanned over time. For example, a moving animal will create low signal disturbance but high Doppler values and will affect a changing set of paths, while a stationary person will create low signal disturbance with low Doppler values, affecting only a fixed set of paths. Still pets will be differentiated from humans based on respiration rate. The next section summarizes the full set of handcrafted features.

5.1 Feature Set

In summary, M-WiFi defines four categories of features resulting in a total of 94 features:

- **Multipath Profile:** eigenvalues of the CSI covariance matrix over receiving antennas, transmitting antennas, and subcarriers.
- **Temporal Features:** eigenvalues of the covariance matrix for successive CSI measurements over time.
- **Frequency-specific Features:** entropy of CSI amplitude and relative phase across subcarriers of different antennas.
- **Minor Channel Variation:** attenuation and reflections caused by a moving or stationary person. This is done through channel variation factor defined as:

$$v = \sqrt{\frac{var(X)}{\frac{1}{T}\sum_{t=1}^{T}|x_t|^2}} \tag{3}$$

where X is the CSI vector for T packets, and $var(X)$ is the variance of vector X. The denominator represents the RMS value of the vector X. We calculate the channel variation factor for every subcarrier of every antenna and then apply the aggregate functions (mean, median, max, min, std) across subcarriers. The resulted features carry the channel variations in both frequency and time dimensions and are calculated for every antenna pair.

5.2 Model Training

To study the generalizability of M-WiFi, we start by developing a *"General Model"* with the goal of automatically detecting occupancy in any environment without any user input. In practice, this will rarely reach the expected performance goals because it is impossible to generalize the size and geometry of every home. For example, a single home in some configurations could be sliced up into three different apartments which would dramatically change the detection requirement for a system installed on a single floor.

To understand the potential performance gap, we first investigate what we expect to be the best-case model in terms of performance, *"Specific Per-House Model"*, where we collect and train data individually on each residence. This is clearly too labor intensive for most users, but it gives us a sense for the performance lost in our generalized model. This then leads to our proposed framework, *"Domain Adaptive Model"*, which uses the generalized model and augments it with a small amount of training data through a transfer learning approach. The domain adaptive model is explained in more detail in the next section.

6 Domain Adaptation at the Edge

Wireless signal propagation is tightly coupled with the physical environment such as the house layout, furniture placements and the position of the WiFi nodes. To detect the presence of a person inside the room, we at least require the knowledge of this profile. However, it is practically impossible to generalize the multipath profile of different indoor spaces. M-WiFi addressed this issue by using transfer learning to tune a pre-trained model to a new building by using a small amount of annotated data from the new house.

However, such model adaptation necessitates two main requirements. First, given the multi-dimensional CSI data and the large sampling rates, we can not assume cloud access to perform this retraining and tuning. To make our approach more amenable to resource-constrained targets, we use a wrapper method which computes the model with a certain subset of features and evaluate the importance of each feature using the training data. Note that this process is performed offline in pre-deployment phase, so it does not affect the real-time performance of the system. We repeat a similar process for pruning the classification model. The second requirement is that retuning the model involves annotating data and thus interaction with the house occupants. On one hand, the more data we can label form the new house, the faster we can tune the model and achieve the expected performance. On the other hand, frequent requests to the occupants for annotation causes user frustration and inconvenience. So, the goal is to minimize the user involvements by only annotating high fidelity data.

M-WiFi starts operating in a new house by using a generalized model trained pre-deployment with a large annotated dataset from different houses. Over the first few days of deployment, the system detects the potential occupancy changes and asks the user to annotate a small amount of data, which is then used for tuning the generalized model. M-WiFi repeats this retuning process in the case

