DMAD: Data-Driven Measuring of Wi-Fi Access Point Deployment in Urban Spaces

JIAXING SHEN and JIANNONG CAO, The Hong Kong Polytechnic University
XUEFENG LIU, Huazhong University of Science and Technology
CHISHENG ZHANG, The Hong Kong Polytechnic University

Wireless networks offer many advantages over wired local area networks such as scalability and mobility. Strategically deployed wireless networks can achieve multiple objectives like traffic offloading, network coverage, and indoor localization. To this end, various mathematical models and optimization algorithms have been proposed to find optimal deployments of access points (APs).

However, wireless signals can be blocked by the human body, especially in crowded urban spaces. As a result, the real coverage of an on-site AP deployment may shrink to some degree and lead to unexpected dead spots (areas without wireless coverage). Dead spots are undesirable, since they degrade the user experience in network service continuity, on one hand, and, on the other hand paralyze some applications and services like tracking and monitoring when users are in these areas. Nevertheless, it is nontrivial for existing methods to analyze the impact of human beings on wireless coverage. Site surveys are too time consuming and labor intensive to conduct. It is also infeasible for simulation methods to predict the number of on-site people.

In this article, we propose DMAD, a Data-driven Measuring of Wi-Fi Access point Deployment, which not only estimates potential dead spots of an on-site AP deployment but also quantifies their severity, using simple Wi-Fi data collected from the on-site deployment and shop profiles from the Internet. DMAD first classifies static devices and mobile devices with a decision-tree classifier. Then it locates mobile devices to grid-level locations based on shop popularities, wireless signal, and visit duration. Last, DMAD estimates the probability of dead spots for each grid during different time slots and derives their severity considering the probability and the number of potential users.

The analysis of Wi-Fi data from static devices indicates that the Pearson Correlation Coefficient of wireless coverage status and the number of on-site people is over 0.7, which confirms that human beings may have a significant impact on wireless coverage. We also conduct extensive experiments in a large shopping mall in Shenzhen. The evaluation results demonstrate that DMAD can find around 70% of dead spots with a precision of over 70%.

CCS Concepts: • **Information systems → Data analytics**; *Expert systems*;

Additional Key Words and Phrases: Wi-Fi AP, AP deployment measuring, data-driven approach, room-level localization

The work described in this article was partially supported by the funding for Project of Strategic Importance provided by The Hong Kong Polytechnic University (Project Code: 1-ZE26). It was also supported by NSFC/RGC Joint Research Scheme with RGC No. N_PolyU519/12 and NSFC Key Grant with Project No. 61332004.

Authors' addresses: J. Shen, J. Cao, and C. Zhang, Department of Computing, The Hong Kong Polytechnic University; emails: {csjshen, csjcao, csczhang}@comp.polyu.edu.hk; X. Liu, School of Computer Science and Technology; email: csxfliu@gmail.com.

https://doi.org/10.1145/3065949

ACM Reference format:
Jiaxing Shen, Jiannong Cao, Xuefeng Liu, and Chisheng Zhang. 2017. DMAD: Data-Driven Measuring of Wi-Fi Access Point Deployment in Urban Spaces. *ACM Trans. Intell. Syst. Technol.* 9, 1, Article 11 (August 2017), 29 pages.
https://doi.org/10.1145/3065949

1 INTRODUCTION

Wireless networks are remarkably important in modern societies, not only just for wireless communication but also as a key enabler of numerous novel applications. One well-known example is wireless based tracking and monitoring systems (Musa and Eriksson 2012; Shen et al. 2016; Bahl and Padmanabhan 2000; Wang et al. 2014). Besides, wireless networks offer many advantages over wired local area networks such as scalability and mobility. It is convenient to access network resources from any locations within the coverage of wireless networks. It can also be set up easily in a quick and expandable way. Last, wireless networks are cost-effective, since wiring costs are eliminated or reduced.

As one of the predominant problems, the wireless networks layout problem and the access point (AP) deployment problem have been extensively studied over decades (Adickes et al. 2002), since strategically deployed wireless networks can achieve multiple objectives like maximizing ratios of traffic offloading (Bulut and Szymanski 2013) and wireless coverage (Chen et al. 2013) and improving indoor localization accuracy (Chen et al. 2013). Existing solutions can be classified as site surveys and simulation approaches. For site surveys, engineers with electronic monitoring equipment such as spectrum analyzers walk throughout the facility to measure wireless signal quality. Based on the information, engineers attempt to identify potential locations for APs that would minimize the disruption of service (Revolutionwifi 2013). However, site surveys require specialized equipment and extensive manpower, which is quite expensive, especially for large areas. Therefore, many simulation approaches (Liao et al. 2011; Meng et al. 2012; While and McDonald 2014) are proposed by modeling the AP deployment problem as an optimization problem. Those simulation methods are mostly built on the basis of propagation loss models that characterize how wireless signal attenuate over distances and different obstacles. Simulation methods usually consist of two stages, an iterative stage to calculate the minimum number of APs, and an optimization stage to find out optimal locations of those APs towards one or more objectives.

However, Wi-Fi signals can be blocked by the human body (Sen et al. 2012), especially in crowded urban spaces. As a result, it could result in unexpected dead spots (areas without wireless transmission coverage) as illustrated in Figure 1(b), where walking people cause real coverage of the on-site AP deployment to shrink to some extent. These dead spots are undesirable, since they degrade the user experience in network service continuity, on one hand, and, on the other hand, paralyze some applications and services like tracking and monitoring when users are in these areas. Nevertheless, it is nontrivial for existing methods to analyze the impact of human beings on wireless coverage. It is too time consuming and labor intensive to measure wireless coverage status for a long time using site survey methods. Also, site surveys may disrupt the ongoing activities (like shopping activities) in the facility. For simulation methods, it is infeasible to consider the impact, since the number of people cannot be determined. Moreover, neither site surveys nor simulation approaches are able to evaluate the severity of different dead spots in a quantitative way.

As explained above, wireless networks can suffer from unpredictable influences of changeable interactions among multiple devices, specific hardware, and human activities. These influences might further lead to a difference between real-world functioning and design-time functioning (Kulin et al. 2016). Recently, the data-driven design of intelligent wireless networks is gaining

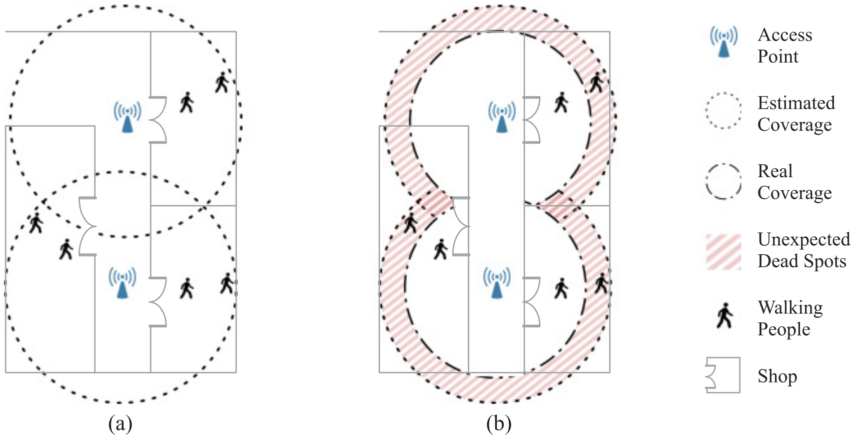


Fig. 1. A simple illustration of the impact of human beings on wireless coverage. (a) Ideal coverage of APs; (b) real coverage of APs in the presence of walking people. The shadow areas in (b) are potential dead spots caused by human beings.

popularity due to its capability to better understand the behavior of complex systems that cannot be easily modeled or simulated. Data science or “data-driven research” is a research approach that uses real-life data to gain insights about the behavior of systems. It enables the analysis of various systems to assess whether they function according to the intended design and as seen in simulations.

In this article, we propose **DMAD**, a **D**ata-driven **M**easuring of Wi-Fi **A**ccess point **D**eployment, to estimate dead spots and quantify their severity based on simple Wi-Fi data collected from the on-site AP deployment and shop data from the Internet. DMAD first classifies static devices and mobile devices with a decision-tree classifier. Then it locates mobile devices to shop-level locations on the basis of two observations of heuristics. (1) We find that the visit duration in different shops differs, for example, people stay longer in restaurants than in clothing shops. (2) Different shops have different popularity in attracting customers, and thus the probability of people appearing in a shop should closely relate to the popularity. These two observations could help to improve the accuracy of shop-level localization. Last, for each area, we estimate the probability of a dead spot in different time slots and derive their severity combining the probability and the number of people. Since if a dead spot appears in an area with more potential users, its severity should be higher.

The contributions of this work are summarized as follows:

- To the best of our knowledge, we are the first to propose the AP deployment measuring problem (ADM).
- We also propose a data-driven approach (DMAD) to solve the ADM problem, which can identify around 70% of dead spots with a precision of over 70%.
- The performance of DMAD is carefully evaluated using data collected from a real AP deployment of a large shopping in Shenzhen.

The remainder of the article is organized as follows. In the next section, we summarize the related work. In Section 3, we give an overview, including the preliminaries, the feasibility of using Wi-Fi data to study wireless coverage, the impact of people on wireless coverage, and the framework of DMAD. Sections 4 to 7 elaborate on each component of DMAD. In Section 8, we present the detailed evaluation of each component and followed by a conclusion.

2 RELATED WORKS

2.1 AP Deployment Problem

AP deployment problem is to find the minimum number of APs and their optimal locations to achieve one or more objectives. Existing solutions can be classified into two categories, site surveys and simulation approaches. Both methods have their own pros and cons. Site surveys are more accurate and robust, but they require sophisticated electronic monitors and extensive manpower, which is time consuming and labor intensive, especially for large areas. Also, site surveys have the potential for disrupting normal operations at the site (Adickes et al. 2002). Simulations are easy and cheap to conduct, but they cannot precisely characterize the radio frequency (RF) propagation due to the vulnerability of wireless signal and the dynamic environment.

2.2 Site Surveys

Site surveys can provide a solid understanding of the on-site RF behavior, identify any dead spots and reveal areas of channel interference (Revolutionwifi 2013). These surveys are usually conducted by engineers with specialized electronic monitoring equipment such as spectrum analyzers. According to objectives, site surveys can be classified into three categories, predictive modeling surveys, pre-deployment surveys, and post-deployment surveys.

Predictive modeling site surveys use software programs to model the facility and RF environment. Those programs can help to outline the required coverage areas using facility floor plans, estimate RF signal attenuation according to different RF environments, and predict the minimum number of APs and their locations. Strictly speaking, predictive modeling surveys belong to simulation methods, as propagation loss models and optimization algorithms are utilized rather than using real equipment to characterize the on-site RF behavior.

Pre-deployment site surveys are often called "AP-on-a-stick" surveys and are performed before setting up a wireless network. In the survey, spectrum analysis is an integral part, which can identify sources of RF interference and dead spots that would cause performance issues. With this survey, a better wireless network design can be achieved by characterizing the RF behavior in the facility, which is uniquely tailored to the physical properties of the environment. It can also be used to verify and adjust a preliminary Wi-Fi network design.

Post-deployment site surveys are performed after the APs have been installed and configured. This type of site survey reflects the RF signal propagation characteristics of the deployed wireless network. The focus is to validate that the performance of the deployed network matches the original network design.

2.3 Simulation Approaches

Since site surveys are time consuming and labor intensive, numerous simulation methods have been proposed to avoid extensive measurements and expensive physical experiments. Simulation methods first emulate the RF propagation in the target environment, then model it as a mathematical problem by using propagation loss models, and, finally, exploit various optimization algorithms to solve it towards one or more objectives.

Once the propagation loss model is determined, given an AP setting, the signal strength of this AP at any locations on the site can be estimated. Then the problem is to find the minimum number of APs and their optimal locations to satisfy some predefined thresholds like the minimum RSS (Received Signal Strength) value. Different simulation methods mainly differ in their propagation loss models, objectives, and optimization algorithms.

