Robust Occupancy Inference with Commodity WiFi

Xiaoxuan Lu[†], Hongkai Wen[†], Han Zou[§], Hao Jiang[*], Lihua Xie[§], and Niki Trigoni[†]

[†]Department of Computer Science, University of Oxford, United Kingdom
[§]School of Electrical and Electronics Engineering, Nanyang Technological University, Singapore
[*]College of Electrical Engineering and Automation, Fuzhou University, China

***Abstract*—Accurate occupancy information of indoor environments is one of the key prerequisites for many pervasive and context-aware services, e.g. smart building/home systems. Some of the existing occupancy inference systems can achieve impressive accuracy, but they either require labour-intensive calibration phases, or need to install bespoke hardware such as CCTV cameras, which are privacy-intrusive by default. In this paper, we present the design and implementation of a practical end-to-end occupancy inference system, which requires minimum user effort, and is able to infer room-level occupancy accurately with commodity WiFi infrastructure. Depending on the needs of different occupancy information subscribers, our system is flexible enough to switch between snapshot estimation mode and continuous inference mode, to trade estimation accuracy for delay and communication cost. We evaluate the system on a hardware testbed deployed in a 600m^2 workspace with 25 occupants for 6 weeks. Experimental results show that the proposed system significantly outperforms competing systems in both inference accuracy and robustness.**

I. Introduction

Occupancy awareness has become a fundamental building block for various ubiquitous services, and attracted a lot of interest from both academia and industry. For instance, building automation systems [7] rely on instant occupancy to dynamically adjust the heating, ventilation and air-conditioning (HVAC) facilities to reduce energy consumption while improving user comfort. Workplace coordination systems [13] leverage the room-level presence and availability of users to enable social interactions and foster potential collaborations. Furthermore, smart homes learn the behaviour pattern of users from their occupancy over the long term, which is the key for many applications such as activity recognition, elderly monitoring and breach detection. There has been a solid body of work on inferring occupancy in indoor environment. Early systems use passive infrared (PIR) sensors [2] and magnetic reed switches [17] to detect motion or door open/shut events. However, those sensors only provide binary information and are inherently noisy, and thus are not suitable for accurate inference. Other systems [9] use only the historical occupancy data to infer live occupancy, but they need long-term calibration to work. On the other hand, systems using information-rich sensors, such as Kinect [11] or cameras [8], are very accurate in tracking human movement, but are privacy-intrusive and not suitable for many environments such as homes and hospitals. In addition, transmitting and processing visual data is typically expensive, which requires significant amount of computation and communication resources. Recently, WiFi infrastructure in buildings has been considered for the task of occupancy inference [15, 4]. For example, [15] harvests beacons emitted from nearby WiFi access points (APs), and is able to determine room-level occupancy information. However, it needs to install a dedicated app on mobile devices and constantly perform WiFi scans, which requires extra user effort and would drain the battery quickly. The system in [4] considers a different approach, which infers user occupancy by looking at the connections between mobile devices and APs. If a device is connected to an AP, the system then assigns the occupant to the area near that particular AP. However, in practice a device is not always connected to the closest APs, which significantly limits its inference accuracy.

In this paper, we make occupancy inference in large indoor space practical and efficient. We propose a novel non-intrusive **Wi**reless Occu**p**ancy **In**ference (WiPin) system, which infers accurate occupancy information with minimum user effort and existing WiFi infrastructure. The idea is that when a user is present in a familiar indoor environment such as her office, the smart devices carried by her (i.e. mobile phones or wearable devices) tend to connect and exchange information with nearby WiFi APs, with or without explicit control of the user. For instance, the user may browse news on web or watch videos on Youtube occasionally, while the smartphone themselves would pull emails and update tweets at certain intervals. Such wireless traffic can be effectively captured by the APs, where our system further analyses the received packets to infer occupancy and identify the users through MAC addresses of their devices at the same time. To balance efficiency and accuracy, the proposed WiPin system consists of two modules, a lightweight front-end which runs on the resource-constrained WiFi APs, and a back-end that resides on the cloud. The front-end senses the live WiFi packets emitted from the user devices, and estimates the current occupancy in real-time, which can be directly fed to the subscribers. In the cases where inference accuracy is the predominant factor, the back-end module of WiPin is activated, which fuses the estimated snapshot

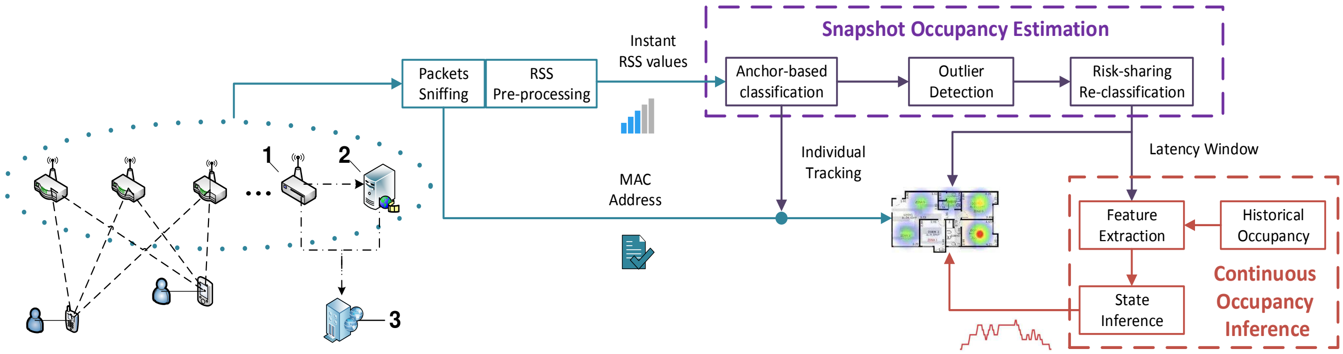

Figure 1: Architecture of the proposed WiPin system. 1. front-end nodes; 2. back-end server; 3. occupancy information subscribers. WiPin includes two major components; snapshot occupancy estimation runs on the front-end nodes while continuous occupancy inference runs on a back-end server.