Algorithm 2 Change Point Detection Algorithm

Input feature set $\{y_{j,t}\}^T$

stopping criterion=Number of required break points(p),grid size $\delta=2$
for feature index j=1:N_f **do**
  signal=$\{y_{j,t}\}^T$
  Initialize $L_j \leftarrow \{\delta, 2\delta, ..., ([T/\delta]-1)\delta\}$
  while stopping criterion is met **do**
    $k \leftarrow |L_j|$ ▷ Number of breakpoints
    $t_0 \leftarrow 0$ ▷ Dummy variables
    Denote by t_i $(i=1,...,k)$ the elements of L_j, $L_j=\{t_1,...,t_k\}$
    Initialize G_j a $(k-1)$-long array. ▷ list of gains
    for $i=1,...,k-1$ **do**
      $c(x_{i=m,...,n}) = \sqrt{\sum_{i=m}^{n}(x_i-\bar{x})^2 \times (n-m)}$
      where $\bar{x} = \frac{1}{n-m}\sum_{i=m}^{n} x_i$
      $G_j[i-1] \leftarrow c(y_{t_{i-1}:t_{i+1}}) - [c(y_{t_{i-1}:t_i}) + c(y_{t_i:t_{i+1}})]$
    $\hat{i} \leftarrow argmin_i G_j[i]$
    Remove $t_{\hat{i}+1}$ from L_j.
$[L_{maj}, conf] = majority\_vote(L_{1:N_f})$
$L_{filt} = L_{maj}(conf[L_{maj}] > 0.5)$ ▷ Confidence filtering
for $i=1:|L_{Filtered}|$ **do** ▷ Proximity filtering
  if $|L_{filt}(i+1) - L_{filt}(i)| \leq \tau$ **then**
    Remove $L_{filt}(i+1)$ from L_{filt}

Output set L_{filt} of estimated changepoint indexes

of detecting a change in WiFi link conditions such as displacement of the WiFi transceivers or significant furniture movements. In the next section, we elaborate on how M-WiFi selects the candidate time periods for annotation.

6.1 Occupancy Transition Detection

The naive approach for data annotation is to ask the user to provide a detailed report about their home occupancy during a deployment phase. In practice, this is too labor intensive and potentially error prone. Instead, M-WiFi uses an unsupervised algorithm to detect the potential occupancy transitions and select the candidate time periods with rich sensing information. This process not only optimizes the user involvement, but also reduces the user error by helping them remember their activities for a specific time period. The intuition is that transition from occupied to unoccupied in a building is accompanied with large movements of occupants (when leaving or entering the house) which appear as abrupt changes on feature values. However, the transition between activities, such as from sleeping to walking, could also result in abrupt changes and so it can be verified by asking the user to annotate the selected periods.

To detect the potential occupancy transitions, M-WiFi collects wireless signals and extracts features on the fly for a given time window. The extracted features are then stored locally during the day creating a multivariate time series signal. After capturing this signal for the entire day, M-WiFi applies an offline change point detection algorithm that chooses the best possible segmentation using a Buttom-up technique [22], explained as Algorithm 2. The algorithm starts by splitting the signal in equal windows of size δ and sequentially merging them until only p change points remain. At every step,

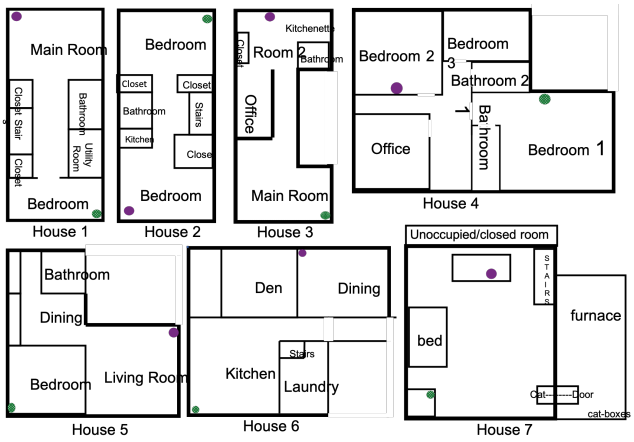


Figure 3: Experimental Setup

all potential change points are ranked by the gain measure defined as:

