Walls Have No Ears: A Non-Intrusive WiFi-Based User Identification System for Mobile Devices

Linsong Cheng, *Student Member, IEEE*, and Jiliang Wang, *Member, IEEE*

***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 access points (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 enable AP to automatically identify legitimate users in indoor environments, 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, and so on, 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 measures similarity for signal sequences while not relying on precise user location information. We implement NiFi on Commercial Off-The-Shelf 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%.**

***Index Terms*— User identification, mobile device, AP, WiFi signals.**

I. Introduction

A. Motivation

NOWADAYS, WiFi has become a fundamental part for providing wireless connection. According to ABI research's report [6], 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, in order to use the network service, a device needs to first connect to a WiFi access point (AP). 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 [8] 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 protection is to set a password. However, according to a security report of Rising Antivirus [1], a large portion of passwords for current APs are too simple. 86% of users never login to the setting page after installing an 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, according to Hong Kong WiFi Adoption and Security Survey 2014 [3], there exist a high portion of users (e.g., 9.1% in home WiFi network), who do not set passwords to protect their WiFi network. Even for complicated passwords, there are cracking softwares and automatic password sharing softwares [7] 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. However, existing methods of those smart APs (e.g., whitelist mechanism) require users to set a white list based on MAC addresses, which are complicated for AP users.

B. Proposed Approach

We propose NiFi, an automatic approach which requires no user configurations, to identify mobile devices of legitimate WiFi users. 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 single location may be similar, signal sequences from users at different areas can be very different.

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

Overall, NiFi can automatically identify legitimate users without user active participation. NiFi can be deployed on commercial WiFi AP while not requiring any modification to

Manuscript received December 18, 2017; revised September 28, 2018; accepted November 25, 2018; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor T. He. Date of publication January 22, 2019; date of current version February 14, 2019. This work was supported in part by the National Key R&D Program of China under Grant 2018YFB1004800, in part by the National Natural Science Fund China for Excellent Young Scholars under Grant 61722210, and in part by the NSFC Key Program under Grant 61532012, Grant 61572277, and Grant 61529202. *(Corresponding author: Jiliang Wang.)*

The authors are with the School of Software and TNLIST, Tsinghua University, Beijing 100084, China (e-mail: chengls14@mails.tsinghua.edu.cn; jiliangwang@tsinghua.edu.cn).

Digital Object Identifier 10.1109/TNET.2018.2886411

user devices such as mobile phones and laptops. Undesired users are difficult to use an AP with NiFi since physical signal sequence is difficult to mimic and crack. Moreover, the traditional user identification approaches (e.g., password) are orthogonal to NiFi and can be conveniently combined with NiFi for a better WiFi protection.

We implement NiFi on a COTS wireless router board (RB912UAG-2HPnD [4]) 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., hotel and 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.

C. 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 signal 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 is only 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 an 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 vertexes or edges. Lastly, we propose a DFS-based path matching algorithm to find the maximum similarity score for the STP in 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 sequences from different phones at the same position may even be different. This is also examined in our experiment in Section VI-C. We further compare different devices and find that the shift between the signals of two different phones in different environments is relatively stable. Based on this finding, we propose a shift-cancelation approach to mitigate the impact of device diversity.

D. Key Contributions

The main contributions of this paper are 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 user identification to a path matching problem in signal space. We further propose effective similarity measurement methods for signal sequences and path matching algorithm for user identification.
- We implement NiFi on COTS routers and evaluate its performance for different mobile devices in different environments.

The remainder of this paper is structured as follows. Section II introduces related works in recent years. Section III presents an overview of NiFi's architecture. Section IV elaborates on data collection and noise removal. Then, Section V and Section VI describe NiFi's identification process in detail. Section VII presents our implementation of NiFi on COTS AP and a comprehensive experimental evaluation. Section VIII discusses NiFi's performance. Finally, Section IX concludes the paper.

II. Related Work

A. 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 [12], Horus [32] and OIL [17] leverage existing WiFi infrastructure to build a fingerprint database for indoor localization. Further, several systems such as LiFS [31], Zee [19] and UnLoc [25] use crowdsourcing to collect signal fingerprints. Meanwhile, those approaches assume multiple APs in the environment and a user device can obtain signals from multiple APs. Some recent works [28], [33] use complex physical layer information, e.g., channel state information (CSI), to obtain more fundamental location related information. There are also approaches using complicated analysis methods or special hardware [15], [20], [30], e.g., antenna array [10], [29]. With fine-grained signal information, Chronos [23] 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 be obtained especially with COTS WiFi APs. Existing CSI measurements obtain CSI with special WiFi network interface cards (e.g., Intel 5300 and Atheros AR9380). Nevertheless, we also find 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. Comparing with existing WiFi localization methods, NiFi aims

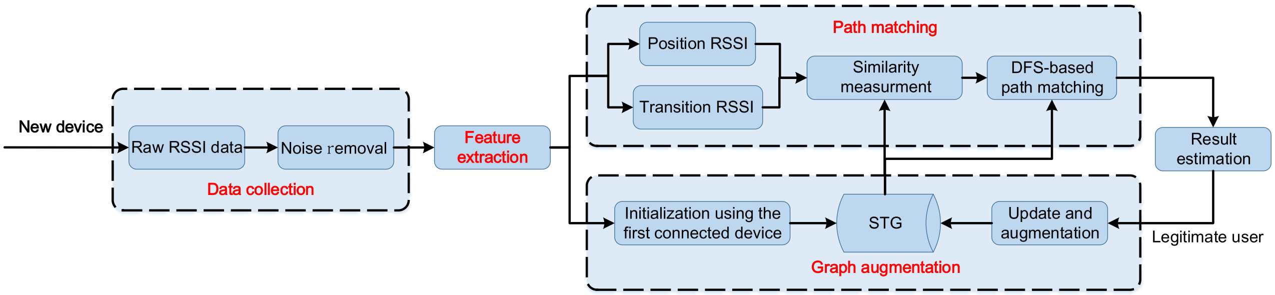

Fig. 1. User identification architecture.

to provide convenient user identification with few requirements. NiFi only needs a single AP to automatically identify legitimate users in different environments with no special hardware requirements.

