CAIL: Cross-calibration for Accurate Indoor Localization Without Extensive Site Survey

Geetali Tyagi
IIIT-Delhi
New Delhi, India
geetali12037@iiitd.ac.in

Shweta Sood
IIIT-Delhi
New Delhi, India
shweta12164@iiitd.ac.in

Vinayak Naik
IIIT-Delhi
New Delhi, India
naik@iiitd.ac.in

Rizwana Ahmad
IIIT-Delhi
New Delhi, India
rizwanaa@iiitd.ac.in

***Abstract*—WiFi-based indoor localization is a complex problem due to high variations of radio frequency (RF) signals in indoor environment. Many popular techniques based on RF fingerprinting require an extensive site survey, which involves time intensive logging of Received Signal Strength (RSS). This paper presents CAIL, a smartphone-based indoor localization system, that utilizes the site survey done by a phone to create an RF fingerprint which is utilized by new phones for location prediction. CAIL neither makes any assumptions about the site and placement of access points (APs) nor does it require any additional infrastructure. CAIL provides these new phones a minimal set of best locations to log at, thereby reducing the war-driving efforts for these phones. CAIL was tested in a building of six floors. Experimental results show that CAIL provides an accuracy of 76%, comparable to the accuracy of 81% on complete site survey, with an 84% reduction in effort.**

I. Introduction

Indoor localization has become an integral part of context-aware applications, such as monitoring user's movements, attendance systems, location-aware search results for content filtering, etc. Many Bluetooth and WiFi-based techniques use data from wireless hardware, such as Simple Network Management Protocol (SNMP) traps and Bluetooth beacons. However, these approaches are unfavorable due to numerous reasons. Bluetooth beacons involve an additional cost of installation of infrastructure and require administrative access to network hardware. SNMP traps are messages sent by mobile phones to SNMP managers, typically used to monitor the connected phones within a network and modify them as required. Traps contain user data that endanger their privacy by exposing information of their connected phones to attacks. RF-based techniques eliminate the additional infrastructure requirements and the privacy concerns by creating an RF fingerprint of the building. The users are able to locate their position by mapping of the signals from their phone onto the WiFi fingerprint. Existing solutions require an extensive site survey to build these RF maps [1].

Traditional fingerprinting techniques involves calibration. Calibration of a phone uses training data from the entire environment, to create a fingerprint for testing of that phone. This paper defines a new technique called *Cross-calibration*, detailed in section III.B. Cross-calibration uses the complete set of training data available for one phone to calibrate another phone, which does not have the training data available for the entire environment. CAIL uses a phone, called $primary$ phone, to log into the site once to create a fingerprint, which is used by other phones to determine their location. This is possible due to the cross-calibration of these new phones against the $primary$ phone. For cross-calibration, these new phones have to log only at the best locations in the environment, identified by CAIL using a technique called recursive bagging, described in III.C. A key advantage of this technique is that it does not make any assumptions about the environment being logged, distribution of APs and their transmit power, unlike earlier work [2], [3]. It is not battery hogging for the phone as the user is required to log only at a few locations that are already identified by the system. This optimization gives accuracy which is comparable to the accuracy when the entire fingerprint for a phone is available.

CAIL finds its application in detecting the presence of users in malls, or in classrooms as a part of an attendance system. The significant contribution of this system is that it provides performance comparable to exhaustive war-driving but with reduced efforts, i.e.,:

- Reduction in war-driving efforts required during fingerprinting of the environment. The cost of training is reduced to the one-time fingerprinting of a $primary$ phone, which is leveraged by the new phones. There is an 84% reduction in data collection efforts, discussed in Section IV.
- CAIL deploys three independent modules comprising indoor localization, cross-calibration, and recursive bagging, detailed in Section III. This makes the system modular as the cross-calibration and recursive bagging algorithms can be applied irrespective of the localization algorithm used.

The rest of the paper is organized as follows: Section II details prior work done in the field of indoor localization, primarily focusing on eliminating user effort for localization. Section III describes CAIL localization and cross-calibration algorithms, and how to select the best locations among the available set, using recursive bagging. The localization algorithm uses k Nearest Neighbor (kNN) [4], detailed in Section III.A. The data collection phase and experiments conducted to evaluate CAIL are described in Section IV. In Section V, we discuss the results of our experiments and the effectiveness of the proposed system in reducing site survey. Section VI presents our concluding remarks.

