Identifying Value in Crowdsourced Wireless Signal Measurements

Zhijing Li, Ana Nika, Xinyi Zhang, Yanzi Zhu, Yuanshun Yao, Ben Y. Zhao, Haitao Zheng
Dept. of Computer Science, UC Santa Barbara
Goleta, CA, USA
{zhijing, anika, xyzhang, yanzi, yao, ravenben, htzheng}@cs.ucsb.edu

ABSTRACT

While crowdsourcing is an attractive approach to collect large-scale wireless measurements, understanding the quality and variance of the resulting data is difficult. Our work analyzes the quality of crowdsourced cellular signal measurements in the context of basestation localization, using large international public datasets (419M signal measurements and ∼1M cells) and corresponding ground truth values. Performing localization using raw received signal strength (RSS) data produces poor results and very high variance. Applying supervised learning improves results moderately, but variance remains high. Instead, we propose *feature clustering*, a novel application of unsupervised learning to detect hidden correlation between measurement instances, their features, and localization accuracy. Our results identify RSS standard deviation and RSS-weighted dispersion mean as key features that correlate with highly predictive measurement samples for both sparse and dense measurements respectively. Finally, we show how optimizing crowdsourcing measurements for these two features dramatically improves localization accuracy and reduces variance.

CCS Concepts

•Networks → Network measurement;

Keywords

Crowdsourcing; transmitter localization; measurement quality

1. INTRODUCTION

As wireless networks continue to grow in size and coverage, network monitoring and management is becoming an increasingly costly and resource intensive task [29]. While it used to be a standard practice to measure wireless performance by covering an area with vehicles and specialized equipment, that is simply impractical today. Instead, companies and research firms are turning to crowdsourcing as a cheap and scalable way to perform network measurements at scale [1].

But just how reliable are these user-contributed measurements? There are obvious reasons to doubt the accuracy and the consistency of user-contributed wireless network measurements. First, unlike specialized measurement tools deployed by network providers, user-contributed measurements tend to be generated using commodity equipment with less accuracy. Second, users are often less tech-savvy, and more likely to introduce errors during operation or through user contexts (*e.g.* driving, phone in pocket). Third, crowdsourced measurements are constrained by the mobility patterns of contributing users. Therefore, measurements will follow user mobility, and are likely uneven in coverage.

With this in mind, it is critical for network providers to understand the value and limitations in crowdsourced network measurements. While crowdsourced measurements can be used for a number of management functions (*e.g.* network performance and coverage measurements [39, 19, 13], transmitter localization and radio map construction [37, 27, 2, 3], spectrum anomaly detection [36]), they are generally not amenable to quantitative analyses, because of the dearth of both measurement data and ground truth datasets.

In this work, we are taking a data-driven, quantitative approach to answering some of these questions, by focusing on the specific application of *basestation localization*. Basestation or transmitter localization is a basic operation in wireless network management, and critical to providers interested in locating misbehaving transmitters or mapping out potential holes in basestation coverage. Besides, nowadays many mobile applications rely on cell tower triangulation to determine user position [4] for lower energy consumption than GPS. However, the public sources of cell tower location are incomplete and inaccurate [2, 5]. Like other management applications, basestation localization uses received signal strength (RSS) measurements gathered by mobile devices. Unlike other applications, analyzing localization performance is tractable today, given the availability of both crowdsourced RSS datasets and ground-truth data on basestation locations.

We are interested in answering several critical questions about user-contributed signal measurements. *First*, how accurately can we locate wireless basestations using RSS measurements and known algorithms, and does accuracy correlate strongly with intuitive properties such as number or density of measurements? *Second*, can machine learning classifiers help improve location accuracy? *Third*, can we develop techniques to identify features or properties of highly accurate measurement instances, and use them to build techniques that produce more accurate results?

Our study uses several large public datasets of crowdsourced RSS measurements gathered by user smartphone apps around the globe through the OpenCellID [2] and OpenBMap [3] projects. They are unique for two reasons: they provide raw signal measurements (compared to aggregate coverage maps), and include ground truth of real basestation locations. In total, we analyze ∼1M cells and 419M signal measurements. Using the ground truth data and

ACM 978-1-4503-4913-0/17/04.
http://dx.doi.org/10.1145/3038912.3052563

	# of Measurements				# of Cells			
	Germany	Poland	Russia	USA	Germany	Poland	Russia	USA
OpenCellID	390M	7.9M	4.7M	15.1M	564K	87K	157K	146K
OpenCellID-GT	13.4M	2.9M	109K	0	10.6K	21.6K	3.5K	0
OpenBMap	1.2M	58K	12.4K	317K	32.4K	1.9K	991	5.2K
OpenBMap-GT	36.6K	8.2K	1.3K	0	773	294	55	0

Table 1: High-level summary of OpenCellID and OpenBMap datasets. Each cell here is uniquely defined by its Cell ID.

existing localization algorithms, we first quantify the predictive quality of crowdsourced data, *i.e.* how accurately can each measurement instance predict the basestation location? We then try to identify and improve the poor localization results by applying supervised learning. Finally, we try to identify key properties of measurement instances that correlate well with localization accuracy, by taking a novel application of unsupervised learning technique we call feature clustering.

We summarize our findings as follows:

- We apply seven popular basestation localization algorithms to our ground truth datasets, and find that localization results have very high variance across a number of factors, including algorithms, datasets, and scenarios. In addition, there is a significant variance in error even across cell instances in the same dataset.
- We apply ML classifiers to improve localization accuracy. While overall accuracy is higher, error variance remains high, and our attempts to find key impactful features produce no clear results.
- We then take a novel application of unsupervised learning to identify hidden correlations in the data, which we call *feature clustering*. We define a distance metric between measurement instances based on similarity of their values in key features. Clustering the entire dataset based on pairwise distances produces key clusters that correlate features with localization accuracy of data inside them. From this, we identify RSS standard deviation and RSS-weighted dispersion mean as independent features that identify highly predictive data instances for sparse and dense measurement datasets.
- Finally, we develop an adaptive crowdsourcing technique using these two features. Applying this technique produces dramatic improvements in both increased localization accuracy and reduced variance. We also show that our results could generalize across datasets and geographic regions.

