WISDOM: An Efficient Framework of Predicting WLAN Availability with Cellular Fingerprints

Shuai Wang
HP Labs China
Beijing, China
shuai.wang@hp.com

Xiaofeng Yu
HP Labs China
Beijing, China
xiaofeng.yu@hp.com

Junqing Xie
HP Labs China
Beijing, China
jun-qing.xie@hp.com

ABSTRACT
Mobile devices with both WLAN adapter and cellular capability, which are also known as dual-mode mobile terminals, are facing various challenges and problems in conventional WLAN discovery mechanisms, including inefficiency in network discovery, unavoidable energy consumption for frequent WLAN scanning, and privacy information leaking in network probing. In this paper, we propose a novel framework called WISDOM (Wireless Indicator Supervised Data Offloading Manipulation), which can efficiently predict the availability of appropriate WLAN access points (APs) for mobile device without the need of turning on its WLAN adapter in advance. WISDOM takes advantage of historical cellular fingerprints (i.e., the pairs of Cell-ID and Received Signal Strength Indicator) to directly model the WLAN coverage, and perform WLAN availability prediction based on the models given a query cellular fingerprint. Similarity and Classification methods are introduced to work in the framework as prediction methods. We have developed a WISDOM prototype and performed simulation and real field tests under various situations. The results showed WISDOM along with the proposed predication methods could reach at least an average of 80% in accuracy and saving 60% of power consumption on average for mobile devices.

Author Keywords
WLAN Availability Prediction; Cellular Fingerprints; Machine Learning

ACM Classification Keywords
H.4.m. Information Systems Applications: Miscellaneous

INTRODUCTION
With the ever growing amount of mobile applications and mobile data traffic, the importance of WLAN availability could not be overemphasized. Generally speaking, for end users, it would be of great convenience if the most favorite WLAN could be timely and intelligently discovered once it is available and in good quality; for the service providers (SP), they are deploying WLAN hotspots for Mobile Data Offloading (MDO) to automatically offload user data traffic from cellular network to WLAN due to its higher network efficiency, lower cost in spectrum and infrastructure, etc.; for enterprises, WLAN is the major wireless infrastructure for deploying BYOD (Bring Your Own Device) and CYOD (Choose Your Own Device) to serve employees' mobile devices at office. For all the above stakeholders, an efficient WLAN discovery mechanism is of great importance.

For most of the time, mobile devices would run in idle mode (screen off) with WLAN scanning period varies from dozens of seconds to minutes or even more for different device models. Long scanning interval makes it impossible to timely discover the availability of WLAN. However, if forcing the WLAN adapter to perform frequent scanning, it would be a waste of battery power since for a great partial of time WLAN is not available [28]. What makes that even worse is that the frequent WLAN probing frames would also lead to privacy issues [24, 28] such as location or other privacy information leaking to adversary or malicious individuals. There are also occasions that users get all the GPS, WLAN adapters turned off in order to save power [2, 4, 8, 29], which makes it even harder to discover the available WLAN network in an efficient manner. These challenges and problems naturally yield the needs of an ideal WLAN discovery mechanism as: only when the interested WLAN is available, the WLAN adapter is turned on to connect to that WLAN, otherwise it is turned off.

In 4G/LTE network infrastructure, a network element called ANDSF (Access Network Discovery and Selection Function) can be used to guide the mobile device for WLAN discovery. However, the default interface elements defined in 3GPP specification [1] only provide limited location or network information, which couldn't guarantee the accuracy of WLAN availability along with WLAN network condition. The inefficiency of the conventional ANDSF assisted WLAN discovery can be described in Figure 1, where 1) let M, M', M" be different mobile devices that are registered to cellular base station Cellx and wish to discover available WLAN; 2) when GPS and WLAN adapters are all turned off, the network side could only provide the same AP list for M, M' and M" as they have the same registered Cell-ID. However, apparently a better solution shall be: a) for M",

UbiComp '15, September 7–11, 2015, Osaka, Japan.

ACM 978-1-4503-3574-4/15/09…$15.00.
http://dx.doi.org/10.1145/2750858.2804275

$AP_m > AP_n$, i.e., AP_m of higher priority; b) for M': $AP_n > AP_m$; c) for M: not recommended, but could try with $AP_n > AP_m$.

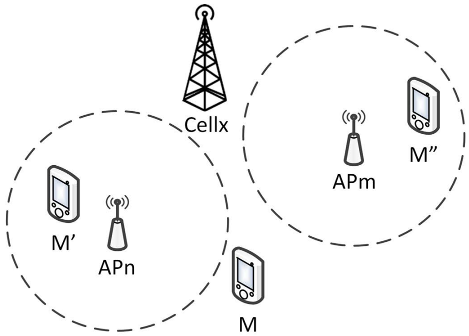

Figure 1. Conventional ANDSF Assisted WLAN Discovery with GPS and WLAN Adapter All Turned Off

Actually, it can be observed that the WLAN coverage is always covered by one or multiple cellular base stations. Moreover, the WLAN coverage area could be further specified by a range of signal strength values of the cellular base stations. To solve the abovementioned problems, we propose a WISDOM (Wireless Indicator Supervised Data Offloading Manipulation) framework on predicting the availability of WLAN. In general, the WISDOM approach performs WLAN coverage modeling with historical cellular fingerprints collected by mobile device during successful WLAN connection, and predicting WLAN availability using the query cellular fingerprints based on the models. We define a *cellular fingerprint* as a set of Cell-ID and Received Signal Strength Indicator (RSSI) pairs collected by the mobile device. For any given time, a cellular fingerprint contains at least one Cell-ID and RSSI pair, which is the registered Cell-ID and its RSSI; it may also contain multiple (usually 0 to 6) neighboring Cell-IDs in the vicinity and their RSSIs. We will also refer a *general fingerprint* to one that contains both the cellular fingerprint and all the available WLAN AP information associated with that cellular fingerprint. The WISDOM framework only tells the availability of user interested WLAN APs (e.g., whether recommended, and the prioritized AP list, etc.), and other work (e.g., authentication, association and other operations on WLAN adapters) will be simply left to implementations of applications.

This paper is organized as follows. Firstly, we review related work and point out their differences and deficiencies in handling the WLAN prediction task. Next we introduce the WISDOM framework and its modeling methods (i.e., Similarity method and Classification method). After that, we introduce the system reference architecture and a WISDOM prototype in different implementation scenarios. Then the performance of WISDOM framework and its modeling methods are evaluated and analyzed. Before conclusion, we make some further discussion on the generalization of WISDOM framework to the complicated conditions under real cases. Our contributions in this paper lie in the following three folds:

- We proposed the WISDOM framework which directly models and predicts WLAN availability using cellular fingerprints. It eliminates the need of any intermediate location mapping and calculation. Moreover, it works in both outdoor and indoor scenarios.
- We investigated two different prediction methods (i.e., Similarity method and Classification method) for WISDOM from three different cases (i.e., Self-Prediction, All-Prediction and Inter-Prediction). These methods have been evaluated on accuracy and run-time performance under different scenarios. The performance results provide guidance for implementation under different system architectures.
- We designed reference system architectures (i.e., Local Device Maintained or LDM architecture, Remote Network Assisted or RNA architecture, and RNA-LDM-Combined architecture) for implementation, built WISDOM prototypes and verified the WISDOM framework working effectively under real cases.

RELATED WORK

To the best of our knowledge, there is no prior technology in modeling WLAN coverage and predicting availability by leveraging cellular fingerprints directly. Izumikawa *et al.* proposed a solution [10] that leveraged the transition of cellular signal quality from outdoor to indoor for indoor WLAN discovery. However, it can only work in the use case as from outdoor to indoor, but can't apply to predict WLAN availability in the outdoor and the complicated indoor cases. Other analogous solutions are mainly focusing on Vertical Handoff (VHO) and cellular RSS based localizations or positioning.

