EAR: Exploiting Uncontrollable Ambient RF Signals in Heterogeneous Networks for Gesture Recognition

Zicheng Chi*
zicheng1@umbc.edu
Computer Science and Electrical Engineering, University of Maryland, Baltimore County

Yao Yao*
of90379@umbc.edu
Computer Science and Electrical Engineering, University of Maryland, Baltimore County

Tiantian Xie*
xtiant1@umbc.edu
Computer Science and Electrical Engineering, University of Maryland, Baltimore County

Xin Liu
xinliu1@umbc.edu
Computer Science and Electrical Engineering, University of Maryland, Baltimore County

Zhichuan Huang
zhihu1@umbc.edu
Computer Science and Electrical Engineering, University of Maryland, Baltimore County

Wei Wang
ax29092@umbc.edu
Computer Science and Electrical Engineering, University of Maryland, Baltimore County

Ting Zhu
zt@umbc.edu
Computer Science and Electrical Engineering, University of Maryland, Baltimore County

ABSTRACT

The exponentially increasing number of Internet-of-Thing (IoT) devices introduces a spectrum crisis in the shared ISM band. However, it also introduces opportunities for conducting radio frequency (RF) sensing using pervasively available signals generated by heterogeneous IoT devices. In this paper, we explore how to leverage the ambient wireless traffic that i) generated by uncontrollable IoT devices and ii sensed by ambient noise floor measurements (a widely available metric in IoT devices) for human gesture recognition. Specifically, we introduce our system EAR, which can conduct fine-grained human gesture recognition using coarse-grained measurements (i.e., noise floor) of ambient RF signals generated from uncontrollable signal sources. We conducted extensive evaluations in both residential and academic buildings. Experimental results show that although EAR uses coarse-grained noise floor measurements to sense the uncontrollable signal sources, the signal sources can be distinguished with an accuracy up to 99.76%. Moreover, EAR can recognize fine-grained human gestures with high accuracy even under extremely low traffic rate (i.e., 4%) from uncontrollable ambient signal sources.

*Authors contributed equally to the paper

SenSys '18, November 4–7, 2018, Shenzhen, China

ACM ISBN 978-1-4503-5952-8/18/11…$15.00
https://doi.org/10.1145/3274783.3274847

CCS CONCEPTS

• **Computer systems organization → Sensor networks**; • **Human-centered computing → User interface design**;

KEYWORDS

IoT, RF Sensing, Heterogenous Networks

ACM Reference Format:
Zicheng Chi, Yao Yao, Tiantian Xie, Xin Liu, Zhichuan Huang, Wei Wang, and Ting Zhu. 2018. EAR: Exploiting Uncontrollable Ambient RF Signals in Heterogeneous Networks for Gesture Recognition. In *The 16th ACM Conference on Embedded Networked Sensor Systems (SenSys '18), November 4–7, 2018, Shenzhen, China.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3274783.3274847

1 INTRODUCTION

The number of Internet-of-Things (IoT) devices is exponentially increasing. It will reach 20 billion by 2020 [31]. These devices will also generate huge amounts of wireless traffic. Based on the Cisco Global Cloud Index [31], the data created by these devices will reach 42.3 ZB (i.e., 4.23×10^{22} bytes) per month and will be 49 times higher than total data center traffic by 2019. The huge amount of data generated by these IoT devices will cause spectrum crisis in the shared Industrial Scientific Medical (ISM) band.

On the other hand, the ubiquitous wireless traffic generated by these IoT devices also introduce potential opportunities for conducting RF sensing (e.g., human gesture recognition). Comparing with the mature voice command systems (e.g., Alexa and Google Home), the scenario of RF signal based gesture recognition is much wider. For example, RF-based system can be used i) in a noisy environment (e.g., when playing loud music, the voice command system does not work well); ii) in a quiet environment (such as a baby bedroom); iii) for disabled people.

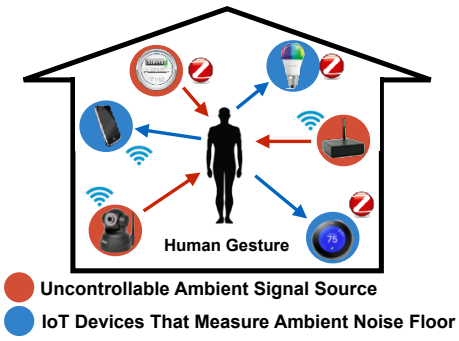


Figure 1: By passively listening to the ambient noise floor from uncontrollable signal sources on IoT devices, EAR recognizes human gestures without introducing extra wireless traffic. Since our system does not require RF signal demodulation, the signal sources can be heterogeneous IoT devices.

Different from existing RF sensing approaches that need the *designated and controlled* senders explicitly send out the RF signals, we propose to explore how to leverage the existing ambient RF signals (i.e., wireless traffic) generated by heterogeneous IoT devices for human gesture recognition. By doing this, we can avoid introducing designated wireless traffic for RF sensing. Therefore, our approach does not introduce extra burden to the overcrowded ISM band and has zero interference with existing wireless traffic.

Figure 1 shows a potential application of our approach in a smart home. Multiple IoT devices (e.g., smart light, cell phone, and thermostats) can seamlessly *work together* to identify the detailed gestures of the occupant based on the signal changes of ambient wireless traffic which is emitted by the uncontrollable signal sources (such as surveillance camera, smart meter, and access point) and bounced back from the occupant's body. These devices can then adjust the environment based on the occupant's gestures (e.g., turning on the light when the occupant is waving his/her hand).

In order to utilize the ambient RF signals for gesture recognition, we need to address the following three unique challenges:

i) Uncontrollable RF signal sources: Since IoT devices are using different radios and protocols (e.g., WiFi, ZigBee, and Bluetooth), they cannot demodulate the wireless traffic generated by different radios. Therefore, when an IoT device receives the ambient RF signals, it is very difficult to tell who sent the signals without demodulation. Moreover, the number of RF signal sources may be different during the human gesture recognition. To address these issues, we propose to use i) complete-linkage clustering for potential source distinguishing and ii) a real-time cluster updating mechanism for merging observations to efficiently maintain the source clusters.

ii) Intermittent RF traffics: Most of the wireless devices, such as wireless access point (AP) and wireless sensor devices, have intermittent RF traffic. However, existing human gesture recognition approaches require the sender to generate continuous signals. In order to leverage the ambient RF signals for human gesture recognition, we need to incorporate the wireless signals from different senders. To address these issues, we propose to use a gesture structure matrix to decompose observed intermittent RF traffics into sparse gesture coefficient, and recover the missing RF traffic from the coefficient.

iii) Asynchronized Measurement: Since IoT devices are using distributed clocks and different protocols (ZigBee, BLE, WiFi, etc.), their clocks are not always synchronized. In order to leverage the ambient RF signals for gesture recognition, we need to coordinate and synchronize the measured RF signal strengths among multiple receivers. To address this issue, we propose an algorithm based on the spacing between reference signals in the environment to synchronize the measurement for each IoT device without explicit time synchronization messages.

Our main contributions are as follows:

- To the best of our knowledge, this is the first work that enables the ambient RF signals from IoT devices for human gesture recognition without introducing any designated RF *sensing* traffic.
- This is also the first work that seamlessly combines multiple pieces of ambient RF signals from multiple uncontrollable senders and multiple receivers for gesture recognition.
- We have extensively evaluated the system with factors (i.e., intermittent traffic rate and type, the number of uncontrollable sources and regularly or randomly deployed nodes and synchronization efficiency) that can potentially affect recognition accuracy. Results show that: i) the recognition accuracy is higher than 90% even under an extremely low traffic rate (i.e., 4%) and different traffic types; and ii) uncontrollable sources can be distinguished with an accuracy up to 99.76%.
- The design of the signal source distinguishing and intermittent RF traffic reconstruction modules are generic. It is possible to extend EAR to be a middleware which "glue" other RF sensing system with uncontrollable ambient signals.

2 SYSTEM OVERVIEW

EAR system is divided into two parts: i) server side and ii) IoT devices side. To save energy on individual IoT device, the server collects data and conducts the main process to recognize human gesture. Figure 2 shows the overview of our system EAR. At the server side, there are three steps:

- **Distinguishing unknown RF signal sources**. To utilize ambient RF signal, EAR distinguishes the unknown signal sources that are characterized by the physical properties of the source and environment. Specifically, we use complete-linkage clustering to identify potential sources and distinguish the source cluster by signal instance similarity among the measurements of different IoT devices. To efficiently maintain the source clusters in real-time, we merge saved signal instances by their centroids (see Section 3).
- **Reconstructing from intermittent RF traffics**. By using a gesture structure matrix, EAR reconstructs a persistent traffic measurement from intermittent RF traffics to capture human motions in the frequency domain (see Section 4).
- **Gesture Recognition**. After above processes, a recognition algorithm is proposed for gesture recognition with ambient RF signals (see Section 5).

To distinguish the signal sources and recognize human gesture, EAR needs the aligned measurements from distributed IoT devices. Instead of using explicit time synchronization messages, we propose

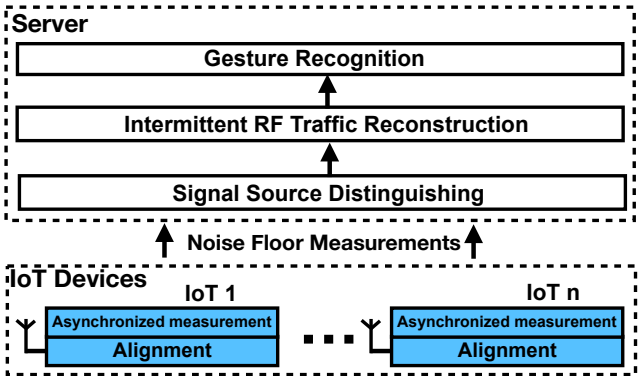


Figure 2: System Overview

to passively listen to the ambient RF signals to align the noise floor measurements on distributed IoT devices by the following approach: •**Asynchronized measurement alignment**. EAR aligns the noise floor measurements by using a folding technique to find the periodical beacon of existing network. To prevent misalignment when multiple beacons exist and have the same (or integer multiple) period(s), we propose to sense and utilize the unique spacing signature among beacons in a distributed manner (see Section 6).