occupancy with historical occupancy data, to infer much finer-grained occupancy and cope with challenging events, e.g. when users change rooms without bringing their phones. In this way, WiPin dynamically tasks the front-end and back-end, to cater for the needs of different subscribers, e.g. building automation systems typically require crisp response to control heating/cooling/lighting, while activity recognition applications would prefer accurate occupancy information, but are more delay tolerant.

Concretely, the technical contributions of this paper are:

- We build WiPin, a practical end-to-end occupancy inference system, which uses commodity WiFi hardware, requires minimum user effort and is able to robustly infer room-level occupancy.
- We design an efficient snapshot occupancy estimation algorithm, which can determine the occupancy of users solely based on the signal strength of their mobile devices, and runs in real-time on resource-constrained wireless routers.
- We propose a continuous occupancy inference approach to further improve the estimated occupancy, by taking the temporal correlations between snapshot occupancy, and historical occupancy data into account.
- We implement the proposed WiPin system in a hardware testbed, and evaluate its performance over 6 weeks. Extensive experiments show that WiPin is superior to the competing systems in both accuracy and robustness.

The rest of the paper is organised as follows. Sec. II provides an overview of the proposed WiPin system. Sec. III presents the proposed snapshot occupancy estimation algorithm, while Sec. IV discusses the continuous occupancy inference approach. Sec. V evaluates our system in real world, and the related work is covered in Sec. VI. Sec. VII concludes the paper and discusses directions of future work.

II. System Overview

To be practically useful, the WiPin system is designed to satisfy the following criteria: a) it should require *minimum effort* from the occupants, and should not seek their persistent participation, e.g. installing dedicated apps or running daemon processes on their devices; and b) it should be *flexible*, and is able to seek the best trade-off between inference accuracy and delay. To this end, we build WiPin as a special wireless distribution system, where the commercial WiFi routers are also *sensor* nodes to sniff wireless traffic generated from the occupants' mobile devices. Fig. 1 illustrates the architecture of the proposed WiPin system.

Front-end: At runtime, the WiFi routers log the Received Signal Strength (RSS) of the captured packets, together with the associated device identifiers (i.e. MAC addresses). The data is then forwarded to the front-end of the proposed WiPin system. In our implementation, the front-end runs at the sink node, which estimates the *snapshot occupancy* based on the RSS measurements, i.e. the current number of occupants within a *zone*, e.g. a room or working area.

Back-end: In the cases where higher quality occupancy information is needed, the estimated snapshot occupancy is forwarded to the back-end for further inference. The back-end caches the stream of estimated snapshot occupancy, and infers the *continuous occupancy* by exploiting their temporal correlations. When historical occupancy data is available, WiPin also incorporates such prior to further improve occupancy inference.

Now we are in a position to present the proposed approaches of snapshot occupancy estimation as well as continuous occupancy inference.

III. Snapshot Occupancy Estimation

Let us assume an indoor environment can be divided into M zones, where a zone can represent an individual room, or a specific area in large open space. For simplicity, we assume that each zone is associated with a WiFi AP

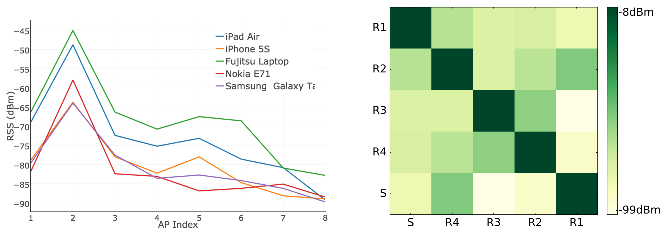

Figure 2: $\{a\}$. WiFi RSS by different mobile devices at the same location (each curve is averaged out from multiple measurements; $\{b\}$. Visualization of RSS matrix of 5 anchors: 1 sink node (S) and 4 remote nodes (R1-R4), where -8dBm is the offline calibrated value for self-sensed RSS.

(referred to as *anchors* hereafter), whose location is known beforehand. Note that although such configuration requires dense infrastructure, it will become standard in the near future, since the next generation WiFi 802.11ad operates on 60GHz, whose communication range is much smaller. Therefore in this case, the problem of estimating occupancy (i.e. determining which zone a use is in) is equivalent with assigning the devices of the occupants to the nearby anchors based on their proximity.

As discussed in the previous section, the proposed WiPin system estimates the snapshot occupancy in real-time given the current observed RSS measurements. However, it is well known that in practice, RSS measurements can vary significantly under environmental dynamics, which makes occupancy estimation inaccurate. To address this problem, WiPin considers a two-step estimation approach, where it firstly performs an initial estimation step by clustering the occupants devices based on their RSS signatures, and then it reconciles the current assignment to remove possible outliers caused by environmental dynamics, e.g. signal blockage.

