Improving Infrastructure-based Indoor Positioning Systems with Device Motion Detection

Huy Tran*†, Abhishek Mukherji†, Nirupama Bulusu*, Santosh Pandey†, Xu Zhang†
*Portland State University, †Cisco Systems
{hptran, nbulusu}@pdx.edu, {abhishmu, sanpande, xuzha2}@cisco.com

Abstract—**Infrastructure-based Indoor Positioning Systems (IIPS) have emerged as critical components of wireless deployments for many enterprises so as to track mobile devices without additional device-side applications or computation. To provide transformative location-based services, it is important that IIPS compute accurate locations over a long period of time, scaling with a large number of tracked devices cost-effectively. In this paper, we present MotionScanner, which includes novel feature-based and end-to-end deep learning motion detection models to detect device motion solely from noisy, temporally sparse, and partial Wi-Fi measurements at access points. We further integrate MotionScanner into an IIPS so that the IIPS can exploit previously computed locations effectively to improve location accuracy, and skip unnecessary location computation. Building on observations of how location estimates of stationary devices scatter over time, we can monitor and enhance the performance of IIPS. We evaluate MotionScanner with data sets collected from real-world deployments of IIPS at two enterprises, and show that MotionScanner achieves 83% motion detection accuracy while saving 80% of computational resources.**

I. Introduction

Infrastructure-based Indoor Positioning Systems (IIPS) leverage the existing wireless LAN (WLAN) infrastructure within enterprise buildings. They aim to provide accurate device tracking to improve enterprise operation at ease and low-cost [8], [12], [22]. IIPS servers localize and track devices such as phones and Wi-Fi tags using only Wi-Fi measurements reported from access points (APs) and not requiring applications running on the tracked devices [8], [20], [34]. However, tracking thousands of devices accurately over time in multipath indoor environments without a significant increase in the cost of IIPS deployment is extremely challenging. This work focuses on detecting and leveraging device motion context (stationary or moving) from the infrastructure side to achieve essential requirements of IIPS deployment.

A. Requirements of IIPS Deployment

Below, we describe the three main requirements of IIPS deployment and explain how device motion context can be leveraged to achieve these requirements.

1. Location accuracy: IIPS need to localize and track a device accurately. Most prior work focuses on improving location accuracy by using only the latest Wi-Fi measurements [20], [37] without considering device motion. When it is known that a device is stationary, more historical measurements or location estimates can be exploited to improve the current location estimate.

2. Scalability: IIPS need to track a large number of devices in real time without reducing location accuracy or increasing the cost significantly. Though extra computational resources could be allocated dynamically by using a cloud computing service, when we deployed the location server on Amazon Web Service, the cost of running a minimal cluster, databases, storage, and bandwidth was high (about \$8,000 per month for tracking $10,000$ devices every four seconds on average). We observe that within many enterprise buildings, people and devices are often stationary for long durations. For example, in office spaces, people are typically stationary 75% of the time [23]. Moreover, Wi-Fi tags are often stationary for long periods of time. Thus, IIPS can scale cost-effectively by not computing locations of stationary devices repeatedly.

3. Location-accuracy monitoring: Monitoring location accuracy of IIPS over space (floor areas) and time is essential in understanding and improving IIPS performance. Traditionally, site-surveys are performed periodically to check how location accuracy of IIPS change. However, performing site-surveys at many enterprise buildings requires extensive human efforts. We often observe that multiple devices are stationary for long periods. Given location estimates of these devices, IIPS can monitor how these estimates scatter over time.

B. MotionScanner: Infrastructure-Side Motion Detection

This work addresses the problem of detecting device motion at the infrastructure side using only Wi-Fi measurements received by the IIPS. Exploiting device sensor data [15], [26], [38] can further improve motion detection accuracy. However, this approach does not fit well with the IIPS design as it requires (i) all tracked devices running motion detection applications (ii) network protocols to support collecting device data (iii) explicit permission of device users for storing their data (iv) correctness and trustworthiness of motion data sent from devices. This work focuses on investigating to what extent detecting device motion using only the Wi-Fi measurements can improve IIPS performance.

We present **MotionScanner**, which enables motion-aware IIPS by using only Wi-Fi measurements (RSSIs and phase vectors) received by the IIPS, and addresses three main challenges: temporally sparse and non-periodic measurements, noisy measurements and missing measurements (Section IV).

C. Contributions

We summarize our contributions below.

• We develop feature-based and deep learning-based models that exploit temporal patterns of measurements from multiple APs to detect device motion accurately in real time (Section V). We focus on the generalizability and simplicity of our models. Our models consist of three main steps: feature extraction for extracting temporal patterns of each AP's measurements, feature aggregation for aggregating temporal features across multiple APs, and modeling for learning the relationship between the features and device motion. The main novelty of our models is the method of computing and aggregating the features, especially phase correlations, effectively regarding temporally sparse, multipath, and missing measurements.

• We evaluate our methods by using dataset collected from real-world deployments of IIPS at two different enterprise settings: a retail and a cafeteria (Section VI). Our results showed that our methods achieve 83% motion-detection accuracy on average by using both features extracted from RSSIs and phase vectors. We also showed that RSSI features proposed in prior work [19], [21], [25] achieve at most 75% accuracy on average with our dataset. Moreover, our feature-based models trained by using data collected in one building can be applied for another building with negligible accuracy reduction. Finally, our deep-learning model exploiting only temporal patterns of RSSI measurements can achieve higher motion detection accuracy compared to the prior work [19], [21], [25].

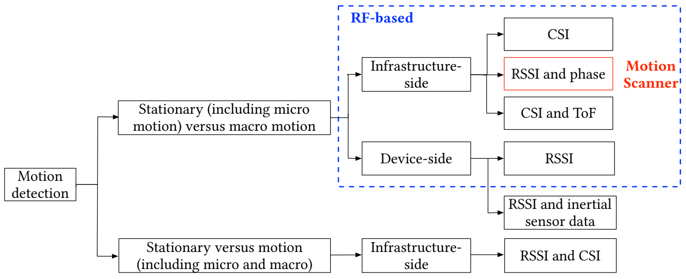
• We show that MotionScanner improves IIPS performance in terms of location accuracy and scalability, as well as enabling location-accuracy monitoring (Section VII). In particular, MotionScanner reduces the number of location computations by as much as 80% without any impact on location accuracy.

II. Related Work

We describe various motion types, categorize the motion detection literature based on these motion types, and highlight where MotionScanner fits in the overall literature (see Figure 1). According to Sun et al. [28], there are three types of object motion: stationary, micro motion, and macro motion. Stationary implies an object is completely static. Micro motion implies the object only moves a small distance (less than one meter) or part of the object (human hand) moves. Macro motion implies the object moves more than one meter. Given these object motion definitions, there are two main categories of motion detection problems that prior work has focused on: (i) stationary (including micro motion) versus macro motion, and (ii) stationary versus motion (including micro and macro). Our work MotionScanner belongs to the first category.