2. DATASETS

Among various public datasets on crowdsourced cellular measurements [2, 3, 6, 7, 8], we use OpenCellID [2] and OpenBMap [3], for our analysis. These two datasets offer raw signal measurements, while the other ones only provide aggregated coverage maps.

OpenCellID. Created to maintain a global database of cellular basestations (identified by their Cell IDs), this dataset was collected by volunteers running a smartphone app that records information of their cellular connections. Each data entry is a single measurement at a particular time and location, containing information on the basestation (country, provider, Cell ID, network type) and the signal (timestamp, GPS, RSS). No user ID is included in any entry.

We group the data by country and select four countries for our analysis (Germany, Poland, Russia and USA). We pick the first three since their datasets come with the ground-truth locations of a portion of the basestations (provided by cellular service providers). We select US because it is similar to Poland in data volume. Together, they form our OpenCellID dataset, including 418M measurements and 954K cell IDs. Later in §3, we use this dataset to identify key characteristics of crowdsourced measurements.

	Weekly		**Monthly**		**Yearly**	
	GSM	**UMTS**	**GSM**	**UMTS**	**GSM**	**UMTS**
OpenCellID	16.7M	1.8M	5.4M	1.2M	725K	481K
OpenCellID-GT	883K	14.8K	292K	10.4K	44.8K	5.5K
OpenBMap	18.3K	52.9K	15.5K	43.5K	11.2K	28.3K
OpenBMap-GT	2K	201	1.6K	171	1K	109

Table 2: Per-cell crowdsourcing instances generated from the four signal measurement datasets in Table 1.

We also create a smaller dataset OpenCellID-GT. It is a subset of OpenCellID and contains only measurements on cells with ground-truth basestation locations. The dataset includes 16.4M measurements and 35.7K cell IDs. We use it to study crowd-based basestation localization (§4, §5). Table 1 summarizes the datasets in terms of the number of measurements and cells covered.

OpenBMap. This dataset is similar to OpenCellID, but significantly smaller in size (4% of OpenCellID). Its data entry has a similar field but no ground-truth basestation locations. We will use it as a secondary dataset to verify our analysis on OpenCellID. Specifically, we consider the OpenBMap data for Germany, Poland, Russia, and USA, in 2014 and 2015. We created two datasets, OpenBMap with 1.6M measurements and 40.5K cells, and OpenBMap-GT with 46K measurements and 1.1K cell IDs. For the latter, we search for the ground-truth basestation locations from OpenCellID-GT based on their unique Cell IDs.

Per-cell Crowdsourcing Instances. From each dataset, we create *crowdsourcing instances* for each cell ID over different time windows (week, month, and year). For each window size, we partition the 2-year data into individual instances for each cell, and remove the empty instances. As a result, each cell will have multiple crowdsourcing instances for a given window size, *i.e.* up to 104 weekly instances, 24 monthly instances, and 2 yearly instances. We also group instances based on their network type (GSM, UMTS[1], LTE, CDMA etc). We find that GSM and UMTS cells dominate in both the OpenCellID (99%) and OpenBMap (95%) datasets. Table 2 summarizes the number of instances for these four datasets. The vast majority of basestations with ground-truth locations are GSM based, *i.e.* 88%-98.8% for OpenCellID-GT and 90% for OpenBMap-GT.

Google Basestation Location Database. We use Google's basestation location database as a reference for our localization analysis. Since 2008, Google has been collecting CellID-GPS pairs for its location-aware services [9, 10]. Also using crowdsourced measurements, they estimate each basestation (identified by the Cell ID) location as the centroid of its measurements [11]. Each estimate comes with an accuracy value ranging from 500m to 5000m, but the metric is undefined. Leveraging Google's Map Geolocation API [12], we crawled the estimated basestation locations for all the cells in OpenCellID-GT and their accuracy level.

3. INITIAL ANALYSIS

We analyze our datasets to identify key properties of crowdsourced cellular measurements. We examine and compare the datasets on measurement count, spatial, and RSS statistics of per-cell measurements. We also present and contrast key results observed on OpenCellID[2] and OpenBMap, as well as consistency of results across countries and between GSM and UMTS cells.

[1] OpenCellID defines UMTS to include UMTS, HSPA and HSPA+.

[2] While OpenCellID-GT is a small subset of OpenCellID, our analysis shows that its structure properties are completely identical to those of OpenCellID (results omitted for brevity).

Localization Methods		Estimated Basestation Location
Non-RSS	Centroid (C) [15]	Geometric center of all the measurements
	Minimum Enclosing Circle (MEC) [31]	Center of the minimum enclosing circle of all the measurements
RSS-based	Weighted Centroid (WC) [15]	RSS-weighted geometric center of all the measurements
	Highest RSS [45]	Location of the measurement with the strongest RSS value
	Model-based [28, 17]	Location of the strongest RSS predicted by the calibrated propagation model
	Grid-based [35]	Center of the grid with the highest likelihood of RSS to be the strongest RSS
	Ecolocation [49]	Location with the highest value on the statistical RSS-distance relationship heatmap

Table 3: Summary of seven commonly-used base station localization methods.

Measurement Count. The number of measurements varies significantly across cells and across instances of each cell (between 1 and 10000), where each instance is a collection of measurements taken over some time window. The majority of cells have a small number of measurements – even over a year, more than 50% of cell instances have less than 20 measurements (for GSM) and 10 (for UMTS). Across countries, cells in Germany tend to have more measurements, and those in Russia have much fewer.

Spatial Distribution. To understand the spatial layout of measurements in each cell, we consider several widely used metrics [40]: average pairwise distance between measurements, diameter, dispersion (the spread of measurements around their center) standard deviation [3], and index of dispersion that quantifies the existence of clusters. For these metrics as well, we observe significant variance across cell instances. For GSM, the diameter ranges from $5mm$ (*i.e.* measurements from a single stationary user) to $68km$, while the dispersion is between $0.1km$ (measurements are near the center) and $10km$ (measurements are widely scattered and form irregular shapes). While these spatial metrics are highly correlated, they have low correlation with the measurement count $(0.02 - 0.1)$. For UMTS, while the cell size is smaller, these spatial metrics still vary widely across cell instances.