A. Initial Estimation

RSS Normalisation: In practice, for a given anchor, the observed RSS measurements of different devices can vary significantly, even when the devices are fixed at the exact same position (see Fig. 2$\{a\}$). This may jeopardise occupancy estimation, since such variation in RSS will introduce significant errors during clustering. To address this issue, WiPin uses the ordinary Procrustes analysis method [23] to normalise the RSS measurements among heterogeneous devices, by exploiting the fact that their RSS variations across different anchors tend to be similar (see the shapes of the lines are very similar in Fig. 2$\{a\}$).

Zone Assignment: At a fixed timestamp, for a given devices, let $\boldsymbol{o}$ be the vector of normalised RSS measurements, where $\boldsymbol{o} = [o_1, ..., o_M]$, and o_i is the observed RSS (of the mobile devices) from the i-th anchor. Given $\boldsymbol{o}$, the task of zone assignment is to find the anchor that is closest to the devices, i.e. assigning this occupant to a particular zone (assuming one to one mapping between anchors and zones). A naive approach could be proximity-based, i.e. just picking up the anchor with strongest RSS. However, this is very unreliable in the presence of wireless signal variations. The state of the art systems (e.g. [15]) consider a fingerprinting-based approach, which cope with the enviornmental dynamics by periodically surveying and re-calibrating the radio map of the indoor space.

WiPin considers a different approach, which uses the anchors as the *references*, to adaptively evaluate the "baseline" of RSS measurements. The key observation here is that an anchor of the WiPin system also generates wireless traffic when operating, which can be overheard by others in the same way as that of the occupants' mobile devices. Therefore, we can compute the pairwise RSS measurements between anchors (as visualised in Fig. 2$\{b\}$). Let $\boldsymbol{r}_i = [r_1, ..., r_M]$ be the reference RSS vector of the i-th anchor, where r_j, $1 \leq j \leq M$ is the RSS measurement made by the j-th anchor. In other words, $\boldsymbol{r}_i$ is essentially the "expected" device signal strength values of the i-th zone, observed by all anchors. In this way, the problem of assigning a device to a zone can be cast into that of finding the anchor whose reference RSS vector $\boldsymbol{r}_i$ is the closest to the RSS measurements $\mathbf{o}$ of that device:

$$f(\boldsymbol{o}) = \operatorname*{argmin}_i D(\boldsymbol{r}_i, \boldsymbol{o}) \tag{1}$$

where $D(\cdot, \cdot)$ is a distance metric (in our case we consider the Euclidean distance), and $f(\cdot)$ is a function that assigns the devices to a zone. Therefore, at the current timestamp, the RSS measurements of all devices assigned to the $i-th$ zone can be represented as:

$$\Psi^i := \{\boldsymbol{o} \in \mathbb{R}^{1\times M} | f(\boldsymbol{o}) = i\} \tag{2}$$

where the population of the i-th zone is the cardinality of the set $|\Psi^i|$. Note that our approach is inherently robust to RSS variations, since the reference RSS vectors $\boldsymbol{r}_i$ are subject to the same signal variation as the RSS measurements of the occupants' devices.

B. Mis-assignment Mitigation

The zone assignment function f essentially assigns the device to the anchor (of a zone) that shares the most similar RSS values. However in practice, such assignment may be inaccurate, i.e. the anchor may not be in the same zone with the device. For instance, as shown in Fig. 3, Alice is an occupant in zone 1 but is assigned to zone 2, since the observed RSS measurements of her devices is more similar with the reference RSS vector of anchor B. In this case, although anchor A is the physically closest one to Alice, the line-of-sight (LOS) path between the her and anchor A is blocked by another occupant, which results in significant RSS attenuation and causes f to make the wrong decision. In the following text, we explain how the proposed WiPin

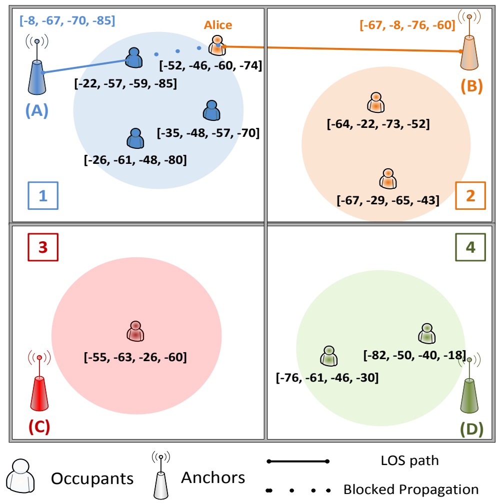

Figure 3: An occupant Alice in zone 1 is misclassified to zone 2 due to signal attenuation. Shaded circles represent the equal-range clusters formed by her peers. Among these centroids of clusters, Alice is closest to the one of zone 1.

system is able to a) for the i-th zone, detect such mis-assignment events; and b) re-assign the devices to the correct zone.