For VHO solutions, Song *et al.* proposed a solution [20] using dynamic interface activating intervals. But these intervals still need to be small enough in order to discover the WLAN availability in a timely manner. Some solutions [9, 14, 21, 30] noticed the pitfalls of continuous scanning or periodical scanning, but they proposed implementations based on or similar to ANDSF thus couldn't guarantee the performance in efficiency and accuracy. Actually, most of the VHO solutions similar to [16, 17] tended to answer the question of handoff to the best wireless network by assuming WLAN information being already available, but not covering when or how WLAN became available. All these VHO related solutions couldn't guarantee the performance in both accuracy and efficiency, and they must also rely on a network assisted manner.

The mobile devices' cellular network RSS (Received Signal Strength) based localization approaches usually include deterministic approaches [6, 12, 13, 15] and probabilistic approaches [3, 5, 6, 15, 19]. Those approaches could be extended to roughly search WLAN APs in vicinity as long as the positions of the APs were known in advance. However, the common drawback of most of those solutions was that they required both the cellular fingerprints and the corresponding GPS information as training data, which had limitations for indoor scenario. For example, Ibrahim *et al.* proposed Cellsense [15] solution which required the GPS information so as to divide the cellular fingerprints into grids for modeling and calculation. However, it needed very rigid data collecting phase (i.e., wardriving) and it was hard to adapt to WLAN or cellular network infrastructure changes. As for modeling, it was also difficult to construct grids for an arbitrary region in an adaptively grained manner since WLAN coverage areas would be irregularly shaped. The major difference between WISDOM and these state-of-the-art solutions is that WISDOM performs WLAN coverage modeling by directly modeling with cellular fingerprints instead of using intermediate locations (or known locations). For other similar solutions, Yacine *et al.* proposed an indoor localization approach [26] based on the use of RSS fingerprints containing data from very large numbers of cellular base stations–up to the entire GSM band of over 500 channels. It utilized machine learning techniques such as Support Vector Machines (SVM) in one-versus-one and one-versus-all configurations for room-level classification. Ye *et al.* proposed a similar indoor localization approach [27] using RSS fingerprints from the GSM network and SVM based classification. Robin *et al.* [18] addressed the RSS based indoor localization problem in a WLAN environment and formulated it as a classification problem using surveyed locations as classes; it presented a discriminatively regularized least square classifier (DRLSC)-based localization algorithm to solve the problem. All the above-mentioned work only handled indoor localization based on either cellular RSS or WLAN RSS. However, WISDOM framework can be well applicable to both indoor and outdoor scenarios.

WISDOM FRAMEWORK

The WISDOM framework takes advantage of the cellular fingerprints for WLAN coverage modeling and availability prediction. Figure 2 illustrates the general concept of WISDOM: 1) assume F_i, F_j, F_k be the current (query) cellular fingerprints collected by different mobile devices; and $F_{m,a}$, $F_{m,b}$, $F_{n,p}$, $F_{n,q}$ represent a number of historic fingerprints modeling the coverage of AP_m and AP_n; 2) even with GPS and WLAN adapter turned off, an efficient prediction can be made as: a) for M'': $AP_m>AP_n$; b) for M': $AP_n>AP_m$; c) for M: not recommended, but could try with $AP_n>AP_m$.

WISDOM framework is mainly composed of (1) *a learning procedure* that collects and updates cellular fingerprints along with AP's information (i.e., SSID, BSSID, RSSI, etc.) under successful WLAN connection, (2) *a modeling procedure* that constructs the models of WLAN coverage (or AP coverage), and (3) *a predicting procedure* that calculates the prioritized AP list as result. The processing procedures of WISDOM framework can be illustrated in the flow charts as Figure 3, and all these procedures could be entirely performed either locally or in a network assisted manner. The general fingerprint F is denoted as:

$$F = \{CellID_r, [<CellID_i, RSS_i>]_{cellular}, [<SSID_j, BSSID_j, RSS_j>]_{AP}\},\ r\in[1, n],\ i=1,..,n,\ j=0,..,m.$$

In F, $CellID_r$ stands for the registered Cell-ID; $[<CellID_i, RSS_i>]_{cellular}$ stands for all the cellular fingerprints including both registered Cell-ID and neighboring Cell-IDs; $[<SSID_j, BSSID_j, RSS_j>]_{AP}$ stands for all the available AP information at current position that includes at least the connected AP and optionally any other available ones. In the predicting procedure, a query fingerprint is a cellular fingerprint containing only $CellID_r$ and $[<CellID_i, RSS_i>]_{cellular}$; while in the learning procedure, the entire F will be used. If a cellular fingerprint has no neighboring Cell-ID available, we call it as a Single-Cell-ID fingerprint; otherwise, we call it as a Multiple-Cell-ID fingerprint.

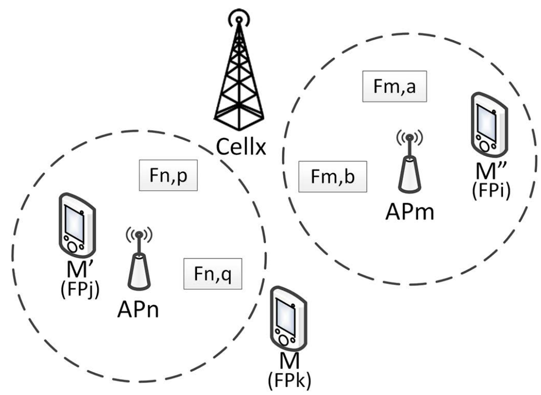

Figure 2. WISDOM WLAN Availability Prediction with GPS and WLAN Adapter All Turned Off

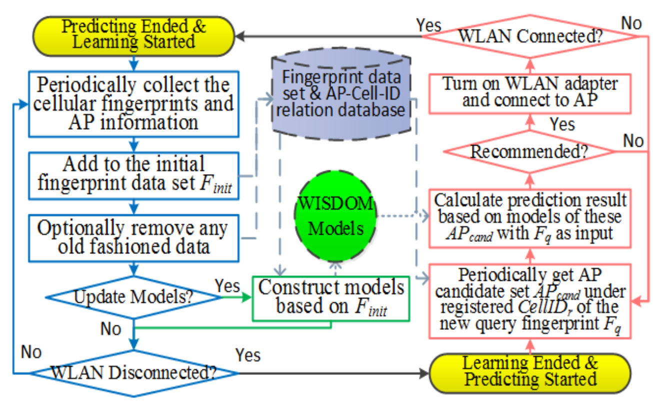

Figure 3. WISDOM Framework General Processing Flows

The WISDOM framework can be applied with different methods for modeling. The following sections introduce the Similarity method and the Classification method that treat WLAN prediction from different perspectives.

Similarity Method

By predicting the mobile device to be within a sub-region of the AP coverage, we can perform WLAN discovery more efficiently, so that when a mobile device is found in a sub-region with relative weak RSS of AP, the WLAN network connection will not be recommended since the network quality will be bad and unstable. By dividing signal strength of an AP into a group of levels, with each level representing certain signal strength range, the AP coverage could be divided into regions in levels. We further observed from field tests that: for outdoor AP, each AP level region can usually geographically be separated based on registered Cell-IDs (as illustrated in Figure 4), which means we can further sub-divide the AP level region into sub-regions according to the registered Cell-ID, so as to achieve a more fine-grained relationship between Cell-ID RSS distribution and the AP coverage. As for indoor AP, we observed many cases where the entire AP level sits within a single registered Cell-ID due to the deployment of indoor Omni-antennas, and the AP region is relatively small enough for modeling. Figure 5 describes how these sub-regions are generated by subdividing each AP RSS level based on registered Cell-IDs.