A. *Stationary (including micro motion) versus macro motion*

In this category, motion detection is performed either at the infrastructure side or at the device side (Figure 1). MotionScanner is performed at the infrastructure side by using RF data that the IIPS measure in a best effort manner (Section III). Therefore, IIPS sample Wi-Fi measurements at a non-periodic and low rate (about 0.2 Hz for RSSIs and 0.15 Hz for phase vectors). This is one of the unique challenges in detecting device motion with IIPS measurements (Section IV).


RF-based
Motion detection
Stationary (including micro motion) versus macro motion
Infrastructure-side
Device-side
CSI
RSSI and phase
Motion Scanner
CSI and ToF
RSSI
RSSI and inertial sensor data
Stationary versus motion (including micro and macro)
Infrastructure-side
RSSI and CSI
Fig. 1: Related work on motion detection

Prior infrastructure-side approaches [28], [35] detect device motion by requiring tracked devices or transmitters to send RF signal periodically at 5Hz and 50Hz, respectively. Moreover, [28] needs to aggregate many measurements in 4 seconds to detect device motion with 98% accuracy. [35] achieves 95% accuracy but assumes there is only one tracked device in an environment. Both these approaches detect device motion using measurements (channel state information or time of flight) reported from a single receiver deployed in test beds (office spaces). MotionScanner combines different temporal correlations of measurements (RSSIs and phase vectors) reported from multiple commercial-off-the-shelf (COTS) APs to detect device motion in real time.

Prior device-side approaches detect device motion using data (RSSIs or inertial sensor data) collected from applications running on a device. Prior work [19], [21] samples RSSI measurements periodically at 0.4 Hz and 3 Hz, respectively. Prior work proposed several RSSI features in both frequency domain and time domain for detecting device motion. Extracting features in the frequency domain requires periodic measurements, while measurements from IIPS are non-periodic. We exploit features in the time domain and show that using these features achieves much lower accuracy in our dataset (Section VI). Prior work [21] also assumes a fixed frequency of a person's movement in a test bed. Without this assumption, and with a low and non-periodic sampling rate, MotionScanner exploits not only a combination of RSSI features but also a phase feature to detect device motion accurately in real-world deployments of the IIPS. Moreover, as we discussed in Section I-B, requiring applications running on every tracked device does not fit well with IIPS design.

B. *Stationary versus motion (including micro and macro)*

Recent non-invasive (device-free) solutions have focused on detecting either micro motion or macro motion of tracked objects for health care applications such as in-home elderly or child monitoring and gesture recognition [17], [18], [33], [36]. Custom hardware [17], [18] or pairs of a transmitter and receiver [33], [36] are required to analyze RF measurement changes caused by motion. These solutions are limited to small, static environments having a single or very few tracked objects (less than 6) [18]. Also, they do not distinguish between micro motion and macro motion.

III. Background

In this work, we use COTS APs [3], [11] that have been deployed at many enterprise buildings. Figure 2 shows that the AP has 4 serving antennas (for transmitting or receiving signals from Wi-Fi devices) and 32 quasi-circular-array antennas. To localize a device and detect the device motion, we use two types of Wi-Fi measurements: received signal strength indicators (RSSIs) measured at the serving antennas and an angle-of-arrival (AoA) phase vector (consisting of phase values) measured at the antenna array. The phase vector is computed from channel state information (CSI) [1] measured at the physical layer of the AP [14], [37]. We emphasize that our motion detection approach can be applied for other COTS APs [9] that have a different number of antennas.

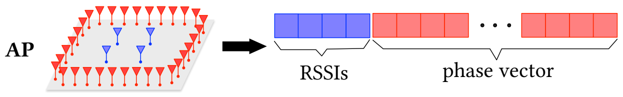


Fig. 2: Measurements at an AP [8]

Figure 3 illustrates an architecture of IIPS on top of an enterprise WLAN infrastructure [4]. Each AP measures Wi-Fi signals emitted from tracked devices and forwards the measurements to a WLAN controller [2], [6]. The controller aggregates and then forwards the measurements received from multiple APs to a location server (LS) deployed on-site or on-cloud. Based on the measurements, the LS uses a combination of RSSI-based trilateration and phase-based AoA method to localize and track the devices [14], [37]. The method achieves median localization accuracy ranging from 1 m to 3 m for real-world deployments in many enterprise buildings (including retail, airports, and workplaces) having about one AP per 15 m x 15 m (which is required to guarantee good Wi-Fi coverage [7]). The method also achieves sub-meter accuracy for areas having line-of-sight between the device and several APs.

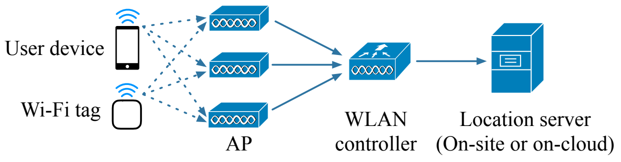


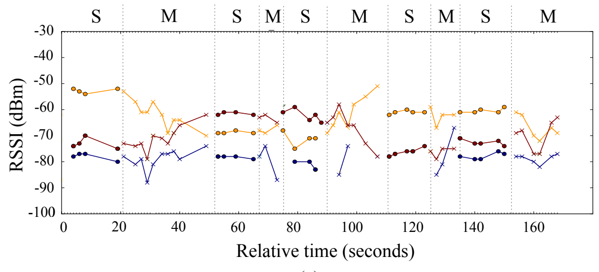
Fig. 3: Illustration of IIPS [4] on top of a WLAN

IIPS samples Wi-Fi measurements in a best effort manner without requiring any specific applications installed on a mobile device. For RSSI measurements, a group of APs in the vicinity of a device measures the RSSI of the signal emitted from the device whenever the device transmits a packet. Such a packet can be a probe request for APs [32], an uplink data packet, or a response to a request from the AP when the device's radio is active [3], [5]. For AoA phase vector measurements, a group of APs measures AoA of signals emitted from a device when a master AP exchanges packets with the device. The master AP selects each of its associated devices sequentially, and each AP in the group is also selected as the master AP in a round robin way [10]. However, for both RSSI and AoA measurements, because of unpredictable interference, the CSMA nature of Wi-Fi networks as well as uncontrolled device behavior, the IIPS can not guarantee that any measurements are conducted at scheduled time instants. Neither can it guarantee that all packet exchanges (e.g., between a device and a master AP) are successful. As a result, we will observe uncertain delay between consecutive Wi-Fi measurements of the same device which we called *temporally sparse and non-periodic measurements*, as well as partial measurements which we also name *missing measurements*.

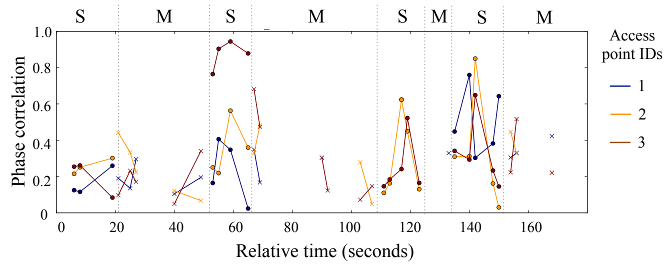
IV. Challenges

There are three main challenges that we address in detecting device motion by using Wi-Fi measurements reported from the infrastructure side.