II. Related Work

The work done in the domain of indoor localization can be classified into primarily two categories:

978-1-5090-5356-8/17/$31.00

A. Model-based techniques

These techniques construct an RF propagation model like the log-distance path loss (LDPL) model, which estimates RF distances using RSS values. Lim *et al.* [5] deploy WiFi trackers at predefined locations to measure RSS values and generate an RSS map using LDPL model. Ji *et al.* [6] also deploy WiFi trackers but instead employ a more sophisticated ray tracing model. However, all these model-based approaches are restricted by the necessity of an available accurate model of the environment or building. Even if such a model is available, the RF signal propagation will be affected by other factors such as the building material of walls. Each such factor leads to increased computation costs, thus making localization nonviable. On the other hand, without the information about the environment, accurate model-based localization is not possible. For example, to reduce laborious site survey, EZ [7] uses a configuration-free indoor localization scheme, without any assumptions about the RF environment, by modeling constraints of wireless propagation and solving using a genetic algorithm. However, these reduced measurements come at the cost of accuracy.

B. RF Fingerprinting based techniques

These techniques require fingerprinting the environment at required locations and then comparing these stored fingerprints with measurements from the phone for testing. RADAR [1] uses RSS values at various locations from APs for deterministic fingerprinting and matching. A more accurate system, Horus [8] uses probability distributions for each AP and location instead of absolute RSS measurements. The matching is done using maximum likelihood.

PlaceLab [9] uses radio beacons for localization. ActiveCampus [10] uses a similar technique but with added assumption of knowledge about the physical AP locations. SurroundSense [11] utilizes ambiance features, such as WiFi signals, camera, sound, and light for indoor localization. Collating all these environmental factors ignores energy considerations and may not work well for locations lacking ambiance diversity like libraries, offices. A peer assisted localization approach [12], leverages acoustic ranging estimates among the peer phones combined with WiFi-based localization to obtain an accurate distance estimate. All these techniques require fingerprinting the entire indoor space and maintenance of RF logs. DAIR [13] tries to eliminate such constraints, but this requires a dense deployment of USB-based adapters attached to desktop machines for monitoring wireless network, hence, increasing the pre-deployment effort.

A system, Zee [14] uses the walks that users take through environments of interest on a daily basis and the inertial sensors without explicit user intervention. It combines this with the constraints provided by a blueprint of the indoor space, taken as an external input, to discard infeasible locations in the environment and finally converge to the true location. A technique WILL [15], substitutes the traditional labor-intensive site survey with a combination of accelerometers to track user movements and WiFi fingerprints. It relies on a physical floor plan for mapping the logical plan constructed from fingerprint data to the actual rooms. Additionally, there is continuous WiFi logging from the phone which may lead to battery drainage and redundancy in the huge amount of data produced.

The technique proposed in this paper doesn't require any external information about the environment, additional sensors such as an accelerometer, a magnetometer, or acoustic sensors to generate location prediction. The reduction in effort in Zee and WILL systems is due to reduced direct manual labor for WiFi measurements, not the actual data collected or time consumed during collection. CAIL shrinks the actual data for the user, in addition to reducing manual labor, to a subset of the total environment and still produces desired results within the whole environment.

III. ALGORITHM

CAIL model for cross-calibration uses two phases of the generic fingerprinting algorithm. *Training phase* comprises data collection at different locations for creating the fingerprint. APs create a network, having a unique id which is the Basic Service Set Identifier (BSSID). BSSID and RSS values for the *primary* phone from wireless APs are recorded for multiple logs. The median RSS value for each BSSID at each location is stored. This creates training log for a phone, which is used by the localization algorithm described in section III.A. *Testing phase* comprises data collection at test locations, which are predicted by the CAIL localization algorithm described in III.A. The cross-calibration of new phones against the training phone for inter-phone evaluation is done using CAIL cross-calibration algorithm described in III.B. CAIL Recursive bagging algorithm described in III.C is used for selecting a subset of locations for cross-calibration. This reduces the effort of data collection during testing phase to just 16%.