3 DISTINGUISHING UNCONTROLLABLE RF SIGNAL SOURCES

EAR utilize the noise floor (NF) measurements to characterize the human gesture. Since the measurements can be from different signal sources, we first discuss why we need to distinguish RF signal sources. Then we propose the method to distinguish RF signal sources by using the NF from distributed IoT devices.

3.1 Why we need to distinguish the sources

Due to the widely adopted carrier-sense multiple access (CSMA) mechanism in wireless communication (e.g., both the WiFi and ZigBee protocols adopt CSMA), different RF senders transmit signals alternately to avoid a collision. Since the RF transmission rate is much faster than human gesture, it is possible that two or more signals present during one gesture.

Figure 3 shows two examples of the same gesture captured by an IoT device (i.e., receiver). During the gesture motion duration, there are two signals from different sources. In setting 1 (Figure 3(a)), the uncontrollable signal source A transmits signal in time duration $T1$ and source B transmits signal in time duration $T2$. In Figure 3(c), the receiver senses the NF from source A and source B in time duration $T1$ and $T2$, respectively. However, in setting 2 (Figure 3(b)), the uncontrollable signal source B transmits in time duration $T1$ and source A transmits in time duration $T2$. And the received NF is shown in Figure 3(d). If we directly use the collected NF measurements for gesture recognition without distinguishing the signal sources, the gesture recognition model obtained by the data (Figure 3(c)) in setting 1 will not be able to recognize the gesture when we collected NF in setting 2. Meanwhile, the gesture models of different sources may not be exactly the same. Therefore, it is important to distinguish the signal sources for accurate human gesture recognition.

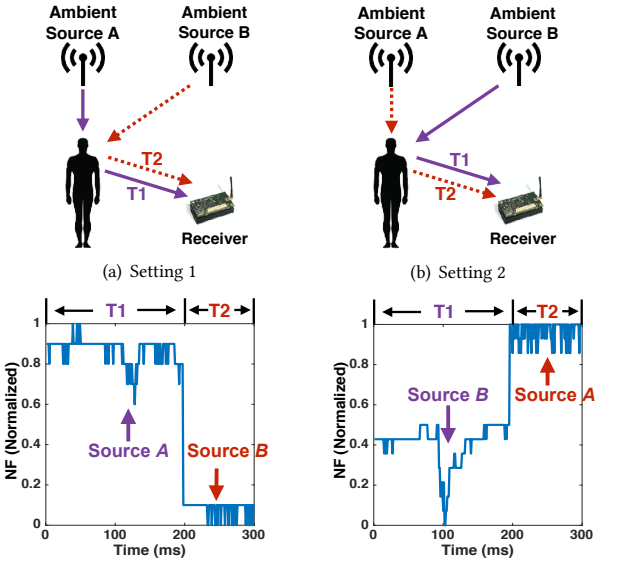


Figure 3: NF measurements for a same gesture under two different settings

	$n = 100$	$n = 500$	$n = 1,000$	$n = 2,000$
P(0.6)	1.3e-11%	1.6e-59%	3.9e-119%	3.3e-239%

Table 1: Probability to be recognized with n samples if there are three ambient signal sources.

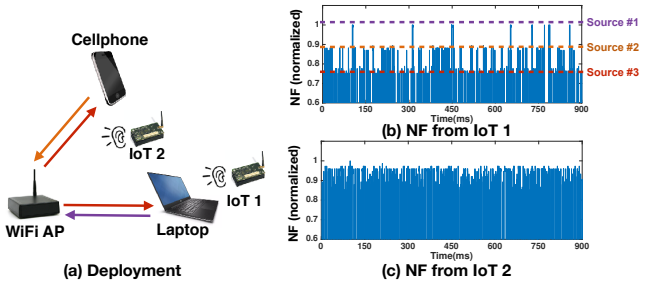


Figure 4: Some IoT devices (i.e., IoT 1) can distinguish signal sources via NF while the others (i.e., IoT 2) cannot.

Formally, suppose a gesture can be recognized by the features extracted from NF measurements. By performing the similarity of matching, NF measurements can be matched to the same gesture if the signal comes from the same source. If there are k signal sources in the environment, and the NF during a gesture motion is composed of n slots that each can be occupied by one source. With the similarity s ($s \in (0, 1]$), the probability $P(s)$ that the NF measurements can be matched to the right gesture can be expressed as follows:

$$P(s) = \sum_{i=sn}^{n} \binom{n}{i} (\frac{1}{k})^i (\frac{k-1}{k})^{n-i} \quad (1)$$

An example in Table 1 shows that when the number of signal sources is three (i.e., $k = 3$). Assuming the gesture can be recognized correctly if $s > 0.6$, we observe that the recognition probability is very low. Thus, we need to distinguish the signal sources.

Algorithm 1 Continuous Signal Source Distinguishing

Input: $\mathcal{S}$, $C = \{\langle C_1, l_1\rangle, ..., \langle C_{n_s}, l_{n_s}\rangle\}$
Output: L, C

1: $\mathcal{E} = PCA(\mathcal{S}, \tilde{k}); d = 0$
2: $\tilde{C} = clustering(\mathcal{E})$
3: **for** $\langle \tilde{C}_i, \tilde{l}_i\rangle$ **in** $\tilde{C}$ **do**
4: $\tilde{l}_i = \underset{l_j:\langle C_j, l_j\rangle \in C}{\text{argmax}} \{\sum_{s\in\tilde{C}_i} \mathbb{1}_{C_j}(s)\}$
5: **if** $\tilde{l}_i = \emptyset$ **then**
6: $d++; \tilde{l}_i = n_s + d$
7: $C = C \cup \{\langle \tilde{C}_i, \tilde{l}_i\rangle\}$
8: **else**
9: $C_j = C_j \cup \tilde{C}_i$
10: **end if**
11: **end for**
12: **for** $\langle C_i, l_i\rangle \in C$ **do**
13: **repeat**
14: $s, s' = \underset{s,s'\in C_i}{\text{argmin}} \{||s - s'||\}$
15: $C_i = C_i \backslash \{s, s'\} \cup \{centroid(s, s')\}$
16: **until** $|C_i| \leq N_e$
17: **end for**
18: $L = label(\mathcal{S}, C)$

3.2 Continuous signal source distinguishing

Since the goal of EAR is to utilize ambient RF signal for recognizing human gesture, the received signal may come from heterogeneous IoT devices. e.g., a ZigBee receiver picks up WiFi signals. Thus, it is possible that the IoT receiver is not able to demodulate the received signal. In other words, the receiver cannot distinguish the signal source by extracting identification (such as node ID or source address) from the demodulated packet. Therefore, how to distinguish RF signal sources by using the widely available noise floor (NF) measurements [1] is challenging.

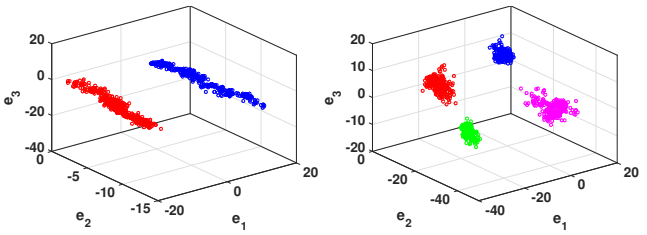
We conducted an experiment with the setup shown in Figure 4(a), where two IoT nodes are passively recording the NF while two devices (i.e., cellphone and laptop) are performing normal Internet access (e.g., watching online videos). The recorded NF readings for IoT 1 and IoT 2 are shown in Figure 4(b) and Figure 4(c), respectively. From figures 4(b), we find that it is possible to distinguish by the measurements from IoT 1 because IoT 1 is at the boundary of the three senders. However, it is very difficult to differentiate the signal sources by IoT 2's measurements (shown in Figure 4(c)) because it is at the center of the three senders.

Given the signal source location is unknown to IoT receivers, we propose to incorporate the NF measurements from multiple IoT devices to distinguish the RF signal sources.

To distinguish the signal source, we design a *Continuous Signal Source Distinguishing (CSSD)* (Algorithm 1). *CSSD* distinguishes the sources based on the similarity among the NF measurements across different IoT receivers. This algorithm has a time complexity of $O(n^2)$ where n is the number of samples. In the evaluation (see Section 7.4), we show that this algorithm can achieve an accuracy up to 99.76% for source distinguishing.

We first introduce the input and output of the algorithm then specify the process of the algorithm.

• **Inputs:** The input $\mathcal{S}$ is an k by i array of received NF measurements:



(a) Two ambient sources (b) Four ambient sources

Figure 5: Source Clustering. The clustering method in CSSD algorithm is effective to separate the signal sources.

$$\mathcal{S} = \begin{bmatrix} s_{11} & \dots & s_{1i} \\ \vdots & \ddots & \vdots \\ s_{k1} & \dots & s_{ki} \end{bmatrix} \quad (2)$$

where each element s_{ki} is the measurement at time i from device k. Thus, each column $\mathbf{s}_{(i)} = [s_{1i}, ..., s_{ki}]^T$ is the measurements at across all devices at a specific time [2]. The second input C includes n clusters. Each pair $\langle C_i, l_i\rangle$ depicts a signal source l_i and its associated NF array set $C_i = \{\mathbf{s}_1, ..., \mathbf{s}_{|C_i|}\}$ that characterizes the observation of physical property for the signal sources.

• **Outputs:** $L = \{l_1, ..., l_w\}$ is the label of the signal sources for $\mathcal{S}$. *CSSD* keeps updating C because the clustering results of the same source may be different over time due to human influence or environment change.

