How Can I Guard My AP? Non-intrusive User Identification for Mobile Devices Using WiFi Signals

Linsong Cheng and Jiliang Wang
School of Software and TNList
Tsinghua University, China
chengls14@mails.tsinghua.edu.cn,
jiliangwang@tsinghua.edu.cn

ABSTRACT

With the development and popularization of WiFi, surfing on the Internet with mobile devices has become an indispensable part of people's daily life. However, as an infrastructure, WiFi APs are easily connected by some undesired users nearby. In this paper, we propose NiFi, a non-intrusive WiFi user identification system based on WiFi signals that enables AP to automatically identify legitimate users in indoor environment such as home, office and hotel. The core idea is that legitimate and undesired users may have different physical constraints, e.g., moving area, walking path, etc, leading to different signal sequences. NiFi analyzes and exploits the characteristics of signal sequences generated by mobile devices. NiFi proposes a practical and effective method to extract useful features and measure similarity for signal sequences, while not relying on precise user location information. We implement NiFi on Commercial Off-The-Shelf (COTS) APs, and the implementation does not require any modification to user devices. The experiment results demonstrate that NiFi is able to achieve an average identification accuracy at 90.83% with true positive rate at 98.89%.

CCS Concepts

•**Networks** → **Network components;**

General Terms

Design, Experimentation, Security, Performance

Keywords

User Identification; Mobile Device; AP; WiFi Signals

1. INTRODUCTION

1.1 Motivation

MobiHoc'16, July 04-08, 2016, Paderborn, Germany

DOI: http://dx.doi.org/10.1145/2942358.2942373

Nowadays, WiFi has become a fundamental part for providing wireless connection. According to ABI research's report [5], WiFi chip shipment reaches near 18 billion from 2015 to 2019. With the development and popularization of mobile devices equipped with WiFi chip, e.g., laptops, tablets and smartphones, WiFi becomes even more important for ubiquitous wireless access.

Normally, a device needs to first connect to a WiFi access point (AP) in order to use the network service. However, a well-known problem is that an AP may be connected and used by some undesired users nearby, which slows down the network speed and brings harm to legitimate users. For example, it has been reported [7] that the Internet fee loss caused by undesired users reached up to 5 billions RMB in China every year. Meanwhile, undesired users may even result in privacy and security issues. The most common way for AP connection control is to set a password. However, according to a security report of Rising Antivirus [1], a large portion of passwords for APs are too simple. 86% of users never login to the setting page after installing a AP, and 92% of users do not change the default password (e.g. "admin" and "root"). 73% of users choose an easy-to-guess or simple password (e.g "12345678"). Even worse, there exist a large portion of users, especially for those non-experts, who never set password for their APs. Even for complicated passwords, there are cracking softwares and automatic password sharing softwares [6] based on crowdsourcing, making password based authorization invalid. Therefore, automatic user identification becomes more and more important, especially as the increasing of non-expert WiFi AP users.

The AP vendors (e.g., Huawei, D-Link, TP-Link) have noticed those problems. They have largely simplified AP password setting process and encourage users to set password for each AP. They are also seeking to design automatic methods for user identification. They propose some smart APs (e.g., MiWiFi from XiaoMi, HiWiFi, etc) that claim to be able to distinguish legitimate users and undesired users. Those smart APs require users to set a white and black list based on MAC addresses.

1.2 Proposed Approach

We propose NiFi, an automatic approach which requires no user configurations, to identify legitimate WiFi users of mobile devices. NiFi seeks to exploit the signal characteristics from different users. We find that legitimate users and undesired users may have very different physical constraints, e.g., moving area, walking path, leading to different signal sequences. Though signal characteristics from users at a s-

ingle location may be similar, signal sequences for users at different areas can be very different.

NiFi analyzes and exploits the characteristics of signal sequence generated by mobile devices, and extracts useful features from different users. Based on those features, NiFi proposes a similarity measurement algorithm for user identification. Meanwhile, NiFi maintains a feature database for legitimate users for similarity measurement and propose a novel graph based online database update method.

Overall, NiFi can automatically identify legitimate users without explicit password. NiFi can be deployed on commercial WiFi AP while not requiring any modification to user devices such as mobile phone and laptop. Undesired users is difficult to use an AP with NiFi since physical signal sequence is difficult to mimic.

We implement NiFi on a COTS wireless router board (RB912UAG-2HPnD [3]) as a prototype system. We also conduct extensive experiments with different devices in different environments. Experiment results show that NiFi achieves an average user identification accuracy at 90.83% with false negative rate at 1.11% and false positive rate at 17.22%. Such a result enables NiFi to be used for many different application scenarios, e.g., for a hotel or restaurant that are willing to provide convenient and exclusive WiFi access for guests. We also provide tunable parameters to adjust the false positive rate and false negative rate. Our current setting favors a low false negative rate while allowing a certain false positive rate.

1.3 Technical Challenges and Solutions

Practically, if each connected user can be precisely located, NiFi is easy to implement. However, precise localization is difficult to obtain, especially when complex physical layer information (e.g., CSI) and pre-collected signal fingerprints are unavailable. NiFi is required to be used without precise localization and thus its design has several challenges.

The first challenge is how to extract useful user features without precise location information. Instead of precise location, NiFi uses the signal sequence from each user. However, a user may spend different time at different positions. Even for two legitimate users, one may stay in one room while the other moves between different rooms, resulting in different signal sequences. We model the signal sequence from each user as a signal transition path (STP), and extract key features based on an iterative change point detection algorithm.

The second challenge is how to construct and maintain the feature database for user identification, and how to measure the similarity. There are limited initial data of legitimate users in order to reduce the deployment overhead. It is also difficult to obtain the training set that covers all possible paths. We design a path merging method to combine multiple paths from legitimate users to a signal transition graph (STG). Then we transfer user identification to the problem of matching a STP in the STG. If a user is legitimate, we can update the STG with the path merging method. To match an STP in the STG, we propose two different methods for measuring the similarity for vertices or edges. Then we propose a DFS-based path matching algorithm to find the maximum similarity score for the STP on the STG.

The third challenge is to deal with device diversity. Different phones may have different transmission power, antenna layout and hardware configuration. Therefore, the received signal strength from different phones at the same position may even be different. This is also examined in our experiment in Section 6.3. We further compare different devices and find that the shift between the signals for two different phones at different environments is relatively stable. Based on this finding, we propose a shift-cancelation approach to mitigate the impact of device diversity.