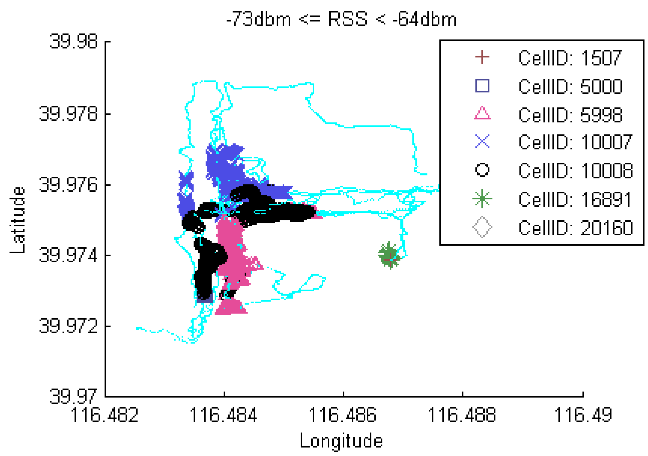

Figure 4. Registered Cell-IDs in AP Level. Lines represent the field test walking paths; bold points show the AP RSS level coverage in range [-73, -64) dBm, with each color or shape standing for a distinct registered Cell-ID.

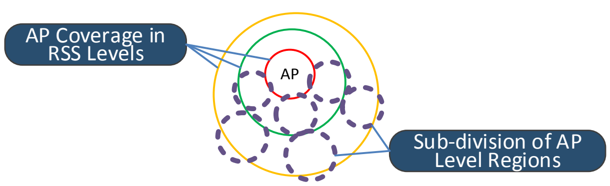

Figure 5. Sub-division of AP Signal Level. AP coverage can be generally divided into levels; dotted circles in each level denote fingerprints of the same registered Cell-ID, and form different sub-regions as modeling units for Similarity method.

The Similarity method collects the cellular fingerprints within the sub-regions of the AP RSS levels to construct the probability mass function (PMF) for every appeared Cell-ID in that sub-region. It then uses the PMFs of these sub-regions to calculate a similarity value for each sub-region based on the given query fingerprint; and takes the max similarity together with related level as the result of the given fingerprint to that AP. After calculation on multiple AP candidates, the final prediction result can be made. The Similarity method generally can be implemented through three algorithms under the WISDOM framework: Algorithm 1 is applied for the modeling procedure, which constructs the probabilistic model per AP in the WLAN for similarity calculation. Algorithm 2 and Algorithm 3 are applied for the predicting procedure.

Algorithm 1: Probabilistic Model Construction per AP

Input: initial AP fingerprint data set F_{init}
1. Define AP RSS range with l levels in step g;
2. Define cellular RSS range with k levels in step h;
3. Group F_{init} to subsets F_l in levels l;
4. For each level l:
   - 4.1. Classify fingerprints in F_l into classes F_s^l based on the registered Cell-ID $CellID_r^l$ of the fingerprint;
   - 4.2. For each class F_s^l of size N:
     - 4.2.1. For each $CellID_t^s$ in F_s^l, calculate PMF_t^s based on occurrence f_k in each level k of all cellular signal strength in set RSS_t^s:

$$PMF_t^s(RSS_t^s) = \begin{cases} \frac{f_k}{N}, f_k > 0 \\ \delta, f_k = 0 \end{cases}$$

Output: PMF_t^s set as model construction result.

Algorithm 2: AP Candidates Calculation

Input: query fingerprint F_q
1. Get AP candidate set AP_{cand} based on the registered Cell-ID of F_q from AP-Cell-ID relation database;
2. For each AP_i in AP_{cand}:
   - 2.1. $<S_i, l_j>$ = **Algorithm 3**(F_q);
3. Sort AP_{cand} according to S_i and l_j to get set $AP_{cand}^{'}$;

Output: $AP_{cand}^{'}$ as final decision;

Algorithm 3: Similarity Calculation per AP

Input: query fingerprint F_q
1. For each subclass s_j of all levels l of AP_i:
   - 1.1. Define the set $CellID_x$, $CellID_y$ and $CellID_z$ as:
     $CellID_x$ = CellIDSet(F_q) ∩ CellIDSet(s_j);
     $CellID_y$ = CellIDSet(s_j) - CellIDSet(F_q);
     $CellID_z$ = CellIDSet(F_q) - CellIDSet(s_j);
   - 1.2. Calculate the similarity D_j of F_q to s_j by cumulating the probability in logarithm for each of all n elements in $CellID_x$ and $CellID_y$ in PMF_t^s, and each of all m elements in $CellID_z$ of probability σ;

$$D_j = \sum_{t=1}^{n} \lg PMF_t^s(RSS_t) + m \cdot \lg \sigma$$

   - 1.3. Use a predetermined minimum probability p_{min} as δ for those PMF_t^s with f_k=0, and for probability σ;
2. Define S_i as the similarity of F_q and AP_i, then;

S_i =min(D_j);

Output: S_i and RSS level l_j;

Classification Method

In addition to Similarity method, we also formulate the WLAN availability prediction problem as a multi-label classification and ranking problem [22][23] from a machine learning perspective. In Classification method, the modeling work is performed based on all the general fingerprints (i.e., cellular fingerprints and corresponding AP information) under the same registered Cell-ID instead of per AP. We take each AP's BSSID as a label that uniquely identifies the AP. And due to signal fluctuation, each cellular fingerprint will be mapped to multiple APs of the general fingerprints for modeling, which means multiple AP labels. To solve the WLAN availability multi-label classification problem, we transform it into one or more single-label classification problems. There are many simple transformations that can be used for the transformation, and we apply the label powerset (LP) method. LP considers each unique set of AP labels that exists in a multi-label general fingerprint training set as one of the AP classes of the new single-label classification problem. Table 1 gives an example of transforming AP labels to AP classes. It can be seen that each cellular fingerprint might have several AP labels (listed in the 2nd column); and these multiple AP labels can be transformed into AP classes (listed in the 3rd column) by treating the entire label set for a cellular fingerprint as a new class. The AP labels AP_1, AP_2, AP_3 and AP_4 have been transformed to AP classes APC1, APC2 and APC3.

FP	Multiple AP Labels	Single AP Class Label
f1	AP_1, AP_2	APC1={AP_1, AP_2}
f2	AP_1, AP_2	APC1={AP_1, AP_2}
f3	AP_1, AP_2, AP_3	APC2={AP_1, AP_2, AP_3}
f4	AP_2, AP_3, AP_4	APC3={AP_2, AP_3, AP_4}
f5	AP_2, AP_3, AP_4	APC3={AP_2, AP_3, AP_4}

Table 1. Sample Cellular Fingerprints, AP Labels and AP Classes. First column FP stands for cellular fingerprint.

Various machine learning techniques could be applied to the transformed single-label classification problem. The one we used is the SVM (Support Vector Machine), which outperforms other single label classifiers during our tests. By taking each Cell-ID in cellular fingerprint as a distinct dimension, each AP class (i.e., a group of APs' coverage areas) could be placed into a multi-dimensional feature space. Figure 6 is an example illustration on relation between two Cell-IDs (X, Y) as the cellular fingerprint vectors and an AP's coverage area as the target class.

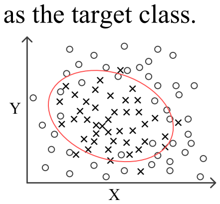

Figure 6. Illustration of AP Coverage. × stands for a cellular fingerprint inside AP coverage; ○ stands for outside. Axis X and Y stand for the signal propagation of two cells. Oval stands for a potential hyper-surface identifying AP coverage.