- *Temporally sparse and non-periodic measurements*: As discussed in Section III, IIPS samples Wi-Fi measurements in a best effort manner. Therefore, measurements are temporally sparse and non-periodic. In our dataset (Figure 9d), the mean sampling rate of RSSI measurements is about one sample per 5 seconds with a standard deviation (SD) of 4 seconds. The average sampling period of phase measurements is about one sample per 6 seconds with an SD of 6 seconds. Figure 5 illustrates how measurements reported by APs in the vicinity of a device over time and the histogram of the measurements. Given the temporally sparse and non-periodic measurements reported by APs, it is challenging to detect device motion at a time step, especially when the device stays or moves in short time intervals.
- *Noisy measurements*: In an indoor environment, Wi-Fi measurements at an AP are often noisy due to unpredictable signal attenuation and multipath signal propagation. For RSSI measurements, Figure 4a shows RSSI measurements vary when a phone is moving. However, there are episodes in which the device is stationary but RSSI measurements fluctuate at a similar level compared to when the device is moving. For phase measurements, Figure 4b shows that the correlations of consecutive phase vectors are relatively higher when the device is stationary but also fluctuate significantly. Therefore, it is challenging to detect the device's motion accurately given the noisy measurements.
- *Missing measurements*: We observe three main cases in which measurements at an AP are missing. First, when a device stays far from the AP or is moving, we often observe intermittent measurements at the AP over time [19]. Figure 4 represents missing measurements over time as white gaps between markers in each motion (stationary or moving) episode. Second, the device may not respond to all request packets from the AP during the phase measurement process (as described in Section III). Figure 5b shows that about 25% of phase vectors having missing phase values. Third, the device can switch between different frequency bands (2.4GHz and 5.0GHz) when sending Wi-Fi signals. The switching frequency is device dependent.

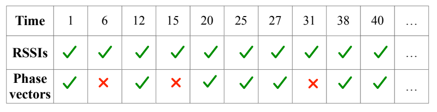


(a)



(b)

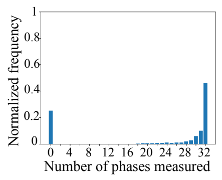
Fig. 4: (a) Measured RSSIs and (b) computed phase correlations at three APs deployed in a cafeteria (shown in Figure 9a) when a phone is carried around by a user alternating between stationary (S) and moving (M).



Time	1	6	12	15	20	25	27	31	38	40	...
RSSIs	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	...
Phase vectors	✓	x	✓	x	✓	✓	✓	x	✓	✓	...

Measurements reported by APs over time (x represents no measurements)

(a)



(b)

Fig. 5: (a) Measurements reported by APs in the vicinity of a device over time. (b) Histogram of the average number of measured phase values per AP. About 25% of the time, APs report only RSSIs. About 75% of the time, the APs report both RSSIs and phase vectors. Within that, about 50% of the phase vectors have full 32 phase values measured at the 32 circular-array antennas of each AP.

V. Device Motion Detection

We first formalize the motion detection problem. Then, we describe our approaches for detecting the device motion.

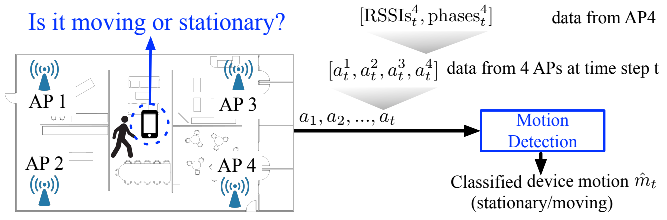
A. Problem statement

Table I introduces the notation we use to define the motion detection problem. Figure 6 depicts the problem.

Symbol	Description
time_t	Timestamp of time step t
RSSIA_t^n	Signal strength measured in 5GHz at an AP n at time_t
RSSIB_t^n	Signal strength measured in 2.4GHz at an AP n at time_t
RSSIs_t^n	$[\text{RSSIA}_t^n, \text{RSSIB}_t^n]$
$\text{phase}_t^{n,p}$	Phase value measured at antenna p at AP n at time_t
phases_t^n	Phase vector measured at an AP n having P antennas $[phase_t^{n,1}, ..., phase_t^{n,P}]$
a_t^n	Data measured at an AP n at time_t $[\text{RSSIs}_t^n, \text{phases}_t^n]$
a_t	Data measured at N APs deployed on a floor at time_t $[a_t^1, a_t^2, ..., a_t^N]$
m_t	Device motion at time step t $m_t = 0$: the device is stationary (negative) $m_t = 1$: the device is moving (positive)

TABLE I: Notations

Online device motion detection. Given the time series $a_1, ..., a_t$ of Wi-Fi measurements per device reported from a set of access points within a floor until the current time t, an IIPS needs to detect (classify) the device motion m_t at the time t as stationary (including micro motion) or moving i.e., macro motion. The definitions of motion types are in the first paragraph of Section II.



Is it moving or stationary?
$[\text{RSSIs}_t^4, \text{phases}_t^4]$
data from AP4
$[a_t^1, a_t^2, a_t^3, a_t^4]$ data from 4 APs at time step t
AP 1
AP 2
AP 3
AP 4
$a_1, a_2, ..., a_t$
Motion Detection
Classified device motion $\hat{m}_t$
(stationary/moving)

Fig. 6: Motion detection problem

B. Feature-based Motion Classifier

The feature-based approach consists of three steps: feature extraction, feature aggregation, and modeling. The feature extraction step extracts temporal features from the measurements reported from each AP until time_t such that each feature correlates with device motion at time_t. To reduce the effect of noisy measurements and missing measurements on the correlation, different temporal features extracted from measurements at multiple APs are aggregated to generate robust spatial-temporal features. Finally, the modeling step trains models to learn the relationship between the features and the device motion. Below, we describe these steps in detail.

1) Feature Extraction: At the current time step t, the feature extraction step extracts temporal features from the Wi-Fi measurements reported from each AP in a time window from $t - w$ until t. The window size parameter w is derived from a training dataset to achieve optimal classification accuracy.

• *RSSI features:* We experimented with a variety of RSSI features. In Table II, we list the RSSI features that contribute to improving the motion detection accuracy together with our key observations for each feature. Other tested features in time domain include Tanimoto distance, Spearman's correlation, and Euclidean distance over RSSIs. However, those features do not contribute significantly to improving the accuracy of motion detection.