1.4 Key Contributions

The main contribution of this paper is as follows:

- We first propose to use signal information collected at AP for user identification. We propose NiFi, an approach that enables automatic AP user identification.
- We present detailed practical analysis for signal from different devices at AP. In NiFi, we transfer the user identification problem to a path matching problem in signal space. We further propose effective similarity measure methods for signal sequences and path matching algorithm for user identification.
- We implement NiFi as on COTS routers and evaluate its performance for different mobile devices in different environments.

The remainder of this paper is structured as follows. Section 2 introduces some related works in recent years. Section 3 presents an overview of NiFi's architecture. Section 4 elaborates on data collection and noise removal. Then, Section 5 and Section 6 describes NiFi's identification process in detail. Section 7 presents our implementation of NiFi on COTS routers and a comprehensive experimental evaluation. Finally, Section 8 concludes the paper.

2. RELATED WORK

2.1 Indoor Localization

A large body of indoor localization approaches based on WiFi signals have been proposed in the past two decades. Many methods such as RADAR [11], Horus [30] and OIL [16] leverage existing WiFi infrastructure to build a fingerprint database for indoor localization. Further, several systems such as LiFS [29], Zee [18] and UnLoc [24] use crowdsourcing to collect signal fingerprints. Meanwhile, those approaches assumes multiple APs in the environment and a user device can obtain signal from multiple APs. Some recent works use more complex physical layer information, e.g., channel state information (CSI), to obtain more fundamental location related information, e.g., the angle-of-arrival. There are also approaches using complicate analysis methods or special hardware [19, 14, 28], e.g., antenna array [27, 9]. With fine-grained signal information, Chronos [22] can even achieve decimeter-level localization with a single WiFi AP.

Our work is inspired by those localization methods. In our work, we seek to achieve user identification from the AP side. We find that precise location information is difficult to obtain especially with COTS WiFi APs. Nevertheless, we also find that that precise location information is not required towards our goal. Accordingly, we also do not use complex physical layer information or infrastructure, which are not commonly available for nowadays WiFi application scenarios.

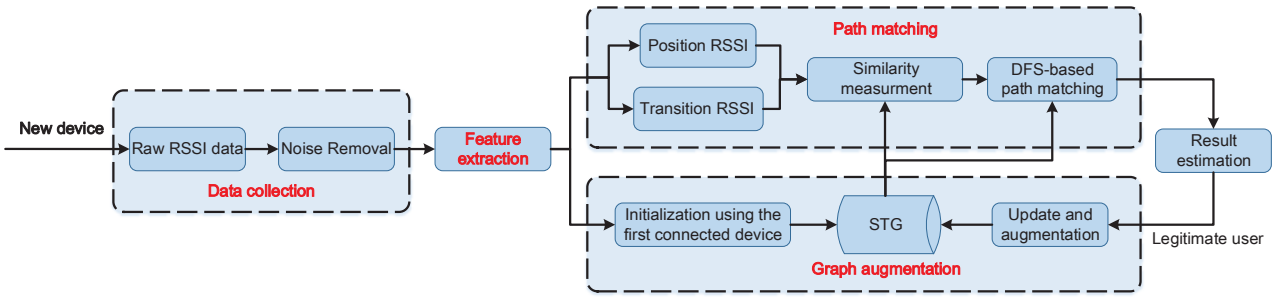

Figure 1: System architecture

2.2 Fine-grain Signal Based Activity Recognition

There are also a large collection of activity recognition approaches based on wireless signals to recognize human activities [20, 21, 17, 15, 8, 10, 23, 25, 26]. Most of those approaches are based on fine-grained channel state information (CSI). More specifically, different user activities may have different impacts on wireless siginal and thus result in different CSI values. For example, some early approaches focus on recognizing macro-movements such as motions(crawling, lying down, standing up, and walking) [20, 21], and gestures [17, 15, 8]. Recently, several works are proposed to recognize micro-movements. WiKey [10] utilizes the patterns in the time-series of CSI values to recognize keystrokes using a laptop with Intel Link 5300 WiFi NIC for CSI collection. WiHear [23] detects and analyzes fine-grained signal reflections from mouth movements while speaking. RF-IDraw [25] constructs a virtual touch screen in the air using RF signals. Those approaches are used for recognize small user activities with special hardware. Meanwhile, they often require a training set. Thus they may not be appropriate for user identification especially on COTS APs.

3. NIFI OVERVIEW

In this section, we introduce NiFi's design goals and an overview of the system architecture.

Overall, the design goals of NiFi are as follows.

- NiFi should reside on the AP side to automatically identify the legitimate user.
- NiFi should be non-intrusive, requiring no user active participation or user device modification.
- NiFi uses physical signal information from users which is difficult to mimic, and it does not require password input from users.

3.1 System Architecture

The overall system architecture is shown in Figure 1. The working process of NiFi consists of four major components.

Data collection: The first component is the Data Collection component. In this component, NiFi collects raw WiFi signal data from different users. Currently, NiFi works on APs with OpenWrt system and uses commands provided by OpenWrt to collect RSSI information. It is noted that NiFi can also leverage other types of signal information. NiFi then groups all the RSSI information according to MAC address of the corresponding packet. Meanwhile, as there exists noise in the collected RSSI sequence, we need to perform noise removal in this component.

Feature extraction: The second component is the feature extraction component. After data collection, NiFi has a RSSI sequence for each user, which contains the RSSI values for users at different positions. Therefore, we partition the RSSI sequence into groups according to user's position and moving status. More specifically, we seek to group RSSI values for the same position together (position RSSI group), and group RSSI values for a user moving between two different positions together (transition RSSI group). We model a position RSSI group as a vertex and a transition RSSI group as a edge. Then the original RSSI sequence can be modeled as a signal transition path (STP) consisting of vertexes and edges in between.

Graph augmentation: In this component, we aim to build a signal transition graph (STG) based on STPs of all legitimate users. We design a graph augmentation algorithm to update the STG with the STP of legitimate users. Initially, we can consider the first connected user as a legitimate user. Accordingly, we use the STP of the first user to construct the initial STG. Then, we augment and update the STG gradually when new legitimate STPs are identified. The detailed graph augmentation algorithm is introduced in Section 6.4.

Path matching: We design a path matching algorithm to solve the user identification problem. After we have built the STG, we can match a new coming STP (the result from feature extraction) on STG. If there is a match for the STP according to the path matching algorithm, the corresponding user is considered as legitimate. It should be noted that, in the path matching algorithm, we have different matching methods for the vertexes and edges according to their properties. For example, for an edge (transition RSSI group), we consider not only its absolute RSSI values and statistics, but also its trend and so on. The detailed path matching algorithm is introduced in Section 6.2.