B. Fine-Grained Signal Based Activity Recognition

There are also a large collection of activity recognition approaches based on wireless signals to recognize human activities [9], [11], [16], [18], [21], [22], [24], [26], [27]. 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, standing up, and walking) [21], [22], and gestures [9], [16], [18]. Recently, several works are proposed to recognize micro-movements. WiKey [11] utilizes the patterns in the time-series of CSI values to recognize keystrokes using a laptop. WiHear [24] detects and analyzes fine-grained signal reflections from mouth movements while speaking. RF-IDraw [26] constructs a virtual touch screen in the air using RF signals. Those approaches are used for recognizing user activities with special hardware. Meanwhile, they often require a training set. Thus they are not appropriate for user identification especially on COTS APs.

III. NiFi Overview

In this section, we introduce NiFi's design goals and an overview of the system workflow and user identification 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 user devices which is difficult to mimic.

A. 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 Section VII-F.

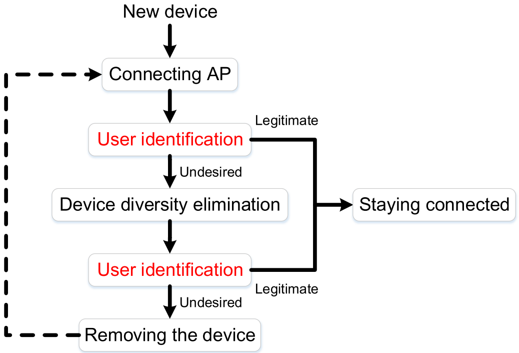

Fig. 2. System workflow.

B. System Workflow

The system workflow of NiFi is shown in Figure 2. When a new user device connects to the AP, NiFi makes user identification using its signal sequence. If the user device is identified as legitimate, NiFi allows it to stay connected. Otherwise, NiFi considers the influence of device diversity and makes user identification again. The detailed device diversity elimination method is introduced in Section VI-C. After eliminating device diversity, if the device is still considered as undesired, NiFi disconnects it. Otherwise, NiFi allows the user device to stay connected.

The overall system workflow of NiFi is flexible and our main work is focused on how to make user identification. For example, when identifying an undesired user, NiFi allows his device to reconnect the AP as the dashed line shows in Figure 2. It is because that NiFi's identification result depends on the areas where the user uses the AP. If an undesired user moves to legitimate areas, he should be considered as a legitimate user now and have a chance to reconnect the AP. On the contrary, we can make a black list and prevent his device to reconnect the AP for some time, which will change the overall workflow.

C. User Identification Architecture

The user identification architecture is shown in Figure 1. The identification 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.

Feature Extraction: The second component is the feature extraction component. After data collection, NiFi has an 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 an 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 VI-D.

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. The detailed path matching algorithm is introduced in Section VI-B.

IV. Data Collection and Noise Removal

In this section, we introduce our data collection and noise removal process. Table I summarizes the symbols used in this paper.

A. 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 [4]) with OpenWrt system. OpenWrt is an operating system based on the Linux kernel, primarily and widely used on many routers and APs.

B. Noise Removal

For the raw RSSI sequence, there are three kinds of noise as we can see in Figure 3(a). 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 which arrives at the AP becomes very small.

TABLE I

Symbols in This Paper

Symbol	Description
f_i	The i^{th} value of collected RSSI sequence F
r_i	The i^{th} value of filtered RSSI sequence R
s_i	The i^{th} value of 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

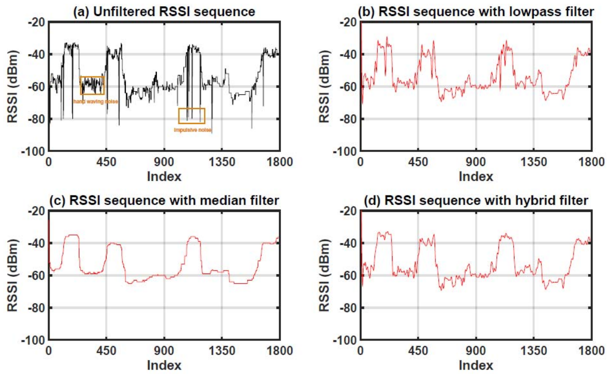

Fig. 3. Unfiltered and filtered RSSI sequence. (a) Unfiltered RSSI sequence. (b) RSSI sequence with lowpass filter. (c) RSSI sequence with median filter. (d) RSSI sequence with hybrid filter.

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 [11], 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 cannot remove the impulsive noise well as Figure 3(b) shows.

For the third type of impulsive noise, a median filter is particularly effective as shown in Figure 3(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,

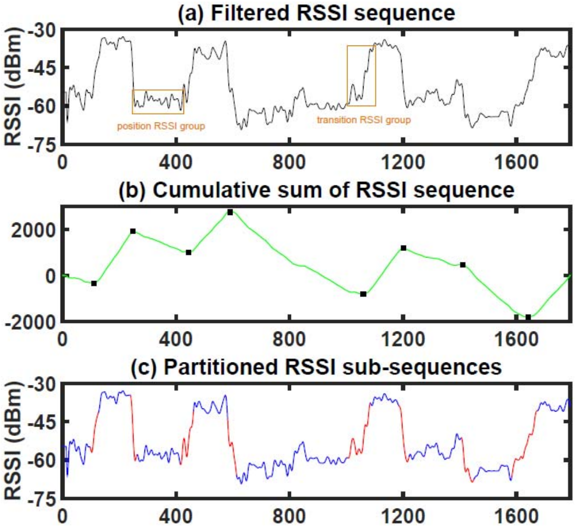

Fig. 4. Partitioning an RSSI sequence. (a) Filtered RSSI sequence. (b) Cumulative sum of RSSI sequenc. (c) Partitioned RSSI sub-sequences.