Detect Mis-assignment: Recall that in the initial estimation, WiPin computes a set of RSS measurements Ψ for each zone. We define the mean of RSS measurements for a given zone as:

$$\bar{\mathbf{o}} = \frac{1}{|\Psi|} \sum_{j=1}^{|\Psi|} \mathbf{o}^j \tag{3}$$

Intuitively, $\bar{\mathbf{o}}$ specifies the centroid RSS measurements of the given zone, and the distance between a RSS measurements vector $\mathbf{o}^j$ and $\bar{\mathbf{o}}_i$ indicates how well the j-th occupant can be clustered into the current zone. If a device with $\mathbf{o}^j$ that is very far away from the computed centroid, it is very likely to be mis-assigned to that zone, i.e. it is an *outlier*. To detect such outliers, let us first define the assignment error of a given occupant j as a random variable $\boldsymbol{e}^j$, where $\boldsymbol{e}^j = \mathbf{o}^j - \mathbf{o}$. Then the assignment error for a given zone can be represented as $\boldsymbol{e} = [\boldsymbol{e}^1, ..., \boldsymbol{e}^{|\Psi|}]$, where $|\Psi|$ is the total number of occupant currently assigned to this zone.

Ideally if there is no mis-assignment, the assignment error (i.e. residual) $\boldsymbol{e}$ for a given zone should be bounded. Formally, we require the following restricted residual (RR) constraint to hold for every zone:

$$P(\|\mathbf{e}\| \geq \xi + \varepsilon) \leq \eta \tag{4}$$

Here ξ is the confidence threshold, ε is a slack variable, and $\eta \in [0, 1]$ denotes the maximum tolerance. It can be shown

Algorithm 1: Sequential Outlier Detection Algorithm

Input: Ψ - The set of RSS vector defined in (2)
ξ - The confidence threshold
η - The maximum tolerant probability of deviation
ε - The slack variable
Output: $\Psi^{outlier}$ - All the outliers
$\widehat{\Psi^i}$ - The inliers of Ψ^i

```
Ψ^outlier ← ∅
for each zone i ∈ {1, 2, ..., M} do
    Ψ̂^i ← Ψ^i                          // Initialize Ψ̂^i
    while |Ψ̂^i| > 2 do
        // Check RR by Equ. (5)
        if ‖Σ_ee + ē^T ē‖ / (ξ+ε)^2 > η/M then
            q ← argmax_{o^j} ‖e^j‖        // Outlier detection
            Ψ̂^i ← Ψ̂^i \ {q}              // Set difference
            Ψ^outlier ← Ψ^outlier ∪ {q}   // Set union
        else
            break
return Ψ^outlier, [Ψ̂^1, ..., Ψ̂^M]
```

that a sufficient condition for Eq. (4) to hold is [19]:

$$\frac{\|\Sigma_{ee} + \bar{\mathbf{e}}^T \bar{\mathbf{e}}\|}{(\xi + \varepsilon)^2} \leq \frac{\eta}{M} \tag{5}$$

where $\bar{\mathbf{e}}$ and Σ_{ee} are the mean and covariance of $\mathbf{e}$ respectively.

Algo. 1 shows how the proposed WiPin system applies the above Eq. (5) to each zone, and detects the outliers. The idea is that we iteratively move the occupants whose RSS measurements are "far away" from the centroid of the current assigned zone to the outlier set, until the RR constraint holds on the inlier set. The complexity of Algo. 1 is cubic in the worst case, but in practice, we found it is very efficient since RR constraint tends to converge in just a few iterations. Note that different ξ and η may affect the performance of WiPin, e.g. smaller ξ and η means WiPin may produce more outliers.

Re-assign Outliers: Given the outliers $\Psi^{outlier}$, WiPin re-assigns them to different zones with a probabilistic membership function. Concretely, for a given RSS measurement $\boldsymbol{o}$, the likelihood that it belongs to zone i is:

$$\mu_i(\mathbf{o}) = \begin{cases} 1 & \mathbf{o} \in \widehat{\Psi^i} \\ \frac{\|\mathbf{o} - \bar{\mathbf{o}}^i\|}{\sum_{i=1}^{M} \|\mathbf{o} - \bar{\mathbf{o}}^i\|} & \mathbf{o} \in \Psi^{outlier} \end{cases} \tag{6}$$

where $\bar{\mathbf{o}}^i$ is the mean RSS measurement of the i-th zone.

Then the number of occupants in zone i can be evaluated as $\sum \mu_i(\mathbf{o})$.

IV. Continuous Occupancy Inference

The snapshot occupancy estimation algorithm discussed in the previous section is extremely efficient, and runs in real-time on commodity WiFi routers. However, it only considers the data captured at single timestamps, and assumes the occupants should carry their mobile devices with them all the time. In practice this may not always be the case, e.g. an occupant might go out for lunch but forget to bring her phone. To cope with such challenging scenarios, the proposed WiPin system also considers a continuous occupancy inference approach, which exploits the temporal correlation of the sequence of estimated snapshot occupancy, and incorporates historical data to improve inference performance.

Let us assume at each timestamp t ($t = 1 : T$ within one day), the frond-end of WiPin generates a collection of snapshot occupancy estimation (i.e. number of occupants) $\boldsymbol{z}_t = [z_t^1, .., z_t^i, z_t^M]$ for all M zones. On the other hand, we also assume that we possess certain historical occupancy information $[\boldsymbol{h}_{1'}, ..., \boldsymbol{h}_{T'}]$, where $\boldsymbol{h}_{t'}$ is the historical occupancy of the indoor environment at time t', obtained through a previous offline calibration phase. Note that here we assume the historical timestamps t' can be mapped uniquely to the current t, i.e. they represent the same time of the day (e.g. 12pm on Monday), but during different weeks. Intuitively, the historical occupancy represents the "expected" occupancy status of the building learned from the past. With a slight abuse of notation, we denote the historical occupancy of the environment as $[\boldsymbol{h}_1, ..., \boldsymbol{h}_T]$ hereafter. Let $\boldsymbol{s}_t$ be the actual occupancy at t, which is unknown. Then the task is to infer the sequence the real occupancy $S = [\boldsymbol{s}_1, ..., \boldsymbol{s}_T]$, given the observations $[\boldsymbol{z}_1, ..., \boldsymbol{z}_T]$ and historical $[\boldsymbol{h}_1, ..., \boldsymbol{h}_T]$. In the following text, we first explain who the proposed WiPin system extracts informative features from the raw observations, and then show how it reliably infers the latent states, i.e. the real occupancy with the extracted features.