3.2 Legitimate Users and Undesired Users

First, we consider there are legitimate areas (e.g., user-defined). All other areas are considered as undesired areas. Users in the legitimate areas are considered as legitimate users. In our implementation, we consider the first connected user as the first legitimate user (e.g., the first user is the

administrator who installs the AP). We discuss the impact of data from the first connected user in 7.5.

4. DATA COLLECTION AND NOISE REMOVAL

In this section, we introduce our data collection and noise removal process. We assume legitimate users and undesired users have different physical constraints, e.g., moving area, walking path. For example, in home environment, legitimate users may appear in rooms of a certain home. Tabel 1 summarizes the symbols used in this paper.

4.1 Data Collection

NiFi collects RSSI signal sequences on AP from different users. NiFi has no special requirements on hardware. We use a wireless router board (RB912UAG-2HPhD [3]) with OpenWrt system. OpenWrt is an operating system based on the Linux kernel, primarily and widely used on many routers and APs.

4.2 Noise Removal

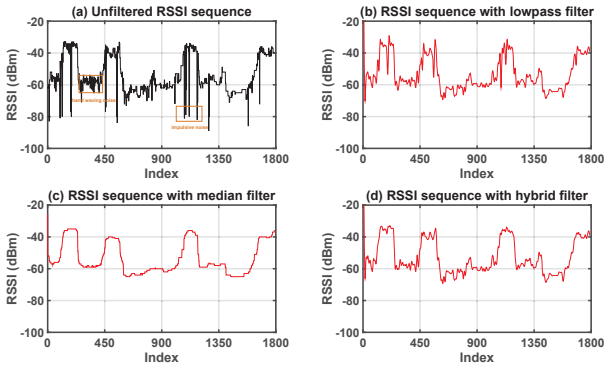

Figure 2: Unfiltered and filtered RSSI sequence

As we can see in Figure 2 (a), there are three kinds of noise in the RSSI sequence. The first type of noise is the slight RSSI fluctuation due to environmental gaussian noise. This can be observed in normal collected RSSI sequence. The second type of noise is caused by small environment changes such as device perturbation in one's hand or people walking nearby. Besides, we also notice there exists some impulsive noise in the RSSI sequence for some type of mobile phones (e.g., Xiaomi Mi3). We investigate the data and think this may be related to the antenna layout and hardware design in this type of phone. For a specific angle, the emitted signal arrived at the AP becomes very small.

For the first two types of noise, we consider them as high frequency and design a lowpass filter (e.g., Butterworth filter) to remove them. For example, we assume that device perturbation on human hands is usually small and quick. More specifically, as in [10], we assume that the hand and finger movement approximately lies between 0.5Hz to 80Hz. As we sample RSSI values at a rate of $F_s = 10$, we accordingly set the cut-off frequency ω_c of the Butterworth filter at $\omega_c = \frac{2\pi \times f}{F_s} = \frac{2\pi \times 0.5}{10} \approx 0.31$rad/s. However, a lowpass filter can not remove the impulsive noise well as Figure 2 (b) shows.

Table 1: Symbols in this paper

Symbol	Description
f_i	Collected RSSI sequence F
r_i	Filtered RSSI sequence R
s_i	Cumulative sums
Δ	Position RSSI group shift
D	Max distance between two empirical distributions
P	a path with n vertexes and n-1 edges
V	# of STG's vertexes
E	# of STG's edges
V_k	The k^{th} vertex of STG
E_k	The k^{th} edge of STG
S	The best matching score
$M(v_i, v_i')$	Similarity of two position RSSI groups
$C(e_i, e_i')$	Similarity of two transition RSSI groups
λ	Threshold value of a legitimate user
γ	Threshold value of Position RSSI group shift
δ_{min}	Low similarity threshold value of two vertexes
δ_{max}	High similarity threshold value of two vertexes
β	A factor for adjusting the upper limit of scoring
σ	Threshold score of terminating the current search

For the third type of impulsive noise, a median filter is particularly effective as shown in Figure 2 (c). However, for a median filter, it's difficult to choose an appropriate window size to remove all the three types of noise without losing detailed characteristics.

Therefore, in this step, we first pass the sequence to a median filter. We use a small window size for the median filter, e.g., 20, in order to maintain the original data characteristics. Then, the result is passed to a lowpass filter. The final result is shown in Figure 2 (d). It should be noted that after filtering, we only remove those high frequency noise. There may still exist low frequency noise, e.g., noise due to surrounding people's movement. Therefore, there may still exist fluctuations after filtering. We will address those remaining fluctuations in Section 5.2.

5. FEATURE EXTRACTION & SIMILARITY MEASUREMENT

In this section, we describe the feature extraction process and similarity measurement methods in NiFi.

5.1 Feature Extraction

To extract features, we first partition a RSSI sequence into groups according to user's position and moving status. As shown in Figure 3 (a), we find that the original RSSI sequence consists of a series of sub-sequences of two types. In the first type of subsequence, most of the RSSI values are approximately at a certain level. We denote RSSIs in the first type as *position RSSI group*. In the second type of subsequence, the RSSIs change from a certain level to another level. We denote RSSIs in the second type as *transition RSSI group*. The position RSSI group and transition RSSI group comprise the features of the RSSI sequence. In this step, we design an iterative change point detection method for feature extraction.

Iterative Change Point Detection: We use a change point detection algorithm based on CUSUM [13]. Denote the original RSSI sequence as $r_1, r_2, \ldots, r_n$, we compute a signal sequence's cumulative sums s_i as:

$$s_i = s_{i-1} + (r_i - \overline{r}) \quad (1)$$

where $s_0 = 0$ and $\overline{r} = \frac{\sum_{i=1}^{n} r_i}{n}$. Based on the slope change

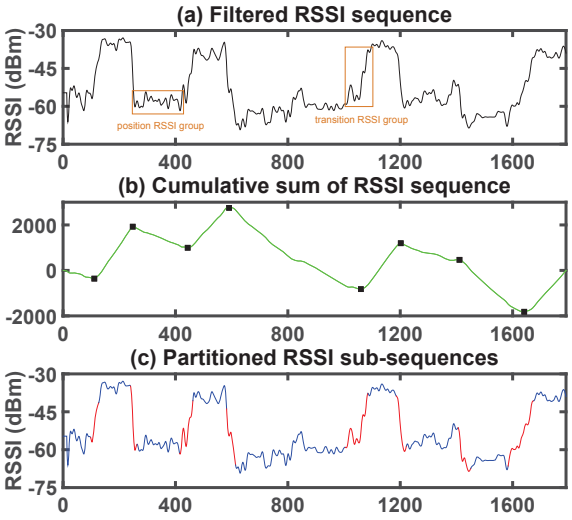

Figure 3: Partitioning a RSSI sequence

of the CUSUM curve s, we calculate the extreme points as original change points. The black squares in Figure 3 (b) show the original change points. The change points usually reflect the position where the original sequence suddenly increases or decreases. However, in the presence of transition RSSI group, the original sequence may vary gradually from one level to another level. Thus a change point corresponds to a transition RSSI group nearby. We need to further extract the transition and position RSSI groups based on those original change points.