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 3(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 V-B.

V. Feature Extraction & Similarity Measurement

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

A. Feature Extraction

To extract features, we first partition an RSSI sequence into groups according to user's position and moving status. As shown in Figure 4(a), we find that the original RSSI sequence consists of a series of sub-sequences of two types. In the first type of sub-sequence, 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 sub-sequence, 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 [14]. Denote the original RSSI sequence as $r_1, r_2, \ldots, r_n$, we compute i^{th} value of a signal sequence's cumulative sums s as:

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

where $s_0 = 0$ and $\overline{r} = \frac{\sum_{i=1}^{n} r_i}{n}$. Based on the slope change of the CUSUM curve s, we calculate the extreme points as original change points. The black squares in Figure 4(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 further extract the transition and position RSSI groups based on those original change points.

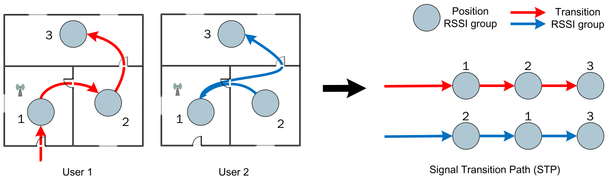

Fig. 5. Feature extraction.

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 4(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 4(c). It should be noted that it is difficult to define the exact start and end of a transition group.

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

Figure 5 shows an example. There are two users walking in three rooms (room 1, room 2 and room 3). User 1 walks from room 1 to room 3 through room 2. User 2 walks from room 2 to room 1 and then to room 3. Meanwhile, each user may stay in each room for some amount of time. We collect an RSSI sequence for each user. Then we partition the original sequence into position and transition RSSI groups corresponding to the vertexes and edges of its STP.

B. Similarity Measurement

Till now, we have an STP for each user. To match the STP on STG, we first measure the similarity between two vertexes 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 move at different speeds, resulting in different number of RSSIs in vertexes or edges. Second, two users may not be able to stay in exactly the same position or move along exactly the same path. For example, two users may stay in two slightly different positions in the same room, leading to difference in RSSI sequences 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.

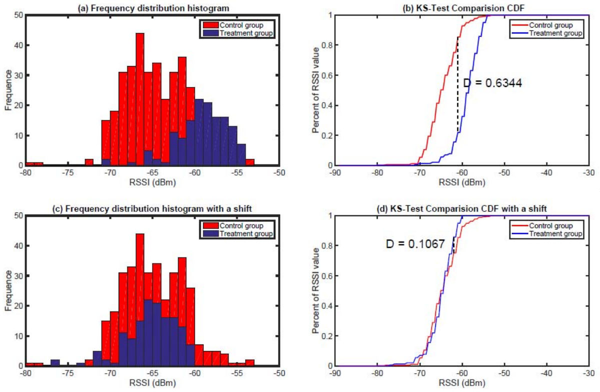

Fig. 6. Similarity measurement of position RSSI groups. (a) Frequency distribution histogram. (b) KS-Test comparison CDF. (c) Frequency distribution histogram with a shift. (d) KS-Test comparison CDF with a shift.

1) Position RSSI Group: For a position RSSI group, we focus on the statistics while ignoring 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 $J(A,B) = \frac{|A \cap B|}{|A \cup B|}$. 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. Figure 6(a) shows the frequency distribution histograms of two position RSSI groups (a control group and a treatment group). In the figure, we use the RSSI sequences of Xiaomi MI3 whose signals are the most unstable compared with three other experimental devices (Samsung Galaxy Note8, iPhone6 and Huawei Honor6) in order to ensure that our similarity measurement method can work for most devices.

We utilize the Kolmogorov-Smirnov test (KS-test) [5] to measure the similarity between two position RSSI groups. KS-test is a nonparametric test and has the advantage of making no assumption about the distribution of data. As Figure 6(b) shows, KS-test quantifies a maximum distance (D statistic) between the empirical distribution curves of two groups. The smaller D statistic is, the higher the similarity is. 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. Once there exists a deviation between two groups, the KS-test will output a poor similarity such as the result in Figure 6 (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:

$$M(R, R') = \max\{(1 - \lambda \times \frac{\Delta}{\gamma}) \times KS(R(\Delta), R')\} \quad (2)$$

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 an integer and γ is small, we enumerate different values for Δ to calculate M. In Figure 6(c) and Figure 6(d), Δ is 6 and $KS(R(\Delta), R') = 1 - 0.1067 = 0.8933$, where we get the final similarity $M(R, R') = (1 - \frac{0.25 \times 6}{10}) \times 0.8933 = 0.7593$.

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 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 [13].

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 [13]. 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:

$$C(R, E_k) = P(R|\psi_k) \quad (3)$$

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

VI. User Identification

After extracting an STP of a user, NiFi conducts continuous similarity measurements without time interval according to the results of iterative change point detection. The rest identification process includes four parts: path scoring, path matching, device diversity elimination and graph augmentation.

A. 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 that users stay in one position but also consider that users move 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) owning a 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 vertexes (n_h) owning a 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 measure the vertexes and edges of an STP sequentially to calculate the path similarity score. For two pathes P and P', we compute their similarity score as:

$$S_n(P, P') = \beta \times ((1-\lambda) \times \frac{\sum_{i=1}^{n} M(v_i, v_i')}{n + n_l} + \lambda \times \frac{\sum_{i=1}^{n-1} C(e_i, e_i')}{n-1}) \tag{4}$$

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).

B. 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 VI-D.

For an 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,

$$S(STP, STG) = \max\{S_n(STP, P)\} \tag{5}$$

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 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 in line 4 and calculate the $currentScore$ in line 5 with Equation 4. 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.

Since the STG may not contain all legitimate paths, we add virtual edges between vertexes that have no edges in STG. The similarity score between a virtual edge and any other edge is 0. Therefore, if a user walks along a path that has never be explored by any legitimate user before, our path matching algorithm can still continue the rest search on STG.