Across countries, the cells in Germany tend to have larger diameter, higher dispersion, and more clustered cells than others. For example, more than 55% of GSM cell instances have diameters larger than $4km$, which reduces to 6%, 20% and 29% for Russia, US, and Poland, respectively.

RSS. Across all measurements, RSS values were evenly distributed between (-112dBm, -51dBm[4]) for GSM and (-120dBm, -60dBm) for UMTS, and distributions were similar across four countries. Per-cell mean RSS and RSS standard deviation values both vary widely across cell instances.

Intuitively, RSS values should correlate inversely with distance to the basestation. We test this hypothesis using ground-truth basestation locations in OpenCellID-GT. Since basestations are generally configured with the same transmit power, we look for this relationship using the Pearson correlation coefficient γ. Ideally, γ should be close to -1. Instead, a large portion of the cells (50% for Germany, 40% for Poland and Russia) display weak correlation $(-0.5 < \gamma < 0.5)$, while 10% of cells in Poland and Russia even display strong positive correlation $(\gamma > 0.5)$. This highly unpredictable relationship between RSS and distance to basestation is somewhat expected in crowdsourced measurements, since so many other factors can have strong impact on RSS values. This underscores the level of randomness present in crowdsourced measurements, and is a key reason why these datasets are less useful than controlled datasets.

User Context. 22% of OpenCellID measurements contain information on moving speed and phone direction. Our analysis on these data shows that the vast majority of measurements came from moving users. For Germany and Poland, many users were traveling at high speeds (in vehicles). The reported phone directions were uniformly distributed. Finally, 3% of measurements report estimated GPS error but the data volume is too small to offer representative results.

OpenBMap vs. OpenCellID. While much smaller in data volume, the OpenBMap datasets have similar per-cell characteristics as those of OpenCellID. The key difference is that for Germany, the measurement diameter and average pairwise distance are much smaller than those in the OpenCellID Germany datasets. As a result, the spatial properties in OpenBMaps become much more consistent across the four countries.

4. LOCALIZATION PERFORMANCE

We now examine whether crowdsourced signal measurements can be used to accurately locate basestations. Our analysis uses the OpenCellID-GT dataset, which we have shown to closely mimic OpenCellID in terms of structural characteristics. We also use the OpenBMap and Google datasets to validate our findings.

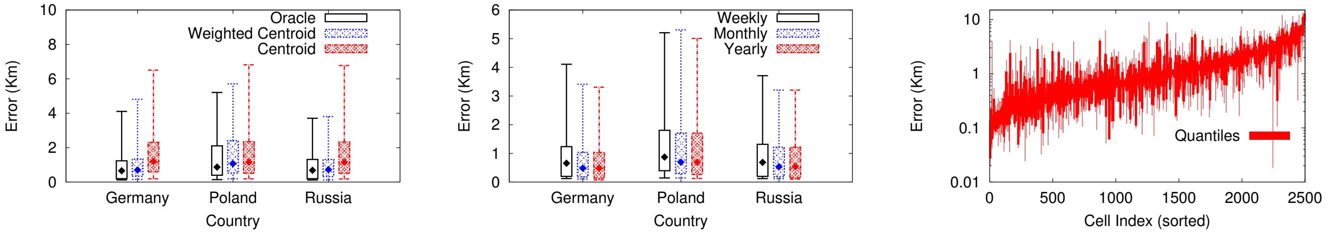
We consider seven commonly known transmitter localization algorithms, summarized in Table 3. Two methods (Centroid and MEC) only use spatial data, *i.e.* measurement location, and the rest five methods use both spatial and RSS data. We also consider the "Oracle" method, which, for each crowdsourcing instance, outputs the best localization result across the seven algorithms. It provides the upper bound on localization accuracy assuming one can always pick the best localization method for a given instance. We apply these algorithms on the OpenCellID-GT dataset, focusing on cell instances with at least three valid RSS measurements. We evaluate these algorithms in terms of the localization error, *i.e.* the distance between the estimated and ground-truth basestation locations. We separate our analysis for GSM and UMTS cell instances, but find that they lead to consistent conclusions. For brevity, we only show the results for GSM cells since they dominate the dataset.

4.1 Key Results

Across the seven algorithms, we found that no single algorithm is consistently the best, but Weighted Centroid is more likely to be the best. As an example, Figure 1(a) plots the distribution quantiles (5%, 25%, 50%, 75%, 95%) of the localization error based on weekly measurements, where Weighted Centroid already closely approximates "Oracle" (the best of the seven algorithms). We also observe that for 67% of the cases, RSS methods outperform non-RSS methods. As shown by Figure 1(a), Centroid has a much longer tail than Weighted Centroid. This means that RSS data does help localization but must be handled carefully.

Large Variance across Cell Instances. The most significant observation from our analysis is that for all the localization algorithms

[3] The standard deviation of the distance between measurement point and their centroid center, a commonly used dispersion metric.
[4] The RSS value is capped by -51dBm in all GSM measurements.

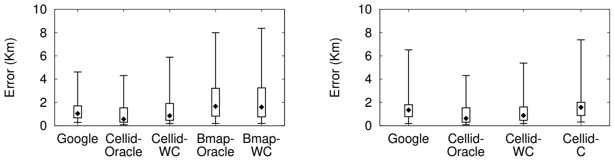


(a) Weekly, Different Algorithms

(b) Oracle, Different Time Windows

(c) Distribution across Instances per Cell

Figure 1: (a) Comparing Centroid, Weighted Centroid and Oracle (best of 7 algorithms), in terms of quantiles (5%, 25%, 50%, 75%, 95%) of the localization error distributions. (b) The localization error of Oracle for different time window sizes. (c) Localization error distribution across crowdsourcing instances for each individual cell.