For each change point, we need to identify the start and end of the corresponding transition RSSI group (as the red sub-sequences in Figure 3 (c)). We develop an iterative change point detection algorithm. We partition the RSSI sequence into segments based on original change points. For each segment, we iteratively search for its change points. For two consecutive segments i and $i+1$, we use the last change point in segment i and the first change point in segment $i+1$ as the start and end of a transition RSSI group. The sequence between the first and last change point in segment i is identified as a position RSSI group. The resulted transition RSSI groups (as the red sub-sequences) and position RSSI groups (as the blue sub-sequences) are shown in Figure 3 (c). It should be noted that it is difficult to define the exact start and end of a transition group.

Based on the transition and position RSSI groups, we transfer the original RSSI sequence to an STP, in which position RSSI groups correspond to vertices and transition RSSI groups correspond to edges.

5.2 Similarity Measurement

Till now, we have an STP for each user. To match a user's STP on STG, we first measure the similarity between two vertices or edges. It is difficult for similarity measurement due to the following reasons. First, users may stay in different positions for different amount of time or moving at different speeds, resulting in different number of RSSIs in vertices or edges. Second, two users may not be able to stay in exactly the same position or moving along exactly the same path. For example, two users may stay in two slightly different positions in the same room, leading to difference

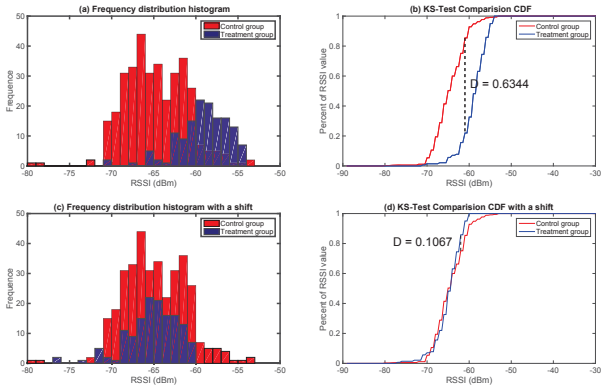

Figure 4: Similarity measurement of position RSSI groups

in RSSI for those two users. Third, due to the influence of the surrounding environment, the sequence of RSSI groups will randomly and slightly fluctuate. To address those difficulties, we introduce similarity measurement methods for position RSSI group and transition RSSI group.

5.2.1 *Position RSSI Group*

For a position RSSI group, we focus on the statistics while ignore the varying trend due to its random fluctuations. Jaccard similarity coefficient is a classical metric for comparing the similarity and diversity of two sets A and B as:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} \tag{2}$$

However, this metric is not feasible because it is used for comparing two sets without duplicates. We calculate the frequency distribution for each group and then calculate the similarity based on frequency distribution. Figure 4 (a) shows the frequency distribution histograms of two position RSSI groups (a control group and a treatment group).

We utilize the Kolmogorov-Smirnov test (KS-test) [4] to compare the similarity between two position RSSI groups. The KS-test is a nonparametric test and has the advantage of making no assumption about the distribution of data. As Figure 4 (b) shows, KS-test quantifies a maximum distance (the D statistic) between the empirical distribution curves of two groups. With the decrease of the D statistic, the similarity becomes higher. For two RSSI sequences $R = \{r_1, r_2, \ldots, r_n\}$ and $R' = \{r'_1, r'_2, \ldots, r'_m\}$, we denote the KS-test result as $KS(R, R') = 1 - D$.

Two users may not be able to stay at exactly the same position or moving along exactly the same path. Once there is a deviation existing between two groups, the KS-test will output a poor similarity such as the result in Figure 4 (b), $KS(R, R') = 1 - 0.6344 = 0.3656$. Therefore, we shift the treatment group with a certain threshold γ and then calculate the KS-test result for the shifted group. For different shifts, we calculate the maximum KS-test result for the similarity M. Denote Δ shifted group of R as $R(\Delta)$, i.e., $R(\Delta) = \{r_1 - \Delta, r_2 - \Delta, \ldots, r_n - \Delta\}$. We define the similarity of two position RSSI groups as:

$$M(R, R') = \max\{(1 - \lambda \times \frac{\Delta}{\gamma}) \times KS(R(\Delta), R')\} \tag{3}$$

where $-\gamma \leq \Delta \leq \gamma$ and λ is a scale parameter between 0 and 1. λ is the threshold value and we set $\lambda = 0.25$ in our experiments. When $|\Delta| > \gamma$, we consider that there is less similarity between two position RSSI groups and we will set $M = 0$. As Δ is a integer and γ is small, we enumerate different values for Δ to calculate M. In Figure 4 (c) and Figure 4 (d), Δ is 6 and $KS(R(\Delta), R') = 1 - 0.1067 = 0.8933$, where we get the final similarity $M(R, R') = (1 - \frac{0.25 \times 6}{10}) \times 0.8933 = 0.7593$.

5.2.2 Transition RSSI Group

Different from position RSSI group, we consider not only the statistics but also the trend of a transition RSSI group, which contains significant information such as walk speed, time and orientation. Here, we adopt Hidden Markov Model (HMM) to measure the similarity between transition RSSI groups of STP and STG.

In HMM, the system is assumed to be a Markov process with N unobserved states $H = \{H_1, H_2, \ldots, H_N\}$. There are M observed states $V = \{V_1, V_2, \ldots V_M\}$ corresponding to all the probable RSSI values in our work. We denote the state at time t as i_t and the transition probability distribution between these N unobserved states as a matrix A. Next, we denote the the probability of observing the symbol v_k given that we are in state j as B. Besides, we use π to denote the initial state. Thus, we can describe an HMM as $\psi = (A, B, \pi)$ according to a classical tutorial of Hidden Markov Model [12].