$$G(i-1) = c(y_{i-1:i+1}) - [c(y_{i-1:i}) + c(y_{i:i+1})] \quad (4)$$

$$c(y_{i=m:n}) = \sqrt{\sum_{i=m}^{n}(y_i-\bar{y})^2 \times (n-m)} \quad (5)$$

where y is the time-series signal, c is the cost function defined based on the standard deviation of the signal every 3 sequential change points. The change points with the lowest gain G are deleted until the stopping criterion (here the number of requested change points) is met. This process is essentially pruning the initial change point vector by merging multiple segments. The same process will be repeated for all the features, resulting in p selected change points per feature index. Finally, M-WiFi applies two heuristics to select the final list of candidate time periods for annotation. First, it performs a majority voting across features and selects the change points that are common between at least half of the features. Second, it applies a proximity filtering to ignore periods shorter than a threshold. The idea is maximize the amount of annotated data for a given annotation request to user. We define this threshold to be 1 hour in our experiments.

7 Evaluation

7.1 Implementation

M-WiFi consists of two main hardware components: (1) a Hub Collection Unit, and (2) a WiFi client, both are TPLink N600 OpenWRT platforms and equipped with Atheros WiFi chipsets, 2 antennas, 128MB of memory, 8GB of local storage, and a 560MHz MIPS 74Kc CPU. We use the CSI tool [24] installed on the hub to obtain the CSI phase and amplitude values of 56 sub-carriers for each received packet per antenna, resulting in a 2x2x56 CSI matrix. During the data collection, the hub queries any active client at a specific rate and extracts the CSI information from the received responses. In addition, to avoid interference with the background traffic, the nodes coordinate to operate on a WiFi channel with minimum possible interference. Most of the data collections are performed in 5GHz frequency band employing an unused 40MHz channel. We later show that the detection performance is independent of the operating frequency or wireless channel and the system can operate in both 2.4 or 5GHz frequency ranges.

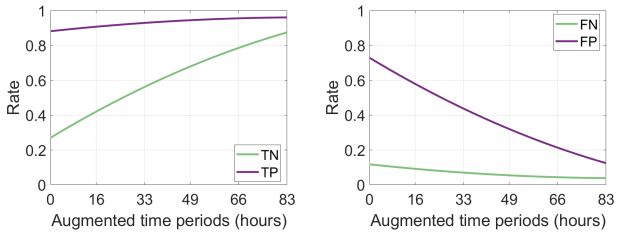


Figure 4: Domain Adaptive performance converges within a few days with an average of 4.5 annotation requests to the user.

We explored different classification models ranging between SVM, decision tree, random forest, KNN, boosted trees, and Multilayer Perceptron. Our empirical results show that Multilayer Perceptron works best for this type of data, given the diversity of occupancy scenarios and the complex relationships in the data. The results provided here are all based on a 2 layer MLP classifier.

7.2 Experimental Setup

We deploy M-WiFi in 7 different homes and a total of 25 different setups, ranging from a small apartment to single houses with multiple rooms and pets. Participants were given no special instructions and followed their routine activities and home occupancy functions. The homes include both single-person and multi-person occupants and the people living in the home include students, professionals, and homemakers. The floorplans for the test spaces, along with the locations of the hub (green) and the client (purple) are shown in Figure 3. The duration of data collection varies between houses, resulting in a total of 100 days-worth of data. The summary of each home conditions are presented in Table 1.

To collect the ground truth, we used a wireless motion activated video cameras that is placed on the outside of all entrances to the home. For this, we used a Netgear ARLO video capture system, since it is battery operated and can run without external internet access[1]. The captured videos are analyzed manually on a weekly basis to generate a file containing occupancy status and timestamps for which this status is held. In addition, we performed over 100 controlled experiments for system debugging, which are short-term measurements of the environments with specified human activities such as unoccupied, occupant sitting still, lying down, walking randomly, or pet presence. These datasets are used in the sensitivity analysis to demonstrate the signatures that the models are learning.