• *Phase correlation feature:* Phase correlation represents the similarity between two phase vectors measured by an AP n at two different time steps. If there is a strong correlation between the phase vectors at the current time step t and the previous time step $t - 1$, a device is likely to be stationary. However, as mentioned in Section IV, we often have missing phase vectors due to the lower sampling of the phase vectors compared to RSSIs. Therefore, instead of considering only the phase vector at $t - 1$, we consider the phase vector at time

RSSI features	Description
Visibility of AP	The ratio between the number of measured RSSIs and the window size w [19]. *We observe that the AP visibility reduces when a device is moving.*
Consecutive RSSIs	The RSSIs at time steps $t-1$ and t, which will be used to compute the correlation of RSSIs across access points in the feature aggregation step.
Consecutive RSSI difference	The absolute difference between two consecutive RSSI measurements at times $t-1$ and t. If an access point does not report RSSI at either of the times, it will not be used in the feature aggregation step.
SD of RSSIs	The standard deviation of RSSIs measured within the window w if the RSSI at t is available and the number of available RSSIs in the window is greater than 3.
SD of RSSI differences	By first computing the absolute differences between the single current RSSI with all previous RSSIs in the window w and then computing the standard deviation of those differences.

TABLE II: RSSI features

step t' in a window of w measurements ($t' \in [t-w, t-1]$). We define the phase correlation below.

$$\text{corr}(\text{phases}_t^n, \text{phases}_{t'}^n) = \frac{|\sum_p e^{2\pi j(\text{phases}_t^{n,p} - \text{phases}_{t'}^{n,p})}|}{c} \quad (1)$$

The phase correlation is computed by using only the pairs of phase values denoted as $\text{phases}_t^{n,p}$ and $\text{phases}_{t'}^{n,p}$ measured at the same antenna p ($p \in [1, 32]$) at the AP n at the time step t and t', respectively. Quite often the AP can not measure phase values at all antennas during the phase measurement process as discussed in Section IV. We normalize the phase correlation by dividing the numerator by c, which is the number of antennas that have phases measured at both time steps t and t'. To ensure the certainty of the phase correlation, we select the phase vector at the latest time step t' in the window such that c is at least 8.

2) Feature Aggregation: The feature aggregation step applies different aggregation operators on the temporal features extracted per AP to generate spatial-temporal features. We describe our aggregation operators below.

• *Average over RSSI feature:* For each RSSI feature except the *Consecutive RSSIs* feature, the operator computes the average of the feature values calculated for the APs having RSSI measurements. For the SD features, the operator considers only the feature values corresponding to the APs that have the highest visibility (*Visibility of AP*). In other words, it discards the feature values corresponding to the APs having many missing measurements.

• *Pearson correlation over consecutive RSSI:* This operator computes the correlation of *consecutive RSSIs* at multiple APs. When a device is stationary, changes of RSSIs are often similar at most APs, which corresponds to a high correlation value.

• *Max of phase correlations:* This operator computes the maximum of *phase correlation* values computed for multiple APs having phase measurements. We select this operator due to two reasons. First, phase vectors measured at an AP are very sensitive to device motion [18], [33]. Moving the device to another location affects the measured phase vectors significantly, which reduces the phase correlations computed at all APs. Second, when the device is stationary, the APs having line-of-sight (LOS) with respect to the device's location often have stable and high phase correlations. Figure 7a compares the CDF of phase correlations computed at LOS APs vs. non-LOS APs when the device is stationary at multiple locations. With the AP density (about 1 AP per 15 m x 15 m) in typical enterprise deployments [7], we expect at least one AP having LOS with respect to a device's location. Figure 7b shows the distribution of the max of phase correlations when a mobile device is moving versus stationary.

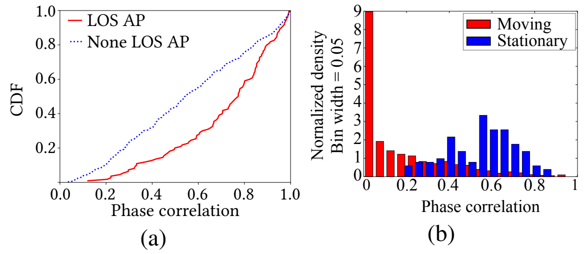


Fig. 7: (a) CDF of phase correlations at the APs having line of sight (LOS) and non-LOS with respect to device locations (b) Histogram of a max of phase correlations when a device is moving versus stationary

Before inputting the spatial-temporal features into models that we describe below, we scale all of the features to the range $[0, 1]$. To further address the challenge of missing measurements as described in Section IV, if a feature computed from measurements in band A (5 GHz) is missing, we replace it with the feature in band B (2.4 GHz) and vice verse. If the features are missing in both bands, we impute them with the average value of the feature in our training set (mean imputation method [13]).

3) Modeling: The modeling step trains different models to learn the relationship between spatial-temporal features and device motion. We considered different feature-based models: Recurrent Neural Network (RNN), Random Forest (RF), and Hidden Markov Model (HMM). For the HMM, we calculate an emission probability and transition probabilities based on the distribution of the aggregated *SD of RSSIs* feature and the distributions of the other aggregated features, respectively.

C. Deep Learning Approach: End-to-End RNN (E2E-RNN)

As shown in Fig. 8, we design a deep neural network E2E-RNN that consists of three components. The first component is an RNN of LSTM cells [16] to capture the temporal correlation of the Wi-Fi measurements at each AP. As described in Equation 2, the output vector s_t^n of the LSTM cell for the AP n at the time step t is a function of s_{t-1}^n which is the output from the cell in the previous time step at the AP n and a_t^n which is the Wi-Fi measurement at the time step t. The length of the output vector m can be optimized by performing a grid search by using a training set. To help the network taking into account the missing data, we impute unmeasured data values with zeros and add a binary indicator $mask_t^n$ into a_t^n to indicate if the measurement is completely missing [24].

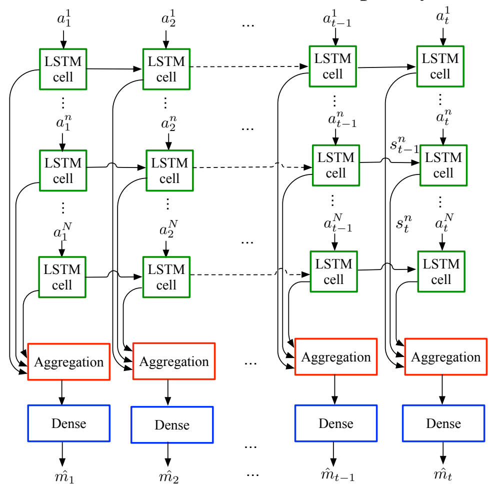


Fig. 8: E2E-RNN approach

$$s_t^n = f(s_{t-1}^n, a_t^n) \quad \text{where} \quad a_t^n = [\text{mask}_t^n, \text{RSSIs}_t^n, \text{phases}_t^n] \tag{2}$$

Given the outputs from N RNNs corresponding to N APs, the aggregation component aggregates the outputs $[s_t^1, ..., s_t^n, ..., s_t^N]$ from the RNNs at each time step. First, we force the network to ignore outputs where the corresponding measurement vector inputs are zero vectors by removing s_t^n if a_t^n is a zero vector for $n \in [1, N]$, and then take the average of the remaining output vectors in each dimension. The output of the aggregation function is a vector s_t of length m.

Given the s_t at the time step t, the dense component [29] outputs the vector $\hat{m}_t$ which consists of the probability of moving and stationary of the device at the time step.