In our work, we set up an HMM ψ_k for every edge of STG with initial parameters A, B and π. Next, for every edge, we adjust its HMM parameters continually and get the optimal unobserved state sequence using the corresponding transition RSSI groups of legitimate users according to the algorithm in [12] . Then, the similarity between a transition RSSI group $R = \{r_1, r_2, \ldots, r_n\}$ and the k'st edge E_k of STG can be calculated as the probability of the observation sequence R in ψ_k:

$$C(R, E_k) = P(R|\psi_k) \tag{4}$$

Specially, if there isn't a corresponding edge on STG for R, we output a low score.

6. USER IDENTIFICATION

After extracting an STP of a user, the rest identification process includes four parts: path scoring, path matching, device diversity elimination and graph augmentation.

6.1 Path Scoring

Before introducing path matching algorithm, we first introduce how to calculate the similarity score of two paths. Basically, the score of legitimate users' STP should be higher than that of undesired users. For two paths $P = (v_1, e_1, v_2, e_2, \ldots, v_{n-1}, e_{n-1}, v_n)$ and $P' = (v'_1, e'_1, v'_2, e'_2, \ldots, v'_{n-1}, e'_{n-1}, v'_n)$, both of which contain n position RSSI groups and $n-1$ transition RSSI groups, we have several rules for similarity score calculation:

1. The final score should be calculated from both vertex similarity score (i.e. $M(v_i, v'_i)$) and edge similarity score (i.e. $C(e_i, e'_i)$) on two paths. We not only consider user staying in one position but also consider user moving from one position to another.

2. The influence of position RSSI group and transition RSSI group to final score should be tunable. Currently, we consider position RSSI group plays a more important role than transition RSSI group. The reason is that position RSSI group is relatively stable in the same position, while transition RSSI group is prone to vary due to different factors such as walking speed.

3. With more vertexes (n_l) having very low similarity score (e.g., less than a threshold δ_{min}), the path similarity score should be lower. We have $n_l = |\{i | M(v_i, v'_i) < \delta_{min}, 1 \leq i \leq n\}|$.

4. With more vertices (n_h) having very high similarity score (e.g., higher than a threshold δ_{max}), the path similarity score should be higher. We have $n_h = |\{i | M(v_i, v'_i) > \delta_{max}, 1 \leq i \leq n\}|$.

Based on those rules and similarity measurement methods for vertexes and edges, we process the vertexes and edges of an STP sequentially to calculate the path similarity score. For two path P and P', we compute their similarity score as:

$$S_n(P, P') = \beta \times ((1 - \lambda) \times \frac{\sum_{i=1}^{n} M(v_i, v'_i)}{n + n_l} + \lambda \times \frac{\sum_{i=1}^{n-1} C(e_i, e'_i)}{n - 1}) \tag{5}$$

where λ is a tuning parameter (according to rule 2) and β is a factor for adjusting the upper limit of scoring (according to rule 4).

Algorithm 1 PathMatching

Require: a STP with n vertexes and n-1 edges, the STG with V vertexes and E edges, current vertex matching index l.
Ensure: The best matching score of the STP on STG.

```
S ← 0;
currentScore ← 0;
for each i < V do
    M ← M(v_l, V_i);
    currentScore ← S_l(STP, P_l);
    if currentScore < σ then;
        return;
    else
        if l = n then
            Output(currentScore);
            if currentScore > S then
                S ← currentScore;
        else
            PathMatching(l+1,STP,STG);
end for
```

6.2 Path Matching

We transfer the user identification into a path matching problem on STG. In this step, we match each STP with the STG to check whether the corresponding user of the STP is legitimate. Initially, we use the STP from the first connected user as the STG. Later, we will introduce how to augment the STG with STPs from legitimate users in Section 6.4.

For a STP with n position RSSI groups, the goal of path matching is to find the maximum similarity score between the STP and all paths P on STG. Thus we have,

$$S(STP, STG) = \max\{S_n(STP, P)\} \tag{6}$$

for any path P with n vertexes on STG.

To reduce the overhead, we develop a pruned DFS-based path matching algorithm as in Algorithm 1. We denote the

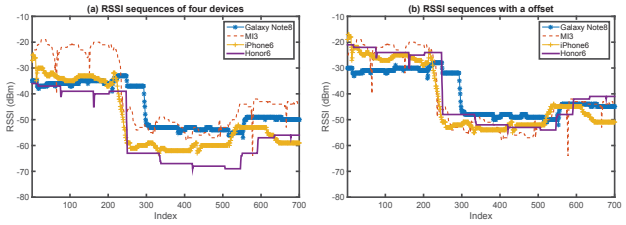

Figure 5: Device diversity of four devices

final path matching score as S and the current matching score between STP and a path $P_l = (v_1, e_1, v_2, e_2, \ldots, v_l)$ as $currentScore$. From line 3 to 15, for each vertex v_l of STG, we calculate the similarity score. We measure the similarity between two vertexes Vp_l and Vg_i in line 4 and calculate the $currentScore$ in line 5 with Equation 5. For line 6 and 7, if the $currentScore$ is smaller than the scoring threshold σ, we will terminate the current search and return. From line 9 to 12, if $l = n$, we will output the $currentScore$ and update the best matching score S. Otherwise, we will recursively invoke Algorithm 1 to search the next position RSSI group. Until searching all branches of STG, the path matching will terminate and get the best matching score S.

6.3 Device Diversity

RSSI is a measurement of the power present in a received radio signal and is susceptible to the device diversity. Figure 5 (a) shows the RSSI sequences of four mobile devices for the same moving path among three rooms.

Obviously, there is some deviation among the RSSI of different devices in the same position. Such a deviation will affect NiFi's identification accuracy for different devices, e.g., using Xiaomi MI3's RSSI data as the initial set to identify other devices.

We also notice that though there may exist a deviation between two types of devices (e.g., Xiaomi MI3 and Huawei Honor6 in Figure 5 (a)), the deviation between those two devices types of is relatively stable across different positions. We calculate the deviations between the RSSIs of Xiaomi MI3 and three other kinds of devices. Further, we calculate the mean, median, maximum, minimum and standard deviation of the deviations for different devices in different rooms as we can see in Table 2. Here, we focus on the mean deviations and the standard deviations. First, the mean deviations for the same device in three rooms are close, especially for Honor6 and iPhone6. Second, all of the standard deviations are quite small so the deviates in the same room are stable. These indicates that NiFi can eliminate the effect of device diversity with an appropriate offset compensation. As shown in Figure 5 (b), the RSSI series of four devices become similar through different offset compensations.