2.3.1 Propagation Loss Models.. Propagation loss models describe how RF signal attenuate over physical distance and through different obstacles that have been studied extensively over decades

(Hashemi 1993; Kreuzgruber et al. 1994; Schoberl 1995). Hashemi conducted a comprehensive survey about mathematical and statistical modeling of individual characteristics of propagation losses in Hashemi (1993). Other researchers studied propagation loss of signal with multipath characteristics using ray-tracing techniques in Kreuzgruber et al. (1994), while Schoeberl modeled the propagation losses combining ray tracing and Monte Carlo simulation in Schoberl (1995). All these works indicate that average received signal power decreases logarithmically with distance, which are described in Equation (1),

$$PL = PL_0 + 10 \cdot \gamma \cdot \log_{10} \frac{d}{d_0} + \sum_{i=1}^{n} N_i \cdot L_i. \tag{1}$$

PL is the total path loss, PL_0 is the path loss at the reference distance d_0, γ is the path loss attenuation factor derived from measurements, d is the length of the path, d_0 is the reference distance, N_i represents the number of a particular type of obstacles, and L_i represents the loss associated with that type of obstacles.

2.3.2 Optimization Objectives.. The objectives are usually wireless coverage (Kouhbor et al. 2006; Liao et al. 2011), offloading ratio (Bulut and Szymanski 2013; Kim et al. 2013), fingerprint differences (Liao et al. 2011; Meng et al. 2012), and so on. Recently, more works (Liao et al. 2011; Chen et al. 2013) focus on achieving the combination of multiple objectives.

2.3.3 Optimization Methods.. The Nelder-Mead simplex algorithm is adopted to find the optimal AP locations for maximizing the coverage ratio in Fortune et al. (1995). In Adickes et al. (2002), a one-by-one trial method has been proposed to find the minimum number of APs needed to cover a given site. For a given number of APs, the genetic algorithm (GA) optimizer is used to perform AP location optimization. The authors in Vilovic et al. (2009) use a neural network approach to perform propagation prediction and adopt an ant colony optimization approach to optimize the AP locations and to maximize the average received power. In Sharma et al. (2010), the simulated annealing (SA) algorithm is utilized to find the minimum number and optimal transmission power of APs, but the AP locations are not optimized. Wang et al. exploited GA to the placement of APs with heterogeneous costs and capacities in Wang and Kao (2012) and Wang and Chen (2012). While and McDonald (2014) proposed a multi-objective evolutionary algorithm for three criteria minimized cost, maximized coverage, and minimized service refusal.

2.4 Using Data Science in Wireless Networks.

Data science, or "data-driven research," is a research approach that uses real-life data to gain insights about the behavior of target systems (Kulin et al. 2016). It enables the analysis of various systems to assess whether they function according to the intended design and as seen in simulations.

Wireless networks can exhibit unpredictable interactions between algorithms from multiple protocol layers, interactions between multiple devices, and hardware-specific influences (Kulin et al. 2016). These interactions may further result in a difference between real-world functioning and design-time functioning. Data science methods can be utilized to detect the actual behavior and hopefully provide insights to improve the system performance.

Numerous research areas like large-scale social networks, advanced business, and healthcare processes have successfully adopted data-driven approaches to analyze networked interactions. In traditional wireless research, it often starts with theoretical models to devise solutions that are then evaluated using a simulator or experimental setup (Kulin et al. 2016). In contrast to traditional approaches, research works such as Crotti et al. (2007), Liu and Cerpa (2011), and Liu and Cerpa (2014) use a data-driven approach, starting from large, real-life wireless datasets, to extract knowledge about wireless systems.

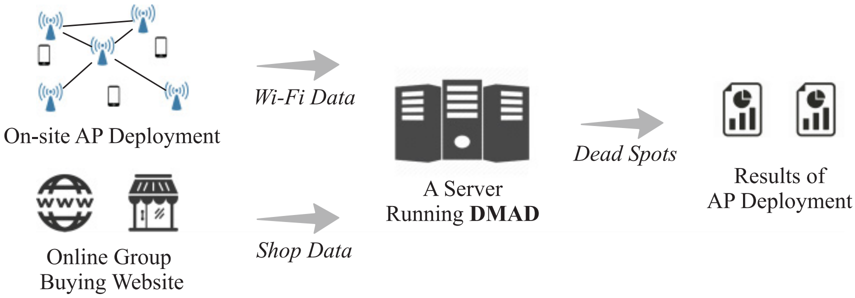


Fig. 2. Workflow of the whole system.

For example, in Crotti et al. (2007), the authors proposed a data-driven solution for fingerprinting wireless devices that can help existing network access control systems to enhance network security by allowing access only for certain devices or device types (devices that have the same hardware configuration). Differing from traditional security mechanisms that rely on device authentication based on public key cryptography and digital certificates, which could be simply transferred to another device. The proposed data-driven approach relies on distinguishing devices by looking into the statistical distribution of inter-arrival times between packets generated by the same device and a particular application. The authors formulated this as a classification problem, proved their hypothesis from two testbeds, and, finally, solved the problem with an artificial neural network model.

3 OVERVIEW

In this section, we give an overview of DMAD by introducing the basics of Wi-Fi AP deployment measuring problem, studying the feasibility of using Wi-Fi data to measure wireless coverage and dead spots, and investigating the impact of people on wireless coverage.

3.1 Preliminaries

DMAD is a data-driven approach to measuring wireless coverage of a given AP deployment. First, we collect Wi-Fi data from deployed APs and shop data from the Internet. Then we conduct a comprehensive analysis on the data to estimate dead spots and quantify their severity. The whole process is depicted in Figure 2. Some of the notions used in this article are listed in Table 1.

3.2 The Feasibility of Using Wi-Fi Data to Measure Wireless Coverage and Dead Spots

Before estimating dead spots, we study the feasibility of using Wi-Fi data to measure AP coverage status and how to represent dead spots.

Figure 3 depicts a simple scenario with one AP and two fixed points. There are two smartphones at both points, respectively, keeping broadcasting Wi-Fi packets. Given transmission time T_t and coverage time T_c of both devices, how do we measure the wireless coverage status at both points?

The transmission time represents the total amount of time of sending packets on the smartphone, while the coverage time means the total amount of time of receiving packets from a smartphone on the AP side. As smartphones would send many packets within 1 minute, we count T_t and T_c in a granularity of 1 minute, which means that if the smartphone sends any packet(s) in the duration of 1 minute, then we increase T_t by 1. Due to packet loss and physical constraint (like distance), we have $T_t \geq T_c$.

Table 1. Notions Used in This Paper

Symbol	Explanation
$\mathcal{A}$	A set of APs, $\mathcal{A} = \{a_1, a_1, \ldots\}$
$\mathcal{S}$	A set of shops, $\mathcal{S} = \{s_1, s_2, \ldots\}$
$\mathcal{D}$	A set of smart devices , $\mathcal{D} = \{d_1, d_2, \ldots\}$
$\mathcal{G}$	A set of non-overlapping grids , $\mathcal{G} = \{g_1, g_2, \ldots\}$
$\mathcal{T}$	A set of time slots, $\mathcal{T} = \{t_1, t_2, \ldots\}$, t_i is a period of time
M_j	Connectivity matrix of d_j, $M_j = [V_j(1) \quad V_j(2) \quad \cdots]$
$V_j(i)$	Connectivity vector of d_j at time i, $V_j(i) = [v_{i1} \quad v_{i2} \quad \ldots \quad v_{i\lvert\mathcal{A}\rvert}]^T$
v_{ij}	Binary variable, $v_{ij} \leftarrow 1$ if a_j hears from the device at time i
ρ	Coverage ratio of an AP, $\rho = T_c/T_t$
T_c	Coverage time, how long an AP can hear from a device
T_t	Transmission time, how long a device sends packets
ζ	Ratio of change, $\zeta \in [-1, 1]$
$\Omega_j(i)$	Coverage ratio vector of d_j during t_i, $\Omega_j(i) = [\rho_1 \quad \cdots \quad \rho_{\lvert\mathcal{A}\rvert}]^T$
$\mathbf{D}_w$	Unlabeled Wi-Fi data, $\mathbf{D}_w = \{\mathcal{E}_1, \mathcal{E}_2, \ldots\}$, $\mathcal{E}_1 = (a_i, d_j, t^k_{start}, t^k_{end})$
$\mathbf{D}^*_w$	Labeled Wi-Fi data, with the label of grid information
$\mathbf{D}_s$	Unlabeled Shop data, $\mathbf{D}_s = \{\mathcal{I}_1, \mathcal{I}_2, \ldots\}$, $\mathcal{I}_1$ is a set of attributes
$\mathbf{D}^*_s$	Labeled Shop data, labels are # of people and their duration time
$\hat{\mathcal{L}}_j(i)$	Estimated location of device d_j at time i
R	$R = (n_{ij})$, n_{ij} is the number of people in shop s_j during t_i
H	$H = (\eta_{ij})$, η_{ij} is number of people in g_j during t_i
$\mathbb{S}_j$	A set of shops that are located in grid g_j
$\mathbb{N}_j$	A set of grids that are neighboring to grid g_j
$\mathbb{V}_j$	A set of connectivity vectors collected in grid g_j

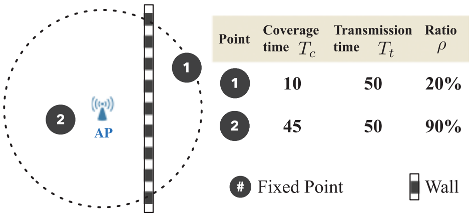
Ideally, if the coverage status is good enough, then T_c would approximate to T_t. Otherwise, T_c would be much smaller than T_t. Based on this intuition, we use a coverage ratio $\rho = T_c/T_t$ to represent the wireless coverage status on a fixed point. In the example of Figure 3, $\rho_1 = 10/50 = 0.2$, $\rho_2 = 45/50 = 0.9$, ρ_k represents the coverage status on the kth point. The larger the ratio, the better the coverage.

Then what is relationship between coverage status and dead spots under this simple scenario? Simply speaking, a point with good coverage status (i.e., large coverage ratio) is impossible to be a dead spot. Instead, those with terrible coverage status are more likely to be dead spots. So we propose a probabilistic representation for dead spots based on coverage ratio as illustrated in Equation (2). $P_{DS}(p_i)$ is the probability that point p_i is a dead spot,

$$P_{DS}(p_i) = 1 - \rho_i \tag{2}$$

Since wireless coverage status would change over time, using deterministic representation of P_{DS} might be error prone, it is better to utilize such a probabilistic representation. Under this definition, $P_{DS}(p_1) = 0.8$ and $P_{DS}(p_2) = 0.1$, which means p_1 is more likely to be a dead spot.

However, the example in Figure 3 just illustrates the simplest case. In real scenarios, we have three imperative issues. First, we can never know the exact transmission time T_t of a smartphone by passively sniffing its Wi-Fi data. Second, a location could be covered by multiple APs with the same SSID. Last, the relation between coverage status and dead spots is much more complex than that of the simplest example.



Point	Coverage time T_c	Transmission time T_t	Ratio ρ
1	10	50	20%
2	45	50	90%

Fig. 3. Example of using Wi-Fi data to represent wireless coverage status. The units of T_c and T_t are both minute. Coverage ratio $\rho = T_c/T_t$.

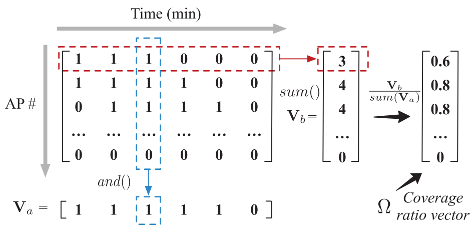
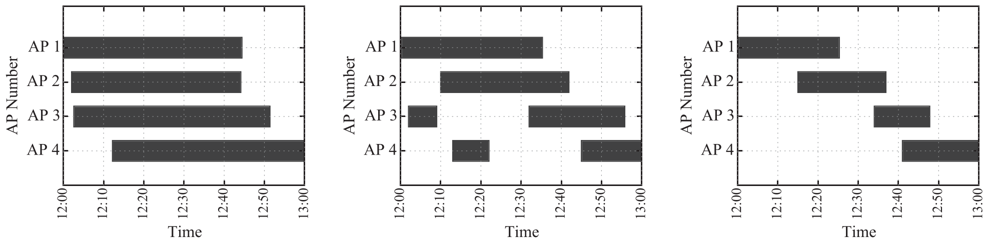


Fig. 4. Illustration of using connectivity matrix M to calculate coverage ratio vector Ω. *and*() does the AND-operation of the column vector.