• **Process:** To i) prevent the "curse of dimensionality" when the number of IoTs increases and ii) retain the most important variance caused by human motion, *fast PCA* is applied (Line 1). Then, each column $\mathbf{s}_{(i)}$ is transformed to $\mathcal{E}(i) = [e_{1i}, e_{2i}, ..., e_{\tilde{k}i}]^T$.

Since we do not make an assumption on the number of potential signal sources and *CSSD* targets at continuous source distinguishing that no hard threshold should be set, we use *complete-linkage clustering* (Line 2) to continuously identify potential source clusters. This algorithm does not require an indication of the number of clusters and has a low time complexity of $O(n^2)$ where n is the number of elements. The key idea of this clustering method is to first assume each element to be its own cluster, then merge pairs of clusters until no two clusters can be merged anymore. Since the spacing between different clusters is large, we break down the clustering into $\tilde{n}$ clusters based on the max common stem height in its dendrogram to make sure that $\tilde{n}$ is larger than the number of sources identified n_s. Figure 5 shows a result from the environments with 2 or 4 sources. We can clearly observe the space among different sources.

So far, we have separated the sources into multiple clusters, but we still do not know which source the cluster belongs. Thus, we match the real-time clusters $\tilde{C}$ with the saved cluster set C and identify the source as the one by the max number of nearest signal instances (Lines 3-11). However, until now we still have two **challenges** to address: i) the matching process between $\tilde{C}$ and C can take too much time when $|C_j|$ becomes large; ii) the clustering results of the same source may be different over time due to human influence or environment changes. To address these two challenges,

[1] Note that sophisticated signal features may not be available on energy constrained IoT devices such as ZigBee.

[2] We assume the measurements from different IoT devices are aligned in terms of time and we will introduce how to align them in Section 6.

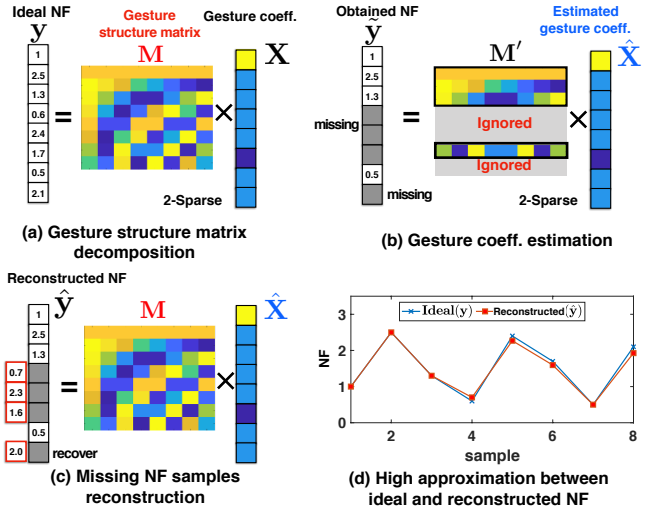


Figure 6: Intermittent Traffic Reconstruction

we dynamically merge elements in previously saved clusters when the elements in saved clusters reach a maximum number N_e and add the newly identified clusters into the saved clusters (Lines 12-17). We can do this because of the following two reasons: i) the fast PC extracts the main components caused by different signal sources rather than variation caused by human influence. ii) the use of the complete-linkage clustering method updates the clusters in real-time even when human movements present. The merging process is as follows: i) find two elements with the shortest distance in the saved cluster (Line 14); ii) replace them by their centroid (Line 15); iii) merging process ends when the element number in the saved and new identified clusters reaches N_e (Line 16).

Since the CSSD algorithm does not require demodulation and based on noise floor measurements, it works with heterogeneous signal sources (e.g., WiFi or ZigBee devices) as long as the signal has a constant transmission power. The noisy signals (such as from a microwave oven or a moving source) will be filtered out because it does not belong to a certain cluster over time.

4 INTERMITTENT TRAFFIC RECONSTRUCTION

In Section 3, we introduced how to distinguish the signal sources. In this section, we propose to deal with the intermittent traffic sent out by IoT devices. Since most of the wireless IoT devices have bursts of intermittent RF traffic, however, the existing human gesture recognition requires a continuous signal measurements to characterize the human motion. To address this challenge, we reconstruct a persistent measurement based on the sparsely observation of NF, whose variance is caused by gesture motion. The key feature we use is that the variance has an underlying structure that can be captured by a proper gesture structure matrix. The advantage of our algorithm includes: **i)** effectively recovering detail information of gesture motions from intermittent traffic; **ii)** performing gesture recognition even the intermittent traffic rate is extremely low. Our evaluation shows that the gesture recognition accuracy is high even under extremely low traffic rate 0.04 (detailed in Section 7.4).

Underlying structure: *An NF measurements stream, which captures gesture motion, can be decomposed by a gesture structure matrix into a sparse gesture coefficient vector.*

Figure 6(a) shows an example of how the underlying structure is used. For an ideal NF stream $\mathbf{y}$ (i.e., an $N \times 1$ column vector), a proper $N \times N$ gesture structure matrix $\mathbf{M}$ can decompose it into a sparse gesture coefficient vector $\mathbf{x}$ (i.e., an $N \times 1$ column vector) in $\mathbf{R}^N$. In this example, $N = 8$. After decomposition, $\mathbf{x}$ is 2-sparse that only the 1st and the 6th coefficient has significant value while other values are close to zero.

Reconstruction: *Even if only partial NF measurements are obtained, the sparse gesture coefficient vector can still be estimated by the gesture structure matrix and used to reconstruct the missing NF values.*

In figure 6(b), the obtained NF measurements stream $\tilde{\mathbf{y}}$ is part of the ideal NF stream $\mathbf{y}$, where only $K = 4$ samples out of 8 are obtained. Thus, the matrix $\mathbf{M}'$ (corresponding to the obtained values $\tilde{\mathbf{y}}$) from $\mathbf{M}$ is used to estimate $\hat{\mathbf{x}}$ which is not full row-rank:

$$\tilde{\mathbf{y}} = \mathbf{M}'\hat{\mathbf{x}} \tag{3}$$

Equation 3 is an under-determined linear equation since the number of equations $K = 4$ is smaller than the number of variables $N = 8$. We model it as a minimization problem:

$$\min_{\hat{\mathbf{x}} \in \mathbf{R}^N} ||\hat{\mathbf{x}}||_1 \quad s.t.\ \tilde{\mathbf{y}} = \mathbf{M}'\hat{\mathbf{x}} \tag{4}$$

Because the ideal gesture coefficient $\mathbf{x}$ is expected to be sparse, this is a l_1 norm minimization problem that can be easily solved using linear programming (LP). After solving the minimization problem, the estimated gesture coefficient $\hat{\mathbf{x}}$ is also 2-sparse and approximates $\mathbf{x}$ well.

Next, we can reconstruct the recovered NF stream $\hat{\mathbf{y}}$ by solving the following equation:

$$\hat{\mathbf{y}} = M\hat{\mathbf{x}} \tag{5}$$

The reconstruct values are obtained as the red boxes shown in Figure 6(c). Figure 6(d) shows that the reconstructed NF values have a high approximation to the ideal measurements.

Gesture structure matrix: Selecting $\mathbf{M}$ is non-trivial, its corresponding inverse must be sufficient to make the gesture coefficient $\mathbf{x}$ a sparse vector. An ideal solution is to obtain $\mathbf{M}$ from ideal NF streams that captured human gesture motions. However, this will introduce extra efforts and the learnt structure matrix might not be universally applicable. Instead, we propose to use a matrix that has potential for gesture recognition purposes. Since human movements induce several high peaks in frequency domain, which were also mentioned by previous researchers ([33],[11]), we select a structure matrix related to frequency transform. After exploration, the matrix we select was previous used for compression, discrete cosine transform (DCT) matrix:

$$\mathbf{M}(i,j) = \begin{cases} \frac{1}{\sqrt{N}} & i = 0, 0 \le j \le N-1 \\ \sqrt{\frac{2}{N}} cos\frac{i \cdot (2j+1)\pi}{2N} & 1 \le i \le N-1, 0 \le j \le N-1 \end{cases} \tag{6}$$

We compare our reconstruction method with an autoregressive model. Figure 7(a) shows the obtained NF measurements with a 0.2 intermittent traffic rate. Figure 7(b) and 7(c) show the original signal and the reconstructed signals by using an autoregressive model and our method, respectively. By visually comparing the results, we observe our technique provides much more details than the autoregressive model even when the traffic rate is only 0.2.

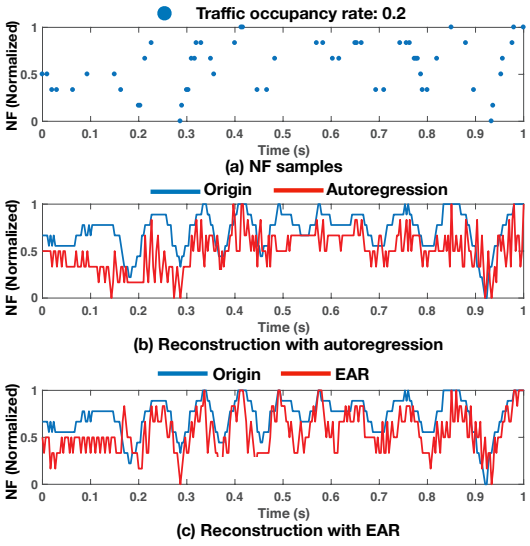


Figure 7: Signal Reconstruction

In the evaluation (Section 7), we extensively evaluate how the reconstructed data impacts the gesture recognition accuracy on different traffic rate.

5 GESTURE RECOGNITION

After the processes of signal source distinguishing and intermittent traffic reconstruction, we propose an algorithm to conduct gesture recognition by using ambient RF signals. Different from previous RF based recognition technique, the recognition algorithm in EAR must support dynamic changes of IoT devices' number which includes the number of signal sources and the number of IoT receivers.