Feature Functions: WiPin considers the following feature functions. $f_1(\boldsymbol{s}_t, \boldsymbol{z}_t)$ and $f_2(\boldsymbol{s}_t, \boldsymbol{h}_t)$ checks how the current observed occupancy $\boldsymbol{z}_t$ and the corresponding historical occupancy $\boldsymbol{h}_t$ support the latent state $\boldsymbol{s}_t$. On the other hand, feature functions $f_3(\boldsymbol{s}_t, \mu([\boldsymbol{z}_{t-w}, ..., \boldsymbol{z}_{t+w}])$ and $f_4(\boldsymbol{s}_t, \sigma([\boldsymbol{z}_{t-w}, ..., \boldsymbol{z}_{t+w}])$ evaluates the comparability between the state $\boldsymbol{s}_t$ and a window of $2w$ observations $[\boldsymbol{z}_{t-w}, ..., \boldsymbol{z}_{t+w}]$, where $\mu(\cdot)$ and $\sigma(\cdot)$ return the mean and standard deviation of the measurements, respectively. Finally, function $f_5(\boldsymbol{s}_{t-1}, \boldsymbol{s}_t, [\boldsymbol{z}_{t-w}, ..., \boldsymbol{z}_{t+w}])$ models the correlation between the observed occupancy during $[t-w, t+w]$ and the state transitions, which is defined as:

$$f_5 = ln\frac{1}{1+\sigma\sqrt{2\pi}} - \frac{(\boldsymbol{z}_t - \boldsymbol{z}_{t-1})(\boldsymbol{s}_t - \boldsymbol{s}_{t-1})}{1+2\sigma^2} \quad (7)$$

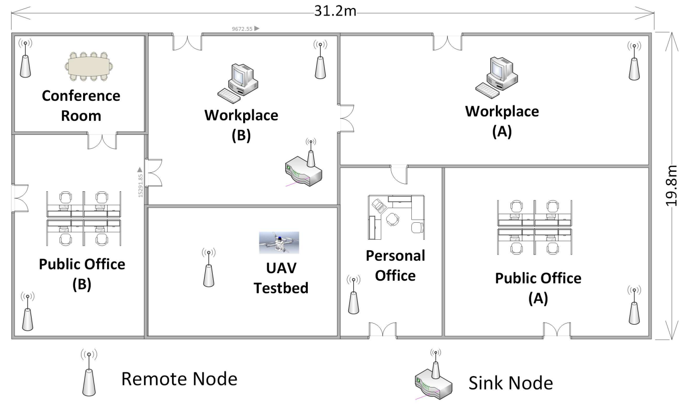

Figure 4: Testbed of the proposed WiPin System.

where σ computes the standard deviation of the measurements.

State Inference: Given the feature functions, the proposed WiPin system considers a state inference approach based on Conditional Random Fields (CRFs) [16]. In practice, CRFs are discriminative models, which directly capture the conditional dependencies between states and observations. Such correlation can be factored as the product of potentials:

$$p(\boldsymbol{s}_{1:T}|\boldsymbol{z}_{1:T}, \boldsymbol{h}_{1:T}) = c^{-1} \cdot \prod_{t=2}^{T} \Phi(\boldsymbol{s}_{t-1}, \boldsymbol{s}_t, \boldsymbol{z}_{1:T}, \boldsymbol{h}_{1:T}) \quad (8)$$

where c is a normalising constant. The potentials Φ is the log-linear combination of the above defined feature functions $\boldsymbol{f} = [f_1, ..., f_5]$, and $\Phi = \exp\{\boldsymbol{w} \cdot \boldsymbol{f}\}$. $\boldsymbol{w}$ is the relative weights between feature functions, and is learned from the data. Finally, our system uses Viterbi decoding to compute the most likely state sequence given the observations and historical data, which is:

$$\boldsymbol{s}^*_{1:T} = \text{argmax}\, p(\boldsymbol{s}_{1:T}|\boldsymbol{z}_{1:T}, \boldsymbol{h}_{1:T}) \quad (9)$$

V. Evaluations

A. Experimental Setup

Experiment Site: We built a 600 m^2 testbed in an office environment, which can be partitioned into 7 zones, as shown in Fig. 4. In the testbed, we deploy 8 commercial off-the-shelf WiFi routers, which forms a star network with the sink node placed in the centre.

Data Acquisition: We recruited 25 volunteers to participate our 6-weeks experiments (there are in total 42 occupants working in the testbed). During this period, the mobile devices of all participants were configured to use WiPin to connect to the Internet. To understand the real occupancy, we considered three different modalities: a) surveillance video clips from the CCTV cameras; b) manual occupancy logging; and c) passive infra-red (PIR) sensors fitted on desks. Ground truth is evaluated by majority voting [21] over data collected from these modalities.