(a) Case I

(b) Case II

(c) Case III

Fig. 5. Typical examples of different coverage statuses at different locations from 12:00 to 13:00. The black bar of an AP indicates the AP can "hear"[1] from the device on that location. Intuitively, the order of coverage status is Case I > Case II > Case III.

For the first issue, we assume that if the smartphone sends any packets, then at least one AP would receive the packet; otherwise, the smartphone is not sending any packets. Based on this assumption, we can calculate an approximation $\hat{T_t}$ of T_t.

For the second issue, when a location is covered by multiple APs, we can transform the Wi-Fi data from user's device d_j into a connectivity matrix M_j using Algorithm 1 described in Section 4. Figure 4 gives an example connectivity matrix and shows how to derive the coverage status from the matrix. Each row vector $V_j(i)$ in M_j is called a connectivity vector. It shows the connectivity information of the device with all APs at a specific time i. For example, $V_j(i) = [1 \quad 1 \quad 0 \quad 0]^T$ means the device d_j is within the coverage of AP a_1 and a_2 at time i. As shown in Figure 4, $\hat{T_t}$ can be calculated by summing up the vector V_a. Then for each AP, we can calculate its coverage time and then derive its coverage ratio. $\Omega = [\rho_1 \quad \rho_2 \quad \cdots]^T$ is a coverage ratio vector containing coverage ratios of all APs.

For the last issue, our basic idea is still that a point with terrible coverage is more likely to be a dead spot, but it requires more meticulous design.

We show three typical coverage status in Figure 5. Intuitively, the order of coverage status should be Case I > Case II > Case III. Since in Case I, the coverage ratios of 4 APs are very large; while in Case III, the ratios are all quite small. This ranking can be explained from another perspective: All large ratios indicate that the location is covered by multiple APs for most of the time, and thus the probability of dead spots is significantly smaller than that of all small ratios.

[1] "Hear" means the AP receives any packet(s) from the device.

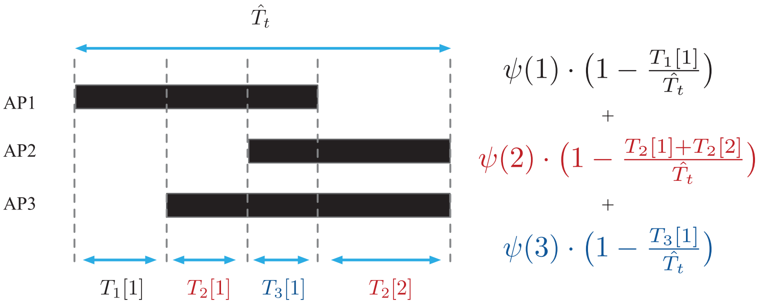

$\hat{T}_t$
AP1
AP2
AP3
$T_1[1]$
$T_2[1]$
$T_3[1]$
$T_2[2]$
$\psi(1)\cdot(1-\frac{T_1[1]}{\hat{T}_t})$
+
$\psi(2)\cdot(1-\frac{T_2[1]+T_2[2]}{\hat{T}_t})$
+
$\psi(3)\cdot(1-\frac{T_3[1]}{\hat{T}_t})$

Fig. 6. Illustration of translating a connectivity matrix into probability of dead spots.

Based on the observation above, we devise Equation (3) to map a connectivity matrix M_j into the probability of dead spots. Actually, $1 - sum(T_i)/\hat{T}_t$ is the coverage ratio when the location or area is covered by i APs. T_i is an array of coverage time when covered by i APs. $sum(T_i)$ sums up T_i. Since $\hat{T}_t$ is just an approximation of the real transmission time as mentioned in the earlier part of this section, we use a decay function $\psi(i)$ to represent the initial probability of dead spots when covered by i APs.

Figure 6 illustrates the idea of Equation (3) by showing an example of applying the equation. $T_2[1]$ means the first coverage time when the location is covered by two APs,

$$P_{DS}(M_j) = \sum_{i=1}^{|\mathcal{A}|}\left(\psi(i)\cdot\left(1-\frac{sum(T_i)}{\hat{T}_t}\right)\right). \tag{3}$$

3.3 More Discussion on Coverage Ratio and Dead Spots

Here we discuss two issues to clarify both concepts and eliminate potential misunderstandings of coverage ratio and dead spots.

The first issue is about measuring coverage status using data collected on the AP side. If an AP can hear from a device, then it is very likely that the device can also hear from the AP. However, even though a device can hear from an AP, the AP sometimes cannot hear back from the device, since the transmit power of an AP is usually larger than that of a mobile device. This indicates that using data collected on APs and data on mobile devices represents different coverage. The coverage from the AP side is a proper subset of the coverage from the device side. DMAD focuses on the former coverage that is more meaningful. If the AP cannot hear from the device, then a range of services and applications residing on the AP side like passive tracking (Musa and Eriksson 2012) cannot work. Worse still, devices cannot access the Internet.

The second issue is about situations where DMAD cannot work. DMAD does not estimate dead spots by directly checking whether there is wireless coverage or not, which is the main idea of site survey. Instead, it estimates the probability of dead spots in a given area (around 20m × 20m) for a period of time based on the coverage status. The coverage status cannot be calculated without coverage time T_c and estimated transmission time $\hat{T}_t$. Both T_c and $\hat{T}_t$ are derived from the packets heard on the AP side. Therefore, if a device has already been on a dead spot, or the device does not send any packets, then DMAD cannot work properly.

However, in real scenarios, both situations are very rare. First, DMAD only estimates the probability of dead spots in expected coverage area, which are supposed to have wireless coverage in normal circumstances. Dead spots in an expected coverage area are caused by the human body and change with on-site people and cannot cover a large area. Therefore, it is quite rare that devices are within dead spots all the time. Second, smartphones keep broadcasting packets even not in use, which is explained in Section 4.

3.4 The Impact of People on Wireless Coverage

We found some static devices that are fixed in location, such as desktops, smart TVs, and IP cameras, in a large shopping mall in Shenzhen. The way we found those devices is well explained in Section 5. For each static device, we transform its Wi-Fi data into 24 connectivity matrices, with each matrix representing the connectivity information for 1 hour. Then we calculate the coverage ratio vector from the connectivity matrix following the procedure in Figure 4.

Usually, early in the morning, there are no people except for few on-duty security officers in a shopping mall. Therefore, the coverage ratio vector $\Omega(s)$ of that period of time reflects the wireless coverage without people, while coverage ratio vector $\Omega(d)$ during the time of 5:00~22:00 indicates wireless coverage in the presence of humans.

We use Equations (4) and (5) to measure the change from $\Omega(s)$ to $\Omega(d)$. The output of the function is a *ratio of change* $\zeta \in [-1, 1]$. The larger ζ is, the poorer the coverage status compared to $\Omega(s)$,

$$\zeta = \left(\frac{\Omega(s)}{sum\left(\Omega(s)\right)}\right)^T \cdot \frac{\Omega(s) - \Omega(d)}{sum(\Omega(s) - \Omega(d))}, \tag{4}$$

$$sum(\Omega(s)) = \sum_{i=1}^{|\mathcal{A}|} \rho_i. \tag{5}$$

In Equation (4), $(\frac{\Omega(s)}{sum(\Omega(s))})^T$ is the transpose of normalized $\Omega(s)$. Those APs with large coverage ratios play dominating roles in coverage status, and therefore their changes should have a larger weight than that of APs with small ratios.

To calculate the ratio of change, we set 3:00~4:00 as $\Omega(s)$ and each hour in 5:00~22:00 as $\Omega(d)$. Then, based on the data collected from the shopping mall in 46 days, we derive the average ratio of change for each hour.

Compared to static devices, it is easier to find mobile devices that are carried by people in the mall. The basic idea is to use the unique MAC address of the mobile device to represent a mobile user. The process is explained n detail in Section 5. So we can also calculate the average number of people in different hours from the accumulated data. The results of correlation analysis of the ratio of change and the number of people are shown in Figure 7. We can see that the Pearson Correlation Coefficient for this is over 0.7, which indicates that people might have a non-negligible impact on wireless coverage and the impact increases with the number of on-site people.

3.5 Framework of DMAD

DMAD has four components, as depicted in Figure 8. The first component is "Data collection," which is to collect desired input data, including Wi-Fi data, shop data, and the floor plan. Then the data are used for "Device classification," which sorts out static devices. After that, we use "Area localization and density calculation" to estimate mobile devices' shop-level locations and analyze human density in different areas and time slots, respectively. Last, "Dead spots estimation" estimates the probability of dead spots and their severity.

4 DATA COLLECTION

Data collection is the first component of DMAD, and it serves as data input for the whole system. We collect two sources of data, Wi-Fi data from deployed APs and shop data from the Internet. The purposes of collecting Wi-Fi data is to determine grid locations, measure wireless coverage,

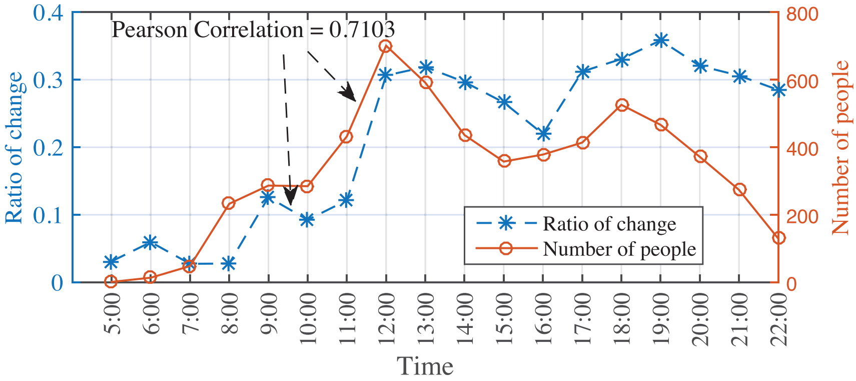


Fig. 7. Correlation analysis of the average *ratio of change* and the average *number of people* from the data collected in a shopping mall in Shenzhen for 46 days.

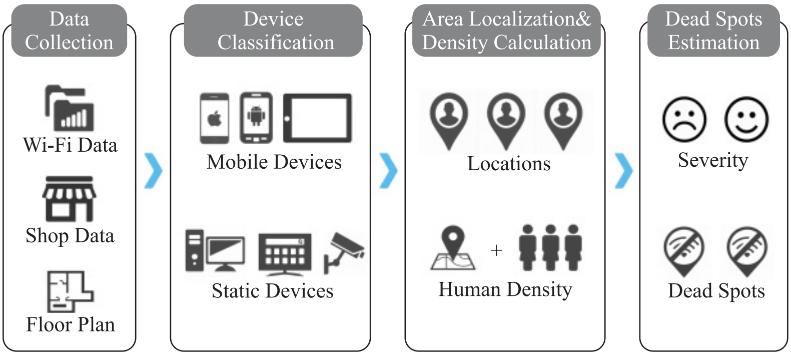


Fig. 8. The framework of DMAD. It consists of four components (data collection, device classification, area localization, and density calculation) and dead spot estimation.

and estimate dead spots of an on-site AP deployment, while shop data can help to improve the accuracy of area localization. In this section, we show that both Wi-Fi data and shop data can be readily collected by introducing details of the data collection processes.

4.1 Wi-Fi Data

Wi-Fi data consist of two parts, a large amount of unlabeled Wi-Fi data $\mathbf{D}_w$ collected from mobile users inside the mall and a small amount of labeled Wi-Fi data $\mathbf{D}_w^*$ from volunteers.

4.1.1 Unlabeled Wi-Fi Data $\mathbf{D}_w$.. $\mathbf{D}_w$ is collected from a large shopping mall in Shenzhen, where we have previously installed 48 APs on five floors. The original purpose of the Wi-Fi network is to provide Internet access for customers in common areas, but we also find that it can be utilized for other applications or services, like indoor localization (Shen et al. 2016). Here we study the problem of measuring Wi-Fi AP deployment in expected coverage areas based on the accumulated Wi-Fi data (46 days in total, starting from May 1, 2015). Figure 9 shows the AP installation and expected coverage area on the ground floor.