A. CAIL Localization Algorithm

CAIL localization algorithm is a smartphone-centric approach for indoor localization. It doesn't depend on the transmit power of the access point but on the strength of the received signal as we only log the RSS, measured in dBm units. The methodology for location detection employs k Nearest Neighbor (kNN), a clustering algorithm, commonly used in machine learning for classification of data points. In our scenario, these data points are median RSS values from all BSSIDs and k =1. Using this technique, we find which location amongst all the locations logged during training time has an RSS value closest to the RSS value in the testing log for a common BSSID. This is repeated for all the BSSIDs common to training and testing logs, and they give their respective predictions for a testing location. Finally, a mode of the locations predicted by these BSSIDs gives the final prediction of the location. Mode is a majority voting of BSSIDs to predict a location, thus it increases the confidence in the prediction.

Let us consider two locations: location 1, location 2 in our training phase. At these locations, let n_1, n_2 be the number of BSSIDs detected, respectively. For all $n_1 \cup n_2$ BSSIDs, we store the median RSS value of all the logged values for that location and BSSID. So for location 1, each of the n_1 BSSIDs has multiple RSS values logged at different times and days. Radio signals undergo irregular multi-path reflections that could cause variation in RSS values of the signals [10].

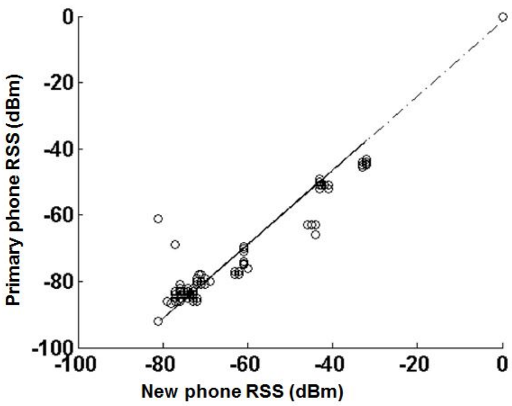


Fig. 1: Linear regression curve to fit a line on RSS values of *primary* versus a new phone for all common BSSIDs at 5 locations.

Storing and using the median of these logged values, helps in eliminating the variation in data and the possibility of error in RSS values arising from the multi-path effects. This is repeated for n_2 BSSIDs at location 2. Finally, we have training data with $n_1 \cup n_2$ BSSIDs and their respective median RSS values at the two locations. If a BSSID is not seen at a location, it's value is set to infinity.

Now, in order to predict the location (i.e. either location 1 or 2) for an unknown test log during the testing phase, CAIL stores the median RSS value for the testing location corresponding to each BSSID in the test data. Only the BSSIDs common to training and testing log are considered.

For a common BSSID j, the median RSS value of test data is compared to median RSS values of the BSSID j at locations 1 and 2. The closer of the two locations is considered the predicted result for BSSID j and placed in result set R. This is done for all common BSSIDs, say m at all locations. So we have a result set R with m values that are either location 1 or 2. The final predicted location is the mode of all m locations in R.

The same approach can be easily scaled to a training phase with more than 2 locations and predicting locations of multiple test logs.

The scope of CAIL is limited to predict locations that were logged during fingerprinting in training phase. Prediction of locations that were never sampled during training phase is out of the scope of this paper.

B. CAIL Cross-calibration Algorithm

It is not necessary that the entire training data for a mobile phone is always available. Constructing such a data is time-consuming and practically infeasible. Instead, one can use *primary* phone for which the entire training data is available, to create a fingerprint that can be used by other phones to find their location. However, two phones at the same location may not see the same RSS values due to a difference in chip configurations, resulting in a signal reception bias. The accuracy of location prediction is very poor, if testing logs and training logs are not from the same phone, as later described in section IV.C and shown in Figure 5(c). This gives rise to the need of cross-calibration of a new phone entering the environment against the *primary* phone, to find a relationship between the RSS values seen by the two different phones. For a new phone, CAIL uses a minimum number of training locations to cross-calibrate. Since data from the new phone only includes RSS values from some selected locations, the RSS values for the other locations need to be derived using values from *primary* phone. For this, linear regression is applied to RSS values of *primary* phone and new phone to find the function to map RSS values of the new phone to those of *primary* phone.