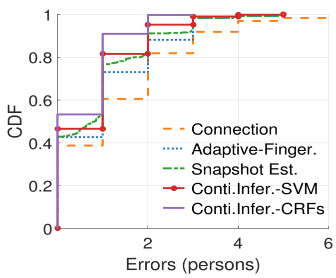

Figure 5: Overall performance comparison.

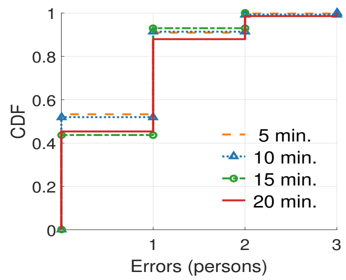

Figure 6: Impact of query intervals.

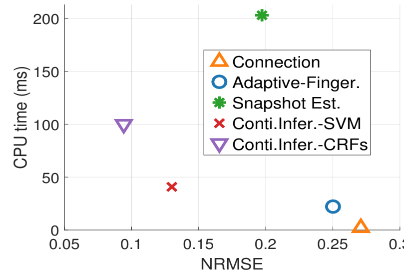

Figure 7: Computational overhead of different approaches.

B. Competing Approaches

Snapshot Estimation: For snapshot estimation, we compare it with two competing approaches: *Connection based-* The WiFi connection based approach is similar to the one adopted in [4], in which WiFi connection logs of APs are used to estimate occupancy levels. *Adaptive-fingerprinting based-* This method can be seen an enhanced version of the WiFi fingerprinting method [15], in which they offline calibrated WiFi fingerprints as references for online matching. To mitigate the effects of environmental dynamics, we use real-time RSS index of anchors as dynamic fingerprints.

Continuous Inference: To validate the superiority of CRFs in continuous inference, we compare it with a *SVM* [20]. Considering the inference is a multi-class problem while basic SVMs only give binary solutions, we employ an one-against-all method [14] to fuse multiple SVMs binary decisions to a multi-class result.

C. Evaluation Methods

Parameter Learning: To train the CRFs and SVM, we partitioned the 30 days of RSS measurements and occupancy data into training and testing sets. 5-fold cross validation is performed on the training data spanning a period of 25 days, randomly picked out from the permutated dataset. Hyper-parameters of WiPin, e.g., thresholds of Algo. 1 regularization weights of classifiers, are derived from grid searching on the basis of validation accuracy.

Performance Metrics: In order to comprehensively quantify the accuracy of WiPin, 5 metrics are adopted in total. 1) *NRMSE*: we use the normalized root mean squared error (NRMSE) [6] to measure the overall accuracy. Given the time series of estimated zonal occupancy $\mathbf{s}_{1:T}$ and ground truth data $\mathbf{g}_{1:T}$, the NRMSE between $\mathbf{s}_{1:T}$ and $\mathbf{g}_{1:T}$ is defined as

$$NRMSE(\mathbf{g}_{1:T}, \mathbf{s_{1:T}}) = \frac{\|\mathbf{g}_{1:T} - \mathbf{s}_{1:T}\|/\sqrt{T}}{max(\mathbf{g}_{1:T}) - min(\mathbf{g}_{1:T})}$$

where $max(\cdot)$ and $min(\cdot)$ takes the max/minimum value of the time series. Same as [5], we study the 2) *mean*, 3) *standard deviation* and 4) *95% percentile* of ℓ_1 errors respectively. We propose a new metric 5) *worst-case ratio* that quantifies the system robustness. The worst case of occupancy inference is when a zone is occupied by at least 1 people but the system estimates empty occupancy. The proportion of such errors in instances is referred as worst-case ratio, which is crucial for real-world implementations.

D. Experiment Results

Front-end: Tab. I summarizes the accuracy of different front-end implementations. It's noteworthy that snapshot estimation outperforms its two competing opponents by 37% and 27% respectively in terms of NRMSE. Meanwhile, its robustness is justified by 95% percentile error and worst-case ratio. Moreover, as shown in Fig. 5, the worst (max) error of the snapshot estimation approach is only 4.66, which is also lower than Connection and Adaptive-fingerprint approaches (5 and 6 respectively). It indicates that by re-labeling outliers, WiPin is able to improve tentative estimations. Note that WiPin achieves this directly on the front-end. The Connection based approach is suffering form the AP handover problem that phones may stick their connections to a router till it loses signal reception totally. Therefore, errors will be introduced since the connected router for a device is not necessary the nearest one. In addition, RSS is a more agile indicator for device mobility, hence the Adaptive-fingerprint based approach performs better than the Connection one (see Fig. 5).

Back-end: SVM and CRFs further reduced the errors of best front-end estimations by 34% and 52% (see Fig. 5 and Tab. I). Notably, the exact classification rate is not increased much by CRFs and SVM, however they restrict deviations thereby to enhance accuracy and robustness compared with snapshot estimation. It indicates that continuous inference is a more desirable way to interpret occupancy dynamics. Beyond this, we also found that CRFs are more suitable for continuous inference in WiPin. SVM converges within 3 persons while CRFs converges within 2 persons for 95% of cases. Except for worst-case ratio in which SVM is slightly better, CRFs comprehensively outperform SVM in terms of all other metrics. Shown in Fig. 8, SVM roughly captured the trends of occupancy transitions. However its predictions change abruptly sometimes and may give false alarms even

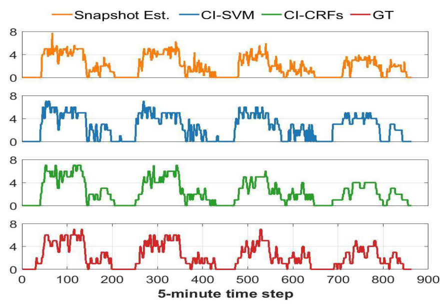

Figure 8: Four-day ground truth (GT) v.s. various predictions of Public Office (B).