Based on these experimental observations, we address this problem with a shift-cancelation approach. First, NiFi performs the path matching for the original STP. Second, if the output result is an undesired user, NiFi performs a new path matching with an offset compensation. For a vertex of the original STP which is matched on STG, the original STP will be shifted according to the mean deviation of these two vertexes. Third, NiFi continues the rest of path matching process as described in Section 6.2. NiFi will repeat the second and third step for every vertex of the STP until a matching score is higher than the legitimate threshold. Otherwise, NiFi outputs a negative result.

	iPhone6			Honor6			Galaxy		
	r1	r2	r3	r1	r2	r3	r1	r2	r3
Mean	11.6	7.2	13.5	16.7	13.6	12.5	13.8	2.3	5.8
Median	12	6	14	18	13	12	14	2	6
Max	16	12	16	25	18	14	18	5	7
Min	1	3	6	2	9	3	1	0	2
Std	2.79	3.26	2.16	3.19	2.52	1.40	2.77	1.19	0.85

Table 2: Deviations between the RSSI sequences of Xiaomi MI3 and three other kinds of devices. R1, r2, and r3 represents three different rooms

6.4 Graph Augmentation

We construct the STG as a database by combining STPs of legitimate users. Initially, we use the STP of the first legitimate user (e.g., the first connected user) to construct the initial STG. Since the initial STP may contain the same position RSSI groups on the path, we combine similar vertexes according the vertex similarity measurement method. Then, we augment and update the STG online when a new legitimate STP is identified. If a vertex or edge of the STP is match with that in the STG, we update the corresponding vertex or edge in the STG. Otherwise, we extend the STG with the vertex or edge, which implies that a new possible legitimate position or moving path is found.

7. EVALUATION

In this section, we present the evaluation results of NiFi.

7.1 Evaluation Methodology

We implement NiFi on off-the-shelf hardware devices. Specially, we use a wireless router board (RB912UAG-2HPnD [3]) with OpenWrt, which works in 802.11n AP mode at 2.4GHz. Since our implementation does not rely on any specific hardware, it can be used for other wireless routers based on OpenWrt, which is widely used in a large collection of wireless routers such as TP-Link, Huawei, D-Link, and HiWiFi [2].

We evaluate the performance of NiFi from different aspects:

- To verify the effectiveness and usability of NiFi in different scenarios, we evaluate NiFi in three environments including an office, a laboratory and a dormitory environment. Figure 6 shows the experiment environments.
- We evaluate the performance of NiFi under different user activities. We test six kinds of user activities for each mobile device. (1) L1: Keeping still in a legitimate area. (2) L2: Walking between two legitimate areas. (3) L3: Walking among all legitimate areas. (4) I1: Keeping still in an undesired area. (5) I2: Walking between two undesired areas. (6) I3: Walking among undesired areas.
- We evaluate the performance of NiFi with different mobile devices, i.e., Samsung Galaxy Note8, Xiaomi MI3, iPhone6 and Huawei Honor6.

For different experiment settings, we evaluate the accuracy of NiFi. For each experiment, we perform 10 runs of tests

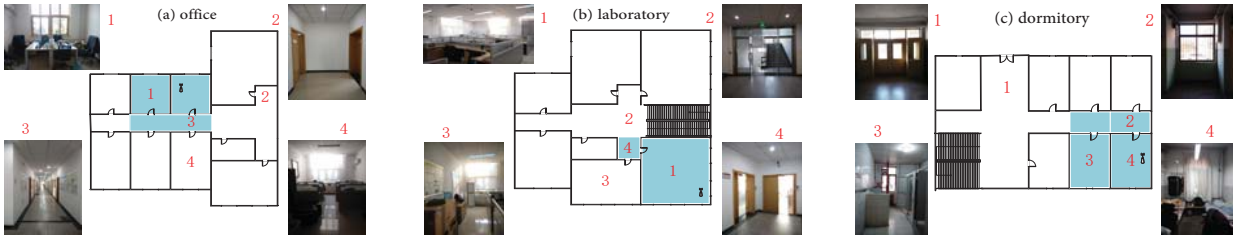

Figure 6: Three experimental environments (office, laboratory and dormitory)

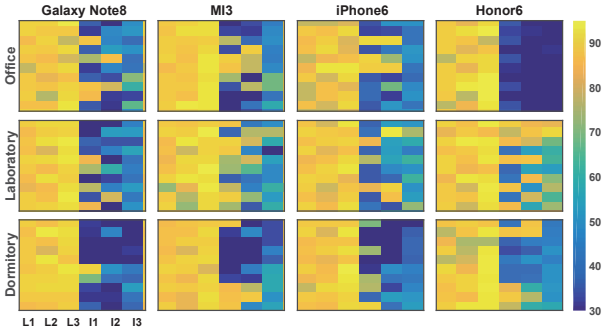

Figure 7: Heat map of matching scores under different environments with different user activities. The color indicates the matching score.

	Office	Laboratory	Dormitory	Average
R_{tp}	100%	98.33%	98.33%	98.89%
R_{fn}	0%	1.67%	1.67%	1.11%
R_{tn}	85.83%	75%	87.5%	82.78%
R_{fp}	14.17%	25%	12.5%	17.22%

Table 3: R_{tp}, R_{fn}, R_{tn}, and R_{fp} in three environments

	Office	Laboratory	Dormitory
Samsung	91.67%	93.33%	98.33%
Xiaomi	88.33%	83.33%	91.67%
iPhone	91.67%	93.3%	91.67%
Huawei	100%	76.66%	85%

Table 4: Identification accuracy of four devices in three environments

and record all the results. Further, we conduct cross validation and address device diversity. Then we evaluate the impact of initial signal samples and different users activities to the performance of NiFi.

In Figure 6, the blue areas are legitimate areas and the rest are undesired areas. The location of AP with NiFi is also indicated in the Figure.

7.2 Accuracy Evaluation

Across all the experiments, NiFi outputs a score according to the path matching algorithm described in Section 6.2. We record all the results and draw the heat map in Figure 7. The figure consists of twelve squares corresponding to four kinds of devices and three kinds of scenarios. The row in each square corresponds to a experiment run and the column corresponds to different user activities (L1-L3 and I1-I3).