(a) Cells in OpenBMap-GT

(b) Cells in OpenCellID-GT

Figure 2: Validation using the secondary datasets (OpenBMap and Google).

including “Oracle”, the localization error varies significantly across crowdsourcing cell instances. For example, Figure 1(b) plots the quantiles of the localization error of “Oracle”, for weekly, monthly, and yearly cell instances. The localization error varies significantly between $0.01km$ and $6km$. The gap between the 75- and 95-percentile values is particularly large, often more than 6x larger than the median error value. The same applies to the other seven localization algorithms.

To understand whether the good (or poor) performance “sticks” to individual cells, we study localization error distribution across each cell’s weekly crowdsourcing instances (for cells with at least 4 weekly instances). Figure 1(c) plots the quantiles of each cell’s localization error, sorted by the median value. The error varies significantly across crowdsourcing instances, often by more than a factor of 10. The same applies to monthly and yearly instances (results omitted for brevity).

Together, these results show that when localizing basestations using crowdsourced signal measurements, the performance varies significantly across cells and crowdsourcing instances within each cell. Such significant variance translates into large, undesirable uncertainty in localization accuracy.

4.2 Validation via Secondary Datasets

We validate our observations using the OpenBMap and Google datasets. First, we consider the Cell IDs covered by OpenBMap-GT, and use its 2-year crowdsourcing measurements to perform basestation localization. Then, for the same set of Cell IDs, we perform localization using the 2-year data in OpenCellID-GT and also find the estimated basestation location from Google’s crowdsourced basestation localization database. For all three localization outcomes, we compare them to the ground-truth and derive the corresponding localization errors. Figure 2(a) plots the quantiles of the localization error across different datasets. While the localization accuracy varies across the datasets and algorithms, they all display significant variance across Cell IDs.

Next, we compare Google and OpenCellID by using the Cell IDs covered by OpenCellID-GT. This is a larger dataset with 23.3K Cell IDs. Figure 2(b) compares the per-Cell ID localization error for Google, and OpenCellID (Oracle, Weighted Centroid, and Centroid). The Google’s result is on par with OpenCellID Centroid. But overall, they again display significant variance across Cell IDs.

5. PREDICTING LOCALIZATION ACCURACY VIA CLASSIFICATION

We have observed significant variance in crowdsourced localization errors. For crowdsourced measurements to be useful, we must find techniques to distinguish “predictive” quality measurement samples from others, where “predictive” is the ability to produce localization values with low error. Our goal is to answer the question: *can we develop techniques to identify the predictive ability of crowdsourced measurement samples, and what if any “features” can help*? The most intuitive feature is the number of measurements in each cell instance, *i.e.* the more the measurements, the better the prediction. However, we confirmed that measurement count is not a reliable metric, and shows no detectable relationship to localization accuracy with weak Pearson correlations [-0.06, 0.02] across different scenarios and datasets.

In this section, we search for good indicators of a instance’s prediction accuracy by applying machine learning classifiers using features of the crowdsourced measurement data. Not only do we seek to develop tools to identify predictive instances, but we also wish to identify and understand the key features associated with accurate measurement data samples. We focus on Weighted Centroid as the localization algorithm since it performs the best in most scenarios.

5.1 Feature Selection and Training

The complex structure of crowdsourced data means that it is unlikely that the localization accuracy is controlled by a single property. Thus we consider a classifier-based approach. For a given localization accuracy requirement, *e.g.* the localization error $< x$, we seek to predict whether a crowdsourcing instance can produce localization results meeting such requirement, while identifying the key features that lead to such good (or poor) performance.

Feature Extraction. We build four categories of features to characterize the crowdsourced datasets: *spatial*, *RSS*, *localization algorithm*, and *combined RSS and spatial*. The spatial features are those used by common spatial analysis [40]. The RSS features represent the statistical distribution of the RSS within each cell. The algorithm features look at the difference between results of different localization algorithms. And the combined RSS and spatial features capture the joint distribution of RSS and spatial properties,

Dimension	Feature Class	Details
Spatial	measurement count	
	diameter	
	clustering: index of dispersion [18]	
	clustering: nearest-neighbor index	
	minimum enclosing circle radius	
	dispersion	max, min, median, mean, StdDev, coefficient of variance
	angular coverage	
	standard deviational ellipse [51]	StdDev(major), StdDev(minor), StdDev(major)/StdDev(minor)
RSS	RSS	max, min, median, mean, StdDev, coefficient of variance
	RSS (power level, dB)	
	% of RSS (power level)> γ	γ=-55, -60, -65, -70 and -80dB
	# of RSS (power level)> γ	
Algorithm	distance between algorithm M and N's location estimates	
RSS-Spatial	RSS-weighted dispersion	max, min, median, mean, StdDev, coefficient of variance
	RSS-weighted standard deviational ellipse	StdDev(major), StdDev(minor), StdDev(major)/StdDev(minor)
	correlation between measurement distance to center and RSS	
	spatial autocorrelation [20]	
	estimated path loss exponent	

Table 4: Features considered in our analysis.

Feature	Feature ranking methods	
	CFS Information Gain	Random Forest
RSS-Weighted Dispersion Mean	0.18	0.27
RSS StdDev	0.15	0.21
RSS-Weighted Dispersion StdDev	0.14	0.18
directional bias: RSS-weighted StdDev(major)/StdDev(minor)	0.09	0.2
distance gap between Centroid and Weighted Centroid (CWC)	0.04	0.13

Table 5: Feature selected by CFS, rankings and importance.

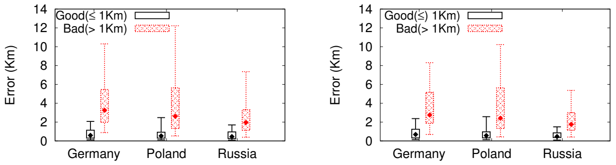
and the spatial properties of the strong measurements. Table 4 lists the features and the detailed descriptions are in the Appendix.

Feature Selection. Our initial feature set in Table 4 is large, and may contain features that are either redundant or irrelevant. To prevent overfitting, we first apply the correlation feature selection (CFS) [22] to identify a subset of relevant features for the classifier. CFS selects features independent of the classifier, and applies two criteria: the feature must be highly indicative, and must be highly uncorrelated with the features which are already selected. Table 5 lists the set of features selected via CFS, which are consistent across countries and time windows.