The unlabeled Wi-Fi data are passively collected from users' smartphones, as smartphones keep broadcasting Wi-Fi packets (Freudiger 2015) that can be sniffed by off-the-shelf APs. Even when users are not using Wi-Fi services, smartphones send out packets (e.g., probe requests)

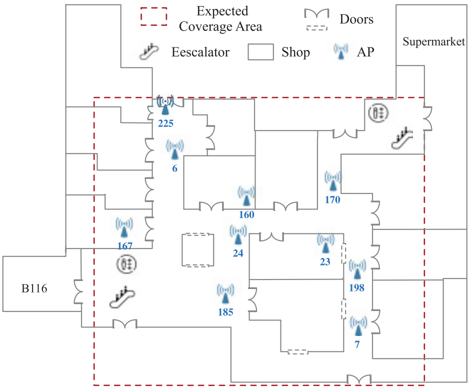


Fig. 9. AP deployment and expected coverage area on the ground floor of the mall.

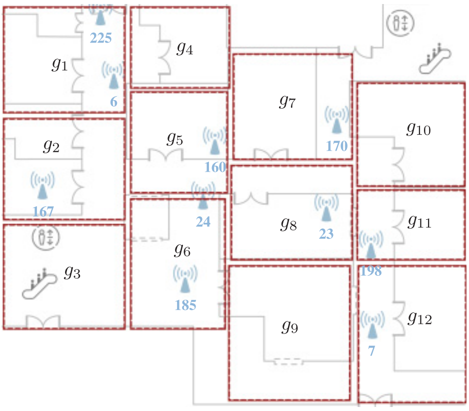


Fig. 10. Grid partition on expected coverage area of the ground floor.

intermittently (Musa and Eriksson 2012). Figure 12 illustrates a simple scenario and lists some descriptive data records.

Each AP works under the OpenWrt[2] system, with a monitor mode virtual network interface[3] enabled. We run Tcpdump (a utility for capturing network traffic) to sniff nearby wireless traffic. More specifically, we use each AP to collect tuples in the format of <AP#, MAC, t_{start}, t_{end}>, and, once the entry is finished, the AP uploads it to the server and then deletes the local entry file. Detailed process is illustrated in the flowchart of Figure 11.

As can be seen from the flowchart, collecting the Wi-Fi data does not require analyzing each packet and extracting the information like with a received signal strength indicator. Instead, we only record the connectivity information, that is, whether the smartphone is under the coverage of an AP. The advantages are twofold; on the one hand, the connectivity information is easy to collect, and it does not add too much of a burden to those APs. On the other hand, it saves much space compared to storing information from every packet, which could be incredibly huge in volume (several Giga bytes form all APs for only 1 day).

Table 2 shows a small fraction of the raw Wi-Fi data. It has four fields as follows: *AP#* shows the id of the AP that hears from the device; MAC is the hashed MAC address of the device; T_{start} is a timestamp that the device is heard for the first time; and, last, T_{end} is a timestamp that the device is last heard by the AP.

Then the raw Wi-Fi data are transformed into a connectivity matrix using Algorithm 1. An example of a connectivity matrix can be found in Figure 4.

4.1.2 *Collecting Labeled Wi-Fi Data* $\mathbf{D}_w^*$..

The label of $\mathbf{D}_w^*$ is the grid information that is manually separated. We separate the expected coverage area of the mall into 60 grids, and Figure 10 illustrates the grid partition of the ground floor. To collect the data, we engage over 20 volunteers in a week with different smartphones, including popular iOS and Android devices. The purpose of $\mathbf{D}_w^*$ is for area localization (in Section 6), which estimates people's area locations based on their Wi-Fi data.

[2] OpenWrt (https://openwrt.org/) is a highly extensible GNU/Linux distribution for embedded devices (typically wireless routers).

[3] Virtual network interface, https://wiki.openwrt.org/doc/networking/network.interfaces.

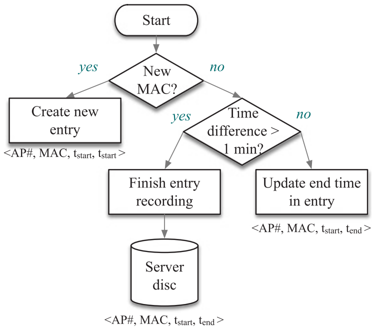
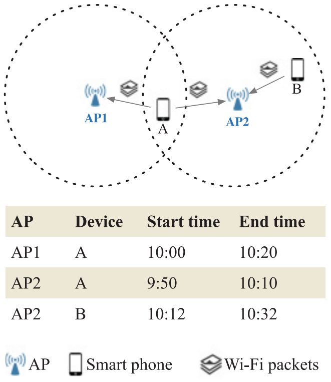


Fig 11. Flow chart of collecting Wi-Fi data in APs.



AP	Device	Start time	End time
AP1	A	10:00	10:20
AP2	A	9:50	10:10
AP2	B	10:12	10:32

Fig 12. An simple illustration of the Wi-Fi data collection.

Table 2. A fraction of raw Wi-Fi data. MAC has been hashed

AP #	MAC	T_{start}	T_{end}
47	3891527	1431652262	1431652271
12	160458	1431721805	1431721810
6	1200164	1431800823	1431800828
30	528517	1431879976	1431880176
6	1585005	1431951873	1431952173
14	398316	1431968982	1431968987
2	685499	1432033589	1432034997
4	102681	1432114009	1432114014
...	...	...	...
22	1114093	1432160871	1432160896
25	1832169	1432514915	1432515031
46	4234664	1432302241	1432302400
8	493476	1432324267	1432324290
22	100731	1432386985	1432387036

ALGORITHM 1: Transform Raw Wi-Fi Data into Connectivity Matrices

```
Data: Raw Wi-Fi data from all APs in a day
Result: $K$ connectivity matrices: $\{M_1, \ldots, M_K\}$
$users \leftarrow$ Group the raw data by the field of MAC address;
for $user_i \in users$ do
    $entries_i \leftarrow user_i$'s raw Wi-Fi data ;
    Create a zero $(|\mathcal{A}| \times 1440)$ matrix $M_i = (m_{ij})$;
    for $entry \in entries_i$ do
        $entry \leftarrow (a_j, d_i, t_s, t_e)$ ;
        transform $t_s, t_e$ into the order of the matrix $s, e$;
        for $k \in [s, e]$ do
            $m_{jk} \leftarrow 1$
        end
    end
end
```

Volunteers are required to collect some “wireless fingerprints” in specific grids following the procedure below. First, they get to the grid, turn on the Wi-Fi function, and record the start time; then they walk around within the grid, after visiting all feasible locations of the grid, record the end time, and turn Wi-Fi off. It usually takes 5 to 10 minutes to finish a collection process.

Since the presence of people can block wireless signals and this will have a negative impact on localization performance, we separated the daytime into several time slots $\mathcal{T} = \{t_1, t_2, \ldots\}$ and collected fingerprints for each time slot. In this way, we collect over 1,500 Wi-Fi data entries for $\mathbf{D}_w^*$.

Although DMAD requires such a labeling process, it takes much less effort compared to site survey. As described in Section 8.1, even simplified site survey usually takes 500~700s to check whether dead spots exist in a grid, while the labeling work takes shorter time (300~600s). More

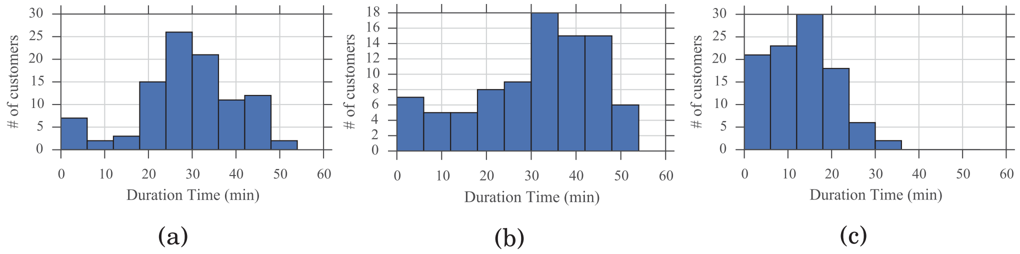


Fig. 13. Histogram of duration time of around 100 customers from three different shops. (a) A fast food restaurant; (b) a traditional Chinese restaurant; (c) a women's accessories shop.

importantly, dead spots are related to human activities, and the detection results may become invalid over time. To detect dead spots next time, site survey needs to start from scratch, while DMAD does not bother to do that, since it merely requires a one-time investment.

4.2 Shop Data

Shop data also consist of two parts, unlabeled shop data $\mathbf{D}_s$ from the Internet and some labeled shop data $\mathbf{D}_s^*$ collected by volunteers.

4.2.1 Collecting Labeled Shop Data $\mathbf{D}_s^*$.. The labels of shop data are the number of people and their visit duration in different shops and time slots, respectively. The number of people is used to calculate a prior probability that people appear in a shop. While visit duration is another kind of "fingerprint," we observe that the time spent in visiting different shops also differs, and we take it as another feature to distinguish users' area locations.

We collect shop data in different time slots in a day, since shops have different popularities during different time slots. For example, restaurants gain more customers during dinner time than clothing shops. To collect the ground truth about the number of people in a shop, we send volunteers to different shops to count the number of customers at different time slots. It usually takes 1 or 2 minutes for volunteers to finish the data collection task. The ground truth is represented in a matrix R in Equation (6), where n_{ij} is the number of people in shop s_j during time slot t_i,

$$R = \begin{bmatrix} n_{11} & n_{12} & n_{13} & \dots & n_{1|\mathcal{S}|} \\ n_{21} & n_{22} & n_{23} & \dots & n_{2|\mathcal{S}|} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ n_{|\mathcal{T}|1} & n_{|\mathcal{T}|2} & n_{|\mathcal{T}|3} & \dots & n_{|\mathcal{T}||\mathcal{S}|} \end{bmatrix}. \tag{6}$$

For visit duration, we use a distribution to represent the duration time in a shop, as it differs from person to person even for the same shop. To collect the data, we ask volunteers to stay near the entrance or the exit of a shop and record the visit duration of customers. Figure 13 shows the duration time of three different shops with around 100 samples. Generally, the distribution can be approximated using a normal distribution.

However, it is too labor intensive and time consuming to collect the distribution of duration time for all shops. We believe that the duration time of a shop is closely related to its type and user ratings. For example, people usually stay in restaurants for around 20 minutes. Also, if the restaurant has a pleasant environment and satisfactory services, customers may choose to stay longer. These observations can be quickly verified from the comparison of the three shops in Figure 13. So we just collect duration time in some typical shops of each category and crawl all shop profiles from

Table 3. A fraction of unlabeled shop data. Some of fields like, floor, location, and average spend is not shown in the table

Name	Type	Likes	Product	Env	Service
Cafe de Coral	Restaurant	41	7.6	7.8	7.6
King of Pastry	Restaurant	3	6.8	6.8	6.9
Benbo	Clothing	2	7.1	7.1	7.1
Shiny Nail	Make-up	32	7.9	8.2	8.3
Costa coffee	Cafe	4	6.8	7.1	7
belle	Clothing	3	6.8	6.8	6.8
Muji	Clothing	8	7.3	7.3	7.3

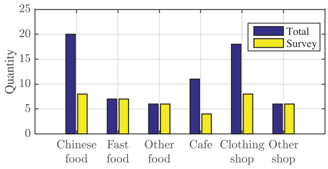


Fig 14. Distribution of total number and surveyed number of shops.

the Internet. Then we utilize machine-learning techniques to predict the distribution of unlabeled shops. A detailed explanation can be found in Section 4.2.2.

4.2.2 Collecting Unlabeled Shop Data $\mathbf{D}_s$.. We collect shop profiles in that mall from Dianping[4] and AutoNavi.[5] For each shop in that mall, we crawl its type (like clothing shop and restaurant), location, the number of positive comments (comments with more than 3 stars), and user ratings about products, environment, and services between May 1, 2015 and June 15, 2015. We exploit Scrapy[6] to crawl the desired data and save them to a local file. A fraction of collected data is shown in Table 3.

There are 68 shops of interest in that mall, and we classify those shops into six categories. The total number and the number of surveyed shops are shown in Figure 14. Among the six categories, Chinese food restaurants, clothing shops, and cafes are the top three categories in terms of total number, and we collect duration time from some of these categories. For other categories, we just collect data from all shops.

To predict mean and standard variance of the distribution is a regression problem. The predictor variables are shop type, location, average spend, and user ratings (include service, product, environment, number of positive comments). The response variables are mean and standard deviation (std) of the visit duration distribution. Both response variables are independent, so we can simply use two regression models to regress them.