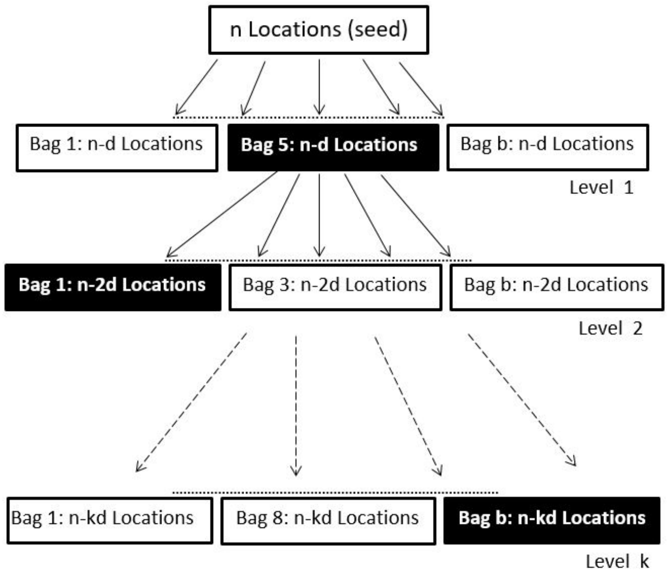


Fig. 2: Recursive bagging demonstrating the selection of best bags at every level, highlighted in black. Every bag at a level is passed through cross-calibration and localization algorithms to determine their accuracy. The best bag selected at a level is then used for generating another set of b bags for the next level, by eliminating d locations. This process converges when the bag size is less than, or equal to d.

In CAIL cross-calibration algorithm, median RSS values for every BSSID logged in the training phase for *primary* phone are stored in array T, and for new phone it is stored in array U. The common BSSIDs between two arrays are retained while the others are discarded from the arrays. Now, linear regression is carried out between T and U. The linear line obtained, as shown in Figure 1, provides the slope and intercept values, which are used for mapping between the *primary* phone and the new phone. If a location p, BSSID j is logged only for the *primary* phone then, RSS cross-calibrated value for new phone, RSS_new_{pj}, is obtained from the corresponding RSS value of the *primary* phone, RSS_pri_{pj}, using the equation:

$$RSS\_pri_{\text{pj}} = m * RSS\_new_{\text{pj}} + c \quad (1)$$

where, m is the slope, c is the intercept obtained from linear regression and RSS_pri_{pj}, RSS_new_{pj} are the median RSS values for *primary* phone, new phone respectively, against BSSID j, location p. This helps in generating a complete set of RSS values for the new phone by mapping values of *primary* phone for the locations for which new phone data is not logged. When test RSS values are compared to this set using the localization algorithm, the accuracy achieved is as good as when all RSS values for new phone are available through logging, as seen in section IV.D.

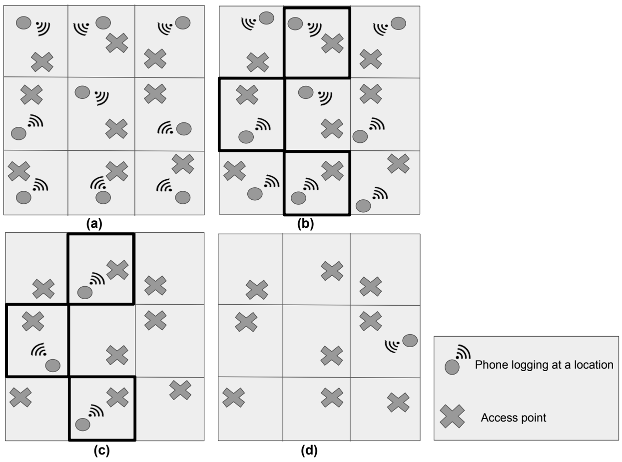


Fig. 3: Deploying CAIL on a hypothetical floor with 9 locations where (a) *primary* phone logs the entire floor (b) *secondary* phone logs the floor and provides the 3 best locations, highlighted as black squares, using recursive bagging (c) *evaluation* phone only logs at the best locations identified and the RSS values for the rest of the locations is obtained by cross-calibration against the *primary* phone (d) *evaluation* phone logs at an unknown location and obtains its location using the CAIL localization algorithm.