The selected feature set is dominated by RSS related features. It is interesting to see that the distance between the localization results of Centroid and Weighted Centroid becomes a key feature. Since Centroid only focuses on spatial characteristics while Weighted Centroid utilizes both RSS and spatial properties, this feature will likely capture the complex interaction of spatial and RSS factors during localization.

Classifier Training and Testing. Using the above features, we build our classifier using multiple methods including Decision Tree, Random Forests (RF), and Support Vector Machine (SVM)[5]. For a given localization accuracy requirement, *e.g.* localization error < x, we prepare the training data based on the localization error obtained using a specific localization algorithm, *e.g.* Weighted Centroid. We label a cell instance whose localization error is less than x as 1 (good) and otherwise as 0 (bad). The trained classifier will output whether a testing instance is good or bad.

Following the above process, we train and test our classifier using 10-fold cross-validation and report classification accuracy, precision, recall and area under ROC curve (AUC)[6]. As expected, Random Forests produce the best classification result, since the crowdsourced data is complicated and noisy. Random Forest is better to handle noise in the data because of ensemble technique.



(a) Weighted Centroid (b) Oracle

Figure 3: The distribution of the localization error for cell instances classified as Good (<1km) and Bad (>1km).

	Germany		Poland		Russia	
	1km	0.5km	1km	0.5km	1km	0.5km
Accuracy	0.84	0.84	0.84	0.88	0.82	0.80
Precision/Recall	0.84/0.85	0.84/0.84	0.84/0.84	0.87/0.88	0.81/0.82	0.79/0.80
AUC	0.91	0.92	0.92	0.93	0.88	0.88

Table 6: Classification results for the Weighted Centroid localization algorithm.

Table 5 lists the top features selected by Random Forest and their ranking in terms of the permutation importance. For comparison, we also list their ranking values computed from CFS, *i.e.* the information gain. For both methods (CFS and Random Forest), the features have similar weights, making it hard to further locate top features among them.

5.2 Classification Results

We now present the detailed classification results using different datasets and scenarios. We build classifiers for each country separately. Our experiment uses OpenCellID for training and testing.

Table 6 shows the classification results when using Weighted Centroid to perform localization, with either x = 0.5km or x = 1km as the accuracy requirement. The results are consistent across countries – the classifier has a reasonable accuracy around 85%. The performance of Russia is slightly worse, potentially due to its smaller data size (10% of the other two countries).

Figure 3(a) plots the quantile distribution of the actual localization error for the good and bad instances predicted by the classifier. Overall, we observe clear separations between the two classes. Using the classifier trained for 1km accuracy, we can identify good crowdsourcing instances that lead to no more than 3km localization error, while the majority (>75%) of these instances produce less than 1km error.

We also repeat our analysis for Oracle (the best of the 7 localization algorithms). Because the corresponding localization algorithm is unknown and will likely be much more complex, the predictability of its localization outcome reduces. As a result, the accuracy, precision, recall reduce 2%-7% and the AUC reduces 1%-4%. Figure 3(b) shows the quantile distribution of the actual error for both classes. We see that the distinction between good and bad cases is more clear in 3(a) than 3(b).

Impact of Training Data. We test the sensitivity of the training data by experimenting with our classifier using different amount of training data, and different types of training data. These include, using the data in one year to predict another year (2014→ 2015,

[5] We use the implementation of these algorthms in WEKA [21] with default parameters.

[6] Higher AUC values indicate stronger prediction power. AUC>0.5 means the prediction is better than random guessing.

% of data used for training	10%	50%	80%	90%	10%	50%	80%	90%
	1km				0.5km			
Germany	0.80	0.83	0.84	0.84	0.80	0.83	0.83	0.83
Poland	0.81	0.83	0.83	0.84	0.85	0.87	0.88	0.88
Russia	0.77	0.78	0.81	0.82	0.76	0.79	0.80	0.80

Training→Testing	Germany		Poland		Russia	
	1km	0.5km	1km	0.5km	1km	0.5km
2014 → 2015	0.83	0.80	0.84	0.86	0.80	0.77
2015 → 2014	0.84	0.82	0.84	0.85	0.81	0.79
Operator 1 → 2	0.82	0.81	0.83	0.85	0.80	0.75
Operator 2 → 1	0.81	0.80	0.82	0.83	0.81	0.79
OpenCellID → OpenBMap	0.84	0.79	0.82	0.85	0.84	0.81

Training	Germany		Poland		Russia	
Testing	Poland	Russia	Germany	Russia	Germany	Poland
1km	0.81	0.81	0.82	0.83	0.77	0.76
0.5km	0.80	0.80	0.81	0.80	0.78	0.78

Table 7: Training data sensitivity analysis, in terms of training data volume, time, cellular operators, datasets, and countries.

2015→ 2014), using the data from one cellular operator to predict another, using OpenCellID dataset to predict OpenBMap dataset, and using the data from one country to predict another country. Here we use OpenBMap only for testing since OpenBMap does not have sufficient data.

Table 7 lists the classification accuracy result. First for different training data amounts, the result shows that accuracy begins to reduce when there are not enough data volumes. Then overall the accuracy of prediction across data time and operators is on par with the original result in Table 6. As for cross-validation across countries, both Germany and Poland can well predict the other two countries, while the accuracy reduces to 0.77 when training with Russia. This is mostly because the Russia dataset is 10 times smaller than the Germany and Poland datasets.

6. FEATURE CLUSTERING

Despite good classifier performance, our efforts to identify and understand key features in predicting localization accuracy using standard ML were unsuccessful. Both information gain metrics and classifiers such as Random Forests produced feature importance rankings that did not clearly distinguish between key features. While these classifiers can identify instances likely to predict location within some error, they do not shed insights on the fundamental features that are indicative of predictive measurement instances.

In this section, we introduce a different approach that applies unsupervised learning to identify underlying correlations between key features and a measurement instance's predictive accuracy of Weighted Centroid. We define a distance metric that captures the similarity between key features of any two data instances. By computing the similarity metric between all pairs of instances, we can apply clustering algorithms to detect clusters of instances that capture features that tend to occur simultaneously. We call our approach *unsupervised feature clustering*.