We conduct regression analysis and show the relation between some predictor variables and both response variables, respectively, in Figures 15 and 16. For mean, we can see that there exists a strong linear-log relation (Benoit 2011) between predictors and the response. So we use ordinary least-squares to estimate the unknown parameters. The regression results indicate that R-squared of the model is 0.810. We also utilize fivefold cross validation to evaluate the accuracy of the regression model. The root-mean-squared error is 4.611.

For std, there does not exist an obvious relation from the perspective of all data, but the data show strong cohesion within the same kind of shops. So, for each category, we use a simple linear regression model to regress the response. We use twofold cross validation to evaluate those linear regression models. The average root-mean-squared error is 3.623.

[4] Dianping (https://www.dianping.com/), a popular Chinese group buying website for locally found consumer products and retail services.

[5] AutoNavi (http://www.gaode.com/), a well-known map website in China.

[6] Scrapy (https://scrapy.org/), an open source and collaborative framework for extracting data from websites.

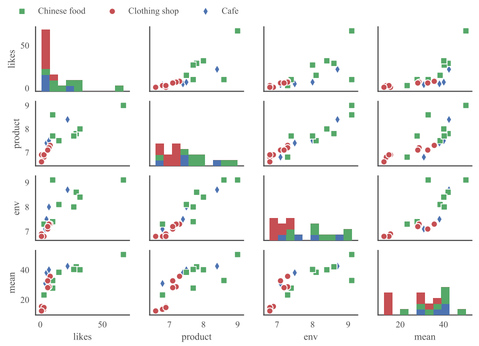


Fig. 15. Regression analysis between some predictor variables and mean of the duration distribution.

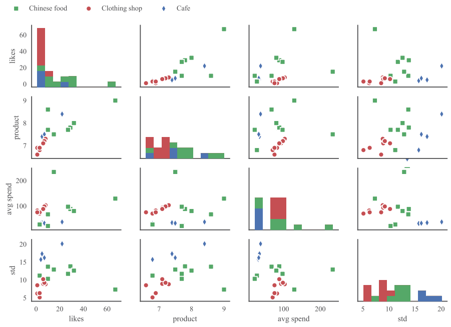


Fig. 16. Regression analysis between some predictor variables and standard deviation of the duration distribution.

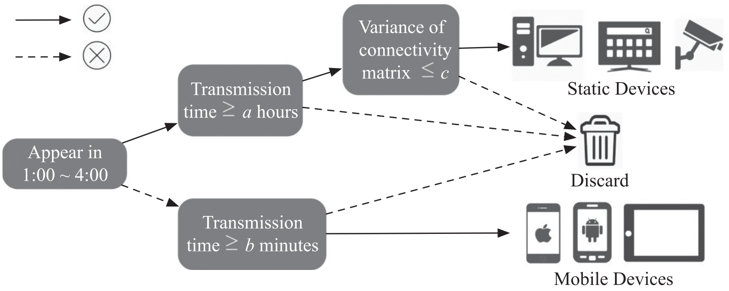

Variance of connectivity matrix $\leq c$
Transmission time $\geq a$ hours
Appear in 1:00 ~ 4:00
Transmission time $\geq b$ minutes
Static Devices
Discard
Mobile Devices

Fig. 17. Decision tree for classifying static and mobile devices.

5 DEVICE CLASSIFICATION

Device classification classifies devices as static devices and mobile devices. Static devices are defined as devices fixed in locations like desktops and IP cameras, while mobile devices could easily change their locations with the help of people, such as smartphones and tablets.

Static devices can be utilized to study the impact of people on wireless coverage status of those fixed locations. The results are demonstrated in Section 3.4. For mobile devices, their Wi-Fi data can be exploited to infer people's locations and mobility patterns.

We propose a decision-tree classifier to classify devices, as illustrated in Figure 17. First, the most distinguishing feature between mobile and static devices is that static devices still work early in the morning, and here we choose 1:00~4:00 AM. Figure 18 visualizes the Wi-Fi data of a static device on June 5, 2015.

Then, for mobile devices, we filter out those devices that may come from passers-by using a threshold of b minutes. For static devices, we use a threshold a to filter out devices with short transmission time. In addition, we check the mobility to remove static devices whose locations changed over time. The mobility can also help to remove mobile devices that are left by some shop owners unintentionally.

To check the mobility, we calculate a variance γ of a connectivity matrix; if the variance exceeds a threshold, then it should not be a static device. Connectivity matrix M_j is shown in Equation (7).

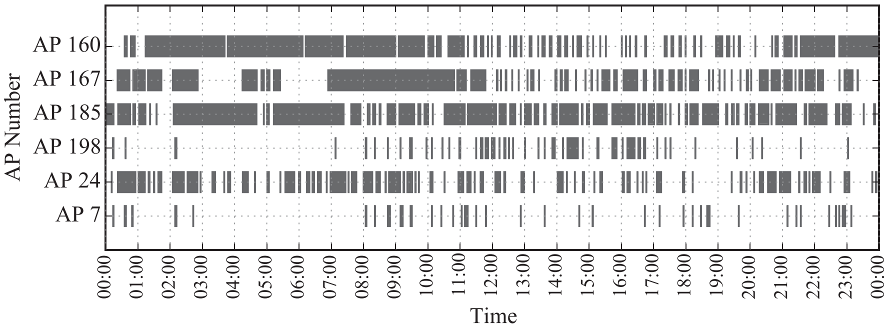


Fig. 18. Raw data of a static device on 5 June 2015.

γ is calculated using Equation (8), where $var(X)$ calculates the statistical variance,

$$M_j = [V_j(1) \quad V_j(2) \quad \cdots \quad V_j(k)] = \begin{bmatrix} v_{11} & v_{21} & v_{31} & \dots & v_{k1} \\ v_{12} & v_{22} & v_{32} & \dots & v_{k2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ v_{1|\mathcal{A}|} & v_{2|\mathcal{A}|} & v_{3|\mathcal{A}|} & \dots & v_{k|\mathcal{A}|} \end{bmatrix}, \tag{7}$$

$$\gamma = \frac{\sum_{i=1}^{|\mathcal{A}|} var([v_{1i} \quad v_{2i} \quad \cdots \quad v_{ki}])}{|\mathcal{A}|}, \gamma \in [0, 0.25]. \tag{8}$$

6 AREA LOCALIZATION AND DENSITY CALCULATION

Area localization and density calculation are the core component in DMAD. In this section, we elaborate on our proposed solutions and demonstrate that although the connectivity information is coarse-grained for fine-grained localization, it is still feasible to locate users to grid-level locations.

We first separate the floor plan into 60 non-overlapping areas (or grids, denoted as $\mathcal{G} = \{g_1, g_2, \cdots\}$) manually. Most of the grids contain one or more shops, and a few of them contain only common areas. Then, based on users' Wi-Fi data, we are able to derive their grid locations. We also have two observations of heuristics that can be utilized to improve the accuracy of area localization. First, different shops attract different numbers of people. Besides, the visit duration in various types of shops differs.

6.1 Area Localization

Wi-Fi-based indoor localization has been extensively studied over the past few decades (Musa and Eriksson 2012; Shen et al. 2016; Bahl and Padmanabhan 2000; Wang et al. 2014). The output of those systems can be classified into geometric locations (represented in coordinates) and semantic locations. Our problem belongs to the latter category, and we find two existing methods that can be used to solve this problem.

The first method is the centroid method (Bulusu et al. 2001), the main idea of which is quite simple. Given a connectivity vector $V_j(i) = [\sigma_1 \quad \sigma_2 \quad \dots \quad \sigma_{|\mathcal{A}|}]^T$, the estimated location $\hat{\mathcal{L}}_j(i)$ can be calculated using Equations (9)–(11), where $\Phi = [\begin{smallmatrix} x_1 & x_2 & \cdots & x_{|\mathcal{A}|} \\ y_1 & y_2 & \dots & y_{|\mathcal{A}|} \end{smallmatrix}]$ is the coordinate vector of all APs. Based on $(\hat{x}, \hat{y})$, the grid location can be determined with ease. However, this method works well only if the density of APs is high enough (zeng Wang and Jin 2009); it may work poorly in our scenario due to low AP density and multiple floors,

$$\hat{\mathcal{L}}_j(i) = \begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix} = \frac{1}{sum\left(V_j(i)\right)} \Phi \cdot V_j(i), \tag{9}$$

$$\hat{x} = \frac{1}{\sum_{k=1}^{|\mathcal{A}|} \sigma_k} \sum_{t=1}^{|\mathcal{A}|} x_t \cdot \sigma_t, \tag{10}$$

$$\hat{y} = \frac{1}{\sum_{k=1}^{|\mathcal{A}|} \sigma_k} \sum_{t=1}^{|\mathcal{A}|} y_t \cdot \sigma_t. \tag{11}$$

Another method is the fingerprinting method (Bahl and Padmanabhan 2000; Wu et al. 2004; Youssef and Agrawala 2005; Shen et al. 2016), which consists of a training phase and a testing phase. The training phase is to construct a fingerprint database that requires a simple site survey to collect the connectivity information of APs in all grids. In the testing phase, given a measured connectivity vector, we compare the vector with that of all grids in the database and use the best match as the estimated user location.

However, the RF signal is vulnerable to environmental disturbances and varies over time, which degrades the performances of deterministic fingerprinting approaches. Some researchers proposed a probabilistic fingerprinting method (Youssef and Agrawala 2005), which is based on statistical inference between the reported signal information and stored fingerprints. Specifically, given a measured connectivity vector V_m, the objective is to find a grid g ($g \in \mathcal{G}$) that maximizes the posterior probability, that is, $\arg\max_g P(g|V_m)$. Traditional probabilistic fingerprinting calculates $P(G|V_m)$ (G is a variable representing all g) using Equation (12),

$$P(G|V_m) = \frac{P(V_m|G) \cdot P(G)}{P(V_m)}. \tag{12}$$

In most cases, previous works regard $P(G)$ as uniform distribution, that is, $P(G) = 1/|\mathcal{G}|$. However, it is not the case in real scenarios. We observe that various shops have different popularities, and the number of customers they attract thus differs. In a similar way, different grids have different attractiveness, and therefore $P(G)$ should differ from grid to grid and time to time. Here we use the number of people in shops to model the popularities of each grid and derive a more practical and accurate estimation of $P(G)$.

We also notice that the length of visit to different types of shops differs. Therefore, besides the Wi-Fi signal, we also exploit the visit duration to distinguish different grids. Equation (13) shows how we calculate the probability of people in all grids, where G is a variable for different grids in $\mathcal{G}$, T is duration time, and W is the measured Wi-Fi data during the period of T. To calculate $P(G|WT)$, we need to know $P(G)$, $P(T|G)$, and $P(W|G)$, which are described in Sections 6.1.1–6.1.3.

$$\begin{aligned} P(G|WT) &= \frac{P(WT|G) \cdot P(G)}{P(WT)} \\ &= \frac{P(W|G) \cdot P(T|G) \cdot P(G)}{P(WT)} \propto P(W|G) \cdot P(T|G) \cdot P(G). \end{aligned} \tag{13}$$

6.1.1 $P(G)$.. $P(G)$ is the probability that people appear in a specific grid. As grids are closely relate to shops, and we use the matrix $R = (n_{ij})$ (in Section 4.2.1) to derive the *a priori* probability. The probability people appear in grid g_j is calculated using Equations (14) and (15). $P(s_j, t_i)$ is the probability that people appear in shop s_j during time slot t_i. $P(g_j, t_i)$ represents the probability people appear in g_j during t_i. $\mathbb{S}_j$ is a set of shops that are in the range of g_j, and $\mathbb{N}_j$ represents a set of grids that are neighboring to g_j. If there are no shops in grid g_l, then we use the average probability from all neighboring grids $\mathbb{N}_l$ of g_l as alternative,

$$P(s_j, t_i) = \frac{n_{ij}}{\sum_{k=1}^{|S|} n_{ik}}, \tag{14}$$

$$P(g_j, t_i) = \begin{cases} \sum_{s_k \in \mathbb{S}_j} P(s_k, t_i), & if\, |\mathbb{S}_j| \neq 0 \\ \sum_{g_l \in \mathbb{N}_j} P(g_l, t_i)/|\mathbb{N}_j|, & if\, |\mathbb{S}_j| = 0 \end{cases}. \tag{15}$$

6.1.2 $P(T|G)$.. $P(T|G)$ is the probability of how long people will stay in a given grid. Similarly to using the number of shops to estimate the number of grids, we calculate the distribution (μ_g and σ_g^2) of the duration time for a grid using Equations (16) and (17). $\mathbb{S}$ is a set of shops that are in the range of grid g. If $|\mathbb{S}_j| = 0$, which means there is no shops in g_j, then the distribution of such grids are collected manually,

$$\mu_g = \frac{1}{|\mathbb{S}|} \cdot \sum_{s_k \in \mathbb{S}} \mu_s, \quad if\, |\mathbb{S}| \neq 0, \tag{16}$$

$$\sigma_g^2 = \frac{1}{|\mathbb{S}|^2} \cdot \sum_{s_k \in \mathbb{S}} \sigma_s^2, \quad if\, |\mathbb{S}| \neq 0. \tag{17}$$

We also have two methods to find the duration time of a user in different grids. The most direct way is to exploit traditional area localization methods to map the Wi-Fi data to grid locations and then based on the locations to derive the duration time. The detailed process is illustrated in Algorithm 2.