[1]The experiments are conducted under appropriate IRB approvals

Houses	Type of house	# of People	# of Pets	# of Rooms	# of Days
H1	Town	1	0	5	14
H2	town	2	2 dogs	5	7
H3	Town	1-5	0	2	9
H4	Single	1-5	1dog 2cat	6	21
H5	Apt	1	0	4	15
H6	single	2	2 dogs	4	9
H7	single	3	2 cats	4	24

Table 1: Experimental Setup

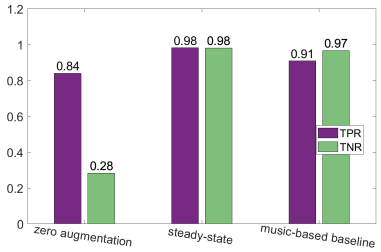


Figure 5: the generalized baseline model performs poorly in new houses due to the lack of multipath profile data (a), but it reaches a steady state performance after few days of augmentation (b) with an average of 98% accuracy, which is comparable to our high-performance baseline (c) with 93% accuracy but much worse run-time performance (Table 2)

(1 day worth of data)	M-WiFi	MUSIC-based baseline
execution time	2.9 hours	23.7 hours

Table 2: M-WiFi shows an 8x runtime performance compared with a MUSIC-based approach at a fraction of the memory.

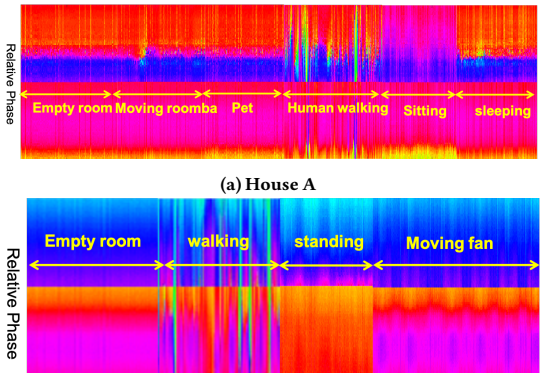
We target the estimation of human sensing over 5-minute intervals, so the features are calculated using a sliding window of 5 minute length with 50% sliding overlap. We discuss the impact of window size on the system performance in the sensitivity analyses. Sensor sampling between our hub and client devices will occur at a rate of 10Hz to capture subtle movements. We also define accuracy as $(TP+TN)/(TP+TN+FP+FN)$, given true/false positive and negative cases.

7.3 Detection Performance

To evaluate our proposed system, we first provide the detection accuracy for the "Domain Adaptive Model", which represents the expected performance when the system is installed in a completely new house. We show that we can tune the generalized model to the new house by using incremental training for as few as 3 days worth of data.

To better understand the effectiveness of this approach, we compare the detection accuracy between the generalized model (zero augmentation), steady-state model (specific per-house) and a high-performance model and resource hungry MUSIC algorithm (hard to imagine running on embedded targets in the near future). For the generalized model, we use a *leave one house out* approach, where in each iteration the data from one house is entirely used for testing and the data from the rest of the houses are merged as training set. This equivalents with the expected performance of the system on the very first day of deployment with zero augmentation. In the steady-state mode, we train and test the model for each individual house using 5-fold cross validation. This provides the long-term steady state performance, where the multipath profile of the environment is fully captured by the models over time. This gives us a sense of the expected performance in the long run by gradually learning the multipath profile using our proposed transfer learning method.

7.3.1 ***Domain Adaptive Model Performance*** We evaluate M-WiFi's performance in a realistic scenario, where the classification models are trained with a large dataset from multiple houses and



(a) House A

(b) House B

Figure 6: CSI phase carries distinctive patterns for pets and moving objects compared to the human body.