We have described our neural network for classifying a device's motion m_t at a time step t. We emphasize that all parameters (weights) in the network are shared across different APs so that the number of parameters in the whole network is small. This approach allows us to train the network with a small training set. We use a gradient descent optimization method that minimizes the loss function that computes average cross-entropies of the classified motions in a time window [30]. We evaluate this approach in Section VI-C1.

VI. Evaluation

In this section, we describe how we evaluate our motion detection approach.

A. Goals and Metrics

We focus on answering the following questions:

1) How do our models classify device motion by using either RSSI measurements or the combination of RSSI measurements and phase vector measurements?
2) How do our models generalize across different device types and enterprise types?
3) What is the run-time overhead added by performing motion classification?
4) What is the tolerance of the IIPS to misclassification of motion classification?

Table III describes our evaluation metrics. These metrics are well-defined in literature, yet we describe them here in the context of device motion classification.

Metric	Description
Accuracy	Fraction of correctly classified samples
Precision	Fraction of samples classified as moving which are correct
Recall	Fraction of moving samples correctly classified
F1 score	Harmonic mean of precision and recall
False Negative Rate (Sensitivity)	Fraction of moving samples incorrectly classified as stationary

TABLE III: Description of metrics used in our evaluation

B. Data Collection

While a person is carrying a phone and walking on a floor, we record Wi-Fi measurements including RSSIs and phases at APs, the corresponding location estimates computed by IIPS, and the corresponding actual motion of the device over time. The device's actual motion at time$_t$ is determined by the actual distance that the person moved from time$_{t-1}$ to time$_t$. As we defined in Section V-A, if the distance is greater than 1 m, the device motion is moving. Otherwise, it is stationary.

To collect the person's actual locations every second, first, we use a mobile application to mark on the application's floor map the points corresponding to the locations where the person makes a turn or stops moving. Then, given any two consecutive marked points at t and $t + T$ as well as the time stamps associated with these points, we interpolate the person's actual locations every second from t, $t + 1$, ..., $t + T$. To ensure the accuracy of the marked points, we require the person to make a turn or to stay at the locations where there are visual landmarks or at intersections. Also, to ensure the accuracy of the interpolated points, we require the person to walk naturally (without changing his or her walking speed).

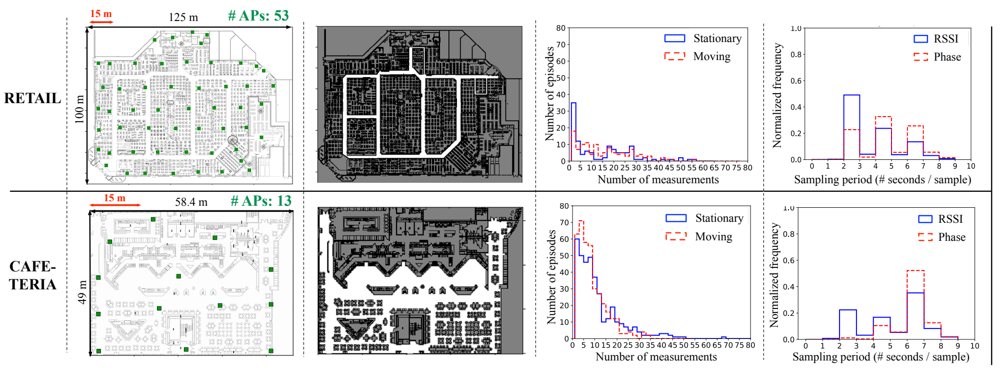
We performed the data collection during the business hours over multiple days at two different enterprise buildings: a retail store and a cafeteria. We describe our dataset in Fig. 9. We use the dataset for training and testing our motion detection models as well as illustrating device motion-based use cases.

C. Methodology, Results and Analysis

Below, we present our evaluation methodology, results, and our analysis for each of the goals described in Section VI-A.

1) Comparison of motion detection models: We compare the accuracy of different models using the same data set in each enterprise building. Given data collected over multiple experiments, we first randomly permute the experiments performed in each building. Then, we put 80% of the data into a training set for training the models and the remaining 20% of the data for testing the models. To find the best hyper-parameter values of each model, we hold out 10% of the training data to evaluate the accuracy of the model with a different combination of the hyper-parameter values.

Feature-based approach. Table IV shows the testing results of our models that use either RSSI features or the combination of RSSI features and the phase correlation to detect device motion. We summarize our analysis of the results below.



(a) AP placement (b) Participants' walking area (c) Histogram of number of measurements per motion episode (d) Histogram of sampling rates of RSSI and phase measurements

Enterprise buildings	RETAIL	CAFETERIA
Traced devices	iPhone 6s, OnePlus	iPhone 6s, iPhone 6s Plus, Nexus 4, Nexus 6
Number of measurements	Moving: 1845, Stationary: 1688	Moving: 3199, Stationary: 4514
Total data collection time (hours)	Moving: 1.6, Stationary: 1.7	Moving: 3.8 Stationary: 6.5
Sampling period (Avg., SD)	RSSI: (3.6, 2.3) Phase: (4.7, 4.9)	RSSI: (5.5, 5.0) Phase: (7.6, 7.1)
Number of devices in the environment	several to couple hundreds	several to couple hundreds

(e) Dataset: summary

Fig. 9: (a) AP placements (green squares) at the retail and the cafeteria (about one AP per 15m x 15m) (b) In each floor map, the white color, black color, and dark grey color represent the area that participants visited, obstacles, and the area that the participant did not visit, respectively (c) Histogram of the number of measurements in each motion (stationary or moving) episode. For retail, a large number of episodes have a small number of measurements (less than 3), which makes it difficult to classify device motion. (d) Histogram of the sampling periods (inverse of sampling rate) of RSSI and phase measurements. Phase measurements have smaller sampling rate. (e) Summary of our dataset.

Feature-based approach	Training set (Cafeteria) -> Testing set (Cafeteria)						Training set (Retail) -> Testing set (Retail)					
	Accuracy	F1	FNR	FPR	Precision	Recall	Accuracy	F1	FNR	FPR	Precision	Recall
RNN	0.81, **0.85**	0.78, **0.82**	0.17, **0.17**	0.19, **0.14**	0.74, **0.80**	0.83, **0.84**	0.71, **0.79**	0.74, **0.80**	0.18, **0.14**	0.41, **0.29**	0.67, **0.75**	0.82, **0.86**
RF	0.81, **0.87**	0.77, **0.84**	0.22, **0.13**	0.16, **0.13**	0.76, **0.81**	0.78, **0.87**	0.70, **0.77**	0.72, **0.78**	0.24, **0.19**	0.36, **0.27**	0.68, **0.76**	0.76, **0.81**
HMM	0.78, **0.82**	0.76, **0.79**	0.16, **0.17**	0.26, **0.19**	0.69, **0.75**	0.84, **0.83**	0.72, **0.74**	0.73, **0.74**	0.24, **0.25**	0.33, **0.27**	0.70, **0.74**	0.76, **0.75**

TABLE IV: Testing results of feature-based models trained by using data collected at the same enterprise type. For each metric, the first number is the result of the model using only RSSI features, the second number (**bold**) is the result of the model using both RSSI features and phase feature. The models that use both features outperform the models that use only RSSI features.