Algorithm 1 PathMatching

```
Require: an 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.
 1: $S \leftarrow 0$;
 2: $currentScore \leftarrow 0$;
 3: for each $i < V$ do
 4:   $M \leftarrow M(v_l, V_i)$;
 5:   $currentScore \leftarrow S_l(STP, P_l)$;
 6:   if $currentScore < \sigma$ then;
 7:     return;
 8:   else
 9:     if $l = n$ then
10:       Output($currentScore$);
11:     if $currentScore > S$ then
12:       $S \leftarrow currentScore$;
13:   else
14:     PathMatching(l+1,STP,STG);
15: end for
```

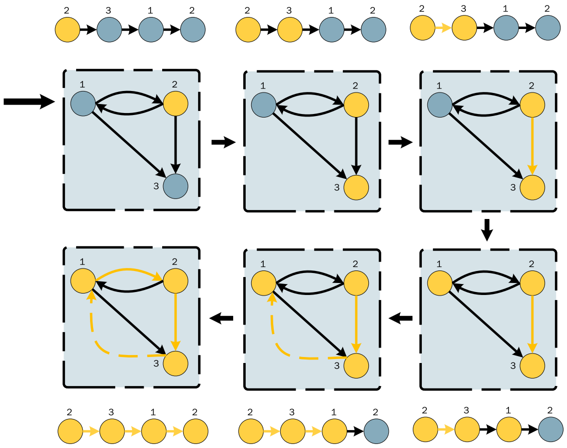

Fig. 7. An example of path matching.

Specially, when a user is not moving, its STP contains only one position RSSI group and no transition RSSI group. NiFi uses the statistic information for user identification. In such a case, the statistic of RSSI values of the legitimate user is very similar with that of the corresponding position RSSI group of STG and that of the illegitimate user is quite different from that of all position RSSI groups of STG.

Figure 7 shows an example of how to match the STP of a user with an actual path 2-3-1-2 on STG. First, we search the most similar vertex on STG of STP's first vertex and the result is vertex 2. Second, we find vertex 3 on STG as a matching for STP's second vertex. Third, taking vertex 2 as the starting point and vertex 3 as the ending point, we measure similarity between STG's edge $\langle 2, 3 \rangle$ and STP's first edge. Next, we repeat these three steps until matching the last vertex and edge of STP. Specially, in this example, we have a virtual

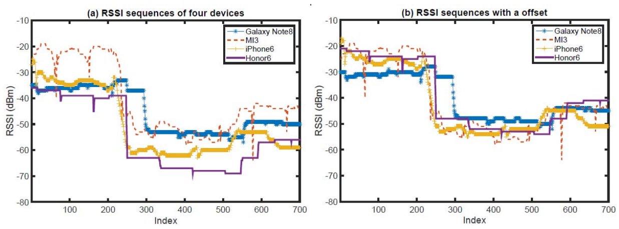

Fig. 8. Device diversity of four devices. (a) RSSI sequences of four devices. (b) RSSI sequences with a offset.

TABLE II

DEVIATIONS BETWEEN THE RSSI SEQUENCES OF XIAOMI MI3 AND OTHER THREE KINDS OF DEVICES. R1, R2, AND R3 REPRESENT THREE DIFFERENT ROOMS

	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

edge $\langle 3, 1\rangle$. This situation will decrease our favor to the current path but won't terminate the search.

C. Device Diversity

RSSI is a measurement of the power present in a received radio signal and is susceptible to the device diversity. Figure 8(a) shows the RSSI sequences of four kinds of mobile devices (Galaxy Note8, MI3, iPhone6 and Honor6) for the same moving path among three rooms.

Obviously, there is some deviation among the RSSI sequences of different devices in the same position. Such a deviation will affect NiFi's identification accuracy for different devices, e.g., using 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., MI3 and Honor6 in Figure 8(a)), the deviation between those two types of devices is relatively stable across different positions. We calculate the deviations between the RSSI sequences of MI3 and other three 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 indicate that NiFi can eliminate the effect of device diversity with an appropriate offset compensation. As shown in Figure 8(b), the RSSI sequences 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 VI-B. NiFi will repeat the second and third steps for every vertex of the STP until a matching score is higher than the legitimate threshold. Otherwise, NiFi outputs a negative result.

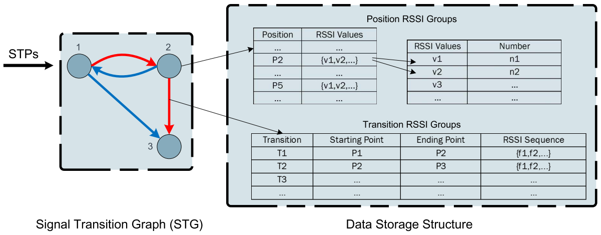

Fig. 9. Construction and storage of STG.

D. 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 to the vertex similarity measurement method. The initial STG may be incomplete. But as NiFi identifies more legitimate users, the information of those legitimate users are used to improve the STG. As Figure 9 shows, we construct an STG using the STPs of two users in Figure 5. For STG's vertexes (position RSSI groups), we count and store their statistical distributions. For STG's edges (transition RSSI groups), we store their starting position, ending position and RSSI sequence.

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. We will add them as a new vertex or edge of the STG.

Figure 11 shows an example of graph augmentation. We match an STP of a user with the path 1-2-3-4 on STG and the score is high due to the high similarity of 1-2-3 part. We update the STG gradually according to the STP. First, if the similarity between the STP's first vertex and the STG's vertex 1 is high, we add the statistical distribution of the STP's first vertex into vertex 1's. Second, we do the same work for the STG's second vertex. Third, if the similarity between the STP's first edge and the STG's edge $< 1, 2 >$ is high, we adjust the HMM parameters of edge $< 1, 2 >$ using the RSSI sequence of the STP's first edge. Next, we repeat these three steps until matching the last vertex and edge of the STP. Because there is not a matching vertex on the STG for the STP's last vertex, we add a temporary vertex (vertex 4) on the STG and store its statistical distribution. At the same time, we add a temporary edge (edge $< 3, 4 >$) and corresponding HMM.