5.1 Gesture Detection

The first step of gesture recognition is to detect the start and end of a gesture (i.e., the boundary of NF measurements). Based on empirical analysis, we find the mean of second principal component $\mathbf{E}(h_2)$ after fast PCA (see Section 3) is stable when no human motion presents. Whereas in the presence of human gesture, $\mathbf{E}(h_2)$ shows large variance. Figure 8 shows an example of the $\mathbf{E}(h_2)$ changes while three different gestures (i.e., "Kick", "Wave", and "Pull") present. We can observe the difference of the amplitude between human gesture not presents and presents. To utilize the mean of second principal component $\mathbf{E}(h_2)$ for gesture detection, we calculate the approximate entropy of $\mathbf{E}(h_2)$ over time. Approximate entropy is a useful technique that quantifies the correlation of fluctuations over time. There are three steps to calculate the approximate entropy $AE(h_2)$: 1) we form a sequence of vectors $\mathbf{x}(1), \ldots, \mathbf{x}(N-m+1)$, where $\mathbf{x}(i) = [h_2(i), \ldots, h_2(i+m-1)]$; 2) we construct $C_i^m(r)$, the probability of $\mathbf{x}(j)$ such that $dis[\mathbf{x}(i), \mathbf{x}(j)] \leq r$ (where $dis[\mathbf{x}(i), \mathbf{x}(j)]$) is the longest distance between any two elements in these two vectors; 3) approximate entropy $AE(h_2)$ can be calculated as follows:

$$\mathrm{AE}(\mathrm{h}_2) = \Omega^m(r) - \Omega^{m+1}(r) \tag{7}$$

where $\Omega^m(r) = (N-m+1)^{-1} \sum_{i=1}^{N-m+1} \log(C_i^m(r))$, m and r are two parameters which depend on the gesture duration and sampling rate. The larger the m and r values, the false positive error is lower

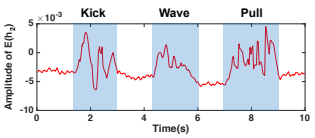


Figure 8: Detect human gesture with $\mathbf{E}(h_2)$

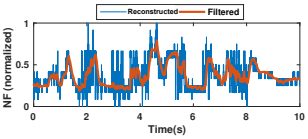


Figure 9: Filtering for a reconstructed source.

and the false negative is higher because the approximate entropy is calculated with a larger window size and higher similarity criterion. For gesture detection, the purpose is to detect all the potential gestures (lower false negative is better).We empirically choose $m = 50$ and $r = 3$ for best results. To detect the gesture, we first train the threshold based on the approximate entropy of the second principal component $AE(h_2)$ for the period with and without activities. During the testing, when $AE(h_2)$ is detected higher than the threshold, we set that time as the start of a gesture. When $AE(h_2)$ is lower than the threshold, we determine that time as the end of a gesture.

5.2 Filtering

After bouncing back from human body, low frequency components are introduced to the RF signal by the body movement [11, 33]. To accurately recognize the human gesture, a Low-Pass Filter (LPF) with steep roll-off is applied to extract the low frequency parts and remove other noises. Moreover, we want to reduce the ripples in the passband in case that they twist the frequency components caused by weak gesture. A second order low pass filter with a cutoff frequency $f_c = 50Hz$ and a 40 dB of stopband attenuation provides good performance in our system. The parameters are chosen empirically. The transfer function is as follows:

$$H(z) = \frac{0.015 - 0.006z^{-1} + 0.015z^{-2}}{1 - 1.771z^{-1} + 0.795z^{-2}} \tag{8}$$

Figure 9 shows a snapshot of reconstructed NF values (blue colored curve) and the values after filtering (red colored curve). We can observe that this LPF keeps the low frequency trend but removes the high frequency noises.

5.3 Recognition

Compared to existing works, EAR has the following three challenges: i) within an individual gesture, the NF measurements may come from different sources; ii) the number of RF signal sources dynamically changes over time; iii) the NF measurements come from different receivers that need to be fused. To address these challenges, the design of recognition algorithm should consider the following:

• *The algorithm should support varying-dimension vectors.* Because the NF matrix has a varying dimension (i.e., the number of signal sources and receivers changes dynamically. For example, the IoT device may lose power.), the models' dimension cannot be fixed.

• *Different sources cannot use one collaborative model.* Namely, the model for the measurements from one signal source cannot be used for the measurements from another source.

• *Low time complexity for updating the model.* Since the number of sources and IoT devices can be large, if the algorithm requires a lot of time for updating, it decreases the capability for EAR to adapt to new situations quickly.

With these considerations, we select Markov chain models as the basic algorithm building block because of its balance between simplicity and performance in our system. Since the impact of different sources and activities on the NF measurements is varying, it is important to assign different weights for different models to optimize the overall recognition accuracy. Therefore, we first introduce our proposed weight optimization methods for multiple signal sources and receivers. Then we present the detailed recognition steps and time complexity.

Weight optimization. Suppose that there are m sources and n receivers, we assign weight $w(j,k)$ for different sources and receivers. The key idea of weight optimization is to learn the optimal weight selection based on the training dataset, then the weight will be updated based on the variances of the NF samples during real-time recognition. We formulate the weight optimization problem as a minimization problem:

$$\begin{aligned} \min \quad & \sum \mathbb{1}_{\chi_i}(\hat{\chi}_i) \\ s.t. \quad & 0 \le w(j,k) \le 1 \quad \forall j \in [1,m], k \in [1,n] \qquad (a) \\ & \sum_{j=1}^{m}\sum_{k=1}^{n} w(j,k) = 1 \qquad (b) \end{aligned}$$

where $\mathbb{1}(\cdot)$ is the indicator function, χ_i and $\hat{\chi}_i$ are the ground truth and recognition result of each sample, respectively. If $\hat{\chi}_i = \chi_i$, $\mathbb{1}_{\chi_i}(\hat{\chi}_i) = 0$; otherwise $\mathbb{1}_{\chi_i}(\hat{\chi}_i) = 1$. The two constraints ensure that the weight is between 0 and 1, and the sum of all the weights is 1. Because the two constraints and the object function are all linear functions, the problem can be solved by linear programming. With the optimal weight selection from training procedure, we update the weight based on the variances of the NF measurements during the testing. If the variance from one source is low, it means that the NF measurements are rarely affected. Therefore, we will decrease the weight when the variance is low or increase the weight when the variance is high.

Recognition steps: Figure 10 shows the six recognition steps: 1) we calculate the frequency component vectors of the reconstructed NF values by using Short Time Fourier Transform (STFT); 2) we apply k-means to cluster frequency component vectors of all the samples in the training dataset; 3) the Markov state transition matrix of each gesture is updated based on the clustered state transitions for all the samples in the training dataset; 4) we apply gesture recognition with the samples in the training dataset to update the weights for different sources and receivers; 5) the c cluster centroids learned from the training process are applied to label the samples in the testing dataset with the shortest Euclidean distance; 6) finally, we calculate the log-likelihood probabilities of each sample for each gesture, and classify the sample as the gesture with aggregated maximum log-likelihood probability from all pairs of source and receiver. By doing this, the dynamically varying dimension does not affect the recognition process.

Time complexity: Assuming the max length of each sample is l. To update the models from a dataset with q samples for each source and device pair requires a time complexity of $O(c \cdot mnq \cdot l \cdot t)$ and $O(c \cdot mnq \cdot l)$ for finding c clusters in t iterations and updating the Markov chain models, respectively. Since the updating of each Markov state transition matrix can be conducted in parallel, and the clusters can be created once, the final updating time complexity is $O(clq)$.

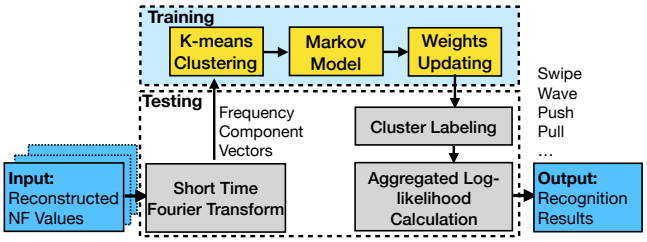


Figure 10: Recognition Steps

6 ASYNCHRONIZED MEASUREMENT ALIGNMENT

In Section 3.2, we proposed the method to distinguish signal sources by the NF measurements from multiple IoT receivers. In order to utilize the NF measurements from different IoT receivers, we need to make sure that each data point from distributed receiver is aligned. Since one design goal of EAR is to avoid introducing extra traffic, instead of using explicitly time synchronization protocols (such as TPSN[12] or FTSP[24]), we propose a measurement alignment technique which only relies on the NF measurement of ambient RF signals.