Feature-based approach	Training set (Retail) -> Testing set (Cafeteria)						Training set (Cafeteria) -> Testing set (Retail)					
	Accuracy	F1	FNR	FPR	Precision	Recall	Accuracy	F1	FNR	FPR	Precision	Recall
RNN	0.78, **0.78**	0.68, **0.67**	0.41, **0.43**	0.10, **0.08**	0.80, **0.83**	0.60, **0.57**	0.70, **0.73**	0.70, **0.73**	0.31, **0.30**	0.29, **0.24**	0.71, **0.75**	0.69, **0.70**
RF	0.83, **0.87**	0.79, **0.84**	0.20, **0.13**	0.15, **0.13**	0.78, **0.81**	0.81, **0.87**	0.72, **0.78**	0.75, **0.79**	0.19, **0.18**	0.37, **0.26**	0.69, **0.77**	0.81, **0.82**

TABLE V: Testing results of feature-based models trained by using data collected in the retail and tested by using data collected in the cafeteria, and vice versa. Compared to Table IV, the performance of the RF models are slightly different though the RF models, in this case, are trained by using data collected in another building.

• The models that exploit the combination of RSSI features and phase correlation outperform the models that only exploit RSSI features in terms of all of the metrics described in Table III. For example, by using RSSI features only, the RF models achieve the accuracy of 0.81 and 0.70 for cafeteria and retail, respectively. By using the combination of RSSI features and phase correlation, the RF models achieve the accuracy of 0.87 and 0.77 for cafeteria and retail, respectively.

• The RF and RNN models have similar performance. This indicates that using RNN models to further extract temporal correlation of features is not necessary.

• Compared to cafeteria, our models have poorer performance for retail. This is due to a large number of motion (stationary or moving) episodes in the retail dataset have a few number of measurements (as shown in Fig. 9c). Given temporally sparse measurements in short periods of stationary or moving, it is more difficult to determine device motion.

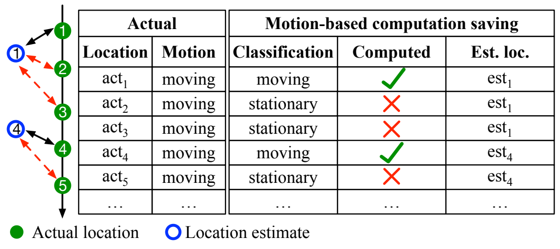
To further analyze the contribution of each RSSI feature into the performance of our models, we train our models by using each RSSI feature and analyze the performance of our RF models. Table VI shows the accuracy of the RF model when applying each of the features (described in Table II). The *SD of RSSIs* feature gives the best accuracy. For *Visibility of AP* feature, though it gives the worst accuracy compared to other features, it is required by the aggregation operator as described in Section V-B2. We note that these features achieve much higher accuracy in the prior work [19]. For example, *Visibility of AP* feature achieves the accuracy of 0.86, but only 0.63 with our dataset. It is probably because the sampling rate of RSSIs in their setups is periodic and doubles the sampling rate of RSSIs in our setups (Section II).

Feature-based approach using RSSI measurements		Cafteria		Retail	
RSSI features	Aggregation operator	Accuracy	F1	Accuracy	F1
SD of RSSIs	Average	0.80	0.76	0.69	0.7
SD of RSSI differences	Average	0.76	0.68	0.66	0.69
Consecutive RSSI difference	Average	0.75	0.69	0.66	0.69
Consecutive RSSIs	Pearson correlation	0.73	0.64	0.59	0.64
Visibility of AP	Average	0.61	0.37	0.64	0.63

TABLE VI: Accuracy of feature-based approach when each RSSI feature is applied separately

End-to-end (E2E) motion detection approach. We implemented our E2E Recurrent Neural Network (E2E-RNN) by using the TensorFlow library. Given the large dataset collected at the cafeteria (Fig. 9e), we are able to train the E2N-RNN described in Section V-C to classify a device motion by using RSSIs. In the training process, we test different hyper-parameter values, we achieve the following results by using Gradient descent optimization method with learning rate 0.2. The number of hidden units in a LSTM cell is 6. For the testing set at the cafeteria, E2E-RNN achieves accuracy of 0.84 and F1 score of 0.79. For the testing set at the retail, E2E-RNN achieves accuracy of 0.70 and F1 score of 0.66. These results show that the approach can achieve better accuracy compared RF model using RSSI. However, this approach's limitation is that it requires vastly more training data. Therefore, given many missing phase vectors due to low sampling rate of phase vector, the model could not learn the relationship between device motion and changes of phase measurements.

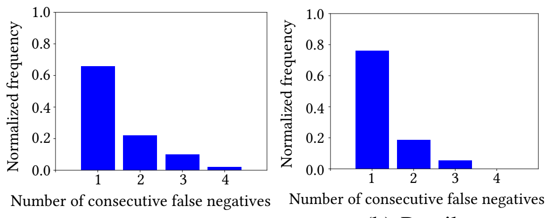
2) Model generalizability: To investigate the generalizability of our feature-based models across enterprise types and device types, we train our models by using data collected in the cafeteria and test the models by using data collected in the retail, and vice versa. Table V shows the performance of our models that use either RSSI features or the combination of RSSI features and phase correlation for classifying device motion. Compared to the testing results of the RF models in Table IV, the RF models achieve very similar performance in terms of all of the metrics (as described in Table III) though the RF models are trained by using data collected in another building. We conclude that our RF models are generalized across enterprise types and device types.



Actual		Motion-based computation saving		
Location	Motion	Classification	Computed	Est. loc.
act_1	moving	moving	✓	est_1
act_2	moving	stationary	✗	est_1
act_3	moving	stationary	✗	est_1
act_4	moving	moving	✓	est_4
act_5	moving	stationary	✗	est_4
…	…	…	…	…

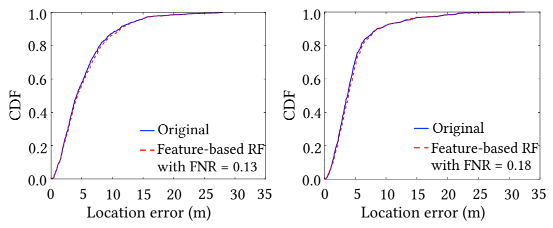
Fig. 10: Motion-based computation saving

3) Computation time: To make IIPS scalable, the running time of detecting a device's motion at a time step needs to be significantly lower than the running time of estimating the device's location. To investigate the overhead added by performing motion detection, we compare the average running time of our motion detection approaches and the localization method (described in Section III), at each time step. We measure these running times on the same computer (Intel CPU @ 3.70GHz). For the feature-based approach that uses the RNN model, extracting six aggregated features and performing the model take about 1.3 ms and 0.7 ms, respectively. Thus, at a time step, the approach takes about 2 ms, which is only 13% of the running time for a location computation, which takes about 15 ms. Therefore, for a device correctly classified as stationary, we can save about 87% of the CPU time spent on each location computation. For E2E-RNN models, the average CPU run-time is also relatively small (about 4 ms).