The edge of AP coverage is generally a hyper surface with some vectors located around the edges which are hard to distinguish whether inside or outside of the AP coverage; but these vectors generally fall within some specific region. The classification task now has been transformed into a non-linear SVM task that is to find the hype-surface for a target class (i.e., the AP coverage, or the AP class) in the multi-dimensional feature space, and return classes with probabilities for a given query cellular fingerprint. The ultimate prediction task for a prioritized AP list could be calculated based on rankings of probabilities of these labels. During our experiments, several kernel methods had been tested and the Sigmoid Kernel [7] was found to be the best in performance of both accuracy and run-time cost.

Sample on How WISDOM Framework Works

A sample is used here to interpret how WISDOM framework works with Similarity method.

(1) *For the learning procedure*: Table 2 shows the data collected on two APs (i.e., different BSSID) with same interested SSID. (2) *For modeling procedure*: according to Algorithm 1, use k=33 levels to divide the Cell RSSI range as [-115, -51] dBm in step h=2 with -115 as a representation of non-existence of Cell ID; use l=5 levels to divide AP RSSI range of [-100, -55] dBm in step g=9; based on AP RSSI *Level*, and registered Cell-ID *RegCell*, we get two sub-regions for AP_1 and two for AP_2 as Table 3, Table 4, Table 5 and Table 6 in PMF distribution on Cells.

	FP	Cell_1 (Level)	Cell_2 (Level)	Cell_3 (Level)	RegCell	AP RSSI (Level)
AP_1	p1	-61 (28)	-115 (1)	-87 (15)	Cell_1	-64 (4)
	p2	-63 (27)	-57 (30)	-85 (16)	Cell_1	-61 (4)
	p3	-63 (27)	-51 (33)	-85 (16)	Cell_1	-64 (4)
	p4	-65 (26)	-53 (32)	-115 (1)	Cell_1	-58 (4)
	p5	-67 (25)	-55 (31)	-81 (18)	Cell_2	-61 (4)
AP_2	q1	-71 (23)	-67 (25)	-75 (21)	Cell_1	-73 (3)
	q2	-115 (1)	-69 (24)	-73 (22)	Cell_1	-70 (3)
	q3	-73 (22)	-65 (26)	-73 (22)	Cell_2	-64 (4)
	q4	-75 (21)	-65 (26)	-71 (23)	Cell_1	-66 (3)
	q5	-75 (21)	-67 (25)	-115 (1)	Cell_1	-70 (3)

Table 2. Sample Data Collected during Learning Procedure

Levels	1	2 to 14	15	16	17 to 25	26	27	28	29	30	31	32	33
Cell_1	0	0	0	0	0	0.25	0.5	0.25	0	0	0	0	0
Cell_2	0.25	0	0	0	0	0	0	0	0	0.25	0	0.25	0.25
Cell_3	0.25	0	0.25	0.5	0	0	0	0	0	0	0	0	0

Table 3. PMF of AP_1 Sub-region $s_{1\text{-}1}$ (*AP Level 4*, *Cell_1*)

Levels	1 to 17	18	19 to 24	25	26 to 30	31	32 to 33
Cell_1	0	0	0	1	0	0	0
Cell_2	0	0	0	0	0	1	0
Cell_3	0	1	0	0	0	0	0

Table 4. PMF of AP_1 Sub-region $s_{1\text{-}2}$ (*AP Level 4*, *Cell_2*)

Levels	1 to 21	22	23 to 25	26	27 to 33
Cell_1	0	1	0	0	0
Cell_2	0	0	0	1	0
Cell_3	0	1	0	0	0

Table 5. PMF of AP_2 Sub-region $s_{2\text{-}1}$ (*AP Level 4*, *Cell_2*)

Levels	1	2 to 20	21	22	23	24	25	26	27 to 33
Cell_1	0.25	0	0.5	0	0.25	0	0	0	0
Cell_2	0	0	0	0	0	0.25	0.5	0.25	0
Cell_3	0.25	0	0.25	0.25	0.25	0	0	0	0

Table 6. PMF of AP_2 Sub-region $s_{2\text{-}2}$ (*AP Level 3*, *Cell_1*)

FP	Cell_1 (Level)	Cell_2 (Level)	*Cell_3 (Level)*	Cell_4 (Level)	RegCell
q	-63 (27)	-51 (33)	*-115 (1)*	-87 (15)	Cell_1

Table 7. Sample Query Fingerprint

(3) *For prediction procedure*: Table 7 gives a sample query fingerprint; the candidate AP list can be returned by Algorithm 2 firstly based on the Cell-AP relation learned previously; since registered Cell ID of the query fingerprint is Cell_1, the candidate AP list will be [AP_1, AP_2]; then Algorithm 3 will be invoked on each AP candidate to

calculate the similarity with that AP by looping through all PMFs of all sub-regions; for sub-region $s_{1\text{-}1}$ of AP_1, three Cell ID sets that defined by Algorithm 3 are as: $CellID_x$ = [Cell_1, Cell_2]; $CellID_y$ = [Cell_3]; $CellID_z$ = [Cell_4]; the similarity of the query fingerprint with sub-region $s_{1\text{-}1}$ can be calculated as:

$$\begin{aligned} D_{s_{1-1}} &= \lg PMF^{s_{1-1}}_{Cell\_1}(RSS_{Cell\_1}) \\ &+ \lg PMF^{s_{1-1}}_{Cell\_2}(RSS_{Cell\_2}) \\ &+ \lg PMF^{s_{1-1}}_{Cell\_3}(RSS_{Cell\_3}) + \lg \sigma \end{aligned}$$

If we take a constant value 1/5000 as the minimum probability p_{min}, then the similarity will be:

$$\begin{aligned} D_{s_{1-1}} &= \lg 0.5 + \lg 0.25 + \lg 0.25 + \lg(1/5000) \\ &= \text{-}5.2041 \end{aligned}$$

Similarities with $s_{1\text{-}2}$ of AP_1, and $s_{2\text{-}1}$, $s_{2\text{-}2}$ of AP_2 can be similarly calculated separately as:

$$D_{s_{1-2}} = \text{-}14.7959,\ D_{s_{2-1}} = \text{-}14.7959,\ D_{s_{2-2}} = \text{-}11.6990$$

For AP_1, since the similarity of $s_{1\text{-}1}$ is larger than $s_{1\text{-}2}$, then $s_{1\text{-}1}$ will be the most probable sub-region that the query fingerprint will be located in, and the return value will be -5.2041 with relative level as 4; for AP_2, the return value will be -11.6990 with relative level as 3; finally, the return values will be collected by Algorithm 2 for order sorting based on similarity value and accepted level; assuming the predetermined value is l_{min}=2, then after filtering and sorting, the prioritized AP list is [AP_1, AP2] with AP_1>AP_2 in priority; based on the recommendation of prioritized AP list output by WISDOM, mobile device will turn on WLAN adapter and discover the interested WLAN.

SYSTEM ARCHITECTURE AND IMPLEMENTATION

WISDOM framework can be implemented and deployed in (1) Remote Network Assisted (RNA) architecture, (2) Local Device Maintained (LDM) architecture, or (3) RNA-LDM-Combined architecture as shown in Figure 7. For LDM architecture, the entire WISDOM framework is implemented and managed locally on mobile devices: the Learn symbol stands for learning procedure when connected to AP and getting cellular fingerprints for model training; the Query symbol stands for query process on getting the predicted result generated by WISDOM model. For RNA architecture, WISDOM modeling is performed on a WISDOM Server or an ANDSF like network component with WISDOM extension; the Learn symbol represents updating training records to the network side; the Query symbol represents query process, but this could also be a pushing behavior from network side to the mobile device.