Metric	Connection based	Adaptive-Finger.	Snapshot Estimation	Conti. Infer. -SVM	Conti. Infer. -CRFs
NRMSE	0.271	0.250	0.197	0.130	0.094
Mean ℓ_1 error total	1.615	1.381	1.055	0.775	0.560
Std. dev. ℓ_1 error	1.412	1.211	0.998	0.894	0.661
95% percen. ℓ_1 error	4	4	3	3	2
Worst-case ratio	1.07	9.5	7.8	4.2	4.8

Table I: Specific Performance Comparison of all metrics.

in obviously unoccupied periods. This is because SVM does not exploit the intrinsic structure among outputs but predicts on separate samples. In contrast, CRFs are able to provide smooth predictions and identify some critical periods, e.g., work hours and off-duty hours. Although such smoothness in CRFs occasionally results in skips of sharp transitions from the ground truth data. It seems like that CRFs act like another filter posed on the SVM's results by considering the structure in the sequence.

Impact of query interval: WiPin is designed to provide system subscribers with accurate occupancy levels when they have different query intervals. To examine this, we downsample measurements of interval sizes from 5 minutes to 20 minutes and investigate its performances. As shown in Fig. 6, the accuracy of continuous inference decreases slightly with the increase of interval length. This is because temporal correlations between occupancy samples decay with the increase of delay length, which blurred the chain structure in outputs. However, when analyzing the error distributions, we found that though the exact classification (0 error cases) rates fluctuate, most errors quickly converge to 1 person for all the interval lengths. As a result, the degree of declining brought by longer delay lengths is limited, and WiPin is still able to infer relatively accurate results. Considering that commonly query intervals of occupancy systems are below 20 minutes [5], WiPin is able to accommodate subscribers with various needs in practice.

Computational Cost: We evaluate the overheads of different methods in WiPin. Specifically, we measure the CPU time when the number of mobile devices is of around 80% capacity in each monitored zone. The result in average is shown in Fig. 7. For front-end occupancy estimation implemented on the router, connection based and adaptive fingerprints are superior to snapshot estimation in terms of computational efficiency. However, the overhead of snapshot estimation is far below the 1-second updating interval which incur very limited delay in processing. When it comes to continuous inference, CRFs outperforms SVM by $\sim 50\%$. The computation bottleneck of SVM lies in that it has to use multiple binary SVMs to give multi-class predictions, while CRFs infer the state directly with Viterbi algorithm.

VI. Related Work

Occupancy Sensing A major distinction among occupancy inference systems lies in different sensors adopted in the system. Popular sensors for occupancy detection include PIR motion sensors [1], magnetic reed switches [17] and CO_2 sensors [5]. Nevertheless, due to the sensor intrinsic limitation, it's difficult for them to detect occupants when they are relatively motionless. To address this problem, POEM [8] combined the ceiling mounted camera and motion sensors to largely improve the occupancy detection accuracy (94%) in a power-efficient way. In the context of radio based sensing, Sentinel [4] leverages the requests received by the RADIUS server and is able to output near real-time occupancy. Sentinel takes the initiative to leverage the existing WiFi infrastructure and requires minimal calibration. However, such WiFi connection based systems suffer from AP handover problem especially in personal networks. Ariel [15] requires intrusive app installation, and its accuracy is affected by environmental dynamics.

In contrast, the proposed system makes use of the existing WiFi infrastructure and sniffs the RSS packets in a non-intrusive way. No applications are required for usage. Moreover, WiPin is self-contained and cost-efficient to be accessed by various occupancy information subscribers.

State Inference Model based inference techniques are well established for occupancy prediction [10, 9] and most of them are developed on the basis of Markov chain theory. The states to be inferred are vectors in which each component represents the number of occupants in each zone. However, these methods solely depend on historical information, which blinds the system to solve the issue incurred by deviations of online occupancy from historical calibration.

WiPin hybridizes real-time measurements and offline calibrations thereby better efficacy is achieved. Moreover, we deeply investigate the patterns among measurements and adopt utility theory to boost system robustness to outliers. To our best knowledge, it's the first time that an occupancy inference system like WiPin integrates risk-sharing concepts.

Fingerprinting-based Indoor Positioning Mobile device based wireless localization approaches typically calibrate and build radio map offline and estimate locations by online fingerprint matching [3, 22]. Not only the fingerprints survey is labor-intensive [12] but the developed radio map gradually becomes obsolete along with environment changing [18]. However, WiPin does not require fingerprinting calibration process at all. The adaptiveness of anchors makes WiPin free of war-driving site surveys and re-calibrations.