6.1 Effectiveness of Distributed NF

Different devices have different local times, the timestamps of NF measurements that capture the same motion are not aligned. Figure 11 shows an example where one ZigBee is sending packages while two sensing devices Rx1 and Rx2 measure NF. Although NF captures the same event, it is not aligned due to a local time difference. In order to align NF, we have:

***Lemma 1**: NF measurements can be synchronized if the line of sight distance between any device and ambient unknown source pair has $d < min\{c(1/f - \tau), r\}$. c is the RF signal speed, f is the NF sampling rate, τ is the multipath delay, and r is the effective communication range of the ambient source.*

Proof. Suppose there are two devices in the effective communication range of a source. After the source sends out an RF signal, the maximum multipath propagation distance to each device is $d^{(1)}_{max}$ and $d^{(2)}_{max}$ (assuming that $d^{(1)}_{max} \ge d^{(2)}_{max}$). In order for the ambient signal to arrive at both devices in the same sampling period while carrying multipath effect that captures human motion, we have $(d^{(1)}_{max} - d^{(2)}_{max})/c < 1/f$. This formula can be satisfied if $d^{(1)}_{max}/c < 1/f$. This formula can be further expressed as $d_1/c + \tau_1 < 1/f$, where $\tau_1 = (d^{(1)}_{max} - d_1)/c$. τ_1 happens to be the multipath time (or the signal pulse receiving duration) for device 1. Moreover, if this device is outside of the ambient source's communication range r, it

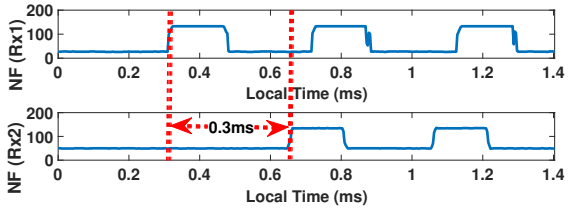
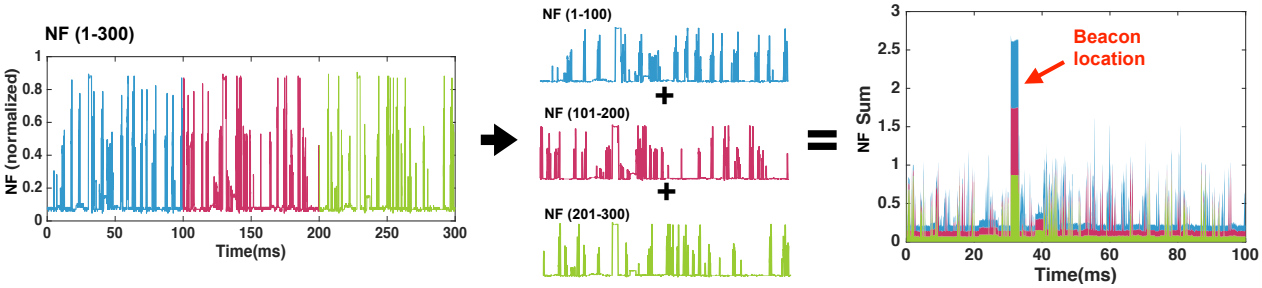


Figure 11: NF captured at the same local time on two sensing devices. Although NF is captured at the same time by two sensing devices Rx1 and Rx2, they are not aligned due to a local time difference of 0.3ms.



(a) NF measurements with beacons (b) Sum up sliced NF measurements (c) Beacon located

Figure 12: The procedure of folding operation

cannot pick up any signal. Therefore, since it applies to any pair of devices, we have lemma 1 proved. □

This lemma assures that if NF is aligned across all the IoT devices, it effectively captures and timestamps the same event on all devices.

6.2 Basic Aligning Approach

One naive solution to align the NF measurements across multiple IoT devices is to conduct time synchronization at each device by sending time synchronization packets with a certain time synchronization protocol (e.g., TPSN[12] or FTSP[24]). However, this introduces too much communication overhead and may interfere with the existing traffic.

Instead of actively synchronize the IoT devices, we propose a novel approach by passively searching the reference signal from ambient sources for data alignment. The key idea is to search a periodical reference signal which can be sensed by all the devices and use this reference signal as a starting point to align the NF measurements from different devices. For example, the beacon packets (which are sent for neighbor discovering purposes) from a WiFi AP are excellent reference signals. Since the beacon packets are sent periodically (e.g., every 100ms) and basically never change, the period parameter in IoT devices can be either manually set when deploying the system, or obtained by cross-technology communication technique (such as [9, 10, 19, 34]) passively sensed and calculated (by using the searching algorithm proposed in [36]) for time synchronization without explicitly sending packets. However, when the traffic in the environment is high, capturing them is a non-trivial problem. To solve this problem, we adopt the *folding* approach originally proposed to search for weak signals in the noisy radio[8, 13, 23].

To identify the location of beacon packets, we denote the sensed NF measurements as $NF(t)$. When these NF readings are sliced with time period P, they form a matrix $NF'(i,j)$, where $i \in [1,P]$, $j = t/P$. Then, we build the histogram $h(i)$ of $NF'(i,j)$:

$$h(i) = \sum_{n=1}^{j} NF'(i,n) \tag{9}$$

Based on the maximum value of $h(i)$, we can identify the location of beacon $i = \arg\max_k(h(k))$, because only the periodical beacon's NF measurements are aggregated together after folding. All devices thus can be synchronized.

Figure 12 shows the procedure of the *folding* operation used to locate the beacon packets. A series of NF measurements (i.e., $NF(1-300)$) sensed by a ZigBee node is shown in Figure 12(a). By looking at the figure, it is hard to find the beacon packets because there are many data and control packets being sent from unknown signal sources along with the beacon packets. Since beacon periods can be easily obtained. In this example, a WiFi beacon with a time period of $P = 100ms$ can be obtained by the configuration of WiFi AP. Thus, we slice the $300ms$ NF series into three pieces (three different colors in Figure 12(a)) and sum them as shown in Figure 12(b). The result of the summed NF measurements is shown in Figure 12(c) and we can clearly identify the beacon's location.

6.3 Advanced Aligning Approach

The folding technique is effective to align the IoT devices by only passively sense the NF even there are multiple beacons exist. However, if multiple beacons have the same or integer multiple beacon period, misalignment will occur.

For example, there are three signal sources Tx_1, Tx_2 and Tx_3 that transmit beacon $B1$, $B2$, and $B3$ at periods $p_1 = 1.6, p_2 = 1.6$ and $p_3 = 3.2$, respectively. Figure 13(a) shows the folding sums from two IoT receivers (i.e., $Rx1$ and $Rx2$) with $P = 1.6$. Because $Rx1$ and $Rx2$ are at different location that the received signal strengths are different, the maximum folding sums for $Rx1$ and $Rx2$ locates beacon $B1$ and $B2$, respectively. Apparently, the folding technique yields a misalignment.

In order to overcome this challenge, we leverage the intuition that although the magnitude of the beacon varies when the communication distance between signal source and IoT receivers are different, the spacing between beacons are unique and can be used as a signature for alignment.

Specifically, we propose a *distributed alignment pilot searching* (DAPS) algorithm (Algorithm 2) with time complexity of $O(nlogn)$ to converge the IoT devices to the same beacon (which is named pilot).

The input of this algorithm is the spacing sequence $\mathbf{s} = \{s_1, s_2, ..., s_n\}$ obtained by the folding technique. Each s_n is the space between two beacons (e.g., the s_1, s_2, and s_3 shown in Figure 13(a)). The output i is the index of the pilot. The process is as follow:

Step 1: Identify Minimal leading sequence: we find the minimal leading sequence from $\mathbf{s}$ (*Line 1*). If $|\mathbf{I}|$ is 1, the algorithm returns with this location (*Lines 2-4*). Take $Rx1$ in Figure 13(b) as an example, since the minimal spacing are $s_2 = 0.3$ and $s_3 = 0.3$, the corresponding set is $\mathbf{I} = \{2, 3\}$.

Step 2: Pilot locating. We calculate the max non-decreasing subsequence length of all minimal leading sequences. Then, the pilot is

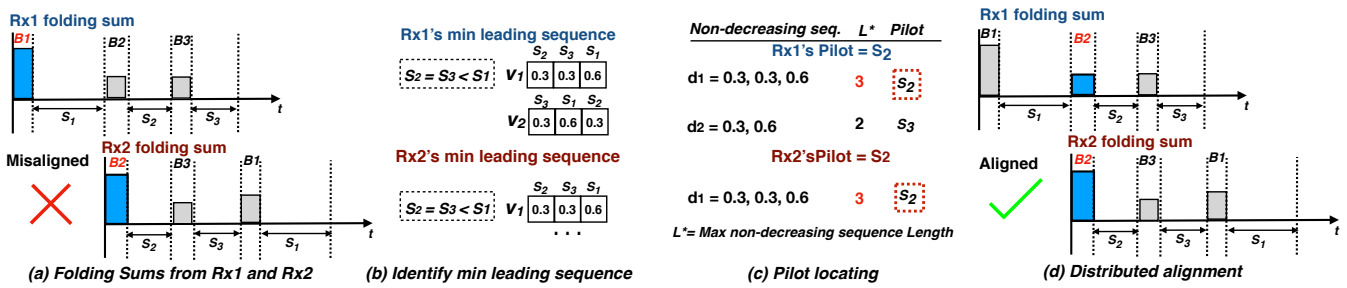


Figure 13: Distributed Aligning based on the beacon spacing signature between passive IoT receivers.

Algorithm 2 Distributed Alignment Pilot Searching

Input: $\mathbf{s} = \{s_1, s_2, \ldots, s_n\}$
Output: i

1: $\mathbf{I}$= argmin$_i \{s_i\}$
2: **if** $|\mathbf{I}| = 1$ **then**
3: **return** $\mathbf{I}_1$
4: **end if**
5: **for** $k \in 1 : |\mathbf{I}|$ **do**
6: Left circular shift $\mathbf{s}$ by $\mathbf{I}_k - \mathbf{I}_{k-1}$ to make $s_{\mathbf{I}_k}$ the first element
7: Calculate $L^*(k)$ by solving Eq. 10 with DP
8: **end for**
9: i = argmax$_k$ $L^*(k)$

located by the one with max value (*Lines 5-8*). If multiple minimal leading sequences are available, we always chose the first one.

Take Rx_1 as an example in Figure 13(b). There are two valid sequences as $\mathbf{v}_1$=[0.3, 0.3, 0.6] starting with s_2, and $\mathbf{v}_2$=[0.3 0.6 0.3] starting with s_3. In Figure 13(c), the max non-increasing subsequence length of $\mathbf{v}_1$ is 3 with the sequence as $\mathbf{d}_1$=[0.3, 0.3, 0.6]. That of $\mathbf{v}_2$ is 2 with a sequence as $\mathbf{d}_2$=[0.3, 0.6]. Therefore, by finding the index that leads to the max length among the alternatives, we can locate the pilot as s_2. The purpose of finding max non-decreasing subsequence is to let the algorithm work even with repetitive spacing sequence like [0.3 0.6 0.3] and [0.3, 0.3, 0.6] that rarely happens in real world settings.