We have implemented prototypes for both RNA and LDM architecture. As all these systems perform similar tasks despite whether performed entirely locally or with network support, we will only take the prototype for LDM architecture as an example for introduction. It is an Android application called WISDOM Wi-Fi, which helps users to predict the most favorite WLAN available. The entire WISDOM framework is implemented on the mobile device; thus the prediction task is performed locally. Moreover, a sharing function is also implemented as an extension, which enables WISDOM Knowledge Base (i.e., the trained models) to be shared across different devices to predict unknown APs or enhance the known models.

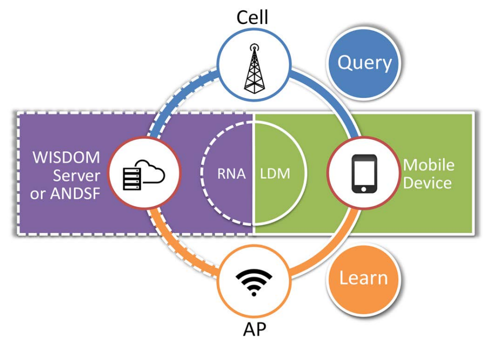

Figure 7. LDM, RNA and RNA-LDM-Combined Reference Architecture. LDM has Cell, AP and Mobile Device; RNA has WISDOM Server/ANDSF, Cell, AP and Mobile Device.

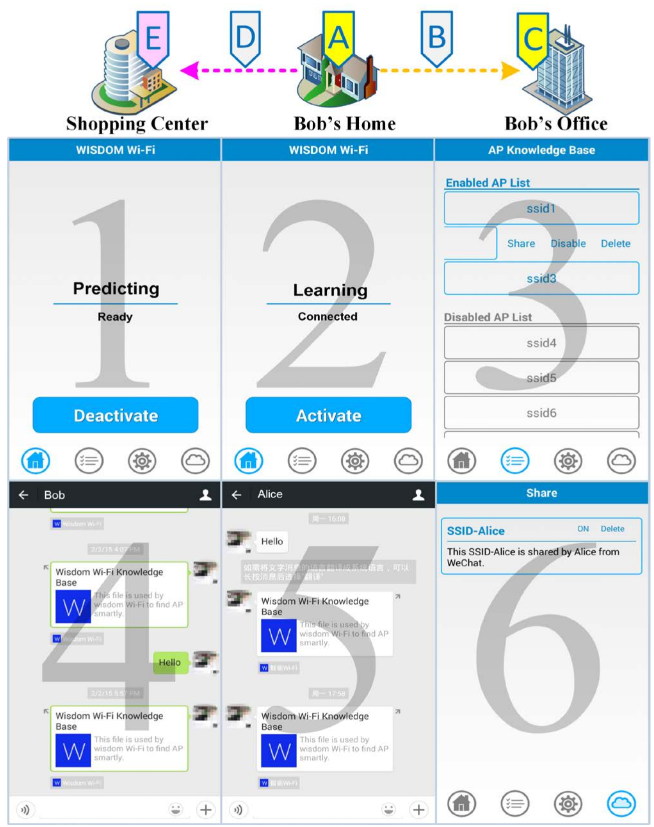

Figure 8. Demonstration and Screenshots

Figure 8 shows the typical user scenarios of WISDOM Wi-Fi with screenshots. Suppose Alice and Bob are friends, and both have used the WISDOM Wi-Fi for some time. Bob has the Knowledge Base of both his home and office, while Alice has that of a Shopping Center. Every morning Bob will leave home **A** for the office **C**. When Bob is on way **B** to office, WISDOM Wi-Fi will turn off his mobile device's WLAN adapter, and work in predicting mode (*as screenshot 1*). Once Bob arrives at the office, the office WLAN will be successfully predicted and his mobile device's WLAN adapter will be turned on and get connected to office WLAN (*as screenshot 2*). When Bob comes back home after work, the similar procedures will occur again with the home WLAN predicted. Now suppose Alice and Bob are chatting via a social networking application (WISDOM Wi-Fi currently supports WeChat [25] and will add support of other applications like Facebook) and Bob says he will go to the shopping center **E** in the weekend. Alice tells Bob that she often goes there, and shares her Knowledge Base AP to Bob through this social networking app (*as screenshots 3 and 4*). Bob simply clicks the sharing message in the application, and easily gets the Knowledge Base with shopping mall AP installed in his local WISDOM Wi-Fi application (*as screenshots 5 and 6*). When Bob goes on the way **D** to the shopping center **E**, WISDOM Wi-Fi will turn off WLAN adapter like **B**. Once Bob arrives at **E**, WISDOM Wi-Fi will successfully predict the shopping center's WLAN, and turn on the device's WLAN adapter for connection.

EXPERIMENTS AND PERFORMANCE EVALUATION

We conducted both simulations and field tests to evaluate the performance of the WISDOM framework. The simulations were performed with both the Similarity and Classification methods based on the collected data sets. And the field tests were performed with the WISDOM Wi-Fi application implemented with Similarity method.

Data Collection

To analyze the WLAN discovery problem under real situations, we collected data of cellular network and WLAN APs for verification for both outdoor and indoor by walking around the office campus (around 720,000 square meters) with certain slightly different walking paths (e.g., distant from one to several buildings) as Figure 4. We used Android devices and had developed an Android application DataCollector for data collection. The mobile devices used for data collecting are listed in Table 8. The DataCollector app had been installed on each testing mobile device, and for each device the records were collected for three to five times each day around the office campus including both outdoor and indoor on Jul. 15 to 17, and Sep. 15 to 18 in 2014. Cellular information was recorded every second, and AP information was recorded every five seconds. For simplicity, we only use China Mobile's GSM cellular network for cellular fingerprints. It works similarly when we use cellular fingerprints of other network modes like 3G or 4G. For WLAN, we will use the same operator's WLAN deployed at places of both indoor and outdoor. Each collected data record (i.e., the general fingerprint) is composed of: (1) registered Cell-ID and its RSSI value that can be returned by all the mobile devices under tests; (2) all neighboring Cell-IDs and corresponding RSSI values; (3) all available AP information in groups: each group contains information of an AP with SSID name, BSSID, RSSI value and channel information. All collected data contained 79 unique APs of the operator's WLAN SSID and 65 unique Cell-IDs of operator's GSM network. The collected data records were organized into training data sets and testing data sets without noisy data records (i.e., records of empty AP list) for WISDOM methods on modeling, performing prediction and results evaluation through simulation.

Device Model	**OPPO Find 5**	**Huawei Honor 3C**	**HP Slate 6**	**Samsung Galaxy S3**
Android Version	*4.2*	*4.2*	*4.2.2*	*4.1*
Cell-ID Capability	*Multiple*	*Multiple*	*Single*	*Single*
Training Set Name	*op#1*	*hw#1*	*hp#1*	*s3#1*
Testing Set Name	*op#2*	*hw#2*	*hp#2*	*s3#2*
Training Set Size	*9730*	*4779*	*3171*	*1915*
Testing Set Size	*334*	*377*	*591*	*516*

Table 8. Mobile Device Models

Evaluation Goals

We will evaluate the performance of the WISDOM framework for both accuracy and run-time cost metrics. For run-time cost, we will investigate the modeling cost and query cost respectively. The following three aspects are evaluated through simulation.

(1) *Performance comparison among different data set selection approaches:* We define three different cases: (a) **Self-Prediction**: using each device's own data sets, i.e., #1 for training, and #2 for testing; (b) **All-Prediction**: training with all devices' #1, and testing with each devices' own #2; (c) **Inter-Prediction**: training with each and all unknown devices' #1 separately (e.g., X3#1 for op stands for all #1of hw, hp and s3), and testing with each one's own #2.