But this method performs poorly, since it relies on existing fingerprinting methods that cannot achieve adequate accuracy. Also, the two parameters are hard to tune.

ALGORITHM 2: A Sliding Window Approach on M_j.

```
Data: Wi-Fi data of user j, $M_j = [V_j(1) \quad \cdots \quad V_j(k)]$
Result: A set of subsets
Determine the length $T_w$ of the sliding window, a threshold $\lambda_w$;
for $i \in range(1, k, T_w)$ do
    for $V_j \in [V_i \quad \cdots \quad V_{i+T_w-1}]$ do
        Estimate $\hat{g}_j$ based on $V_j$, using $P(G|V_m)$ ;
    end
    Calculate the percentage of $\hat{g}_j$ among all estimated $\hat{g}$;
    if the percentage of $\hat{g}_j \geq \lambda_w$ then
        The grid of all this window is $\hat{g}_j$;
    end
end
Merge the neighboring windows with same grid information as a subset;
```

Another method is to apply subsequence time series clustering techniques. Subsequence clustering is performed on a single time series to group interesting subsequence time series data in the same cluster (Chen 2005). There are also several methods to solve the subsequence clustering problem, like hierarchical clustering, partitioning clustering, density-based clustering, and and so on. Different methods have different advantages and disadvantages, and a detailed explanation can be found in Zolhavarieh et al. (2014).

Here we choose hierarchical clustering, and one of the reasons is its generality, since it does not require any parameters, such as the number of clusters. The procedure of the algorithm is shown in Algorithm 3.

ALGORITHM 3: Hierarchical Clustering on M_j.

```
Data: Wi-Fi data of user $j$, $M_j = [V_j(1) \quad \cdots \quad V_j(K)]$
Result: Clusters, $\mathbf{C}$
Calculate the Euclidean distance matrix $M_D$ of $M_j$;
while not every $V_j(l)$ in clusters do
    Find two $\mathbf{C}_i$ or $V_j(l)$ with minimum Euclidean distance;
    Merge the two $\mathbf{C}_i$ or $V_j(l)$ to produce a new cluster;
    Update $M_D$ by calculating distances between new cluster and other clusters;
end
```

6.1.3 $P(W|G)$.. W is a set of connectivity vectors of a device d_p, $W = \{V_p(1), \ldots, V_p(K)\}$, where K is the size of the cluster. Given a connectivity vector $V_p(i)$, the probability that it is within g_j can be calculated using Equation (18). $\mathbb{V}_j$ is a set of connectivity vectors (also called fingerprints) collected in g_j and $\mathbb{V}_j(l)$ means the lth vector. $||V_p(i) - \mathbb{V}_j(l)||/|\mathcal{A}|$ calculates the normalized Euclidean distance between $V_p(i)$ and the lth fingerprints in grid g_j. We use average probability of all connectivity vectors in W to represent $P(W|G)$ in Equation (19),

$$P(V_p(i)|G = g_j) = 1 - \frac{1}{|\mathbb{V}_j|} \sum_{l=1}^{|\mathbb{V}_j|} \frac{||V_p(i) - \mathbb{V}_j(l)||}{|\mathcal{A}|}, \tag{18}$$

$$P(W|G = g_j) = \frac{1}{k} \sum_{i=1}^{k} P(V_i|G = g_j). \tag{19}$$

6.2 Density Calculation

Density calculation is quite simple compared to area localization. Based on the grid information derived from area localization, for each time slot and each grid, we count the number of people in that grid as density information (H), which is represented in Equation (20). η_{ij} is the number of people that are within grid g_j during time slot t_i. $\omega_j(i)$ represents the density of g_j during t_i and can be calculated using Equation (21). $|\mathcal{T}|$ is the number of time slots, and $|\mathcal{G}|$ is the number of grids,

$$H = \begin{bmatrix} \eta_{11} & \eta_{12} & \eta_{13} & \cdots & \eta_{1|\mathcal{G}|} \\ \eta_{21} & \eta_{22} & \eta_{23} & \cdots & \eta_{2|\mathcal{G}|} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \eta_{|\mathcal{T}|1} & \eta_{|\mathcal{T}|2} & \eta_{|\mathcal{T}|3} & \cdots & \eta_{|\mathcal{T}||\mathcal{G}|} \end{bmatrix}, \tag{20}$$

$$\omega_j(i) = \frac{\eta_{ij}}{\sum_{i=1}^{|\mathcal{T}|} \eta_{ij}}. \tag{21}$$

7 DEAD SPOTS ESTIMATION

After area localization, each connectivity vector $V_j(i)$ is associated with grid information. We separate all connectivity matrices according to time slot and grid. Then, for each grid and time slot, there is a set of connectivity matrices. Given this information, a key issue here is how to translate those connectivity matrices into the probability of dead spots, which will be introduced in this section, and how to quantify their severity.

7.1 Translating Connectivity Matrices into Probability of Dead Spots

In Section 3.2, we have proposed Equation (3) to transform a connectivity matrix into probability of dead spots $P_{DS}(M_j)$. However, the procedure just converts one connectivity matrix to the probability of dead spots. The question then is: How do we handle multiple connectivity matrices? We believe that devices with larger transmission time are more reliable for estimating dead spots. In extreme cases, when the transmission time of a device is very short, its coverage ratio could be highly biased. A reasonable explanation is that larger transmission time corresponds to larger sampling sizes and thus is more reliable.

Given all connectivity matrices $\{M_1(i), \ldots, M_K(i)\}$ during time slot t_i in grid g_j, we calculate $\tau_j(i)$ (the probability of dead spots in g_j during t_i) using Equations (22) and (23), where $\hat{T}_t(k)$ is the estimated transmission time of M_k,

$$\tau_j(i) = \sum_{k=1}^{K} w_k \cdot P_{DS}(M_k), \tag{22}$$

$$w_k = \frac{\hat{T}_t(k)}{\sum_{p=1}^{K} \hat{T}_t(p)}. \tag{23}$$

7.2 Severity of Dead Spots

Different dead spots have different severity, and if the authorities of the facility want to fix some of them, then they must want to start with the most critical ones. Obviously, the higher the probability of the dead spots, the more severe it is. If the probability of two locations is the same, then what matters is the number of people. Therefore, the severity of a dead spot is not only related to its possibility but also closely associated with the number of potential users around that dead spot.

Here we combine the probability of dead spots $\tau_j(i)$ and human density $\omega_j(i)$ to derive severity of dead spots. β is the significance factor for human density,

$$\lambda_i^j = \beta \cdot \omega_j(i) + (1 - \beta) \cdot \tau_j(i). \tag{24}$$

8 EXPERIMENTS AND RESULTS

In this section, we first introduce the experimental setup and then present the evaluation of each component. Specifically, we carefully study the performance of device classification, area localization, and dead spot estimation. For each of them, we introduce evaluation metrics, baseline approaches, parameter selection, final results, and further discussions if any.

8.1 Experimental Setup

We carry out experiments in a large shopping mall in Shenzhen with five floors (ground and the first to fourth floors) and a total area of over 30,000 m^2. There are 68 shops and 48 APs in the mall, and the floor plan and AP deployment of the ground floor is shown in Figure 9. We manually separate the mall into 60 grids, and most of the grids contain at least one shop, few grids contain only common areas. The partition on the ground floor is shown in Figure 10. There are a few shops, like $B116$ on the floor plan, that are not within the expected coverage areas, so we do not take them into consideration. During the period of 46 days, we collect $|\mathbf{D_w}| = 8,268,462$ Wi-Fi data entries from 726,920 devices.

Table 4. Details of the Testing Data

Issue	Data from task	Data format	# of data
Device classification	I, II, III	<MAC, mobile/static>	249
Area localization	II	<MAC, visit record>	456
Dead spots estimation	III	$< g_j$, t_i, Boolean-DS>	5,127

Table 5. Confusion Matrix of Device Classification

		Predicted condition		
		Static	Mobile	Others
True condition	Static	N_{11}	N_{12}	N_{13}
	Mobile	N_{21}	N_{22}	N_{23}
	Others	N_{31}	N_{32}	N_{33}

To evaluate the performances of different components of DMAD, we engage over 20 volunteers to collect testing data for a period of 1 week. Below shows the tasks that are conducted by volunteers. Table 4 lists detailed information of the testing data for different issues.

I Put some smartphones, including both iOS and Android devices, which keep broadcasting Wi-Fi packets, in some predefined locations, like counter desks and store rooms, for a whole day.
II Do window shopping as usual without preassigned destinations. Record their visiting histories, including the visited grid, start time, and visit duration.
III Conduct simplified site survey with smartphones. Check if there any dead spots in a specific grid during a specific time slot.

As for the simplified site survey, it is conducted using smartphones rather than spectrum analyzers. To detect whether a grid g_j has dead spots or not during time slot t_i, we ask volunteers to go and test every feasible points within g_i. The granularity of test points is about 4m. Generally, there are around 25 points in a grid. For each point, volunteers are required to go there and turn on their Wi-Fi and check the AP list. If the target SSID (“Intown_Free_Wi-Fi”) is not in the list, or the network cannot be associated, then that test point is a dead spot. Usually, it takes 20~30s to finish testing one point.

8.2 Evaluation of Device Classification

Device classification classifies a smartphone as a static device or a mobile device using a decision tree classifier. Here we study the evaluation metric, discuss the selection of system parameters, and show the final results.

8.2.1 Evaluation Metric.. Since this is a classification problem, we use *precision* and *recall* to evaluate the performance, where $precision_i = N_{ii} / \sum_j N_{ji}$, $recall_i = N_{ii} / \sum_j N_{ij}$, and N is a confusion matrix as explained in Table 5.

8.2.2 Parameter Selection.. In this component, we have three parameters, a threshold of transmission time a for static devices, a threshold of transmission time for mobile devices b, and a threshold for variance of connectivity matrix c.

The precision and recall of device classification are not sensitive to parameters a and b. But different values of a and b can affect of the number of static and mobile devices that we can derive

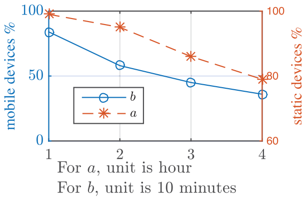


Fig. 19. Impact of different a and b on the percentages of static and mobile devices from D_w.

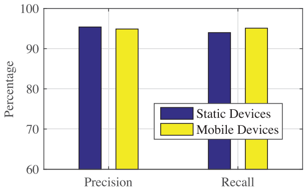


Fig. 20. Precision and recall of static device classification and mobile device classification.

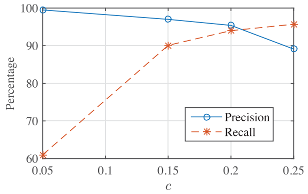


Fig. 21. Precision and recall of static device classification with different c.

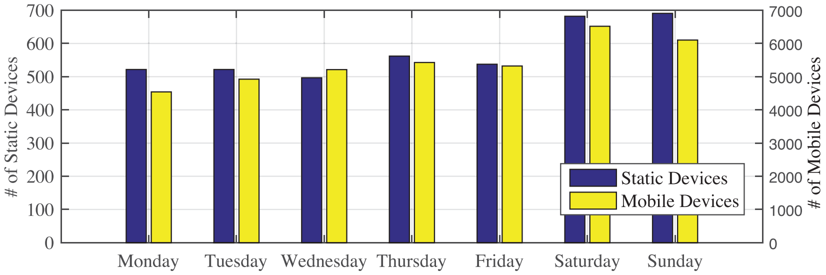


Fig. 22. Average number of static and mobile devices of each day in a week.

from D_w. Figure 19 shows the percentages of static and mobile devices under different value of a and b.