6.1 Algorithm

Feature clustering groups data instances together based on their similarity across a small group of key features. In doing so, we are searching for possible clusters of measurement instances in the feature space, indicating a natural correlation between key features that may not be clear from other types of analysis.

By avoiding user-defined assumptions or constraints, feature clustering reveals inherent correlations between features, and allows us to identify natural combinations of features that produce highly predictive samples. Intuitively, this approach makes the assumption that a specific combination of features tends to coexist in highly predictive samples. If this assumption holds, then clusters of these features will be easily identifiable, and examining clusters will reveal key features that most strongly correlate with highly predictive measurement instances.

The process is as follows:

1. Select a small group of representative features from measurement data.
2. Define a pair-wise similarity metric between two instances based on these features.
3. Identify clusters in the measurement dataset using the similarity metric.
4. Search for correlation between identified clusters and intended outcome (in this context, prediction accuracy).
5. If strong correlation exists, use features in cluster to develop predictors for prediction accuracy.

Features and a Distance Metric. To identify a set of representative features from our measurement data, we rely on our prior results for feature selection using correlation-based feature selection (CFS) [22]. Applying CFS to a wide range of features produced a small set of features, including RSS standard deviation, RSS-weighted dispersion mean, RSS-weighted dispersion standard deviation, RSS-weighted directional bias, and CWC Gap (distance between Centroid and Weighted Centroid localization results).

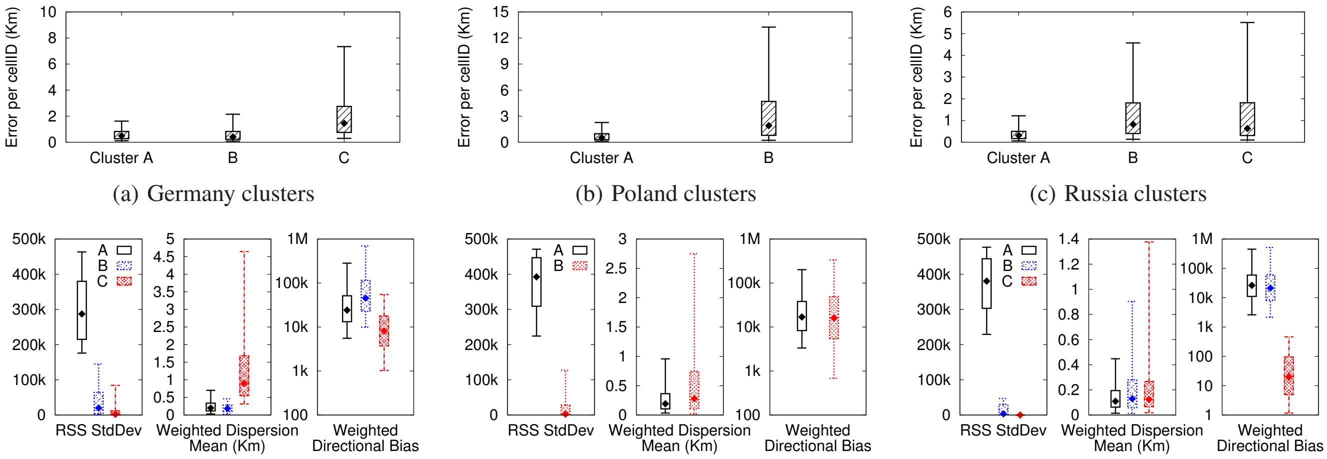
To combine the selected features into a single distance metric, we compute a five-tuple for each measurement sample (all measurement values pertaining to a single basestation). We normalize values for each feature using min-max normalization, i.e. normalized to the max value across all tuples. Finally, we generate a single distance metric by computing the (unweighted) Euclidean distance between the feature vectors of any two instances (the L2 norm of feature vectors).

Clustering. Given the distance metric, we can detect the natural clustering of measurement instances relative to our chosen features. A number of clustering algorithms are available, including hierarchical clustering [24], K-means, and METIS [25]. Since we wish to find natural correlation clusters, *i.e.* not a specific target number of clusters, we use hierarchical clustering, and optimize for modularity across all clusters. As we computed clusters for larger datasets, hierarchical clustering became a computational bottleneck. We switched to K-means for all results in this section and beyond, because it achieved nearly-identical results with an order of magnitude lower computation. We chose these clustering methods because they are commonly used and perform well in our experiments. While the choices of clustering methods could be further optimized, we leave this to future work.

After clusters are generated, we identify the most important features by computing each feature's chi-square statistics [16] and its difference between clusters. This quantifies how different the feature's values are distributed inside and outside a cluster, and in effect captures how important each feature is to distinguishing instances in a given cluster from the rest.

6.2 Results

We perform clustering on measurement datasets for Germany, Poland and Russia respectively, and plot key results in Figure 4. First, Figure 4(a–c) show that each of our three key datasets are dominated by 2 or 3 large clusters. More importantly, these clusters correlate strongly with our primary outcome, localization accuracy. In each case, one of the feature clusters identifies a group of



(a) Germany clusters

(b) Poland clusters

(c) Russia clusters

(d) Germany top features

(e) Poland top features

(f) Russia top features

Figure 4: Clusters and top features using OpenCellID-GT (candlestick graphs).

measurement samples that produce localization results with both low error and low variance (there are two such clusters in Germany). Measurement instances in the remaining clusters produce both much higher errors and higher error variance in their localization results. These results are extremely promising, because they point to the strong correlation of these key features with localization accuracy.

A closer look at the clusters shows that the key feature distinguishing the clusters is RSS standard deviation, and RSS-weighted dispersion mean also plays a role. This is somewhat unexpected, as intuition says that signal dispersion or directional bias might be better indicators for predictive measurement instances. We plot the values for top three features: RSS standard deviation , RSS-weighted dispersion mean and RSS-weighted directional bias in Figure 4(d–f) (we omit the other two features because their values are similar across clusters). In all three datasets, it is very clear that the cluster with the lowest localization error and lowest error variance is defined by RSS standard deviation. In Germany, we also find a second highly predictive cluster defined by low values for RSS-weighted dispersion mean and RSS standard deviation.