To efficiently find the max non-decreasing subsequence length for a sequence, $\mathbf{s}$, Dynamic Programming (DP) is used to efficiently solve the problem as:

$$\mathbf{l}(i) = \begin{cases} 1 + max(\mathbf{l}(j)), & if\ s_j \le s_i, 1 < j < i \\ 1, & if \text{ no such } j \text{ exists} \end{cases} \quad (10)$$

where $\mathbf{l}(i)$ denotes the max length ending at index i. Then the max length of shifting $s_{\mathbf{I}_k}$ to the first element $L^*(k) = max(\mathbf{l})$. Then the pilot can be identified as argmax$_k$ $L^*(k)$, where $k \in \mathbf{I}$. Since the minimal beacon spacing is a small constant, the time complexity is $O(nlogn)$.

After running the algorithm, Rx_1 can be correctly aligned to the pilot from $B2$ as shown in Figure 13(d). Meanwhile, Rx_2 can also use DAPS to converge to the same pilot $B2$ as illustrated in Figure 13(d).

7 PERFORMANCE EVALUATION

In this section, we first introduce the implementation and experimental setup. Then we show the overall gesture recognition results as well as the impact of different parameters. In addition, we testify that EAR works well under various traffic rates and performance contribution of major EAR components. Finally, we evaluate a group of gross activities to show the generality of EAR.

7.1 Implementation

The implementation of EAR is divided into two parts:
Server: the signal source distinguishing (introduced in Section 3), intermittent RF traffic reconstruction (introduced in Section 4), and gesture recognition (introduced in Section 5) modules are implemented on a DELL XPS 9550 laptop.
IoT devices: the data acquisition and alignment (introduced in Section 6) is implemented on distributed ZigBee (TelosB nodes [1]) and WiFi devices (Atheros AR5xx WiFi card).

7.2 Experimental Setup

To find out how ambient RF signals work with EAR, we deployed EAR in two different scenarios as follows:
A meeting room: The first scenario is a meeting room (measured 23 feet in length and 8 feet in width) on the third floor of an academic building. As shown in Figure 14, there is a conference table and 14 chairs in the meeting room as well as a large flat screen on the wall. we deployed two laptops and two smartphones as signal sources and four EAR receivers as well.
An apartment: The second scenario is used to test with multiple heterogeneous wireless devices in an apartment. As shown in Figure 15, the apartment is measured 16 feet in length, 13 feet in width, and 7 feet in height. There is regular furniture (such as drywall, bed, table, counter, etc.) in the apartment. We deployed a WiFi camera, a WiFi AP, a laptop, a wireless camera, a smartphone, a smart TV, a desktop computer, and a robot vacuum (wireless communication enabled). In the meantime, we also deployed ZigBee devices, including a smart meter, a smart thermostat, and smart lights. These devices communicate with each other to create ambient RF traffics on 2.4 GHz (WiFi Channel 11 and overlapped ZigBee channels).

We have 10 volunteers to perform gesture for the evaluation of EAR. The heights of the volunteers range from 168cm to 190cm. We collected 11250 samples in total. The experiment is recorded by a camera to obtain ground truth. We have obtained IRB to conduct the above procedures. To avoid overfitting, a *10-fold cross validation* is used.

7.3 Recognition Results

We first show the gesture recognition accuracy in this section. A broad range of gestures is selected to test the system. The gesture type and the corresponding notations are listed in Table 2. Figure 16 shows the results in an academic building. The overall accuracy

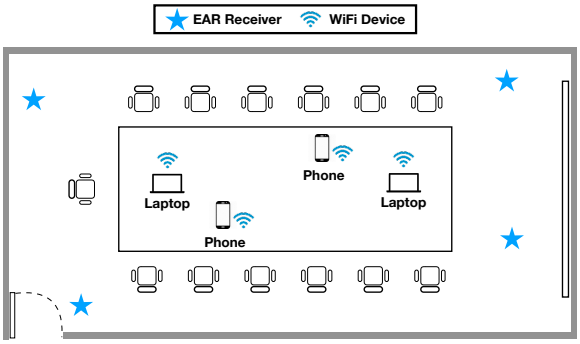


Figure 14: The Layout of A Meeting Room

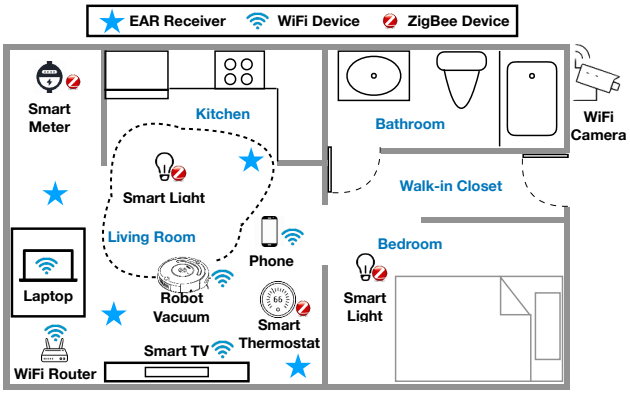


Figure 15: The Layout of An Apartment

Gesture	Notation	Gesture	Notation
Slight Kick	KK	Pull	PL
Wave	WE	Swipe	SP
Punch	PN	Raise Arms	RA
Circle	CE	Stand Still	ST
Push	PH		

Table 2: Gesture and notation

	KK	WE	PN	CE	PH	PL	SP	RA	ST
KK	86.0	1.0	0.0	1.0	2.0	10.0	0.0	0.0	0.0
WE	0.0	95.0	0.0	0.0	2.0	2.0	0.0	1.0	0.0
PN	5.0	3.0	88.1	0.0	1.0	3.0	0.0	0.0	0.0
CE	0.0	1.0	1.0	97.0	0.0	1.0	0.0	0.0	0.0
PH	0.0	2.0	0.0	0.0	92.1	5.0	0.0	1.0	0.0
PL	1.0	1.0	0.0	0.0	3.9	94.1	0.0	0.0	0.0
SP	1.0	1.0	7.8	1.0	1.0	0.0	88.3	0.0	0.0
RA	0.0	3.0	1.0	0.0	5.0	1.0	0.0	90.0	0.0
ST	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0

Figure 16: Confusion matrix of gestures recognition in the meeting room. The overall accuracy is 92.2%.

is 92.2%. For each individual gesture, the highest ("Stand Still") can achieve up to 100%. Figure 17 shows the confusion matrix of the recognition results in the apartment. The highest accuracy is "Stand Still", which achieves 100% accuracy. The lowest gesture is "Punch" which can still achieve an average of 83.2% accuracy. The possible reason is that "Punch" is too similar to gestures "Wave" and "Pull". Overall, the recognition accuracy achieves 92.63% across all types of gestures. The reason that the recognition accuracy in the apartment is slightly higher than the meeting room is that the signal sources

	KK	WE	PN	CE	PH	PL	SP	RA	ST
KK	88.0	0.0	0.0	0.0	1.0	11.0	0.0	0.0	0.0
WE	1.0	92.0	0.0	0.0	4.0	2.0	0.0	1.0	0.0
PN	5.9	4.0	83.2	1.0	0.0	5.0	0.0	1.0	0.0
CE	0.0	0.0	0.0	97.0	0.0	3.0	0.0	0.0	0.0
PH	0.0	1.0	0.0	1.0	96.0	2.0	0.0	0.0	0.0
PL	0.0	0.0	0.0	0.0	4.9	95.1	0.0	0.0	0.0
SP	0.0	1.0	4.9	1.9	2.9	0.0	89.3	0.0	0.0
RA	1.0	3.0	0.0	0.0	2.0	1.0	0.0	93.0	0.0
ST	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0

Figure 17: Confusion matrix of gestures recognition in the apartment. The overall accuracy is 92.63%.

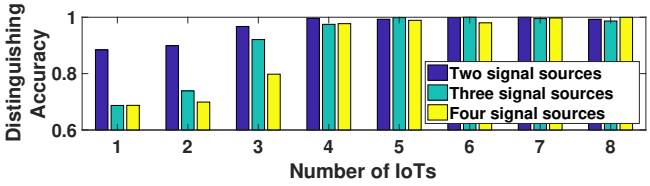


Figure 18: Source distinguishing accuracy.

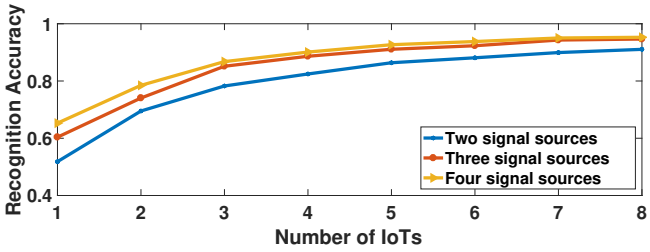


Figure 19: Impact of the number of signal sources.

in the apartment is more than those in the meeting room. We will further evaluate the impact of signal sources' number in Section 7.4.

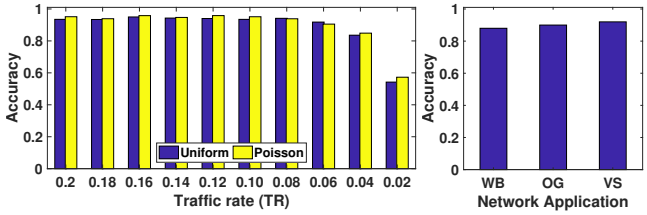
The reason EAR can achieve such a high accuracy is because that: i) EAR can accurately distinguish the ambient signal source; ii) EAR can reconstruct the intermittent RF traffic; and iii) EAR can detect the start and end of every single gesture. We will evaluate these individual modules in following sections.

7.4 Module Evaluation

In this section, we evaluate the performance of different modules as introduced in Section 3, 4, 5, and 6.