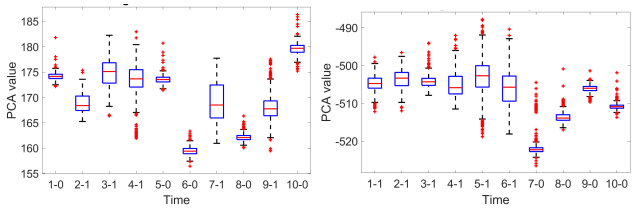


Figure 7: the distinctive PCA distributions of the two occupancy classes confirm the effectiveness of the handcrafted features.

tested on a completely new house with the addition of sparse augmentation. In each iteration, we leave one dataset out for testing and use the rest as training set. Then, we gradually add partially annotated data from the test house to tune the general model. Figure 4 shows the impact of this sequential learning as the model receives more augmented data. We can see that the model tunes to the new house and achieves higher accuracy over time. With an average of 3 days' worth of data, M-WiFi can achieve around 90% true detection, and less than 15% false detection. It should be noted that the self-tuning duration mainly depends on the volume of annotated data and can be shortened if the users can label more data. In addition, since the transfer learning is applied in real-time and on-device, the system can still detect the occupancy in the initial days of deployment and during the domain adaptation phase.

7.3.2 ***Steady-state Performance*** Next, we compare the steady-state performance with two baselines: generalized model, and MUSIC-based learning. Generalized model uses *leave one house out* and zero augmentation, in which an average true positive of 84% and true negative of 28% is achievable. The reason for the weak performance of the generalized model is the lack of multipath profile knowledge for the test houses. So, without any augmentation and the multipath profile of the environment, the model has a hard time detecting the unoccupied periods. This demonstrates the importance of multipath signals and the tuning process on robust human sensing.

Steady-state mode shows the model performance in long term when enough the domain adaptation converges to steady performance. For this, given the short duration of some datasets, we emulate this scenario by training and testing for each individual house using 5-fold cross validation. As shown in Figure 5, M-WiFi can achieve

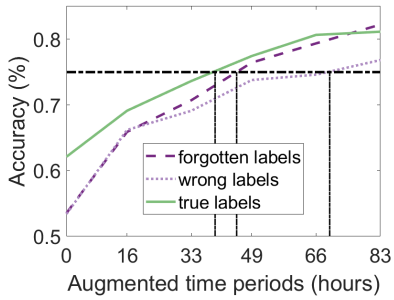


Figure 8: User errors on data annotation delays the model adaptation.

around 98% accuracy in correctly detecting occupied and unoccupied states. It should be noted that occupants in each house had their regular activities during the experiments and the datasets include periods of inactivity (such as during periods of sleeping or watching TV), pets alone at home, or moving furniture around the house. So, we can see that the proposed framework is robust to all these corner cases. We further studied the miss-classifications and realized that the majority of false positive and false negatives occurred at the occupancy transitions. This is mainly due to inaccurate timestamps in ground-truth data and can be easily handled by post-processing or biased toward occupied or unoccupied depending on the application objectives.

We also compare the steady-state performance with the MUSIC-based baseline [17], which also uses multipath profile but without resource constraint considerations. We see that M-WiFi performance is comparable to the baseline, while it achieves 8X higher runtime performance (Table 2).

7.3.3 Underlying Signatures To better understand the underlying reason for the robustness of M-WiFi, we performed extensive controlled experiments with known human activities. In these experiments, a single user enters an apartment and performs different occupancy scenarios such as sitting, lying, and walking for a fixed period of time both in LoS and NLoS of the WiFi nodes. We also perform different experiments with moving objects such as fans and vacuum cleaner, or pets (with and without the presence of a person inside the house). Figure 6 shows a snapshot of CSI relative phase between receiving antennas for different scenarios such as having a pet (here a medium size dog) alone at home or a moving fan. We can see that the moving Roomba or the dog has minimal effect on the CSI measurements. Due to the size and their unique movement patterns, the wireless reflections are disturbed minimally compared to when a person is inside the house. However, a moving fan close to the WiFi nodes can create a distinctive pattern in wireless reflections because of the height and metallic surface. The self-tuning and self-calibration module of M-WiFi detects such potential occupancy transitions that are unknown to the model and adapt it accordingly through user verification.