C. CAIL Recursive Bagging: Selecting a Subset of Locations for Cross-calibration

Finding the minimum number of "good" training locations for cross-calibration is a challenging task. To identify the good locations for cross-calibration, we employ a technique of recursive bagging on a phone called *secondary* phone. This phone provides the best locations to log in an environment. We have shown in section IV.D that the locations identified as good, for the secondary phone can be used for any other phone, say *evaluation* phone, to produce satisfactory accuracy. CAIL recursive bagging algorithm starts with a set or bag of locations of a size say, S as seed. From this, some locations are randomly picked without replacement, to generate a new bags of size S'. b such bags are generated. These b bags form the Level 1 of the algorithm. Algorithm picks one of the b bags and uses the locations of selected bag as an input to the CAIL cross-calibration algorithm described in III.B. This generates the RSS values for the locations that are not present in the bag and creates the complete RSS fingerprint. The complete RSS fingerprint is used to predict the test log location using the CAIL localization algorithm described in III.A. This is repeated for all the b bags. The bag that gives the best accuracy at level 1, becomes the new set of locations from which the next level of b bags can be created. This recursive process reduces the size of the bag by a fixed amount d at every level and converges when the size becomes equal to or less than d.

Randomization in generating b bags at each level reduces the effort to find the best bag. This is done by choosing b, such that the there is a sufficient representation of all locations at a level, among the b bags in the next level. Further, the parameters b and d along with the seed bag size, S can be adjusted to speed up the process. Thus, the technique can be applied to even larger environments with more locations without an increase in cost. On the other hand, brute force to find a bag of similar good locations in such a case would lead to an exponential effort. For our problem, b was chosen to be 10, d was 5 and the seed bag size was 30. So at each level, 10 random bags of 5 fewer locations than the previous level are created. Figure 2 describes this technique.

Model	CPU
Motorola Moto E XT1022	Dual-core 1.2 GHz Cortex-A7
Samsung S6802 Galaxy Ace Duos	Broadcom BCM21553 832 MHz
Samsung 19300 Galaxy S III	Quad-core 1.4 GHz Cortex-A9

TABLE I: Configuration of mobile phones used in the experiments.

IV. Evaluation

For evaluation of CAIL, a total of 3 phones have been used: *primary* phone for creating RSS fingerprint; *secondary* phone for finding the best locations, which is 5 in our case; *evaluation* phone, for testing the efficacy of locations identified by *secondary* phone and finding the accuracy of indoor localization. Figure 3 describes the CAIL indoor localization system for a hypothetical floor with 9 locations, illustrating how any new phone leverages the efforts of a *primary* phone to generate its RSS fingerprint, and finally gets a prediction for an unknown location.

Section IV.B, IV.C, and IV.D detail the performance of CAIL for the three phones. The positional error is described in blocks. These blocks are mapped to a size of 7x8 feet^2. This size could vary depending upon the size of the environment being logged and the precision required. For our test environment, this block size worked well. Since CAIL's target applications are based on environments such as malls, classrooms, we haven't described the error explicitly in terms of distance. Our use-cases include attendance application, that will require the student to be present in the classroom, irrespective of where the student is sitting in the classroom.