(a) Cafeteria (b) Retail

Fig. 11: Histogram of number of consecutive false negatives



(a) Cafeteria (b) Retail

Fig. 12: CDF of location accuracy of moving device with and without skipping location computation when device motion is classified as stationary

4) Tolerance to misclassification: We discuss the impact of each misclassification type, namely, false positive rates (actual stationary, classified as moving), and false negative rates (actual moving, classified as stationary), respectively. False positives (FPs) have no impact on location accuracy as the location computation is triggered for the sample. On the other hand, false negatives (FNs) may impact location accuracy. In Fig. 10, we depict how FNs impact accuracy. The estimated location for act_2, act_3, will be the *last computed location* (=est_1), and for act_5 it will be est_4. As a result, if the number of consecutive FNs is large, it may have a significant impact on location accuracy.

Fig. 11 shows that we rarely have more than 3 consecutive FNs and 70% of FNs are not consecutive. In addition, Fig. 12 shows that the FNR of our RF model (about 0.16 on average) has a negligible impact on the location accuracy as the CDFs of location errors when a device is moving with and without skipping location computations mostly overlap.

VII. Motion-Based Enhancements

In this section, we demonstrate three important use cases of motion classification models towards achieving the essential requirements of the IIPS as described in Section I.

A. Motion-based Computation Saving

Towards achieving high scalability, proactive location computation savings can be achieved by only computing location when a device is classified as moving. Thus, in an enterprise IIPS setup, that tracks thousands of devices, depending on the fraction of tracked devices that are stationary, a significant computation saving can be achieved. Computation savings is directly proportional to the *true negative rate* (TNR) of the motion classifier. Tables IV and V list the FPR of our proposed motion classifiers. For the motion classifiers using RSSIs and phases, TNR = (1 - FPR) $\simeq$ 80%, i.e., 80% of stationary points can be correctly identified for computation saving. The

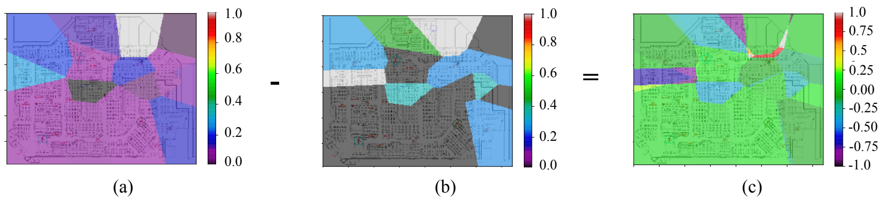


Fig. 13: Location accuracy at a retail represented as heatmaps generated from (a) site-survey approach (b) our approach. (c) The difference between these heatmaps is small.

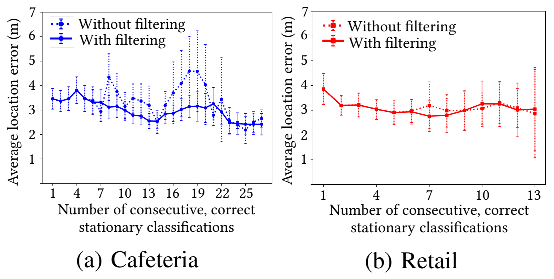


Fig. 14: Motion-based filtering to improve accuracy

potential risks of misclassification are discussed in Section VI-C4 where we establish that the impact of misclassification is not significant. Further, a variety of computation saving schemes can be realized ranging from most aggressive (not re-compute for devices classified as stationary) to conservative ones (selectively re-compute if a device is classified as stationary yet skip computation for longer than certain duration).

B. Motion-based Filters

By leveraging motion detection, location accuracy can be improved by applying certain filters such as Particle Filter or Kalman Filter [31]. In particular, motion classification based filter can be applied to the location estimates to mitigate the errors. Further, different filters can be applied for stationary vs. moving. To demonstrate this idea, we perform motion classification over our collected data sets. For simplicity, we apply a cumulative median filter for stationary devices. Fig. 14 shows the average of location errors with and without applying the filter on different numbers of consecutive location estimates that are correctly classified as stationary. The error bars represent the 90th percentile confidence intervals of the corresponding average errors. For the cafeteria (Fig. 14b), the filtered location estimates are much less scattering and closer to the corresponding actual locations. For the retail, the location accuracy with and without applying the filter are similar. The reason could be the number of consecutive samples classified as stationary is smaller, and participants are actually stationary for shorter periods of time (Fig. 9c).

C. Motion-based location accuracy monitoring

Monitoring location accuracy across a floor over time is vital for analyzing and improving the performance of an IIPS deployment. However, this process is often performed by doing site surveys *repeatedly*, which requires an intensive labor effort. For each site survey, we need to place multiple devices at different physical locations across a floor and then collect the location estimates of each device over a period of time so that we can measure the differences between devices' actual locations and their corresponding location estimates.

In our work, we propose an approach towards automating this process without requiring doing the site surveys or using additional infrastructures such as beacons which are placed statically on a floor. Our observation is that there are often several user devices (phones, laptops, etc.) on a floor that are stationary for a long period of time. The key idea is to apply motion-based filters on the location estimates of the stationary devices to approximate the actual locations of these devices. Given the approximated actual locations and the location estimates, we can estimate how the location estimates scatter over time at each physical location across the floor.

Fig. 13 shows location accuracy monitoring achieved by doing a site survey and by our approach. In Fig. 13a, the location accuracy across a floor is represented as a heatmap in which the colors indicate location accuracy from very good accuracy (black color) to very bad accuracy (red color). The white area indicates there is no site survey data in the area. We generated the heatmap by first applying nearest interpolation [27] on location errors measured at several locations during the site survey and then normalizing the results to the range $[0, 1]$. Fig. 13b shows our approximated heatmap generated by using the approximated actual locations as we described above. Fig. 13c shows that the difference between these heatmaps is small for most of the areas. Thus, our approach can enable monitoring location accuracy over time with low human effort.

VIII. Conclusion

In summary, we have shown that MotionScanner can exploit the temporal patterns of noisy, temporally sparse, and partial measurements from IIPS to detect device motion accurately. We also demonstrated that MotionScanner can enhance the performance of IIPS in terms of scalability and location accuracy, as well as enabling location-accuracy monitoring. We envision that MotionScanner can enable other interesting use cases for enterprises such as client behavior analytics and identifying regions of interest based on how long and how frequently user devices stay or move in particular areas.

IX. Acknowledgement

The authors would like to thank the Connected Mobile Experiences team and Wireless AP teams at Cisco for their support in the data collection process and clarifying explanations on how Wi-Fi measurements from Cisco's APs are used to perform localization and tracking. At PSU, this research was supported by Cisco through grants CG 594206 and CG 1008109.