Signal Source Distinguishing: We first evaluate the signal source distinguishing accuracy. Figure 18 shows that when there are only two signal sources, EAR can achieve at least 90% accuracy to identify them even with only one receiver. When the number of receivers is greater than three, EAR can easily distinguish all sources with an accuracy up to 99.76% ($mean$ = 99.20%) regardless whether of the number of sources. Overall, EAR can distinguish different sources by cooperating the IoT receivers.

Furthermore, we evaluate how the number of signal sources and IoT receivers impact the overall gesture recognition performance in terms of recognition accuracy. As shown in Figure 19, with more ambient signal sources and receivers, the overall accuracy increases up to 96.14%. The results illustrate that EAR not only effectively distinguishes the signal source, but also utilizes the



(a) Uniform and Poisson Distribution (b) Real Applications

Figure 20: Accuracy Comparison under different traffic type (uniform, Poisson distribution and real network application) and traffic rates (from 0.02 to 0.2).

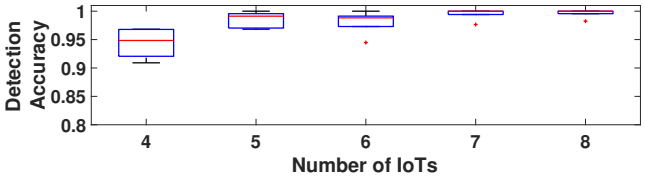


Figure 21: Detection accuracy: When the number of IoT receivers is greater than four, EAR can achieve detection accuracy up to 99.6%.

data from different signal sources and IoT receivers to increase the recognition performance.

Intermittent RF Traffic Reconstruction: The traffic type and rate in the air will affect the recognition performance. To mitigate the impact of traffic rate, we designed the intermittent RF traffic reconstruction module (introduced in Section 4). In Figure 20(a), we evaluate how this module impact the recognition performance when the traffic is under uniform and Poisson distribution with the traffic rate changes from 0.2 to 0.02. To conduct the experiments, we tune the traffic rate by controlling the generated traffic (by using iPerf tool) on the laptop. We used an USRP to measure the traffic rates (defined as channel busy time divided by total time) in the air. The result shows the overall gesture recognition accuracy is higher than 90.0% when the traffic rate is larger than 0.04 for both of the two types of traffic distribution. We further evaluated the system under three types of real network applications: i) web browsing (WB); ii) online gaming (OG); and iii) video streaming (VS). The result in Figure 20(b) shows that all of them can achieve the recognition accuracy higher than 90%.

Gesture Detection: To evaluate the gesture detection accuracy , we utilize the data from different numbers of IoT receivers in the environment. The overall detection accuracy of ten participants is shown in Figure 21. We observe that EAR has rather robust performance over different participants. The detection accuracy increases with the number of receivers for most of the participants. When the receiver number is larger than four, the overall detection accuracy reaches up to 99.6%. The reason is that with more dimensions (each dimension is the data stream from one receiver), the PCA (introduced in Section 5) provides larger variance between absence and presence of human motion. Thus, the accuracy increases as the number of IoT receivers increases.

Asynchronized Measurements Alignment: We evaluate the time efficiency of asynchronized measurement alignment (Section 6) in this section. Intuitively, with higher traffic rate, the alignment approach proposed in Section 6 is harder because the folding sum of beacons stand out. Figure 22 shows the probability distribution of successful alignment under different traffic rate (TR). Overall, the NF measurements from distributed IoT receivers can be synchronized with a median of 3 beacon periods (corresponding to 300ms) . Even though the number of beacons required to align the measurement increases with the increasing of TR, at most 7 beacon periods (corresponding to 700ms) are required.

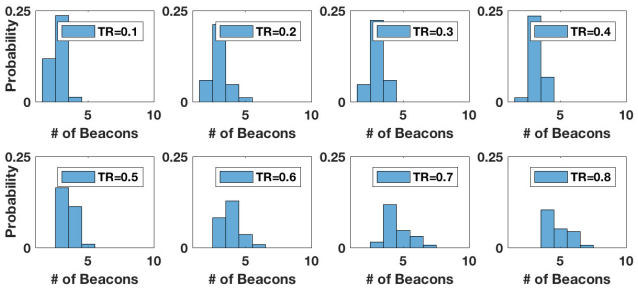


Figure 22: Alignment efficiency: EAR can finish the alignment within 7 beacon periods (corresponding to 700ms) even in the worst case (TR=0.8).

Motion	Notation	Motion	Notation
Dodge	DE	Deep Squat	DS
Jump	JP	Raise Dumbbell	RD
Walk	WK	Sit Down	SI
Run	RN	Stand Still	ST

Table 3: Gross Human Motion and Notation

	DE	JP	WK	RN	DS	RD	SI	ST
DE	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
JP	5.9	91.1	1.0	0.0	2.0	0.0	0.0	0.0
WK	0.0	0.0	99.0	1.0	0.0	0.0	0.0	0.0
RN	0.0	0.0	10.0	90.0	0.0	0.0	0.0	0.0
DS	0.0	1.9	0.0	0.0	97.1	0.0	1.0	0.0
RD	2.0	0.0	0.0	0.0	0.0	98.0	0.0	0.0
SI	1.0	1.0	0.0	0.0	4.0	0.0	94.0	0.0
ST	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0

Figure 23: Confusion matrix of gross human motion. The overall recognition accuracy is 96.15%.

7.5 Gross Human Motion

To show the generality of EAR, we also evaluated a group of gross human motion (i.e., movement and coordination of the arms, legs, and other large body parts). The motion types and corresponding notations are shown in Table 3. From the results in Figure 23, we can observe that the recognition accuracies for all the gross motions are higher than 90%. The overall accuracy is 96.15% which is slightly higher than gesture recognition. The reason is that the gross motion makes more reflection of the RF signals.

8 RELATED WORK

Gesture recognition has lots of potential applications (e.g., smart home, gaming, and remote control). Traditional approaches using cameras may reveal private information [6]. The mature voice command systems (e.g., Alexa and Google Home) are limited in some scenarios such as i) in a noisy environment (e.g., when playing loud music, the voice command system does not work well); ii) in a quiet environment (such as a baby bedroom); iii) for disabled people. Researchers have developed many wearable device systems to recognize human activities [17, 27]. In addition, some researchers have utilized the wearable devices for human activity recognition to provide real-time authenticating service [18, 22]. On the other hand, there are a lot of challenges (e.g., energy efficiency and users' comfort) before making them widely adopted. To overcome these issues, lots of device-free human gesture recognition systems have been proposed by utilizing wireless signals. For example, [26] uses a microphone array, passive infra-red sensors, and illumination sensors to detect activities in a single office room. AALO [15] uses in-home sensors to recognize human activities from room to room. Recently, researchers have proposed to use radio frequency (RF) signals for human gesture recognition. Based on the features of the RF signals, these RF based recognition systems can be divided into three categories:

RSS-based. By demodulating and decoding the received packets, the receiver monitors the change of Received signal strength (RSS) from a specific sender caused by human movement. Harmony [11] achieves up to 90% activity recognition accuracy . As presented in [3, 29, 30], researchers also utilize RSS to recognize gestures. In addition, RSS can be utilized to monitor respiration [2, 16].

CSI-based. Compared with the RSS-based approach, Channel state information (CSI) is a fine-grained measurement and implemented in some specific wireless communication systems (e.g., WiFi). The receiver extracts the CSI information from the received signal to conduct recognition. By leveraging CSI, the recognition accuracy is higher than RSS-based system [33, 35]. Wisture [14] utilizes only WiFi beacons to recognize gesture on a smartphone. Moreover, CSI information is used to hear humans talk [32], recognize keystrokes [7], and obtain gait information [20, 21].

Specific hardware-based. Researchers have also developed specific hardware to recognize activities and gestures [4]. WiSee uses USRP to extract Doppler shift from WiFi signals to perform gesture recognition. By using a large size FFT and two antennas system, Mudra [38] increases the frequency resolution to extract the subtle changes caused by finger. To utilize signals from different WiFi sources, Mudra decodes the WiFi packet to distinguish the incoming signals. By using some specific antenna, it is even achievable to monitor human breathing and heart rate [5]. Some designated RFID sensors can also be utilized to recognize human activities [25] and gestures [28]. Aegis [37] is proposed to recognize human activities while preventing privacy leaking to the adversary.

Different from the existing work, by sensing the noise floor, our work is able to utilize ambient uncontrollable RF signal sources from heterogeneous IoT devices for gesture recognition without generating designated sensing traffic. EAR addresses three unique challenges that arise from this setting, including i) how to recognize and distinguish signal sources from different IoT devices without demodulating and decoding the packets by using the noise floor measurements from low-end IoT devices; ii) how to mitigate the impact of extremely low traffic rate; and iii) how to align the measurements from different receivers without explicitly sending time synchronization messages. Penitentially, EAR can be developed to be a middleware which bridges the input of ambient uncontrollable RF signal sources and existing RF sensing systems.

9 DISCUSSION

9.1 Developing EAR to Middleware

The most important contribution of EAR is utilizing uncontrollable ambient signals for human gesture recognition and the most important designs are distinguishing uncontrollable RF signal sources (Section 3) and intermittent traffic reconstruction (Section 4). Since the design of these two parts are generic that i) the design has no limit on the input of signal sources' number and receivers' number and ii) the design does not assume the purpose of the output signals, it is possible to plug in EAR between existing RF sensing system and the data source. As long as the RF sensing system is built upon signal strength measurements, EAR can help the system to combat ambient sources and multiple sources.

9.2 Negligible Traffic Introduced

EAR is designed to sense human gesture without introducing designated RF sensing traffic. To collect the data on distributed IoT receiver, the introduced traffic is negligible because of the following three aspects: i) EAR works with as little as two receivers which means only the data on one device needs to be transmitted to the other; ii) the volume of noise floor measurements is very tiny (only 8 bits on a time point while CSI is 64bits*2(complex number)*30(groups)*9(3×3 MIMO antennas)) that can be easily piggybacked on existing traffic; and iii) EAR can also be deployed on existing wire connected platform such as desktop because it uses widely available noise floor measurements.