Two Primary Features. These results indicate that two primary features can effectively distinguish predictive measurement instances from others. First, RSS-weighted dispersion mean is a feature that measures the mean distance from each measurement location to the estimated location of the basestation, weighted by each measurement's RSS value. So a high value is not likely generated by measurements near the basestation with high RSS values. It effectively captures the ideal scenario, where there are sufficient strong measurements close to the actual basestation. We note that this cluster only appears in our Germany dataset, which is dense, and contains a large number of measurements in urban settings. In contrast, the Poland and Russia datasets don't show this cluster in our results, because they are much sparser, and much less likely to have samples of dense measurements close to the basestation.

In the absence of well placed measurements with sufficient strong RSS values, our results show that a crowdsourced instance can produce accurate results if the measurements contain high standard deviation in RSS values. This is not an obvious result, but captures the idea that RSS measurements near the actual basestation are more diverse. The diversity comes from the signal propagation in which RSS value changes more dramatically as the receiver (smartphone) gets closer to the signal source (basestation), and user context. Regardless of whether measurements are dense (Germany) or sparse (Poland and Russia), RSS standard deviation provides a strong signal to help guide the search for predictive instances.

7. IMPLICATIONS & APPLICATIONS

Given our insights from the previous analysis, we now consider implications on analysis of crowdsourced wireless measurements. In this section, we consider two questions. First, how can we use our insights to improve crowdsourced measurements for better accuracy? Second, we wish to test the generality of our findings by extending our approach to larger datasets in Europe and the US.

"Filtering" Crowdsourced Instances. Our key result is that RSS standard deviation and RSS-weighted dispersion mean are dominant features for detecting an instance's predictive accuracy. Generally, high RSS standard deviations and low mean RSS-weighted dispersion are both indicative of accurate localization results.

Here, we leverage these features to adaptively improve the quality of crowdsourced measurements. Our methodology is to adaptively monitor these two features as crowdsourced measurement values are gathered over time. Once a measurement instance has met either one or both of these features, we consider it sufficient. For measurement instance who has met neither target, we consider it a low-confidence instance and wait for additional measurements. We will design more complicated noise/anomaly detection in crowdsourcing measurement as a future work.

From results in Section 6, we set the bar of RSS standard deviation as 100k, and the bar of RSS-weighted dispersion mean to 0.5km. We use monthly data and look at instances that fail both targets in the three countries. We gradually add more data in the following months until the bar is reached. When getting more data, we combine the measurements of different months that maximize the RSS standard derivation and minimize the RSS-weighted dispersion mean. Figure 5 shows the results after this "instance filtering" process. It is clearly evident that across all countries, our filtering process dramatically lowers mean error (often by over 50%), and lowers error variance even more significantly (often by over 60%). The resulting instances are more accurate and predictable.

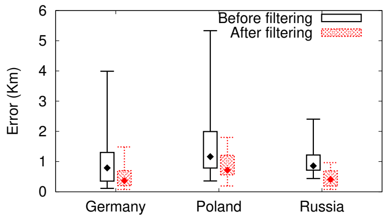


Figure 5: Localization results before and after filtering.

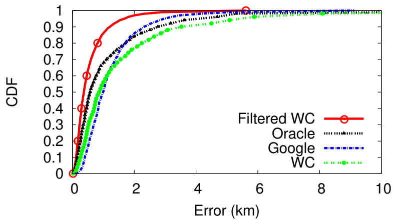


Figure 6: Localization results via filtering & other methods (Germany).

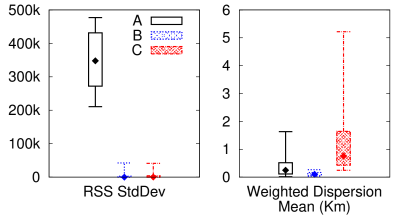
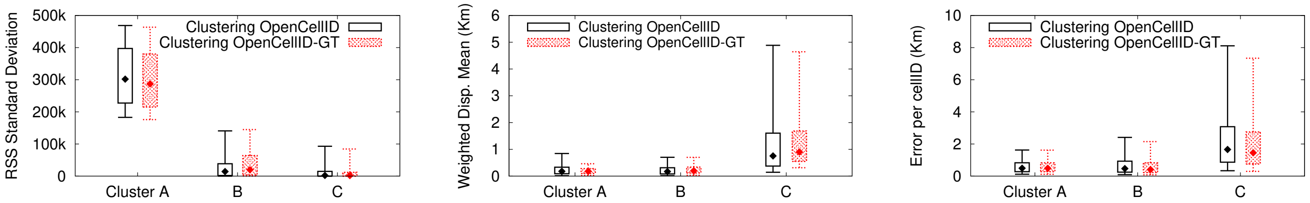


Figure 7: Distribution of top features for clusters in OpenCellID US.



(a) RSS Standard Deviation

(b) RSS-weighted Dispersion Mean

(c) Localization Error

Figure 8: (a)(b) Feature value distributions in clusters of Germany OpenCellID and OpenCellID-GT. (c) Comparing localization error between clusters using OpenCellID-GT and OpenCellID (cells with ground truth location in each cluster are selected).

Next, we compare the accuracy improvements gained by instance filtering against various localization methods, for ground truth data in Germany (results for Poland and Russia are also consistent). In Figure 6, we use a CDF to show the accuracy improvement across all portions of the error distribution. Not only does cleaning reduce error for the bulk of all instances, but it dramatically reduces the long tail of high error instances compared to all methods.

Generalizing Results. Despite these results, one might question whether our findings are a result of artifacts specific to a given dataset or location. Our hypothesis is that our conclusions are general, and high RSS standard deviation and low RSS-weighted dispersion mean should be sufficient to identify highly predictive measurement instances in different settings.

To test this, we extend our methodology to different and larger datasets. In Figure 7, we apply our feature clustering technique to the OpenCellID US dataset. While we do not have ground truth values for this dataset, we can see that the clustered results match Germany almost perfectly (Figure 4(d)). There are three clusters, two clusters with low localization errors that match the high RSS standard derivation and low RSS-weighted dispersion mean features, and one cluster with high localization error.