We set $a = 2$, which means static device should send packets for at least 2 hours. Since if a is too small, we cannot calculate the change of coverage ratio. When a is too large, we may miss many static devices.

For mobile devices, we set $b = 10$. If b is too large, then it may miss a large number of mobile users. On the contrary, if b is too small, those devices with small transmission time may have a side effect on DMAD, as their data may be collected from passers-by, which could be highly biased.

For c, we set it to 0.2, which is derived from $\mathbf{D}_w^*$, since if c is too large, then it cannot restrict the mobility of static devices, while if c is too small, then it cannot tolerate errors.

8.2.3 Results.. Figure 20 shows the precision and recall of device classification for both static devices and mobile devices. Also, we study the impact of different c on static devices as illustrated in Figure 21. The precision slightly reduces when c increase.

The results of device classification over $\mathbf{D}_w$ indicate that among 726, 920 devices, the majority of them (83.1%) are from passers-by of the mall, while 14.98% of them are mobile devices and only (1.92%) of them are static devices.

Figure 22 shows the average number of static and mobile devices of each day in a week. Interestingly, most the of days, the number of mobile devices is around 10 times larger than that of static devices. During weekends, both static devices and mobile devices increase, since more people go shopping in holidays. Also from Figure 23, we can see that during dinner time (12:00 and 18:00) the number of people peaks.

We also analyze the duration time of non-static devices of each day in a week, and the results are shown in Figure 24. From the results, we can find that the majority (80%) of people stay in the

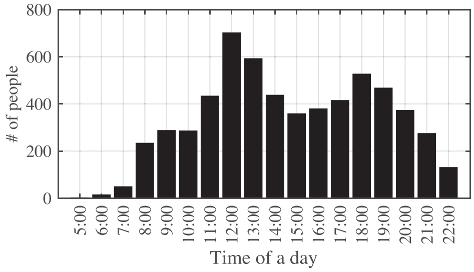


Fig. 23. Average number of people appear in different hours of a day.

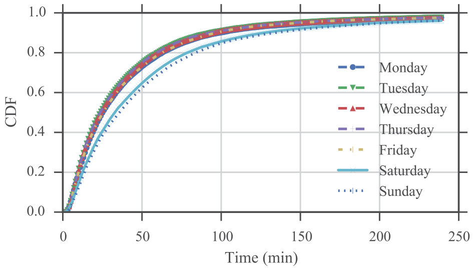


Fig. 24. Comparison of CDFs of total duration time of non-static devices in each day.

shopping mall for less than 1 hour during weekdays, while, during weekends, people stay there for longer times.

8.3 Evaluation of Area Localization

Area localization is to determine users' grid locations according to their connectivity information. We look into evaluation metric and baseline approaches of area localization as well as the performance of grid localization and floor localization. Floor localization is to determine users' floor information, which is more coarse-grained than grid information.

8.3.1 Evaluation Metric.. Area localization is essentially a classification problem, where each grid can be regarded as a class. So we use $accuracy = N_c/N_t$ to measure the performance of floor localization and area localization. N_c is the number of correctly estimated test cases, while N_t is the total number of test cases.

To have a comprehensive understanding of different methods, we also evaluate the $accuracy = N_c^k/N_t$ of top-k results for some methods, where N_c^k is the number of test cases that the top k estimated results cover the true results.

8.3.2 Baselines.. The baseline approaches used for area localization are the centroid method and the fingerprinting method.

The centroid method is denoted as "CEN," and for this method we need to transform the estimated location $\hat{\mathcal{L}}$ to estimated grid $\hat{g}$ by returning the grid to which $\hat{\mathcal{L}}$ belongs. It also happens when $\hat{\mathcal{L}}$ are calculated from multiple APs from different floors. In this case, we determine the floor information by using the closest floor that $\hat{\mathcal{L}}$ is close to.

Another baseline series is probabilistic fingerprinting methods. We denote the traditional method without $P(G)$ and $P(T|G)$ as "FIN." "FIN-G" is a method considering non-uniformed $P(G)$, and "FIN-GT" is our proposed method in MDAD that considers both non-uniformed $P(G)$ (shop popularity) and $P(T|G)$ (visit duration).

8.3.3 Results.. The evaluation results of localization are shown in Figure 25 and Figure 26, which indicate that our proposed approaches ("FIN-G" and "FIN-GT") outperform the centroid method and conventional fingerprinting method by over 10%. The potential reasons are that for the centroid method, it works well when the AP deployment density is high, but the requirement can hardly be satisfied in real scenarios. Also, for conventional fingerprinting methods, due to similar fingerprints in different grids and the vulnerability of the wireless signal, coarse-grained wireless fingerprints alone cannot achieve high localization accuracy.

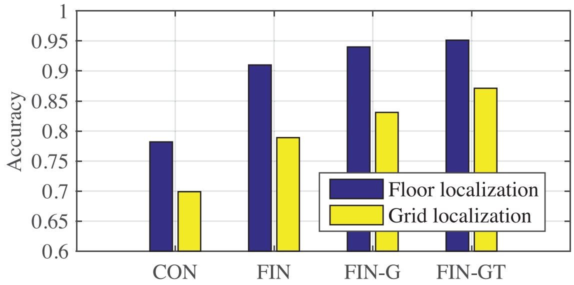


Fig. 25. Accuracy of floor localization and grid localization for different methods.

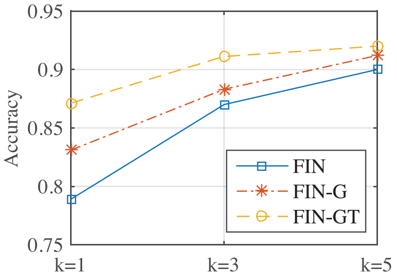
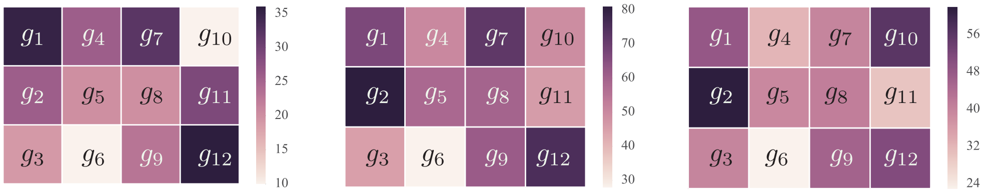


Fig. 26. The impact of different k on the accuracy.



(a) (b) (c)

Fig. 27. Heat map of human density on the ground floor in different time slots. (a) 9:00; (b) 12:00; (c) 18:00.

"FIN-G" and "FIN-GT" utilize a more realistic *a priori* probability of people appearing in different grids. Besides, "FIN-GT" exploits an additional feature of visit duration to separate grids with similar wireless fingerprints.

The human density of the ground floor at different time slots in a day is visualized in Figure 27. We can see that, at the different times in a day, different grids have varied popularity. For example, g_{10} is a supermarket, which has more customers at night than in the mooring. But, compared to other grids, some grids like g_6, which is a common area, have a small group of people all the time. Generally, we find the following rules from the human density data.

—Grids that are close to entrances or exits are likely to have more people.
—The number of people in grids that contain restaurants peaks at dinner time, that is, 12:00 and 18:00.

8.3.4 Further Discussion.. Since different mobile devices may have different transmitting powers, DMAD collects the fingerprints in all grids using devices from different manufacturers. Here we discuss the impact of fingerprints from devices of different manufacturers on the accuracy of area localization. Figure 28 shows the distribution of collected fingerprints and the number of devices used to collect fingerprints.

Figure 29 shows the localization accuracy of using different kinds of devices for training and testing. We can see that using the same kinds of devices for training and testing can achieve better performance, because different kinds of devices generate different fingerprints. The results are from part of the grids, as we do not have fingerprints of all three devices in all grids.

Among cases where the same kinds of devices must be used for training and testing, Apple devices outperform other devices. One of the reasons is that we have only iPhone5s and iPhone6, and the fingerprints collected from both models are quite similar. For Huawei, we have four models (Mate2, Mate7, P7, and P8). The differences between fingerprints are larger than that of Apple

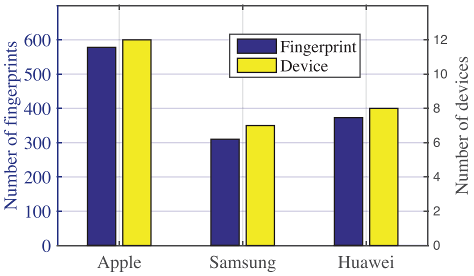


Fig. 28. Distribution of fingerprints and devices used in collecting fingerprints. We use devices from three manufacturers to collect fingerprints for area localization.

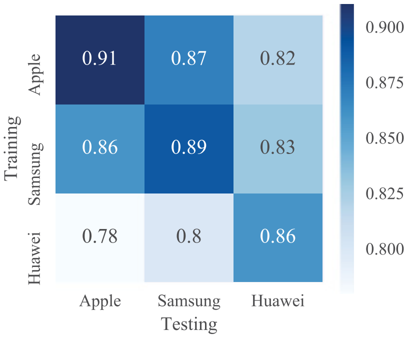


Fig. 29. Confusion matrix of localization accuracy using different kinds of devices for training and testing.

devices. This indicates that if we use more devices to collect the fingerprints, then we can achieve higher localization accuracy. However, collecting so many fingerprints is too time consuming to implement. So there is a tradeoff between accuracy and simplicity of the system.

8.4 Evaluation of Dead Spot Estimation

Dead spot estimation is to estimate the probability of dead spots at a specific grid during a period of time. We study the evaluation metric, parameter selection, and the final results of this component.

8.4.1 Evaluation Metric.. Estimation of dead spots is a binary classification problem, so we use $precision$, $recall$, and F_{score} to evaluate its performance. $precision = tp/(tp + fp)$, $recall = tp/(tp + fn)$, and $F_{score} = 2 \cdot precision \cdot recall/(precision + recall)$. tp are cases where dead spots are predicted as dead spots; tn are cases where there are no dead spots and they are predicted as no dead spots; fp are cases that are predicted to be dead spots, but there are none; and fn are cases where there are dead spots but they are predicted as no dead spots.

8.4.2 Parameter Selection.. We have two parameters in this component: $\psi(n)$ is a decay function and β is the significance factor for human density. $\psi(n)$ models the probability of a dead spot when the location is covered by n APs.

Here we choose an exponential decay function $\psi(n) = 1/(2^n)$, since the best performance of an exponential decay function is better than that of a linear decay function $\psi = 1/(2 * n)$, as shown in Figure 33.

We set $\beta = 0.5$, which means we regard the human density and the probability of dead spots as equally important. β has nothing to do with the accuracy of dead spot estimation, it serves as an importance factor of human density when calculating the severity of a dead spot. If the administrator thinks the number of potential users should be the focus, then β can be set to a larger value.

Besides, we also need a threshold e to determine the existence of dead spots if $P_{DS} \geq e$. We set $e = 0.3$, since the performance peaks under this value.

8.4.3 Results.. The results of dead spot estimation are shown in Figure 30, which demonstrate that when $e = 0.3$, DMAD can identify around 70% of dead spots with a precision of over 70%. Also, Figure 31 shows the CDF of P_{DS} of grids in different floors in all time slots. From the results,

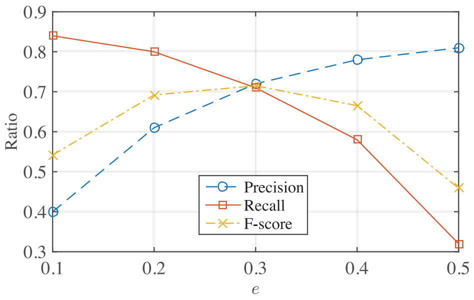


Fig. 30. Precision, recall, and F-score of dead spots estimation under different e.