10 CONCLUSION

This paper presents EAR, a system that leverages existing ambient RF signal from uncontrollable signal sources for high accurate gesture recognition. EAR achieves this without introducing any extra RF traffics to the IoT infrastructure. Therefore, EAR meets the pressing need for spectrum efficiency in the era of ever increasing number of IoT devices while enabling high accurate gesture recognition.

With the exponentially increasing number of IoT devices and the huge amount of data traffic generated by these devices, we have designed the system EAR to recognize human gesture and gross motions by only listening to the existing ambient RF traffic. We have conducted extensive experiments in a real-world apartment and performed 11,2250 instances of gestures and gross motions. The signal sources distinguishing and human gesture recognition accuracies are up to 99.76% and 92.63%, respectively.

ACKNOWLEDGMENTS

This project is supported by NSF grants CNS-1539047 and CNS-1652669. We also thank anonymous reviewers and shepherd for their valuable comments.

REFERENCES

[1] [n. d.]. http://tinyos.stanford.edu/tinyos-wiki/index.php/TelosB.

[2] Heba Abdelnasser, Khaled A. Harras, and Moustafa Youssef. 2015. UbiBreathe: A Ubiquitous non-Invasive WiFi-based Breathing Estimator. In *MobiHoc*.

[3] Heba Abdelnasser, Moustafa Youssef, and Khaled A Harras. 2015. Wigest: A ubiquitous wifi-based gesture recognition system. In *INFOCOM*.

[4] Fadel Adib, Zachary Kabelac, Dina Katabi, and Robert C. Miller. 2014. 3D Tracking via Body Radio Reflections. In *NSDI*.

[5] Fadel Adib, Hongzi Mao, Zachary Kabelac, Dina Katabi, and Robert C. Miller. 2015. Smart Homes That Monitor Breathing and Heart Rate. In *CHI*.

[6] J.K. Aggarwal and M.S. Ryoo. 2011. Human Activity Analysis: A Review. In *ACM Comput. Surveys*.

[7] Kamran Ali, Alex X. Liu, Wei Wang, and Muhammad Shahzad. 2015. Keystroke Recognition Using WiFi Signals. In *MobiCom*.

[8] Z. Chi, Z. Huang, Y. Yao, T. Xie, H. Sun, and T. Zhu. 2017. EMF: Embedding multiple flows of information in existing traffic for concurrent communication among heterogeneous IoT devices. In *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*. 1–9. https://doi.org/10.1109/INFOCOM.2017.8057109

[9] Zicheng Chi, Yan Li, Hongyu Sun, Yao Yao, Zheng Lu, and Ting Zhu. 2016. B2W2: N-Way Concurrent Communication for IoT Devices. In *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM (SenSys '16)*. ACM, New York, NY, USA, 245–258. https://doi.org/10.1145/2994551.2994561

[10] Z. Chi, Y. Li, Y. Yao, and T. Zhu. 2017. PMC: Parallel multi-protocol communication to heterogeneous IoT radios within a single WiFi channel. In *2017 IEEE 25th International Conference on Network Protocols (ICNP)*. 1–10. https://doi.org/10.1109/ICNP.2017.8117550

[11] Zicheng Chi, Yao Yao, Tiantian Xie, Zhichuan Huang, Michael Hammond, and Ting Zhu. 2016. Harmony: Exploiting coarse-grained received signal strength from IoT devices for human activity recognition. In *ICNP*.

[12] Saurabh Ganeriwal, Ram Kumar, and Mani B. Srivastava. 2003. Timing-sync Protocol for Sensor Networks. In *SenSys*.

[13] T. Hao, R. Zhou, G. Xing, and M. Mutka. 2011. WizSync: Exploiting Wi-Fi Infrastructure for Clock Synchronization in Wireless Sensor Networks. In *2011 IEEE 32nd Real-Time Systems Symposium*. 149–158. https://doi.org/10.1109/RTSS.2011.21

[14] Mohamed Abudulaziz Ali Haseeb and Ramviyas Parasuraman. 2017. Wisture: RNN-based Learning of Wireless Signals for Gesture Recognition in Unmodified Smartphones. *CoRR* abs/1707.08569 (2017). arXiv:1707.08569 http://arxiv.org/abs/1707.08569

[15] Enamul Hoque and John Stankovic. 2012. AALO: Activity recognition in smart homes using Active Learning in the presence of Overlapped activities. In *PervasiveHealth*.

[16] Ossi Johannes Kaltiokallio, Hüseyin Yigitler, Riku Jäntti, and Neal Patwari. 2014. Non-invasive Respiration Rate Monitoring Using a Single COTS TX-RX Pair. In *IPSN*.

[17] B. Li, M. Ayazoglu, T. Mao, O. I. Camps, and M. Sznaier. 2011. Activity recognition using dynamic subspace angles. In *CVPR 2011*. 3193–3200. https://doi.org/10.1109/CVPR.2011.5995672

[18] S. Li, A. Ashok, Y. Zhang, C. Xu, J. Lindqvist, and M. Gruteser. 2016. Whose move is it anyway? Authenticating smart wearable devices using unique head movement patterns. In *2016 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. 1–9. https://doi.org/10.1109/PERCOM.2016.7456514

[19] Yan Li, Zicheng Chi, Xin Liu, and Ting Zhu. 2018. Chiron: Concurrent High Throughput Communication for IoT Devices. In *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys '18)*. ACM, New York, NY, USA, 204–216. https://doi.org/10.1145/3210240.3210346

[20] Yan Li and Ting Zhu. 2016. Gait-Based Wi-Fi Signatures for Privacy-Preserving. In *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security (ASIA CCS '16)*. ACM, New York, NY, USA, 571–582. https://doi.org/10.1145/2897845.2897909

[21] Y. Li and T. Zhu. 2016. Using Wi-Fi Signals to Characterize Human Gait for Identification and Activity Monitoring. In *2016 IEEE First International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE)*. 238–247. https://doi.org/10.1109/CHASE.2016.20

[22] A. A. Liu, N. Xu, W. Z. Nie, Y. T. Su, Y. Wong, and M. Kankanhalli. 2017. Benchmarking a Multimodal and Multiview and Interactive Dataset for Human Action Recognition. *IEEE Transactions on Cybernetics* 47, 7 (July 2017), 1781–1794. https://doi.org/10.1109/TCYB.2016.2582918

[23] R Lovelace, J Sutton, and E Salpeter. 1969. Digital Search Methods for Pulsars. In *Nature*.

[24] Miklós Maróti, Branislav Kusy, Gyula Simon, and Ákos Lédeczi. 2004. The Flooding Time Synchronization Protocol. In *SenSys*.

[25] Matthai Philipose Michael Buettner, Richa Prasad and David Wetherall. 2009. Recognizing Daily Activities with RFID-Based Sensors. In *UbiComp*.

[26] Homin Park, Jongjun Park, Hyunhak Kim, Jongarm Jun, Sang Hyuk Son, Taejoon Park, and JeongGil Ko. 2015. ReLiSCE: utilizing resource-limited sensors for office activity context extraction. In *IEEE Transactions on Systems, Man, and Cybernetics: Systems*.

[27] Nam Pham and Tarek Abdelzaher. 2008. Robust Dynamic Human Activity Recognition Based on Relative Energy Allocation. In *Distributed Computing in Sensor Systems*, Sotiris E. Nikoletseas, Bogdan S. Chlebus, David B. Johnson, and Bhaskar Krishnamachari (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 525–530.

[28] Zhou Shangguan, L.and Z. and Jamieson K. 2017. Enabling Gesture-based Interactions with Objects. In *MobiSys*.

[29] S. Sigg, U. Blanke, and G. Tröster. 2014. The telepathic phone: Frictionless activity recognition from WiFi-RSSI. In *PerCom*.

[30] Li Sun, Souvik Sen, Dimitrios Koutsonikolas, and Kyu-Han Kim. 2015. WiDraw: Enabling Hands-free Drawing in the Air on Commodity WiFi Devices. In *MobiCom 2015*.

[31] Cisco Systems. 2014-2019. Cisco Global Cloud Index: Forecast and Methodology, 2014-2019 White Paper.

[32] Guanhua Wang, Yongpan Zou, Zimu Zhou, Kaishun Wu, and Lionel M. Ni. 2014. We Can Hear You with Wi-Fi!. In *MobiCom*.

[33] Wei Wang, Alex X. Liu, Muhammad Shahzad, Kang Ling, and Sanglu Lu. 2015. Understanding and Modeling of WiFi Signal Based Human Activity Recognition. In *MobiCom*.

[34] W. Wang, T. Xie, X. Liu, and T. Zhu. 2018. ECT: Exploiting Cross-Technology Concurrent Transmission for Reducing Packet Delivery Delay in IoT Networks. In *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*.

[35] Yan Wang, Jian Liu, Yingying Chen, Marco Gruteser, Jie Yang, and Hongbo Liu. 2014. E-eyes: Device-free Location-oriented Activity Identification Using Fine-grained WiFi Signatures. In *MobiCom*.

[36] Y. Xiong, R. Zhou, M. Li, G. Xing, L. Sun, and J. Ma. 2014. ZiFi: Exploiting Cross-Technology Interference Signatures for Wireless LAN Discovery. *IEEE Transactions on Mobile Computing* 13, 11 (Nov 2014), 2675–2688. https://doi.org/10.1109/TMC.2014.2309610

[37] Y. Yao, Y. Li, X. Liu, Z. Chi, W. Wang, T. Xie, and T. Zhu. 2018. Aegis: An Interference-Negligible RF Sensing Shield. In *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*.

[38] Ouyang Zhang and Kannan Srinivasan. 2016. Mudra: User-friendly Fine-grained Gesture Recognition Using WiFi Signals. In *CoNext*.