(2) *Performance comparison of Multiple-Cell-ID and Single-Cell-ID fingerprints*: The data sets we collected include both Multiple-Cell-ID and Single-Cell-ID fingerprints. When performing the comparison, for modeling and predicting with Single-Cell-ID, we only use the registered Cell-ID in the Multiple-Cell-ID fingerprint.

(3) *Performance comparison of Similarity method and Classification method.*

Evaluation Criteria for Prediction Accuracy

The evaluation criteria for accuracy are based on the returned prioritized AP list. For consistency and simplicity, we use the first five APs of the returned list for evaluation. In addition, we use a concept similar to nDCG (Normalized Discount Cumulative Gain) ranking measurement for evaluation. Normally, DCG is calculated with the following formula. In WISDOM context, p is the number of APs in the result; and rel_i is the relevance of the i^{th} AP for ranking.

$$DCG_p = rel_1 + \sum_{i=2}^{p} \frac{rel_i}{\log i}$$

The result APs will be matched to the ground-truth APs with levels (mapping to the divided RSSI ranges) of the matched

AP as relevance; for any returned APs that couldn't be found in ground-truth APs, zero is used as relevance. Moreover, instead of using the default of base 2 for logarithm calculation in DCG, we will use n=1.8 (any 1<n<2 will work) as logarithm base to avoid the exceptional cases that exchanging items at 1st and 2nd ranking place will get the same DCG during the evaluation. And then, nDCG is calculated with the following formula:

$$nDCG_p = \frac{DCG_p}{IDCG_p}$$

We preprocess the testing data set by merging all the same cellular fingerprints so that any unique cellular fingerprint maps to a group of APs, where each AP contains several level occurrence and count of each occurred level. We induce the ultimate AP orders based on AP's level and the counts of that level. The *DCG* for this unique cellular fingerprint will be calculated and used as the *IDCG* for the predicted results. In this way, a higher *nDCG* will indicate a better accuracy. In addition, the testing data records of unknown registered Cell-ID are ignored during prediction, and won't be counted into overall success rate, which is the ratio of records with *nDCG*>0 of the whole testing data set.

Evaluation Environment

For the evaluation, we run the WISDOM framework implementation with the collected data sets on an HP EliteBook 8470p with Windows 7 Enterprise, which is equipped with CPU of Intel i5-3320M, 8GB memory and 500GB disk. For Similarity method, which we implemented purely in Java, the Double.MIN_VALUE [11] is used as p_{min} empirically. For Classification method, we adopted a Java library of LibSVM version 3.2.0 [7] for machine learning, and use the Sigmoid Kernel method with default parameters.

Here are some legends used in the figures of evaluation results. For Figure 9 and Figure 10: *S* stands for *Similarity* method, *C* for *Classification* method; *Multiple* (and *M* in Figure 12) stands for the fingerprints used as Multiple-Cell-ID, *Single* (and *S* in Figure 12) for Single-Cell-ID.

Self-Prediction

The overall performance of self-prediction is shown as Figure 9. The successful rate ranges from 59.68% to 100% with an average of 87.06%. By looking into the training and testing data of these devices, it can be found that most of the failed prediction could be led by the signal fluctuation or by limited historic information (i.e., too few records for training). The case of *op#2* has relative low accuracy; it is caused by the reason that the walking path of testing set varied too much from those in training data set. Thus it can still be treated as limited historic information provided by training data set. On the contrary, the case of s3 having nearly the same walking path of testing data set as the training data set, thus it has relatively high accuracy. So in general these cases could be improved by collecting more records covering more regions. Figure 12 shows more details of the performance of self-prediction under Similarity method. The nDCG range has been divided into 12 different ranges with 0 stands for failure and 1 stands for 100% correctness. The successful rate is composed of ranges larger than zero.

All-Prediction

In contrast to self-prediction, the all-prediction case is to perform modeling with all cellular fingerprints collected from all different devices for prediction. The performance of all-prediction is shown as Figure 10. The successful rate ranges from 77.42% to 100% with an average of 89.77%. For most cases the accuracy has been improved. However, for s3 the accuracy has been dragged down, which is caused by reasons of (1) records of testing data set *s3#2* having registered Cell-ID as unknown in training data set *s3#1* but known in training data set of all *#1* yet with only partial availability being successfully predicted, and (2) records of training data set *s3#1* get diluted in modeling with training data of all other *#1*, to result in a lower probability of the AP to be correctly predicted or even eliminated from the final prediction result. These reasons also apply to other cases that get lower accuracy in Figure 10 than in Figure 9.

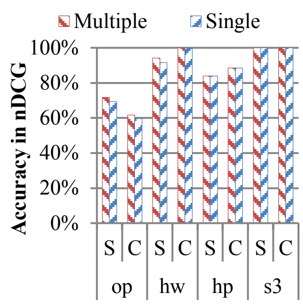

Figure 9. Self-Prediction

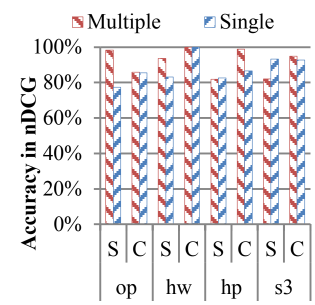

Figure 10. All-Prediction

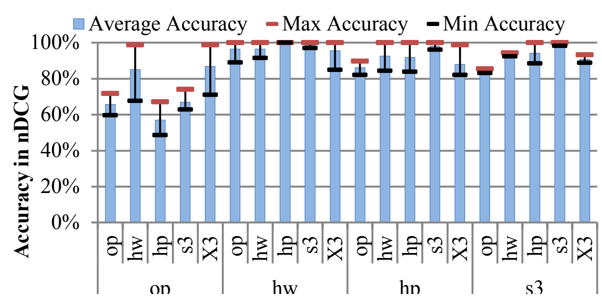

Figure 11. Inter-Prediction Results

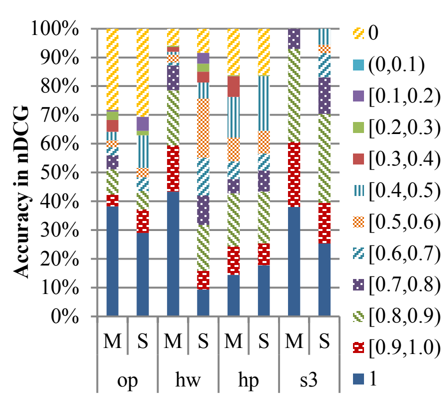

Figure 12. Self-Prediction Insight under Similarity Method

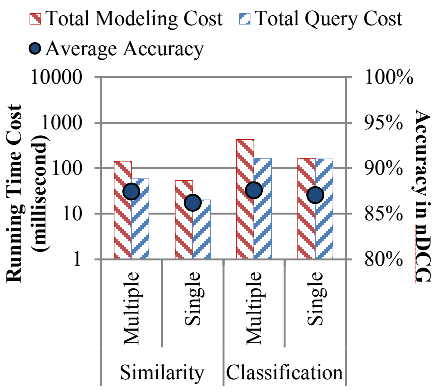

Figure 13. WISDOM Methods Comparison on Self-Prediction

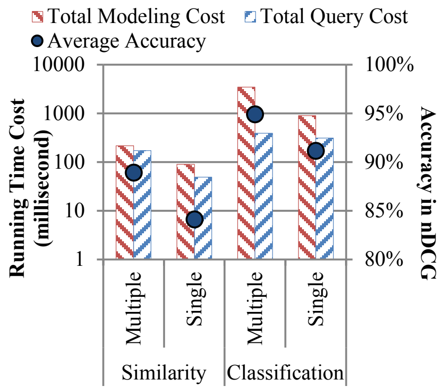

Figure 14. WISDOM Methods Comparison on All-Prediction

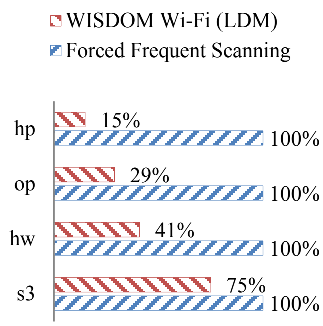

Figure 15. Power Consumption Comparison

Inter-Prediction

The inter-prediction is to verify the feasibility of enabling learned models to be shared and reused between different devices. Figure 11 shows the performance of predicting using different devices' training data sets. In the figure, training data set *X3* is from all other mobile devices (e.g., *X3* for *op* stands for all *#1* of *hw, hp* and *s3*). While the bar stands for the average accuracy, the maximum and minimum accuracy are also marked over each bar. It can be observed that the average accuracy ranges from 56.93% to 100% with an overall average of 88.34%.