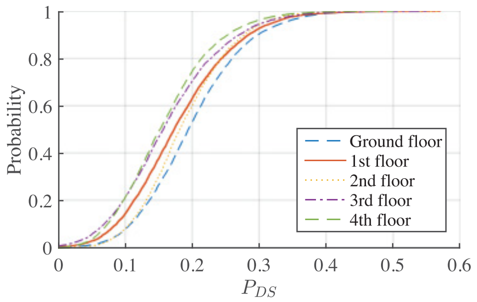
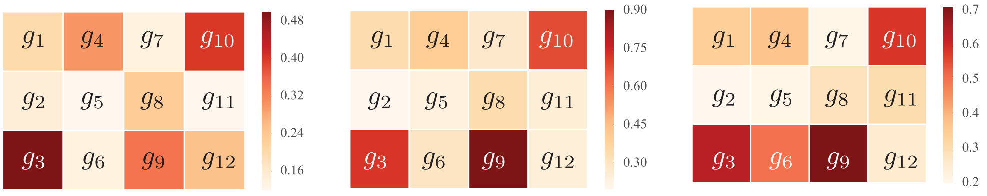


Fig. 31. CDF of P_{DS} of grids from different floors.



(a) **(b)** **(c)**

Fig. 32. Heat map of severity of grids on the ground floor in different time slots. (a) 9:00; (b) 12:00; (c) 18:00.

the lower the floor is, the more dead spots it has. One possible explanation is that more people appear in the lower floors and cause more dead spots.

We also derive the normalized severity $\lambda_i^j/max(\lambda_i^j)$ of different grids during different time slots. Figure 32 shows the severity of grids on the ground floor in different time slots. We can find that some grids, like g_3, g_9, and g_{10}, are more severe over time, which deserves more attention.

8.4.4 Further Discussion.. Since the number of people on weekends is obviously larger than on weekdays, here we compare dead spots on weekdays and weekends. First, we calculate average performance during weekdays and weekends. As shown in Figure 34, the performance during weekends is slightly lower than that of weekdays.

We also count the number of dead spots during a whole day among all grids, and the average number for weekdays is 972. For weekends, it is reported to have 18.8% more dead spots on average. This result is reasonable, since dead spots are closely related to people, whereby more people will result in more dead spotd. Interestingly, as illustrated in Figure 35, we find that the distribution of those additional dead spots obeys a "70-30 rule," which means that 70% of the additional dead spots are generated by 30% of the grids.

9 CONCLUSION

In this article, we propose DMAD, a data-driven measuring of Wi-Fi AP deployment to estimate dead spots and quantify their severity using both Wi-Fi data and shop data.

Based on the collected data, we first classify static devices and mobile devices using a decision-tree classifier. The most distinguishing feature between them is whether they work early in the morning.

Then we locate these devices to shop-level locations based on two observations of heuristics. On the one hand, the duration of visit in different shops differs, for example, people stay longer

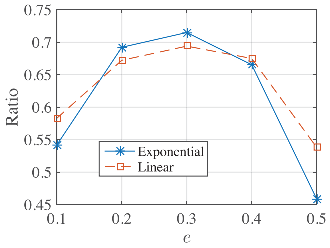


Fig. 33. F-score of linear and exponential decay functions under different *e*.

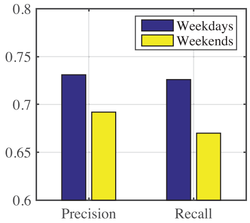


Fig. 34. Average precision and recall of weekdays and weekends.

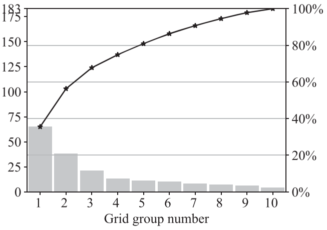


Fig. 35. Pareto chart of additional dead spots. The sorted grids are equally separated into 10 groups.

in restaurants than in clothing shops. On the other hand, different shops have different popularity in attracting customers at different time slots, for example, restaurants attract more people during lunch time than clothing shops. These two features can be exploited to distinguish locations with similar wireless fingerprints.

Last, for each location, we estimate the probability of dead spots in different time slots and derive their severity combining the dead spots probability and human density. Since if a dead spot appears in a place with a lot of potential users, this dead spot must be more severe.

We carefully study the performance of different components of DMAD using real data collected from a large shopping mall. The evaluation results demonstrate that DMAD can identify around 70% of dead spots with a precision over 70%.

REFERENCES

Martin D. Adickes, Richard E. Billo, Bryan A. Norman, Sujata Banerjee, Bartholomew O. Nnaji, and Jayant Rajgopal. 2002. Optimization of indoor wireless communication network layouts. *IIE Trans.* 34, 9 (2002), 823–836.

Paramvir Bahl and Venkata N. Padmanabhan. 2000. RADAR: An in-building rf-based user location and tracking system. In *Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'00)*, Vol. 2. IEEE, 775–784.

Kenneth Benoit. 2011. Linear regression models with logarithmic transformations. *London School of Economics, London* 22, 1 (2011), 23–36.

Nirupama Bulusu, John Heidemann, and Deborah Estrin. 2001. Adaptive beacon placement. In *Proceedings of the 21st International Conference on Distributed Computing Systems 2001*. IEEE, 489–498.

Eyuphan Bulut and Boleslaw K. Szymanski. 2013. WiFi access point deployment for efficient mobile data offloading. *ACM SIGMOBILE Mobile Comput. Commun. Rev.* 17, 1 (2013), 71–78.

Jason R. Chen. 2005. Making subsequence time series clustering meaningful. In *Proceings of the 5th IEEE International Conference on Data Mining (ICDM'05)*. IEEE.

Qiuyun Chen, Bang Wang, Xianjun Deng, Yijun Mo, and Laurence T. Yang. 2013. Placement of access points for indoor wireless coverage and fingerprint-based localization. In *Proceedings of the 2013 IEEE 10th International Conference on High Performance Computing and Communications and the 2013 IEEE International Conference on Embedded and Ubiquitous Computing (HPCC_EUC'13)*. IEEE, 2253–2257.

Manuel Crotti, Maurizio Dusi, Francesco Gringoli, and Luca Salgarelli. 2007. Traffic classification through simple statistical fingerprinting. *ACM SIGCOMM Comput. Commun. Rev.* 37, 1 (2007), 5–16.

Steven J. Fortune, David M. Gay, Brian W. Kernighan, Orlando Landron, Reinaldo A. Valenzuela, and Margaret H. Wright. 1995. WISE design of indoor wireless systems: Practical computation and optimization. *IEEE Comput. Sci. Eng.* 2, 1 (1995), 58–68.

Julien Freudiger. 2015. How talkative is your mobile device?: An experimental study of wi-fi probe requests. In *Proceedings of the 8th ACM Conference on Security & Privacy in Wireless and Mobile Networks*. ACM, 8.

Homayoun Hashemi. 1993. The indoor radio propagation channel. *Proc. IEEE* 81, 7 (1993), 943–968.

JaYeong Kim, Nah-Oak Song, Byoung Hoon Jung, Hansung Leem, and Dan Keun Sung. 2013. Placement of wifi access points for efficient wifi offloading in an overlay network. In *Proceedings of the 2013 IEEE 24th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC'13).* IEEE, 3066–3070.

Shahnaz Kouhbor, Julien Ugon, A. Rubinov, Alex Kruger, and M Mammadov. 2006. Coverage in WLAN with minimum number of access points. In *Proceedings of the 2006 IEEE 63rd Vehicular Technology Conference*, Vol. 3. IEEE, 1166–1170.

Peter Kreuzgruber, Thomas Brundl, Wolfgang Kuran, and Rainer Gahleitner. 1994. Prediction of indoor radio propagation with the ray splitting model including edge diffraction and rough surfaces. In *Proceedings of the 1994 IEEE 44th Vehicular Technology Conference.* IEEE, 878–882.

Merima Kulin, Carolina Fortuna, Eli De Poorter, Dirk Deschrijver, and Ingrid Moerman. 2016. Data-driven design of intelligent wireless networks: An overview and tutorial. *Sensors* 16, 6 (2016), 790.

Lin Liao, Weifeng Chen, Chuanlin Zhang, Lizhuo Zhang, Dong Xuan, and Weijia Jia. 2011. Two birds with one stone: Wireless access point deployment for both coverage and localization. *IEEE Trans. Vehic. Technol.* 60, 5 (2011), 2239–2252.

Tao Liu and Alberto E. Cerpa. 2011. Foresee (4C): Wireless link prediction using link features. In *Proceedings of the 2011 10th International Conference on Information Processing in Sensor Networks (IPSN'11).* IEEE, 294–305.

Tao Liu and Alberto E. Cerpa. 2014. Temporal adaptive link quality prediction with online learning. *ACM Trans. Sensor Netw.* 10, 3 (2014), 46.

Weixiao Meng, Ying He, Zhian Deng, and Cheng Li. 2012. Optimized access points deployment for WLAN indoor positioning system. In *Proceedings of the 2012 IEEE Wireless Communications and Networking Conference (WCNC'12).* IEEE, 2457–2461.

ABM Musa and Jakob Eriksson. 2012. Tracking unmodified smartphones using wi-fi monitors. In *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems.* ACM, 281–294.

Revolutionwifi. 2013. Wi-Fi Site-Surveying 101. Retrieved from *http://www.revolutionwifi.net/revolutionwifi/2013/08/wi-fi-site-surveying-101.html.*

T. Schoberl. 1995. Combined monte carlo simulation and ray tracing method of indoor radio propagation channel. In *Proceedings of the 1995 IEEE MTT-S International Microwave Symposium Digest.* IEEE, 1379–1382.

Souvik Sen, Romit Roy Choudhury, and Srihari Nelakuditi. 2012. SpinLoc: Spin once to know your location. In *Proceedings of the 12th Workshop on Mobile Computing Systems & Applications.* ACM, 12.

Chhavi Sharma, Yew Fai Wong, Wee-Seng Soh, and Wai-Choong Wong. 2010. Access point placement for fingerprint-based localization. In *Proceedings of the 2010 IEEE International Conference on Communication Systems (ICCS'10).* IEEE, 238–243.

Jiaxing Shen, Jiannong Cao, Xuefeng Liu, Jiaqi Wen, and Yuanyi Chen. 2016. Feature-based room-level localization of unmodified smartphones. In *Smart City 360.* Springer, 125–136.

Ivan Vilovic, Niksa Burum, and Zvonimir Sipus. 2009. Ant colony approach in optimization of base station position. In *2009 3rd European Conference on Antennas and Propagation.* IEEE, 2882–2886.

Chen-Shu Wang and Yi-Dung Chen. 2012. Base station deployment with capacity and coverage in WCDMA systems using genetic algorithm at different height. In *Proceedings of the 2012 6th International Conference on Genetic and Evolutionary Computing (ICGEC'12).* IEEE, 546–549.

Chen-Shu Wang and Li-Fang Kao. 2012. The optimal deployment of wi-fi wireless access points using the genetic algorithm. In *Proceedings of the 2012 Sixth International Conference on Genetic and Evolutionary Computing (ICGEC'12).* IEEE, 542–545.

Yan Wang, Jie Yang, Yingying Chen, Hongbo Liu, Marco Gruteser, and Richard P. Martin. 2014. Tracking human queues using single-point signal monitoring. In *Proceedings of the 12th Annual International Conference on Mobile Systems, Applications, and Services.* ACM, 42–54.

Lyndon While and Chris McDonald. 2014. Optimising wi-fi installations using a multi-objective evolutionary algorithm. In *Proceedings of the Asia-Pacific Conference on Simulated Evolution and Learning*. Springer, 747–759.

Chao-Lin Wu, Li-Chen Fu, and Feng-Li Lian. 2004. WLAN location determination in e-home via support vector classification. In *Proceedings of the 2004 IEEE International Conference on Networking, Sensing and Control*, Vol. 2. IEEE, 1026–1031.

Moustafa Youssef and Ashok Agrawala. 2005. The Horus WLAN location determination system. In *Proceedings of the 3rd International Conference on Mobile Systems, Applications, and Services.* ACM, 205–218.

Ji zeng Wang and Hongxu Jin. 2009. Improvement on APIT localization algorithms for wireless sensor networks. In *Proceedings of the International Conference on Networks Security, Wireless Communications and Trusted Computing 2009 (NSWCTC'09)*, Vol. 1. IEEE, 719–723.

Seyedjamal Zolhavarieh, Saeed Aghabozorgi, and Ying Wah Teh. 2014. A review of subsequence time series clustering. *Sci. World J.* 2014 (2014).

Received November 2016; revised February 2017; accepted March 2017