Overall, we have the following observations.

- We can see that each square is divided into two parts: most blocks in the left three columns have a higher score than those in the right three columns. This shows that NiFi can correctly identify the legitimate users (L1-L3) from the undesired users (I1-I3).
- A device may have different performance under different scenarios. For example, in the laboratory environment, we can see that most devices have the lowest performance. This is because the laboratory environment is a single large room, in which our STG contains little path information of legitimate users for identification.
- Different types of devices also have different performance even under the same scenario. For example, in the office experiment, we can see Huawei Honor6 outperforms other devices.
- Different user activity may also influence the performance of NiFi. For example, the performance in the 4th column (I1: keeping still in an undesired area) is lower than that in the last column (I3: moving among all the illegal areas). This is because NiFi has more information for identification in I3.

Further, we explain the details and quantify the results under different scenarios. We quantify the accuracy in terms of true positive rates (R_{tp}), false positive rates (R_{fp}), true negative rates (R_{tn}) and false negative rates (R_{fn}). The result is shown in Table 3. As we known, the R_{fn} indicates a legitimate user is mis-identified as an undesired user and vice verse the R_{fp}. We have two observations. First, R_{fn} is negligible. In our experiment, we prefer to reduce the R_{fn} as much as possible because we do not want to affect legitimate users. In fact, we can adjust the matching score method to tradeoff the R_{fp} and R_{fn} for different application goals. So if the R_{fp} is acceptable, it means our identification program does work. In the worst case, R_{fp} is 25% and NiFi can still get rid of 75% of the undesired devices. Second, as mentioned above, NiFi performs slightly worse in the laboratory, so the error rate is related to the experimental environments.

We also quantify the identification accuracy for different devices in different environments as we can see in Table 4. The results demonstrate NiFi can achieve a high accuracy of all users, i.e. 90.83% in average.

7.3 Device Diversity

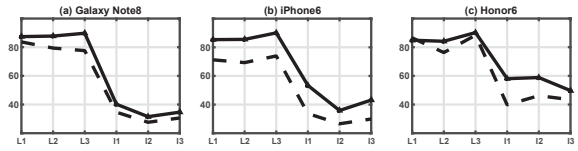

Figure 8: Average scores of self-validation and cross-validation

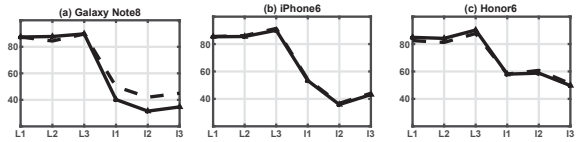

Figure 9: Average scores with device diversity elimination

To quantify the impact of device diversity, we use Xiaomi MI3's RSSI sequence to construct the initial STG and calculate cross-matching score of other devices. As a comparison, for each device, we use its own RSSI sequence as the initial data to calculate the self-matching score. In the experiment, we perform 10 runs for each test under each user activity and average the scores of all 10 runs. In Figure 8, the solid and dashed lines denote the average self-matching score and cross-matching score. In some cases (e.g., the right half of Figure 8 (a)), the two lines are close to each other. However, in most cases (e.g., Figure 8 (b)), the dashed lines are lower than the solid lines, indicating a possible error for user identification due to device diversity.

Then we repeat the above experiment with our device diversity elimination method. As Figure 9 show, in most cases, the dashed lines are close to the solid lines, which demonstrates that NiFi can effectively address device diversity.

7.4 Impact of Different User Activities

To further investigate why R_{fp} is a little high in some cases, we study the influence of different user activities. In our experiment, we perform six kinds of tests for different users with different spatial states (L1-L3 and I1-I3). We calculate the average score and error rate for each kind of test as shown in Figure 10.

There are three observations. First, the average scores of L1, L2, and L3 are much higher than those of I1, I2, and I3, which demonstrates the effectiveness of our identification approach. Second, the average score gradually increases for L1-L3, which coincides the fourth scoring rule. Third, the error rates of I1 and I2 are higher than that of I3. It is because that the information NiFi can use for identification in I1 and I2 is less than that of I3. Once a user frequently moves, the identification accuracy accordingly increases.

7.5 Impact of Initial Signal Samples

The initial data will also affects NiFi's usability and accuracy because it determines the integrality of the initial STG. An incomplete STG may make NiFi outputs a low score for a legitimate user, which will increase the R_{fn} of NiFi. We conduct an experiment to study the impact of changing the initial RSSI sequence. There are three kinds of initial RSSI sequences for three user activities in one legitimate area, two legitimate areas and multiple legitimate areas. Then, we calculate the average scores for L1-L3. As we can see in Figure 11, the average score becomes higher when the initial data becomes more complete. When the initial RSSI sequence only contains one legitimate area, the score is also low for legitimate users. However, even though the initial STG is incomplete, NiFi will make itself perform better and better with the graph augmentation in Section 6.4.

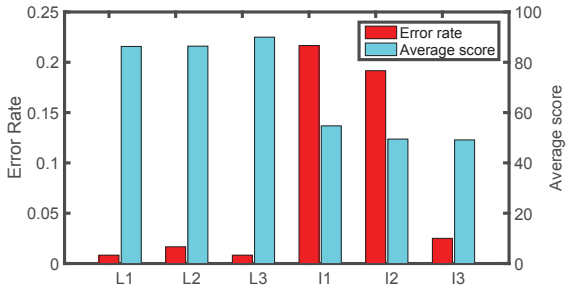

Figure 10: Error rate and average score for each user activity

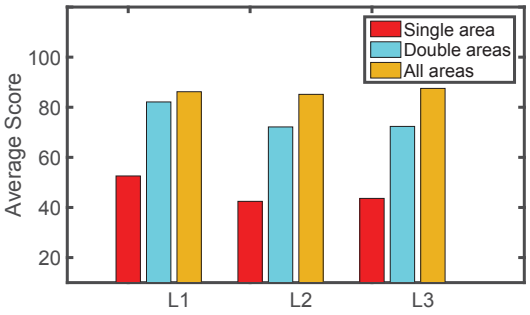

Figure 11: Average scores of L1-L3 under different initial data

8. CONCLUSION

In this paper, we propose NiFi, the first attempt for a non-intrusive, automatical user identification approach using WiFi signals on WiFi APs. NiFi can be deployed on most COTS WiFi routers and requires no special software and hardware support. It also does not require any modification to user devices. We implement NiFi on WiFi routers and evaluate its performance with different mobile devices in different environments. The results demonstrate NiFi can achieve an accuracy up to 90.83% in average. We believe that such a result enables NiFi be appropriate for various application scenarios such as home and hotel environment that are willing to provide convenient and exclusive WiFi access. In future, we will work on further improving the accuracy of NiFi with more physical layer information that can be obtained on COTS WiFi APs.