VII. EVALUATION

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

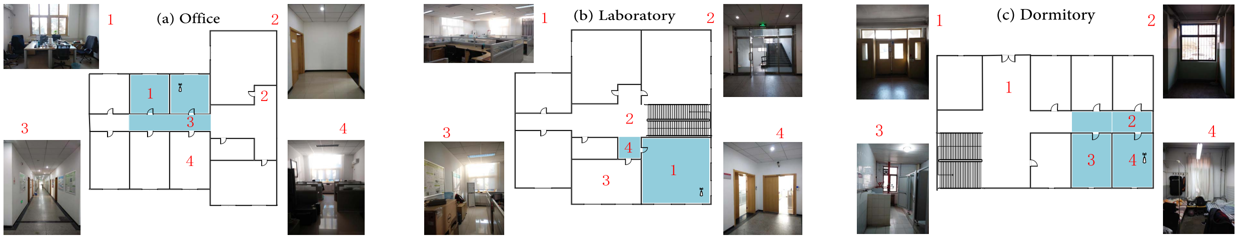

Fig. 10. Three experimental environments (office, laboratory and dormitory). (a) Office. (b) Laboratory. (c) Dormitory.

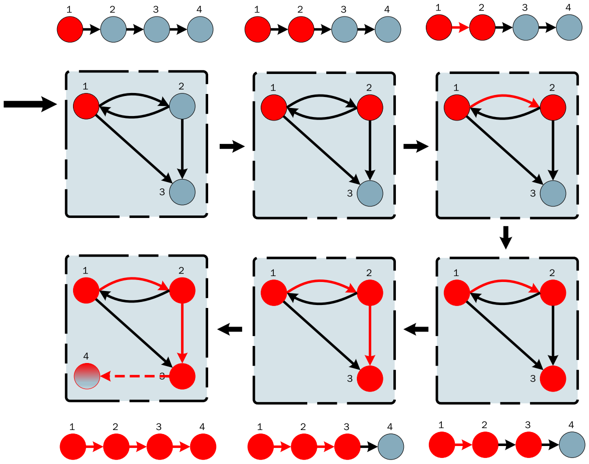

Fig. 11. An example of graph augmentation.

A. Evaluation Methodology

We implement NiFi on off-the-shelf hardware devices. Specially, we use a wireless router board (RB912UAG-2HPnD [4]) 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 10 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 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 for the performance of NiFi. Besides, we record the computing time of above runs and the average computing time is less than 1s. There are four users walking simultaneously in above experimental environments.

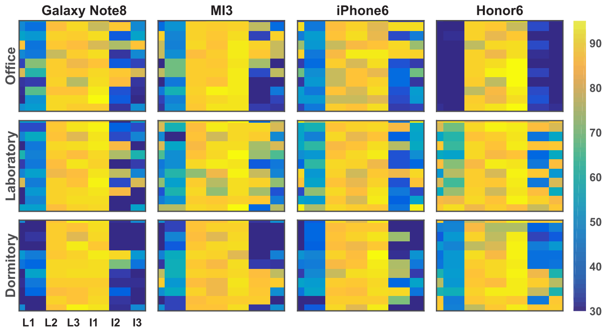

Fig. 12. Heat map of matching scores under different environments with different user activities. The color indicates the matching score.

In Figure 10, the blue areas are legitimate areas and the rest is undesired areas. The location of AP with NiFi is also indicated in the figure.

B. Accuracy Evaluation

Across all the experiments, NiFi outputs a score according to the path matching algorithm described in Section VI-B. We record all the results and draw the heat map in Figure 12. The figure consists of twelve squares corresponding to four kinds of devices in three kinds of scenarios. Every block in each square corresponds to an experiment run and the column corresponds to different user activities (L1-L3 and I1-I3).

Overall, we have the following observations.

- 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 efficiently identify the legitimate users (L1-L3) from the undesired users (I1-I3).
- A device may have different performances 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 performances even under the same scenario. For example, in the office experiment, we can see Honor6 outperforms other devices. Through observing the signal information

TABLE III

R_{tp}, R_{fn}, R_{tn}, AND R_{fp} IN THREE ENVIRONMENTS

	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 IV

IDENTIFICATION ACCURACY OF FOUR DEVICES 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%

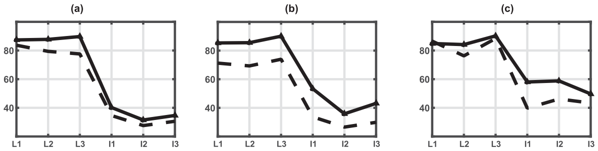

Fig. 13. Average scores of self-validation and cross-validation. (a) Galaxy Note8. (b) iPhone6. (c) Hornor6.

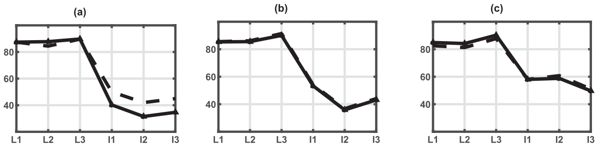

Fig. 14. Average scores with device diversity elimination. (a) Galaxy Note8. (b) iPhone6. (c) Hornor6.

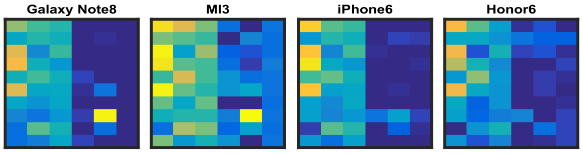

Fig. 15. Average values of RSSI sequences in a laboratory.

of different devices, we find the RSSI sequences of Honor6 are more different in different rooms, which is in favor of NiFi's identification.

- Different user activities 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 undesired 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 III. As we know, the R_{fn} indicates a legitimate user is mis-identified as an undesired user and vice versa 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. If 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, the accuracy of NiFi is related to the experimental environments. We analyze that application scenarios can be clarified into two kinds: multi-room scenario and open-ended scenario. For multi-room scenario (e.g., Figure 10(a) and (c)), the signal characteristics in different areas is quite different, so our system achieves high accuracy. For open-ended scenario (e.g., Figure 10(b)), our STG contains little path information of legitimate users, so NiFi performs a little worse.