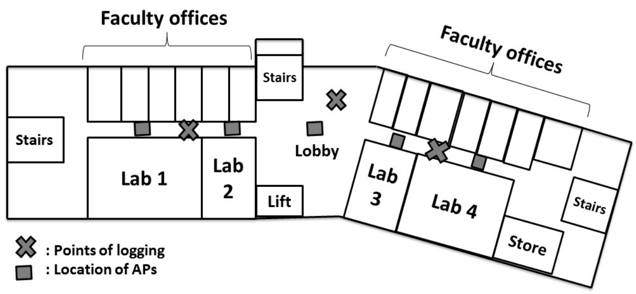
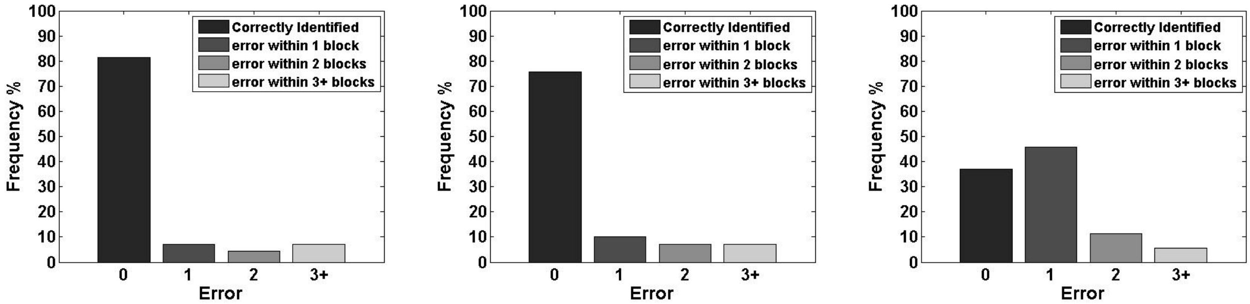


Fig. 4: Floor map of second floor R&D wings with AP and logging locations.

A. Collection of Data

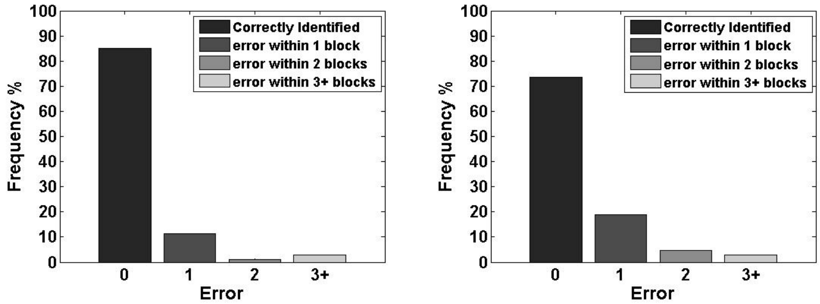
The data was collected for 30 uniformly distributed locations inside the Academic Block, the main building of IIIT-Delhi. The specifications of the 3 mobile phones used are tabulated in Table I.

Locations included places, such as canteen, classrooms, and offices that have furniture, such as chairs and desks, to open areas like corridors and lobbies. The building contains APs catering to a 1000+ student body, administrative staff, and faculty. These locations were uniformly distributed across the six floors of the building. Figure 4 illustrates the floor map of the second floor R&D wings of the building. The logging varied from 10 AM in the morning to 11 PM at



(a) (b) (c)

Fig. 5: Bar graph showing the percentage of locations correctly identified and those predicted within a few blocks of the original location for a) $secondary$ phone intra-phone experiment b) $secondary$ phone inter-phone experiment c) $secondary$ phone without cross-calibration against $primary$ phone; a block is approximately 7x8 feet^2.



(a) (b)

Fig. 6: Bar graph showing the percentage of locations correctly identified and those predicted within a few blocks of the original location for a) $evaluation$ phone intra-phone experiment b) $evaluation$ phone inter-phone experiment; a block is approximately 7x8 feet^2.

Phone	Accuracy for bag of size 5	Effort in logging	Accuracy of bag of size 10	Effort in logging	Accuracy of bag of size 15	Effort in logging	Baseline Accuracy	Effort in logging
$secondary$	75.71	25 min	68.57	50 min	65.71	75 min	81.43	150 min
$evaluation$	73.58	25 min	67.92	50 min	66.03	75 min	84.90	150 min

TABLE II: Accuracy obtained on recursive bagging on $secondary$ phone and using these locations for $evaluation$ phone for cross-calibration.

night. The RSS values in dBm were logged at a frequency of $1/6$ Hz, i.e., 1 WiFi scan per 6 seconds, multiple times over a period of 3 months at all the locations. This is done to eliminate any variations in RSS signal arising due to changes in the environment, from factors like number of people in a location at any given time, furniture, or time of the day. These variations are eliminated by storing only the median RSS values. Each data log collected at a location spanned 5 minutes for the training set and 2 minutes for the testing set. The data contained 106 testing data samples for $evaluation$ phone and 70 for $secondary$ phone, logged at the 30 locations, at different times, over multiple days.