VII. Conclusions

In this paper, we design WiPin, a robust occupancy inference system based on commercial WiFi hardware. The proposed system contains a front-end and a back-end, where the front-end runs directly on the WiFi routers, and estimates the snapshot occupancy of the indoor environment in real-time. On the other hand, when accuracy is the primary concern, our system is able to exploit the temporal correlations between the estimated snapshot occupancy, and incorporate historical data to improve inference accuracy, at the cost of certain fixed delay. We implemented and tested the proposed WiPin system on a real-world testbed, and experimental results show that: a) The front-end of our system outperforms the existing approaches significantly, by reducing the inference error up to 37%; b) when the back-end is activated, it is able to further double the inference accuracy; c) our system is robust to environmental changes and works well under different parameter settings, without incurring much computational overhead; d) in certain cases, the proposed system is accurate enough to infer room-level positions of individual occupants. For future work, we plan to improve WiPin, so it is able to cope with certain extreme cases, e.g. when the first occupant arrives without her phone.

Acknowledgments. This work was supported in part by the EPSRC through grants EP/J012017/1 and EP/L00416X/1; in part by the Republic of Singapore National Research Foundation (NRF) through a grant to the Berkeley Education Alliance for Research in Singapore (BEARS) for the Singapore-Berkeley Building Efficiency and Sustainability in the Tropics (SinBerBEST) Program; in part by the Republic of Singapore NRF under Grant NRF2013EWT-EIRP04-012 and Grant NRF2011NRF-CRP001-090.

References

[1] Y. Agarwal, B. Balaji, S. Dutta, R. K. Gupta, and T. Weng. Duty-cycling buildings aggressively: The next frontier in hvac control. In *IEEE IPSN*, 2011.

[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] P. Bahl and V. N. Padmanabhan. RADAR: An in-building RF-based user location and tracking system. In *IEEE INFOCOM*, 2000.

[4] B. Balaji, J. Xu, A. Nwokafor, R. Gupta, and Y. Agarwal. Sentinel: occupancy based HVAC actuation using existing WiFi infrastructure within commercial buildings. In *ACM SenSys*, 2013.

[5] C. Basu, C. Koehler, K. Das, and A. K. Dey. PerCCS: person-count from carbon dioxide using sparse non-negative matrix factorization. In *ACM UbiComp*, 2015.

[6] A. Beltran, V. L. Erickson, and A. E. Cerpa. Thermosense: Occupancy thermal based sensing for hvac control. In *ACM BuildSys*, 2013.

[7] R. Clark, W. Cronje, and M. A. v. Wyk. Design optimization of a hybrid energy system through fast convex programming. In *International Conference on Intelligent Systems, Modelling and Simulation*, 2014.

[8] V. L. Erickson, S. Achleitner, and A. E. Cerpa. POEM: Power-efficient occupancy-based energy management system. In *IEEE IPSN*, 2013.

[9] V. L. Erickson, M. Á. Carreira-Perpiñán, and A. E. Cerpa. OBSERVE: Occupancy based system for efficient reduction of hvac energy. In *IEEE IPSN*, 2011.

[10] V. L. Erickson, Y. Lin, A. Kamthe, R. Brahme, A. Surana, A. E. Cerpa, M. D. Sohn, and S. Narayanan. Energy efficient building environment control strategies using real-time occupancy measurements. In *ACM BuildSys*, 2009.

[11] X. Gao and S. Keshav. Optimal personal comfort management using SPOT+. In *ACM BuildSys*, 2013.

[12] S. He, S.-H. G. Chan, L. Yu, and N. Liu. Calibration-free fusion of step counter and wireless fingerprints for indoor localization. In *ACM UbiComp*, 2015.

[13] E. Horvitz, P. Koch, C. M. Kadie, and A. Jacobs. Coordinate: Probabilistic forecasting of presence and availability. In *Proceedings of the Eighteenth conference on Uncertainty in Artificial Intelligence (UAI)*, 2002.

[14] C.-W. Hsu and C.-J. Lin. A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks*, 13(2):415–425, 2002.

[15] Y. Jiang, X. Pan, K. Li, Q. Lv, R. P. Dick, M. Hannigan, and L. Shang. Ariel: Automatic wi-fi based room fingerprinting for indoor localization. In *ACM UbiComp*, 2012.

[16] J. Lafferty, A. McCallum, and F. C. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, 2001.

[17] J. Lu, T. Sookoor, V. Srinivasan, G. Gao, B. Holben, J. Stankovic, E. Field, and K. Whitehouse. The smart thermostat: using occupancy sensors to save energy in homes. In *ACM SenSys*, 2010.

[18] X. Lu, C. Yu, H. Zou, H. Jiang, and L. Xie. Extreme learning machine with dead zone and its application to WiFi based indoor positioning. In *IEEE ICARCV*, 2014.

[19] X. Lu, H. Zou, H. Zhou, L. Xie, and G.-B. Huang. Robust extreme learning machine with its application to indoor positioning. *IEEE Transactions on Cybernetics*, 46(1), 2016.

[20] B. Scholkopf and A. J. Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2001.

[21] H. Wen, Z. Xiao, A. Markham, and N. Trigoni. Accuracy estimation for sensor systems. *IEEE Transaction on Mobile Computing*, 14(7):1330–1343, 2015.

[22] M. Youssef and A. Agrawala. The horus wlan location determination system. In *ACM MobiSys*, 2005.

[23] H. Zou, B. Huang, X. Lu, H. Jiang, and L. Xie. A robust indoor positioning system based on the procrustes analysis and weighted extreme learning machine. *IEEE Transactions on Wireless Communications*, 15(2):1252–1266, 2016.