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

C. Device Diversity

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 user activity and average the scores of all 10 runs. In Figure 13, solid lines denote the average self-matching score and dashed lines denote the average cross-matching score. In some cases (e.g., the right half of Figure 13(a)), the two lines are close to each other. However, in most cases (e.g., Figure 13(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 experiments with our device diversity elimination method. As Figure 14 shows, in most cases, the dashed lines are close to the solid lines, which demonstrates that NiFi can effectively address device diversity.

D. Effectiveness of Scoring Method

We use two experiments to demonstrate the effectiveness of our scoring method.

First, we use a simple method, identifying users with the average values of their RSSI sequences to replace our scoring method. Figure 15 shows the heat map of the RSSI average for each test in the laboratory environment. We can find the distinction between the left half and the right half of every square is slight. It's difficult to separate these two parts well with an average threshold value. And this scoring method cannot guarantee a low R_{fn} . In contrast, our scoring method is usable and indispensable for user identification, which can satisfy NiFi's requirement.

Second, we follow fingerprint-based localization to repeat the above user identification experiments with four devices in three environments. We construct an RSSI fingerprint database of legitimate areas according to the average values of RSSI sequences. When a new user connects the AP, we collect

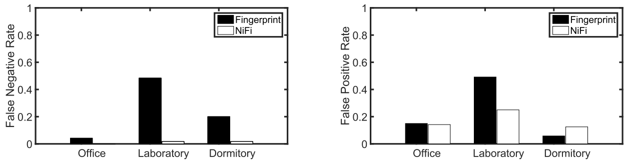

Fig. 16. R_{fn} and R_{fp} with a deviation threshold 5dBm.

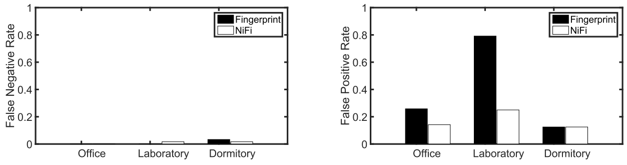

Fig. 17. R_{fn} and R_{fp} with a deviation threshold 10dBm.

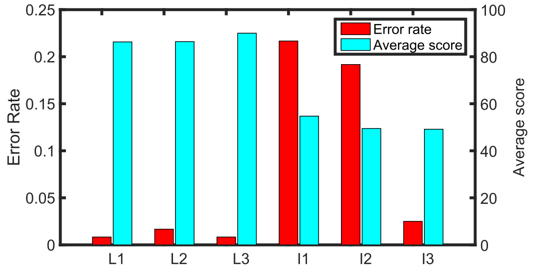

Fig. 18. Error rate and average score for each user activity.

RSSI information and match the closest fingerprint. If the deviation between the device's RSSI average value and this fingerprint is smaller than a threshold, we consider this user as a legitimate user.

As Figure 16 shows, when we use a small deviation threshold 5dBm, R_{fn} is 24.17% and R_{fp} is 23.33% in average. R_{fn} is too high, which will affect legitimate users. However, when we use a big deviation threshold 10dBm to decrease R_{fn} as Figure 17 shows, R_{fp} will increase, up to 39.17%. In laboratory environment, R_{fp} is 79.17%, which means this identification method is useless. By comparison, NiFi owns an effective scoring method, which can prove a tiny R_{fn} and a low R_{fp} at the same time.

E. 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 18.

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 that NiFi can use for identification in I1 and I2 is less than that of I3. Specially, NiFi may make a mistake with low probability, when the statistic information of RSSI values of a static undesired user is very similar with that of the RSSI values in a legitimate area. When the user moves, more information can be collected by NiFi and the accuracy can be further improved.

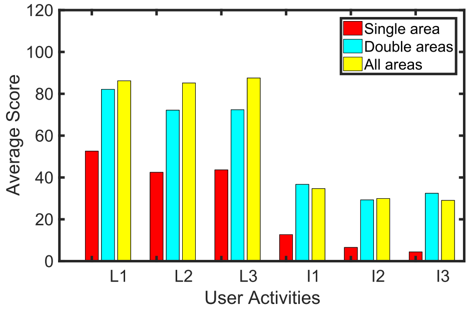

Fig. 19. Average scores of different user activities under different initial data.

F. Impact of Initial Signal Samples

The initial data will also affect 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 different user activities. As we can see in Figure 19, the average scores of legitimate users become higher when the initial data becomes more complete. The scores of undesired users are low all the time. 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 VI-D.

G. Evaluation of Iterative Change Point Detection

We choose ten RSSI sequences with different lengths to evaluate our iterative change point detection method. First, we partition these RSSI sequence according to the actual moving paths. This can be treated as the ground truth according to the real behavior and used to evaluate our partition method. For manual partition results, we calculate two statistics: N_p (numbers of position RSSI groups), N_t (numbers of transition RSSI groups). For the partition results of our iterative change point detection method, we calculate six statistics: N_p, N_t, P_{fp} (false positive rates of position RSSI groups), P_{fn} (false negative rates of position RSSI groups), T_{fp} (false positive rates of transition RSSI groups), T_{fn} (false negative rates of transition RSSI groups). Here, the P_{fp} means the proportion of the sequence part, which belongs to a position group according to iterative change point detections result, is not contained in the corresponding position group according to manual partition result. And vice versa the P_{fn}. Accordingly, we calculate the T_{fp} and T_{fn} of a transition group in the same way.