B. Intra-phone Experiments

Using the CAIL localization algorithm, the baseline accuracy was calculated for the $secondary$ phone by running the testing data of the $secondary$ phone against its own complete fingerprint of 30 locations. A similar baseline accuracy was calculated for the $evaluation$ phone separately. The accuracy for both phones was quite high with 81% of the testing locations correctly identified in $secondary$ phone and 85% in the $evaluation$ phone.

C. Inter-phone Experiments

Inter-phone experiments involved running testing data of $secondary$ and $evaluation$ phones against training data of $primary$ phone. To achieve the same accuracy, albeit with minimal training effort, the cross-calibration algorithm used 5 locations, a subset of the total 30 locations, as input. These 5 locations were obtained after recursive bagging of the training data for the $secondary$ phone. The linear curve derived from cross-calibration between RSS values of 5 locations of $primary$ and $secondary$ phones was used to derive the remaining for $secondary$ phone from those of $primary$ phone. Next, CAIL localization algorithm was deployed on $secondary$ phone using the testing data. The accuracy after cross-calibration on $secondary$ phone was found to be 76% shown in Figure 5(b). This was comparable to the accuracy of 81% obtained with a complete logged fingerprint for 30 locations as shown in Figure 5(a). Further, this is much higher than 37% accuracy when test data of $secondary$ phone is tested against training data of $primary$ phone without cross-calibration as shown in Figure 5(c). This shows that the cross-calibration eliminates the signal reception bias across different phones.

D. Results on a New Mobile Phone

To test the efficacy of the subset of locations achieved by recursive bagging of $secondary$ phone, the inter-phone experiment was carried out on the $evaluation$ phone using training logs at the subset of locations (15, 10 and 5). The $evaluation$ phone's accuracy of localization using 5 locations derived for $secondary$ phone, was 74% as shown in Figure 6(b). The accuracy was 85% when tested with its own complete fingerprint of 30 locations as shown in Figure 6(a). This shows that the 5 locations obtained for one phone are indeed sufficient to make an accurate location prediction, independent of the phone. Thus, we do not require the entire fingerprint of a new phone to achieve good accuracy, greatly diminishing the task of data collection on new phones.

V. Results and Observations

The intra-phone experiments yielded results with an accuracy of more than 80%. The inter-phone experiment on $secondary$ phone yielded a comparable accuracy of 70% on an average after cross-calibration using subsets of locations. Furthermore, when applied to $evaluation$ phone, the same locations gave accuracy as high as 74%. The data and time required for logging 5 locations is 1/6$^{\text{th}}$ that of logging for the entire building of 30 locations. A reduction from a total of 150 minutes to merely 25 minutes has been achieved, as tabulated in Table II. The accuracy given in the table are the values when the prediction was on spot. A detailed report of the error in prediction for cross-calibration for $secondary$ phone is given in Figure 5(b) and for $evaluation$ phone in Figure 6(b). The error is represented in terms of blocks of separation between actual and predicted location. The effects of recursive bagging are clearly seen on $secondary$ phone results. As expected, the accuracy increases as we downsample locations from 15 to 5. Removal of outliers at each downsampling leads to a better set of locations for cross-calibration, giving highest accuracy for the cross-calibration using only 5 locations. The locations chosen using recursive bagging on $secondary$ phone yielded an accuracy of 74% on $evaluation$ phone. This reinforces the existence of "bad" locations for WiFi logging that must be avoided and certain "good" locations that work for most of the phones. Dynamic changes in the environment, such as nearby moving objects may contribute to accuracy drop that is largely handled by periodic logging and storing the median RSS values. Thus, for a drop of 84% in work, there is just a drop of 10% in accuracy.

This approach when scaled to enterprises with larger environments and hundreds of cross-calibrated phones would lead to a huge cut-down of effort while maintaining a minor accuracy drop among all phones.