While visualizing all the features is not possible due to the large dimensions, we apply PCA on the feature set to study the effectiveness of feature engineering. Figure 7 shows the variations of the first principal component across 10 equally sized time periods in two different houses. The labels on the x axis are in this format: "period#-Occupancy#", where the first parameter shows the time period number and the second parameter shows the occupancy of the

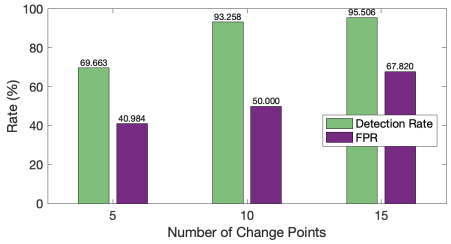


Figure 9: Change point detection performance used to identify time periods that users should label

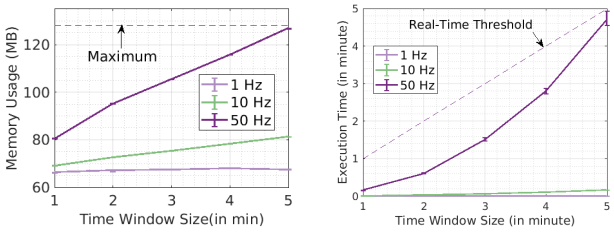


Figure 10: Memory usage and execution time for different sampling rate and time window sizes

house during this time period (0 for unoccupied and 1 for occupied). As we can see, the PC distributions are quite distinctive between the two occupancy classes, which confirm the effectiveness of the handcrafted features in distinguishing the two classes.

7.4 Self-tuning Performance

In the previous section, we saw that the Domain Adaptive Model reaches the steady-state performance after a few days of data augmentation. However, this process relies on the users' input to verify and annotate the tuning data, which may not be accurate all the time. Some users may forget when exactly they left the house or come back, or they may enter the wrong labels. So, to evaluate the impact of user error on model tuning, we imitate these errors by randomly selecting some candidate time windows in each iteration of domain adaptation and assigning either "unknown" or the opposite labels to the tuning data. As we can see in Figure 8, even in the presence of user errors, the model can reach the steady state, but with a lower slope. This indicates that it may require longer augmentation phase to reach the expected performance.

7.4.1 ***Transition Detection Performance*** Another component of the self-tuning algorithm is to detect potential occupancy changes and send the candidate time periods to the user for labeling. However, it is possible that the change detection algorithm confuses changes in the occupants' activity with occupancy transitions. For example, transition from sleeping to daytime activity could be detected as a potential occupancy change while in both cases the home is occupied. For a better performance of the system, we expect near zero false negatives in change detection, but higher false positive is acceptable as it only increases the number of requests to the user. However, this defines a trade-off between user comfort and system performance.

One parameter to control this trade-off is the initial number of change points. With a larger initial value, the algorithm can find change points more accurately, but at the expense of larger number of false positives. We evaluate the impact of this parameter on change detection across all the houses for windows of 24 hours. as shown in Figure 9, we found the best trade-off to happen at initial change point equal 10, which results in detection rate of 93% in exchange of 50% false positive. This corresponds to an average of 5 verification requests to users in a 24 hour period. It should be noted that M-WiFi only requests for user input in the first days of deployment for a maximum of a week, and does not need to repeat this regularly after the deployment phase. So, the goal here is to further reduce the user discomfort in the deployment phase by improving the system performance as fast as possible.

7.5 Workload Benchmarks

One of the main features of M-WiFi is its ability to perform the entire pipeline of feature extraction, learning, and detection on an embedded platform with limited resources. The most significant execution time and memory usage factor is the size of the time window for which the features are extracted along with its corresponding sampling rate. As shown in Figure 10, the larger the time window is, the higher memory and execution time can be expected. However, even with a time period of 5 minutes, the average memory usage is 110MB, which is practically achievable on most application class embedded platforms. This efficiency is primarily due to the optimized set of handcrafted features and M-WiFi's ability to extract these features on the fly. In addition, the average time to execute feature extraction and classification is always smaller than the corresponding time window, which means that the data collection and processing can be efficiently paralleled with zero runtime overhead. Therefore, M-WiFi can reliably run on any embedded platform in real time.