TABLE V

STATISTICAL RESULTS OF MANUAL PARTITION AND ITERATIVE CHANGE POINT DETECTION

	Actuality		**Iterative Change Point Detection**					
	N_p	N_t	N_p	N_t	P_{fp}	P_{fn}	T_{fp}	T_{fn}
S1	1	0	1	0	0.000	0.000	0.000	0.000
S2	2	1	2	1	0.003	0.000	0.000	0.071
S3	2	1	2	1	0.074	0.004	0.024	0.476
S4	2	1	2	1	0.040	0.000	0.000	0.100
S5	3	2	3	2	0.042	0.006	0.024	0.274
S6	3	2	3	2	0.037	0.003	0.006	0.213
S7	3	2	4	3	0.003	0.040	0.010	0.177
S8	4	3	4	3	0.022	0.000	0.000	0.135
S9	8	7	8	7	0.044	0.070	0.249	0.153
S10	10	9	10	9	0.045	0.000	0.000	0.169

As Table 5 shows, we have two observations. First, we find that our iterative change point detection method can get correct N_p and N_f except sequence 7. The reason for sequence 7's error is that our algorithm partitions its first position group into two groups and get an extra transition group. For the extra transition group, we cannot find an appropriate matching. However, this only slightly impacts the final result, as the statistical distributions of extra position groups are still similar with that of sequence 7's first group. Second, we can find that P_{fp}, P_{fn} and T_{fp} are small, lower than 4% in average. Because transition groups are short and our method makes a conservative choice to the start and end of transition groups, T_{fn} is a little higher. This is addressed by NiFi. Even when a small part of a transition RSSI group is lost, NiFi still identify the correct path through the score calculation algorithm.

VIII. DISCUSSION

Accuracy: NiFi's identification accuracy is relative to the user activity. In our experiments, the total accuracy is 90.83% in average. When the user frequently moves, NiFi can achieve a high accuracy up to 98.33%. So the upper limit of NiFi's accuracy depends on how much useful signal information can NiFi get. In order to achieve a higher accuracy, we intend to collect some other information such as transmission rate for auxiliary identification. On the other hand, as long as complex physical layer information (e.g., CSI) can be obtained on COTS APs, our approach can be easily extended to extract more features for identification. And we can get the final identification result through majority voting on the decisions of all subcarriers.

The main weakness of NiFi is that a few of legitimate users may be considered as undesired users. NiFi's false negative rate is only 1.11%, but the desired value is zero. To deal with this problem, even if a user is considered as an undesired user, we allow his device to reconnect the AP and be identified again as we mentioned in III-B. It is possible for an undesired user to crack NiFi's protection by following the moving path of a legitimate user. But mimicking the physical signal information is much more difficult than cracking a password.

Reliability: We conduct extensive experiments to study the influences of user gestures of holding the mobiles, the orientations of the antennas in the devices, and different channels. We find that the influences are slight in real applications and our similarity measurement methods for position RSSI groups and transition RSSI groups in V-B have considered how to eliminate small deviations. Besides, the users switch between 2.4 G and 5G may affect the RSSI readings a lot. We solve this problem through treating the STG of different bands differently. For example, a mobile phone using 2.4G will only use the STG constructed by information in 2.4G.

Scalability: In order to ensure that our graph augmentation method is scalable, we address the augmentation problem in the following aspects. First, only the STPs of legitimate users with a high score will be used to update the STG. Second, if a position RSSI group or transition RSSI group of the chosen STP is matched with a vertex or edge in the STG, the corresponding vertex or edge of STG is updated only when the matching score is high. Under these conditions, the statistical distributions of STG's vertexes are stable and the HMMs of STG's edges perform better with the increasing of added position RSSI groups and transition RSSI groups.

Optimization: Some partial methods and parameters can be optimized further. For example, we use an iterative change point detection algorithm to partition an RSSI sequence. Actually, it is difficult to define the exact start point and end point of a position RSSI group or a transition RSSI group. In other ways, we can set a search window and move it gradually to detect the transition RSSI groups with the cumulative sum of slope change according to the method in [11]. In our experiments, we did not carefully select the best parameters, which means we can change some parameters to achieve a higher accuracy with our user identification approach. For example, for open-ended scenario with less number of path restrictions, NiFi relies more on the difference of legitimate areas' signals and undesired areas'. We can adjust the weights of position RSSI groups and transition RSSI groups in Section VI-A for different environments to achieve a higher accuracy.

Overhead: The overhead of NiFi consists of the following parts: 1) computation overhead. To reduce the computation overhead, we develop a pruned DFS-based path matching algorithm in our method. 2) memory overhead. For a signal sequence of n values, the memory overhead of STG's data storage is $O(n)$ and the memory overhead of HMM-based similarity measurement is $O(n^2)$. Our experiments demonstrate that our system running on a AP will not degrade its performance.

IX. 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 environments, 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.

REFERENCES

[1] *Chinese Network Security Report*. Accessed: Sep. 2015. [Online]. Available: http://www.rising.com.cn/about/news/rising/2014-07-30/2014report-s.pdf

[2] *Devices That Are Supported by OpenWrt*. Accessed: Aug. 2015. [Online]. Available: http://wiki.openwrt.org/toh/start

[3] *Hong Kong Wi-Fi Adoption and Security Survey 2014*. Accessed: Oct. 2016. [Online]. Available: https://www.researchgate.net/publication/271723117_Hong_Kong_Wi-Fi_Adoption_and_Security_Survey_2014

[4] *RB912UAG-5HPnD Routerboard INFO*. Accessed: Aug. 2015. [Online]. Available: http://routerboard.com/RB912UAG-5HPnD

[5] *Tools for Kolmogorov-Smirnov Test*. Accessed: Sep. 2015. [Online]. Available: http://www.physics.csbsju.edu/stats/KS-test.html

[6] *WiFi Chip Shipment*. Accessed: Oct. 2015. [Online]. Available: https://www.abiresearch.com/press/wi-fi-chipset-shipments-will-near-18-billion-chips/

[7] *WiFi Master Key*. Accessed: Oct. 2015. [Online]. Available: http://en.wifi.com/#firstPage/

[8] *WiFi Security*. Accessed: Oct. 2015. [Online]. Available: http://www.bj.xinhuanet.com/bjyw/2014-08/20/c_1112148691_2.htm

[9] H. Abdelnasser, M. Youssef, and K. A. Harras, "Wigest: A ubiquitous WiFi-based gesture recognition system," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Hong Kong, Apr./May 2015, pp. 1472–1480.