Data Set Selection Approaches Comparison

In general, the self-prediction results indicate the feasibility of entirely locally managed mobile applications. The inter-prediction results prove the device independent capability of WISDOM. And for all-prediction result, it provides the guidance for implementation and deployment of network assisted solutions in that it could achieve better performance when modeling according to device capabilities (i.e., Multiple-Cell-ID or Single-Cell-ID devices); and the more training data collected the more accuracy it will achieve. However, for the model construction at early stage lacking of training data, all data from different device models could be used for modeling since it could also achieve an acceptable average accuracy.

Multiple-Cell-ID and Single-Cell-ID Comparison

The difference of modeling with Multiple-Cell-ID or Single-Cell-ID fingerprints can be observed from Figure 9 and Figure 10. For Similarity method, the Multiple-Cell-ID modeling tends to be better for Multiple-Cell-ID devices (i.e., *op* and *hw*) in accuracy; and Single-Cell-ID modeling tends to be better for Single-Cell-ID devices (i.e., *hp* and *s3*). For Classification method, Multiple-Cell-ID modeling always tends to be better than Single-Cell-ID modeling.

Similarity and Classification Methods Comparison

Figure 13 and Figure 14 show the average prediction accuracy and the total run-time cost of all devices in Multiple-Cell-ID and Single-Cell-ID cases for the two methods. It shows that the two methods have little difference (less than 1%) in accuracy for self-prediction case; while for all-prediction case the Classification method has higher accuracy. However, the Similarity method always performs better on cost for both cases. The cost of Classification method has a relative significant increase for the all-prediction case due to the increasing size of training data set. This also indicates that Similarity method would be more suitable for Local Device Maintained architecture; while Classification method would be more suitable for Remote Network Assisted architecture, which can take advantage of Cloud computing platforms.

Field Test

Field tests had also been performed using the WISDOM Wi-Fi application with certain modification to verify the performance of WISDOM under real use case. WISDOM Wi-Fi always output the prediction result as: (1) Unknown: for any query fingerprint with unknown registered Cell-ID; (2) Recommended: with the prioritized AP list; (3) Not-Recommended: WISDOM predicts the availability of WLAN in bad quality, but still returns the prioritized AP list for any forced attempt. We modified WISDOM Wi-Fi by turning on WLAN adapter for (2) and (3) to verify all the positive cases (i.e., non-empty prediction) with acceptable signal strength as -91dBm (i.e., the lowest acceptable level defined by most Android kernel implementation). Since the WLAN adapter will get turned off for empty prediction result, the negative cases wouldn't be verified for the field test under real case, but this could be resolved by simulation, and actually the evaluation results in simulation section has already contained all the negative cases. We count the number of results (2) and (3) as *T*, and define the Success (*S*) and Failure (*F*) of our evaluation for Prediction (*P*) and Real Case (*R*) as Table 9. Hence the successful rate of prediction for our field test can be evaluated as *S*/*T* to reflect the overall performance of WISDOM Wi-Fi under real case. We installed and used the modified WISDOM Wi-Fi and let the mobile device connected to the office and home WLAN (i.e., the learning) the first day. And the next day, we carried the mobile device (e.g., Samsung S3) on the way from home to office and vice versa, to expect the office and home WLAN network to be successfully discovered and get connected to once on our arrivals. Table 10 shows the field test statistics.

R \ P	Recommended	Not Recommended	Unknown
No Predicted AP	*F*	*S*	-
Predicted AP RSS >= -91dBm	*S*	*F*	-
Predicted AP RSS < -91dBm	*F*	*S*	-

Table 9. Instruction on Success and Failure Conditions

Total Prediction				1479
Unknown Cases (i.e., Empty Prediction)				1374
Non-Empty Prediction	**Not Recommended**	**Success (*S*)**	87	105 (*T*)
		Failure (*F*)	3	
	Recommended	**Success (*S*)**	12	
		Failure (*F*)	3	
Successful Rate			94.29%	

Table 10. Field Test Statistics on Predicting WLAN

The result showed that field test could reach 94.29% in accuracy for home and office WLAN prediction under a daily routine use case. Of course, the results of the field tests and simulations reflect the specific conditions under specific locations, walking paths, wireless conditions, time of day, device models, size of the training data, etc. It may be different when given different impact factors. Figure 15 also shows the power consumption of mobile devices under field tests compared to cases by keeping WLAN adapter performing frequent WLAN scanning with 10 seconds interval. It can be observed that WISDOM could even save mobile device's power (60% on average) while guarantee the efficiency of WLAN discovery.

FURTHER DISCUSSION AND FUTURE WORK

The impact factors on the effectiveness of WISDOM framework under real cases are complicated, and can be generally categorized as the following factors. Each impact factor actually covers a set of sub-topics, and we only discuss a few of them for the generalization of WISDOM framework to the complex conditions under real cases.

(1) Wireless condition factors (changes in infrastructure, environment, etc.): deployment changes on updating cellular base stations and WLAN APs may happen occasionally; under such cases, WISDOM framework can be implemented to adapt to the wireless infrastructure changes by adding elimination operation (e.g., remove data older than weeks) to learning and modeling procedures to reflect the latest WLAN conditions. The different cellular network modes (i.e., GSM, UMTS and LTE, etc.) serving in parallel for backward compatibility normally have different cellular fingerprints from each other. To solve the issue, we could separately construct the WISDOM models under different network modes, or simply model all Cell-IDs from different networks together. Weather conditions, traffic and people crowds will also bring fluctuation to the radio signal strength, but this factor actually will be easily conquered by putting it into a distribution or an average condition in the long run.

(2) Location factors (urban and rural areas with different infrastructure deployment density, etc.): some rural areas may have sparse density in cellular coverage like a single base station covering area in miles, and WISDOM will indeed perform poor in prediction for relative small scale WLAN coverage under such cellular coverage. However, since user's moving paths always fall in similar patterns, this will only result in more false-positive prediction results, in other words, some compromise in power consumption.

(3) Mobile device factors (models, firmware, drivers, etc.): different devices models are usually equipped with different wireless chips and antennas, and this will result in different measurement capability and report behaviors. However, the evaluation results on inter-prediction prove the feasibility of reusing the trained models from different device models for prediction while achieving an acceptable accuracy. This also motivates the idea of instant learning by sharing WISDOM models among mobile devices to predict previously unknown WLAN without tediously learning process.

(4) User's behavior factors (preference settings, user's moving behaviors, data collection scale, etc.): the prediction performance at the initial learning stage for new AP is closely related to user's collecting behaviors; walking in different paths for learning is usually a good practice to collect all possible Cell-IDs to reduce false-negative cases. User preference settings such as enabling or disabling the interested WLANs for discovery will be implementation dependent in order to meet specific user requirements under different application scenarios. Other factors such as network bandwidth, backhaul condition, etc., may also be considered on how the prediction result of WISDOM framework shall be adapted for specific application implementation.