References

[1] Y. Chapre, A. Ignjatovic, A. Seneviratne, and S. Jha. Csi-mimo: Indoor wi-fi fingerprinting system. In *39th Annual IEEE Conference on Local Computer Networks*, pages 202–209, Sept 2014.

[2] Cisco. Wireless LAN. https://www.cisco.com/c/m/en_ae/solutions/wireless-lan.html, 2009.

[3] Cisco. Cisco Aironet access point wireless security module (WSM). https://goo.gl/kqjt9R, October 2014.

[4] Cisco. CMX deployment models. https://goo.gl/6EyDZv, December 2015.

[5] Cisco. CMX fastlocate deployment guide mse release 8.0. https://goo.gl/fM6Da3 , January 2015.

[6] Cisco. Cisco wireless controllers. https://goo.gl/TSgCLk, 2016.

[7] Cisco. Hyperlocation: Best Practices and Troubleshooting Guide. https://goo.gl/J127P5, December 2016.

[8] Cisco. Hyperlocation. https://goo.gl/B7RoRq, December 2017.

[9] Cisco. Cisco Aironet 4800 Access Points Data Sheet. https://goo.gl/bSVgg9, August 2018.

[10] Cisco. Cisco CMX Configuration Guide. https://goo.gl/NcZxGT, June 2018.

[11] Cisco. Cisco Hyperlocation module with advanced security data sheet. https://goo.gl/thA29i, December 2018.

[12] Cisco. Customer stories. https://goo.gl/R2ixEY, August 2018.

[13] P. J. García-Laencina, J.-L. Sancho-Gómez, and A. R. Figueiras-Vidal. Pattern classification with missing data: a review. *Neural Computing and Applications*, 19(2):263–282, 2010.

[14] B. D. Hart, P. J. Stager, S. Pandey, D. Kloper, D. Lyons, and M. A. Silverman. Angle of arrival location sensing with antenna array, Nov. 21 2017. US Patent 9,823,330.

[15] S. Hemminki, P. Nurmi, and S. Tarkoma. Accelerometer-based transportation mode detection on smartphones. In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, SenSys '13, 2013.

[16] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, Nov. 1997.

[17] C.-Y. Hsu, Y. Liu, Z. Kabelac, R. Hristov, D. Katabi, and C. Liu. Extracting gait velocity and stride length from surrounding radio signals. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pages 2116–2126, New York, NY, USA, 2017. ACM.

[18] K. Joshi, D. Bharadia, M. Kotaru, and S. Katti. Wideo: Fine-grained device-free motion tracing using RF backscatter. In *12th USENIX Symposium on Networked Systems Design and Implementation (NSDI 15)*, pages 189–204, Oakland, CA, 2015. USENIX Association.

[19] K. Kavitha Muthukrishnan, B. van der Zwaag, and P. Havinga. *Inferring motion and location using WLAN RSSI*, pages 163–182. Lecture Notes in Computer Science. Springer Verlag, 9 2009.

[20] M. Kotaru, K. Joshi, D. Bharadia, and S. Katti. Spotfi: Decimeter level localization using wifi. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, SIGCOMM '15, pages 269–282, New York, NY, USA, 2015. ACM.

[21] J. Krumm and E. Horvitz. Locadio: inferring motion and location from wi-fi signal strengths. In *The First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services, 2004. MOBIQUITOUS 2004.*, pages 4–13, Aug 2004.

[22] G. Kul, T. Özyer, and B. Tavli. Ieee 802.11 wlan based real time indoor positioning: Literature survey and experimental investigations. *Procedia Computer Science*, 34:157 – 164, 2014. The 9th International Conference on Future Networks and Communications (FNC'14)/The 11th International Conference on Mobile Systems and Pervasive Computing (MobiSPC'14)/Affiliated Workshops.

[23] Office employees should be on feet for four hours of working day, study says. https://goo.gl/HL2Xf8, June 2015.

[24] Z. C. Lipton, D. C. Kale, and R. Wetzel. Modeling missing data in clinical time series with rnns. *Machine Learning for Healthcare*, 2016.

[25] K. Muthukrishnan, M. Lijding, N. Meratnia, and P. Havinga. Sensing motion using spectral and spatial analysis of wlan rssi. *Smart Sensing and Context*, pages 62–76, 2007.

[26] T. Nick, E. Coersmeier, J. Geldmacher, and J. Goetze. Classifying means of transportation using mobile sensor data. In *Neural Networks (IJCNN), The 2010 International Joint Conference on*, pages 1–6. IEEE, 2010.

[27] O. Rukundo and H. Cao. Nearest neighbor value interpolation. *International Journal of Advanced Computer Science and Applications(IJACSA)*, 3(4), 2012.

[28] L. Sun, S. Sen, and D. Koutsonikolas. Bringing mobility-awareness to wlans using phy layer information. In *Proceedings of the 10th ACM International on Conference on Emerging Networking Experiments and Technologies*, CoNEXT '14, pages 53–66, New York, NY, USA, 2014. ACM.

[29] Dense layer. https://www.tensorflow.org/api˙docs/python/tf/layers/dense, December 2017.

[30] Gradient descent optimizer. https://goo.gl/6p6cH6, December 2017.

[31] S. Thrun. Particle filters in robotics. In *in Proceedings of the 17th Annual Conference on Uncertainty in AI (UAI*, 2002.

[32] E. Union Agency for Network and Information Security. Passive wifi surveillance and access point hijacking. https://www.enisa.europa.eu/publications/info-notes/passive-wifi-surveillance-and-access-point-hijacking, 2015.

[33] H. Wang, D. Zhang, J. Ma, Y. Wang, Y. Wang, D. Wu, T. Gu, and B. Xie. Human respiration detection with commodity wifi devices: Do user location and body orientation matter? In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '16, pages 25–36, New York, NY, USA, 2016. ACM.

[34] X. Wang, L. Gao, and S. Mao. Phasefi: Phase fingerprinting for indoor localization with a deep learning approach. In *2015 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6, Dec 2015.

[35] C. Wu, Z. Yang, Z. Zhou, X. Liu, Y. Liu, and J. Cao. Non-invasive detection of moving and stationary human with wifi. *IEEE Journal on Selected Areas in Communications*, 33(11):2329–2342, Nov 2015.

[36] J. Xiao, K. Wu, Y. Yi, L. Wang, and L. M. Ni. Fimd: Fine-grained device-free motion detection. In *2012 IEEE 18th International Conference on Parallel and Distributed Systems*, pages 229–235, Dec 2012.

[37] J. Xiong and K. Jamieson. Arraytrack: A fine-grained indoor location system. In *Proceedings of the 10th USENIX Conference on Networked Systems Design and Implementation*, nsdi'13, pages 71–84, Berkeley, CA, USA, 2013. USENIX Association.

[38] M.-C. Yu, T. Yu, S.-C. Wang, C.-J. Lin, and E. Y. Chang. Big data small footprint: the design of a low-power classifier for detecting transportation modes. *Proceedings of the VLDB Endowment*, 7(13):1429–1440, 2014.