9. ACKNOWLEDGMENTS

We thank anonymous reviewers for their insightful comments. This work is supported in part by NSFC 61572277, 61529202, 61532012.

10. REFERENCES

[1] Chinese Network Security Report of Rising Antivirus in the first half of 2014. http://www.rising.com.cn/about/news/rising/2014-07-30/2014report-s.pdf.

[2] Devices that are supported by OpenWrt. http://wiki.openwrt.org/toh/start.

[3] RB912UAG-5HPnD routerboard info. http://routerboard.com/RB912UAG-5HPnD.

[4] Tools for Kolmogorov-Smirnov test. http://www.physics.csbsju.edu/stats/KS-test.html.

[5] WiFi Chip Shipment. https://www.abiresearch.com/press/wi-fi-chipset-shipments-will-near-18-billion-chips/.

[6] WiFi Master Key. http://en.wifi.com/#firstPage/.

[7] WiFi Security. http://www.bj.xinhuanet.com/bjyw/2014-08/20/c_1112148691_2.htm.

[8] H. Abdelnasser, M. Youssef, and K. A. Harras. Wigest: A ubiquitous wifi-based gesture recognition system. In *2015 IEEE Conference on Computer Communications, INFOCOM 2015, Kowloon, Hong Kong, April 26 - May 1, 2015*, pages 1472–1480, 2015.

[9] F. Adib, Z. Kabelac, and D. Katabi. Multi-person localization via rf body reflections. In *Proceedings of the 12th USENIX Conference on Networked Systems Design and Implementation*, pages 279–292. USENIX Association, 2015.

[10] K. Ali, A. X. Liu, W. Wang, and M. Shahzad. Keystroke recognition using wifi signals. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, pages 90–102. ACM, 2015.

[11] P. Bahl and V. N. Padmanabhan. Radar: An in-building rf-based user location and tracking system. In *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 2, pages 775–784. Ieee, 2000.

[12] L. R. R. F. Ieee. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[13] A. A. Mahimkar, H. H. Song, Z. Ge, A. Shaikh, J. Wang, J. Yates, Y. Zhang, and J. Emmons. Detecting the performance impact of upgrades in large operational networks. *ACM SIGCOMM Computer Communication Review*, 41(4):303–314, 2011.

[14] R. Nandakumar, K. K. Chintalapudi, and V. N. Padmanabhan. Centaur: locating devices in an office environment. In *Proceedings of the 18th annual international conference on Mobile computing and networking*, pages 281–292. ACM, 2012.

[15] R. Nandakumar, B. Kellogg, and S. Gollakota. Wi-fi gesture recognition on existing devices. *CoRR*, abs/1411.5394, 2014.

[16] J.-g. Park, B. Charrow, D. Curtis, J. Battat, E. Minkov, J. Hicks, S. Teller, and J. Ledlie. Growing an organic indoor location system. In *Proceedings of the 8th international conference on Mobile systems, applications, and services*, pages 271–284. ACM, 2010.

[17] Q. Pu, S. Gupta, S. Gollakota, and S. Patel. Whole-home gesture recognition using wireless signals. In *The 19th Annual International Conference on Mobile Computing and Networking, MobiCom'13, Miami, FL, USA, September 30 - October 04, 2013*, pages 27–38, 2013.

[18] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan, and R. Sen. Zee: zero-effort crowdsourcing for indoor localization. In *Proceedings of the 18th annual international conference on Mobile computing and networking*, pages 293–304. ACM, 2012.

[19] S. Sen, J. Lee, K.-H. Kim, and P. Congdon. Avoiding multipath to revive inbuilding wifi localization. In *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*, pages 249–262. ACM, 2013.

[20] S. Sigg, M. Scholz, S. Shi, Y. Ji, and M. Beigl. Rf-sensing of activities from non-cooperative subjects in device-free recognition systems using ambient and local signals. *IEEE Trans. Mob. Comput.*, 13(4):907–920, 2014.

[21] S. Sigg, S. Shi, F. Büsching, Y. Ji, and L. C. Wolf. Leveraging rf-channel fluctuation for activity recognition: Active and passive systems, continuous and rssi-based signal features. In *The 11th International Conference on Advances in Mobile Computing & Multimedia, MoMM '13, Vienna, Austria, December 2-4, 2013*, page 43, 2013.

[22] D. Vasisht, S. Kumar, and D. Katabi. Decimeter-level localization with a single wifi access point. In *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16)*, pages 165–178, 2016.

[23] G. Wang, Y. Zou, Z. Zhou, K. Wu, and L. M. Ni. We can hear you with wi-fi! In *The 20th Annual International Conference on Mobile Computing and Networking, MobiCom'14, Maui, HI, USA, September 7-11, 2014*, pages 593–604, 2014.

[24] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, and R. R. Choudhury. No need to war-drive: unsupervised indoor localization. In *Proceedings of the 10th international conference on Mobile systems, applications, and services*, pages 197–210. ACM, 2012.

[25] J. Wang, D. Vasisht, and D. Katabi. Rf-idraw: virtual touch screen in the air using RF signals. In *ACM SIGCOMM 2014 Conference, SIGCOMM'14, Chicago, IL, USA, August 17-22, 2014*, pages 235–246, 2014.

[26] L. S. Z. Z. Y. L. Xiaolong Zheng, Jiliang Wang. Smokey: Ubiquitous smoking detection with commercial wifi infrastructures. *INFOCOM*, 2016.

[27] J. Xiong and K. Jamieson. Arraytrack: A fine-grained indoor location system. In *NSDI*, pages 71–84, 2013.

[28] J. Xiong and K. Sundaresan. Tonetrack: Leveraging frequency-agile radios for time-based indoor wireless localization.

[29] Z. Yang, C. Wu, and Y. Liu. Locating in fingerprint space: wireless indoor localization with little human intervention. In *Proceedings of the 18th annual international conference on Mobile computing and networking*, pages 269–280. ACM, 2012.

[30] M. Youssef and A. Agrawala. The horus wlan location determination system. In *Proceedings of the 3rd international conference on Mobile systems, applications, and services*, pages 205–218. ACM, 2005.