7.6 Wireless Co-existence Testing

One concern is that adding additional wireless traffic might impact existing home networks or that stray signals from neighbors might cause interference. In order to measure the impact of our system on the existing wireless ecosystem, we run speed tests before, during and after our experimentation period in order to detect any network performance side effects. We observe the average packet loss rate of 0%, 0.5% and 4% for 1, 10 and 100Hz sampling rates, respectively. Therefore, 10Hz sampling rate seems to be the sweet spot for both the network and system performance.

It should be noted that M-WiFi's framework is capable of automatically switching to a wireless channel with the least background traffic to further minimize interference. To understand the impact of channel switching on detection performance, we performed an experiment in one of the houses, where the system was running on a 5Ghz wireless channel for 5 days and then switched to 2.4Ghz for the next 3 days. We used the 5Ghz data for training and 2.4Ghz data for testing. We observe that the model correctly detects the human presence with 91% accuracy. This confirms that the feature vector and the classification model are channel independent and so the sensing system can seamlessly transition to any available wireless channel if need be.

7.7 Impact of Time Window Size

The time window size used for feature extraction is another factor affecting M-WiFi performance. We measure the average human presence accuracy for different window sizes of 1, 5, and 10 minutes. We can see that the accuracy slightly reduces as we increase the window size (98.6%, 97.7%, and 96.1%, respectively). This is due to

the aggregate functions applied on raw data for feature extraction. However, very small window sizes can increase the computational cost of the system due to higher rate of feature extraction execution. So, we select 5 minutes as the best window size to balance system and runtime performance.

8 Future Work and Conclusion

This paper presents a domain adaptive model for WiFi human presence detection with learning capabilities at the edge. We show that our proposed baseline model with some additional input from the user can achieve high levels of accuracy on completely unseen environments, even in the presence of pets and moving objects. In our future work, we plan to explore how M-WiFi can interact with spatial features to further improve the performance in long periods of motionless occupancy (like when sleeping). We believe that the last known location of motion (potentially a motion trajectory) can be used as a hint to determine the home occupancy. Using sparse training of locations during the installation step of the system, motion can be potentially classified into a few discrete zones within the home, which can be used to construct the geographical model of transition zones.

9 Acknowledgements

This work was funded in part by ARPA-E grant DE-AR0000932. We thank our shepherd and the anonymous reviewers for their valuable feedback.

References

[1] TPLink N600. https://openwrt.org/toh/tp-link/tl-wdr3600, 2020. [Online; accessed 10-January-2020].

[2] Y. Agarwal, B. Balaji, R. Gupta, J. Lyles, M. Wei, and T. Weng. Occupancy-driven energy management for smart building automation. In *ACM BuildSys*, 2010.

[3] I. B. Arief-Ang, F. D. Salim, and M. Hamilton. Da-hoc: semi-supervised domain adaptation for room occupancy prediction using co2 sensor data. In *ACM BuildSys*, 2017.

[4] A. K. Das, P. H. Pathak, J. Jee, C.-N. Chuah, and P. Mohapatra. Non-intrusive multi-modal estimation of building occupancy. In *ACM SenSys*, 2017.

[5] M. Farhadi, M. Ghasemi, S. Vrudhula, and Y. Yang. Enabling incremental knowledge transfer for object detection at the edge. In *CVPR Workshop*, 2020.

[6] L. Gong, W. Yang, Z. Zhou, D. Man, H. Cai, X. Zhou, and Z. Yang. An adaptive wireless passive human detection via fine-grained physical layer information. *Ad Hoc Networks*, 38, 2016.

[7] W. Jiang, C. Miao, F. Ma, S. Yao, Y. Wang, Y. Yuan, H. Xue, C. Song, X. Ma, D. Koutsonikolas, et al. Towards environment independent device free human activity recognition. In *ACM MobiCom*, 2018.

[8] M. Kotaru, K. Joshi, D. Bharadia, and S. Katti. Spotfi: Decimeter level localization using wifi. In *ACM SIGCOMM Computer Communication Review*, volume 45, 2015.