We are now applying the WISDOM framework into the 4G/LTE core network to enhance Mobile Data Offloading (MDO) for heterogeneous network selection. In the future, we will further enhance the WISDOM framework for a more complete solution from the abovementioned aspects.

CONCLUSION

This paper introduces WISDOM framework on leveraging cellular fingerprints for direct WLAN coverage modeling and availability prediction. Results of experiments show the efficiency of WLAN discovery can be guaranteed with power saving gains. Different impact factors are discussed for the generalization of WISDOM framework under real cases. Improvements to the performance on prediction can basically be achieved by further data collection. WISDOM framework can be used to enhance SP's MDO solution (e.g., enhancement to ANDSF), the enterprise's BYOD and CYOD solutions (e.g., network finder), and various mobile applications (e.g., enhanced connection manager) that require efficient WLAN discovery, power saving and protection on privacy information from leaking.

ACKNOWLEDGEMENTS

Special thanks to the engineers Kuan Shu, Meng Guo and intern Jie Zhao in HP Labs China for their hard work on the research investigation, prototype development, experiments and analysis conducted for the WISDOM project.

REFERENCES

1. 3GPP TS 24.312. Access Network Discovery and Selection Function (ANDSF) Management Object (MO) (Release 12)
2. A. Carroll and G. Heiser, “An analysis of power consumption in a smartphone,” in Proc. of USENIX ATC, 2010.
3. A. Schwaighofer, M. Grigoras, V. Tresp, and C. Homann. GPPS: A Gaussian process positioning system for cellular networks. In Advances in Neural Information Processing Systems 16. MIT Press, 2003.
4. Ahmad Rahmati and Lin Zhong: Context-for-Wireless: Context-Sensitive Energy-Efficient Wireless Data Transfer, in MobiSys’07.
5. B. Ferris, D. Hähnel, and D. Fox. Gaussian processes for signal strength-based location estimation. In Robotics: Science and Systems. The MIT Press, 2006.
6. Chen, M. Y., Sohn, T., Chmelev, D., Haehnel, D., Hightower, J., Hughes, J., ... & Varshavsky, A. (2006). Practical metropolitan-scale positioning for gsm phones. In UbiComp 2006: Ubiquitous Computing (pp. 225-242). Springer Berlin Heidelberg.
7. Chih-Chung Chang and Chih-Jen Lin, LIBSVM : a library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2:27:1--27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm
8. F. L.M. and N. M., “Investigating the energy consumption of a wireless network interface in an ad hoc networking environment,” in Proc. of the Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies, Anchorage, USA, April 2001.
9. Farid, F., Shahrestani, S., & Ruan, C. (2013, December). Application Based Handover: An Energy-Efficient Approach for Heterogeneous Networks. In Computational Science and Engineering (CSE), 2013 IEEE 16th International Conference on (pp. 1185-1190). IEEE.
10. Izumikawa, H., Kishi, Y., & Nagareda, R. (2009, April). Novel WLAN coverage area estimation leveraging transition of cellular signal using multi-mode mobile terminal for heterogeneous networks. In Wireless Communications and Networking Conference, 2009. WCNC 2009. IEEE (pp. 1-6). IEEE.
11. JDK 7 API Online Docs. http://docs.oracle.com/javase/7/docs/api/java/lang/Double.html#MIN_VALUE
12. K Li, J Bigham, E Bodanese, L Tokarchuk, “Outdoor Location Estimation in Changeable Environments,” IEEE Commun. Lett., vol. 17, no. 11, pp. 2072-2075,November 2013
13. Li, K., Bigham, J., Bodanese, E., & Tokarchuk, L. (2013). Location estimation in large indoor multi-floor buildings using hybrid networks. In Wireless Communications and Networking Conference (WCNC), 2013 IEEE (pp. 2137–2142).
14. Lim, W. S., Kim, D. W., Suh, Y. J., & Won, J. J. (2008, November). Efficient WLAN Discovery Schemes Based on IEEE 802.21 MIH Services in Heterogeneous Wireless Networks. In Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE (pp. 1-5). IEEE.
15. M. Ibrahim and M. Youssef "CellSense: An accurate energy-efficient GSM positioning system", IEEE Trans. Veh. Technol., vol. 61, no. 1, pp.286 -296 2012
16. Mandru, N. P., Bavirisetti, D. P., & Khara, S. An Efficient Vertical Handoff Technique for Two-Tier Heterogeneous Networks. International Journal of P2P Network Trends and Technology-Volume2 Issue2-2012.
17. Nam, M., Choi, N., Seok, Y., & Choi, Y. (2004, September). WISE: energy-efficient interface selection on vertical handoff between 3G networks and WLANs. In Personal, Indoor and Mobile Radio Communications, 2004. PIMRC 2004. 15th IEEE International Symposium on (Vol. 1, pp. 692-698). IEEE.
18. Robin Wentao Ouyang, Albert Kai-Sun Wong, and Kam Tim Woo. "Indoor Localization via Discriminatively Regularized Least Square Classification". International Journal of Wireless Information Networks June 2011, Volume 18, Issue 2, pp 57-72.
19. S. Thrun, D. Fox, W. Burgard, and F. Dellaert. Robust Monte Carlo localization for Mobile Robots. Articial Intelligence, 128:99-141, 2001.
20. Song, Q., & Jamalipour, A. (2006, September). A time-adaptive vertical handoff decision scheme in wireless overlay networks. In Personal, Indoor and Mobile Radio Communications, 2006 IEEE 17th International Symposium on (pp. 1-5). IEEE.
21. Triantafyllopoulou, D., Guo, T., & Moessner, K. Energy-Efficient WLAN Offloading through Network Discovery Period Optimization.
22. Tsoumakas, G., Katakis, I., Vlahavas, I. "Mining Multi-label Data", Data Mining and Knowledge Discovery Handbook, O. Maimon, L. Rokach (Ed.), Springer, 2nd edition, 2010.
23. Tsoumakas, Grigorios; Katakis, Ioannis. "Multi-label classification: an overview". International Journal of Data Warehousing & Mining 3 (3): 1–13.
24. User Privacy in iOS and OS X, http://devstreaming.apple.com/videos/wwdc/2014/715xx4loqo5can9/715/715_user_privacy_in_ios_and_os_x.pdf

25. WeChat Android Edition. http://www.wechat.com/en/
26. Yacine Oussar, Iness Ahriz, Bruce Denby and Gérard Dreyfus. "Indoor localization based on cellular telephony RSSI fingerprints containing very large numbers of carriers". EURASIP Journal on Wireless Communications and Networking, 2011, 2011:81.
27. Ye Tian, Bruce Denby, Iness Ahriz Roula, Pierre Roussel, Rémi Dubois and Gérard Dreyfus. Practical Indoor Localization using Ambient RF. IEEE International Instrumentation and Measurement Technology Conference (I2MTC), May 2013, United States. pp. 1125-1129.
28. Yohan Chon, Suyeon Kim, Seungwoo Lee, Dongwon Kim, Yungeun Kim, Hojung Cha, "Sensing WiFi Packets in the Air: Practicality and Implications in Urban Mobility Monitoring", UBICOMP '14, SEPTEMBER 13 - 17, 2014, SEATTLE, WA, USA
29. Yuvraj et al: Wireless Wakeup Revisited: Energy Management for VoIP over WLAN Smartphones, in MobiSys'07
30. Zhang, D., Ren, P., Wang, Y., Du, Q., & Sun, L. (2014, October). Energy management scheme for mobile terminals in energy efficient heterogeneous network. In Wireless Communications and Signal Processing (WCSP), 2014 Sixth International Conference on (pp. 1-5). IEEE.