VI. Conclusion

In this paper we presented CAIL, a novel approach to significantly reduce the effort required for indoor localization using existing WiFi infrastructure. The users of CAIL can easily leverage the one-time fingerprinting done by a single phone and almost instantly use the system for location prediction, irrespective of the mobile phone model they carry. Unlike traditional RF fingerprinting approaches, CAIL does not require any pre-deployment effort or constraints extraneous to the phone. It identifies the best locations to log at, using recursive bagging and employs cross-calibration to map a new phone against the phone used for site survey. Both cross-calibration and recursive bagging modules of CAIL can be used with any existing localization algorithm that uses RF, making CAIL a modular system. We have evaluated the proposed algorithm on data logged from user's phone at just 5 locations in a building and it has shown promising results with an accuracy of around 76%, comparable to 81% accuracy on complete site survey.

Future work will extend the localization algorithm to predict locations that were not logged during the training phase, using interpolation based techniques to generate RSS values for these locations. We will also research other ways of identifying the good locations for cross-calibration. These could be locations showing less variation in RSS values, and those with a larger density of APs.

References

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

[2] Y. Gwon and R. Jain, "Error characteristics and calibration-free techniques for wireless lan-based location estimation," in *Proceedings of the second international workshop on Mobility management & wireless access protocols*. ACM, 2004, pp. 2–9.

[3] H. Lim, L.-C. Kung, J. Hou, and H. Luo, "Zero-configuration, robust indoor localization: Theory and experimentation," in *Proceedings IEEE INFOCOM 2006. 25TH IEEE International Conference on Computer Communications*.

[4] T. M. Mitchell, *Machine Learning*, 1st ed. New York, NY, USA: McGraw-Hill, Inc., 1997.

[5] H. Lim, L.-C. Kung, J. C. Hou, and H. Luo, "Zero-configuration indoor localization over ieee 802.11 wireless infrastructure," *Wireless Networks*, vol. 16, no. 2, pp. 405–420, 2010.

[6] Y. Ji, S. Biaz, S. Pandey, and P. Agrawal, "Ariadne: a dynamic indoor signal map construction and localization system," in *Proceedings of the 4th international conference on Mobile systems, applications and services*. ACM, 2006, pp. 151–164.

[7] K. Chintalapudi, A. Padmanabha Iyer, and V. N. Padmanabhan, "Indoor localization without the pain," in *Proceedings of the sixteenth annual international conference on Mobile computing and networking*. ACM, 2010, pp. 173–184.

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

[9] A. LaMarca, Y. Chawathe, S. Consolvo, J. Hightower, I. Smith, J. Scott, T. Sohn, J. Howard, J. Hughes, F. Potter *et al.*, "Place lab: Device positioning using radio beacons in the wild," in *International Conference on Pervasive Computing*. Springer, 2005, pp. 116–133.

[10] W. G. Griswold, P. Shanahan, S. W. Brown, R. Boyer, M. Ratto, R. B. Shapiro, and T. M. Truong, "Activecampus: experiments in community-oriented ubiquitous computing," *Computer*, vol. 37, no. 10, pp. 73–81, 2004.

[11] M. Azizyan, I. Constandache, and R. Roy Choudhury, "Surroundsense: mobile phone localization via ambience fingerprinting," in *Proceedings of the 15th annual international conference on Mobile computing and networking*. ACM, 2009, pp. 261–272.

[12] H. Liu, J. Yang, S. Sidhom, Y. Wang, Y. Chen, and F. Ye, "Accurate wifi based localization for smartphones using peer assistance," *IEEE Transactions on Mobile Computing*, vol. 13, no. 10, pp. 2199–2214, 2014.

[13] P. Bahl, J. Padhye, L. Ravindranath, M. Singh, A. Wolman, and B. Zill, "Dair: A framework for managing enterprise wireless networks using desktop infrastructure." HotNets, 2005.

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

[15] C. Wu, Z. Yang, Y. Liu, and W. Xi, "Will: Wireless indoor localization without site survey," in *INFOCOM, 2012 Proceedings IEEE*. IEEE, 2012, pp. 64–72.