[9] R. B. Lehoucq, D. C. Sorensen, and C. Yang. *ARPACK users' guide: solution of large-scale eigenvalue problems with implicitly restarted Arnoldi methods.* SIAM, 1998.

[10] X. Li, S. Li, D. Zhang, J. Xiong, Y. Wang, and H. Mei. Dynamic-music: accurate device-free indoor localization. In *ACM UbiComp*, 2016.

[11] J. Liu, Y. Wang, Y. Chen, J. Yang, X. Chen, and J. Cheng. Tracking vital signs during sleep leveraging off-the-shelf wifi. In *ACM MobiHoc*, 2015.

[12] W. Liu, X. Gao, L. Wang, and D. Wang. Bfp: Behavior-free passive motion detection using phy information. *Wireless Personal Communications*, 83(2), 2015.

[13] X. Liu, J. Cao, S. Tang, J. Wen, and P. Guo. Contactless respiration monitoring via off-the-shelf wifi devices. *IEEE TMC*, 2015.

[14] K. Qian, C. Wu, Z. Yang, Y. Liu, and Z. Zhou. Pads: Passive detection of moving targets with dynamic speed using phy layer information. In *ICPADS*. IEEE, 2014.

[15] R. Schmidt. Multiple emitter location and signal parameter estimation. *IEEE transactions on antennas and propagation*, 1986.

[16] O. Shih, P. Lazik, and A. Rowe. Aures: A wide-band ultrasonic occupancy sensing platform. In *ACM BuildSys*, 2016.

[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.

[18] E. Soltanaghaei, A. Kalyanaraman, and K. Whitehouse. Multipath triangulation: Decimeter-level wifi localization and orientation with a single unaided receiver. In *ACM MobiSys*, 2018.

[19] E. Soltanaghaei and K. Whitehouse. Walksense: Classifying home occupancy states using walkway sensing. In *ACM BuildSys*, 2016.

[20] E. Soltanaghaei and K. Whitehouse. Practical occupancy detection for programmable and smart thermostats. *Applied Energy*, 220, 2018.

[21] T. Teixeira, G. Dublon, and A. Savvides. A survey of human-sensing: Methods for detecting presence, count, location, track, and identity. *ACM Computing Surveys*, 5(1), 2010.

[22] C. Truong, L. Oudre, and N. Vayatis. Selective review of offline change point detection methods. *Signal Processing*, 167, 2020.

[23] P. Van Dorp and F. Groen. Feature-based human motion parameter estimation with radar. *IET Radar, Sonar & Navigation*, 2008.

[24] Y. Xie, Z. Li, and M. Li. Precise power delay profiling with commodity wi-fi. *IEEE TMC*, 18(6), 2018.

[25] 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*, 2(3), 2018.

[26] D. B. Yang, L. J. Guibas, et al. Counting people in crowds with a real-time network of simple image sensors. page 122. IEEE, 2003.

[27] Z. Yang and B. Becerik-Gerber. Cross-space building occupancy modeling by contextual information based learning. In *ACM Buildsys*, 2015.

[28] F. Zhang, C. Wu, B. Wang, H.-Q. Lai, Y. Han, and K. R. Liu. Widetect: Robust motion detection with a statistical electromagnetic model. *UbiComp*, 3(3), 2019.

[29] Z. Zhou, Z. Yang, C. Wu, L. Shangguan, and Y. Liu. Towards omnidirectional passive human detection. In *2013 Proceedings IEEE INFOCOM*, pages 3057–3065. IEEE, 2013.

[30] H. Zou, J. Yang, Y. Zhou, L. Xie, and C. J. Spanos. Robust wifi-enabled device-free gesture recognition via unsupervised adversarial domain adaptation. In *ICCCN*. IEEE, 2018.

[31] H. Zou, Y. Zhou, J. Yang, H. Liu, H. P. Das, and C. J. Spanos. Consensus adversarial domain adaptation. In *AAAI*, 2019.