[10] F. Adib, Z. Kabelac, and D. Katabi, "Multi-person localization via RF body reflections," in *Proc. 12th USENIX Conf. Netw. Syst. Design Implement.*, 2015, pp. 279–292.

[11] K. Ali, A. X. Liu, W. Wang, and M. Shahzad, "Keystroke recognition using WiFi signals," in *Proc. 21st Annu. Int. Conf. Mobile Comput. Netw.*, 2015, pp. 90–102.

[12] P. Bahl and V. N. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," in *Proc. 19th Annu. Joint Conf. IEEE Comput. Commun. Soc. (INFOCOM)*, vol. 2, Mar. 2000, pp. 775–784.

[13] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.

[14] A. A. Mahimkar *et al.*, "Detecting the performance impact of upgrades in large operational networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 4, pp. 303–314, 2011.

[15] R. Nandakumar, K. K. Chintalapudi, and V. N. Padmanabhan, "Centaur: locating devices in an office environment," in *Proc. 18th Annu. Int. Conf. Mobile Comput. Netw.*, 2012, pp. 281–292.

[16] R. Nandakumar, B. Kellogg, and S. Gollakota. (Nov. 2014). "Wi-Fi gesture recognition on existing devices." [Online]. Available: https://arxiv.org/abs/1411.5394

[17] J.-G. Park *et al.*, "Growing an organic indoor location system," in *Proc. 8th Int. Conf. Mobile Syst., Appl., Services*, 2010, pp. 271–284.

[18] Q. Pu, S. Gupta, S. Gollakota, and S. Patel, "Whole-home gesture recognition using wireless signals," in *Proc. 19th Annu. Int. Conf. Mobile Comput. Netw.*, Miami, FL, USA, Sep./Oct. 2013, pp. 27–38.

[19] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan, and R. Sen, "Zee: Zero-effort crowdsourcing for indoor localization," in *Proc. 18th Annu. Int. Conf. Mobile Comput. Netw.*, 2012, pp. 293–304.

[20] S. Sen, J. Lee, K.-H. Kim, and P. Congdon, "Avoiding multipath to revive inbuilding wifi localization," in *Proc. 11th Annu. Int. Conf. Mobile Syst., Appl., Services*, 2013, pp. 249–262.

[21] 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. Mobile Comput.*, vol. 13, no. 4, pp. 907–920, Apr. 2014.

[22] 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 *Proc. 11th Int. Conf. Adv. Mobile Comput. Multimedia (MoMM)*, Vienna, Austria, Dec. 2013, p. 43.

[23] D. Vasisht, S. Kumar, and D. Katabi, "Decimeter-level localization with a single WiFi access point," in *Proc. 13th USENIX Symp. Netw. Syst. Design Implement. (NSDI)*, 2016, pp. 165–178.

[24] G. Wang, Y. Zou, Z. Zhou, K. Wu, and L. M. Ni, "We can hear you with Wi-Fi!" in *Proc. 20th Annu. Int. Conf. Mobile Comput. Netw. (MobiCom)*, Maui, HI, USA, Sep. 2014, pp. 593–604.

[25] H. Wang *et al.*, "No need to war-drive: unsupervised indoor localization," in *Proc. 10th Int. Conf. Mobile Syst., Appl., Services*, 2012, pp. 197–210.

[26] J. Wang, D. Vasisht, and D. Katabi, "RF-IDraw: virtual touch screen in the air using RF signals," in *Proc. ACM SIGCOMM Conf.*, Chicago, IL, USA, Aug. 2014, pp. 235–246.

[27] X. Zheng, J. Wang, L. Shangguan, Z. Zhou, and Y. Liu, "Smokey: Ubiquitous smoking detection with commercial WiFi infrastructures," in *Proc. INFOCOM*, Apr. 2016, pp. 1–9.

[28] Y. Xie, Z. Li, and M. Li, "Precise power delay profiling with commodity Wi-Fi," in *Proc. 21st Annu. Int. Conf. Mobile Comput. Netw. (MobiCom)*, Paris, France, Sep. 2015, pp. 53–64.

[29] J. Xiong and K. Jamieson, "ArrayTrack: A fine-grained indoor location system," in *Proc. NSDI*, 2013, pp. 71–84.

[30] J. Xiong, K. Sundaresan, and K. Jamieson, "Tonetrack: Leveraging frequency-agile radios for time-based indoor wireless localization," in *Proc. 21st Annu. Int. Conf. Mobile Comput. Netw.*, 2015, pp. 537–549.

[31] Z. Yang, C. Wu, and Y. Liu, "Locating in fingerprint space: Wireless indoor localization with little human intervention," in *Proc. 18th Annu. Int. Conf. Mobile Comput. Netw.*, 2012, pp. 269–280.

[32] M. Youssef and A. Agrawala, "The Horus WLAN location determination system," in *Proc. 3rd Int. Conf. Mobile Syst., Appl., Services*, 2005, pp. 205–218.

[33] H. Zhu, Y. Zhuo, Q. Liu, and S. Chang, "π-splicer: Perceiving accurate CSI phases with commodity WiFi devices," *IEEE Trans. Mobile Comput.*, vol. 17, no. 9, pp. 2155–2165, Sep. 2018.

Linsong Cheng (S'18) received the B.E. degree in software engineering from Tsinghua University, Beijing, China, where he is currently pursuing the Ph.D. degree. His current research interests are indoor localization, edge computing, and wireless sensing.

Jiliang Wang (M'18) received the B.E. degree in computer science and technology from the University of Science and Technology of China and the Ph.D. degree in computer science and engineering from The Hong Kong University of Science and Technology. He is currently an Associate Professor with the School of Software and TNLIST, Tsinghua University, China. His research interests include wireless and sensor networks, Internet of Things, and mobile computing.