Finally, we test our methodology on the whole Germany dataset (all measurement samples, including those without ground truth locations). In Figure 8, we compare the properties of the clusters produced from the whole dataset to those from the ground truth samples. The whole dataset produced the same number of clusters as the dataset of samples with ground truth, and all key properties of each cluster match the ground-truth clusters nearly perfectly.

While these results are not as strong as our earlier results because we lack ground truth, they clearly support our hypothesis that a) the clustering results and key features are not specific to the ground truth data subsets, and b) the results could generalize across datasets and geographic regions. We hope to see our methodology tested for other crowdsourced measurements in future work.

8. RELATED WORK

Localization using Wardriving. Initial studies explored wardriving as an approach to collect RSS measurements for localizing WiFi Access Points (AP) [15, 23, 26, 43] or cellular towers [14, 45, 47]. Kim et al. [26] concluded that state-of-art localization algorithms can produce erroneous results and this will cause inaccurate estimates of WiFi coverage. Yang et al. [47] studied the accuracy of cell tower localization using wardriving data and showed that frequency, antenna height, and propagation environment make cell tower localization different from WiFi AP localization.

Localization using Crowdsourcing. Since wardriving is cumbersome and does not provide large-scale coverage, recent studies leverage crowdsourcing for indoor localization of WiFi APs [37, 41, 48] or outdoor cell tower localization [34, 44, 38]. [34] examined several localization algorithms using only 950 measurements and showed that the grid-based approach is the best. [44] studied cell tower localization using the OpenCellID dataset and validated the results with data from only 250 users. [38] applied different localization algorithms on a small portion of OpenCellID dataset. Unlike prior studies, we examine and compare seven popular localization algorithms on two *large-scale* datasets and show that there is no algorithm that performs consistently the best. We also examine the key factors that lead to such performance variance.

Quality of Measurements. A few works addressed issues and challenges in crowdsourcing measurements. [19, 30] studied the impact of user context in crowdsourcing based cellular network measurement systems. [33] considered the problem of crowdsourced measurement distribution and data density in network coverage prediction. Li et al [27] identified that data density and environment diversity have major impact on indoor WiFi localization. In contrast, our work examines the quality issues of outdoor cellular crowdsourced measurements using large-scale datasets, focusing on basestation localization. We found that data density does not matter much to localization results. [32] investigated ways to identify true information and reliable users in real-world crowd sensing applications like air quality sensing. They require users to take measurements at the same locations which is not practical in our scenario.

For applications like web page mining, existing works (*e.g.* [50, 46, 42]) tried to remove noise and anomaly in data. Our work differs by providing a systematic framework to examine the key characteristics of crowdsourced cellular measurements and to quantify the usability of this data for basestation localization. *To the best of our knowledge, we are the first to provide a comprehensive study on the usefulness of crowdsourced wireless measurements.*

9. CONCLUSION

Our work analyzes the value of large user-contributed signal measurements in the context of basestation localization, using large-scale RSS datasets from OpenCellID and OpenBMap. We find that even machine learning techniques cannot reduce the variance in localization results, nor can they identify key features (RSS StdDev, RSS-weighted dispersion mean) that correlate strongly with highly predictive data instances. Instead, we apply a *feature clustering* technique to detect natural correlation patterns between measurement features, and use them to identify types of measurement data that correlate well with high or low prediction accuracy. We show that these clustering results are general across datasets, and that we can dramatically improve localization results using our identified features. We hope these results shed light on other types of crowdsourced measurements, and will test the applicability of this approach to other applications in ongoing work.

Appendix: Detailed Description of Features

- *Dispersion*: shows the spread or spatial variability of measurements and calculates their distances to center. When weighted by RSS, the dispersion mean is defined as $\bar{d} = \frac{\sum_i d(i,center)*RSS_i}{\sum_i RSS_i}$, $center$ is the estimated basestation location using Weighted Centroid. The std is $\sqrt{\frac{\sum_i (d(i,center-\bar{d}))^2*RSS_i}{\sum_i RSS_i}}$.
- *Angular coverage*: measures how measurements distribute around the estimated center from the angular point of view.
- *Standard deviational ellipse*: measures the dispersion in two dimensions. The major axis is defined as direction of maximum spread of the distribution. The minor axis is perpendicular to major axis and defines the minimum spread.
- *Estimated path loss exponent*: by fitting the log-normal propagation model, the estimated path loss exponent shows the relationship between RSS and distance.
- *Spatial autocorrelation*: measures the correlation among one point and its relatively close points. Positive spatial autocorrelation occurs when similar values occur near one another. Negative spatial autocorrelation occurs when dissimilar values occur near one another.

Acknowledgement

We thank the anonymous reviewers for their constructive feedback. This work was supported in part by NSF grants AST-1443956, AST-1443945 and CNS-1224100. Any opinions, findings, and conclusions or recommendations expressed in this material do not necessarily reflect the views of any funding agencies.

10. REFERENCES

[1] http://www.cnet.com/news/verizon-t-mobile-att-sprint-who-is-the-fastest-carrier-in-the-nation/.

[2] http://www.opencellid.org/.

[3] http://openbmap.org/.

[4] http://www.skyhookwireless.com/.

[5] http://www.antennasearch.com/.

[6] http://opensignal.com/.

[7] http://www.rootmetrics.com/.

[8] http://www.netradar.org/.

[9] http://googlemobile.blogspot.com/2008/06/google-enables-location-aware.html.

[10] http://jonspinney.com/thoughts/2008/3/24/how-does-googles-mylocation-really-work.html.

[11] http://franciscokattan.com/2010/02/06/dynamic-cell-id-clever-way-to-block-google-but-will-it-backfire/.

[12] https://developers.google.com/maps/documentation/geolocation/.

[13] Achtzehn, A., et al. Crowdrem: Harnessing the power of the mobile crowd for flexible wireless network monitoring. In *Proc. of HotMobile* (2015).

[14] Chen, M. Y., et al. Practical metropolitan-scale positioning for gsm phones. In *Proc. of Ubicomp* (2006).

[15] Cheng, Y.-C., Chawathe, Y., LaMarca, A., and Krumm, J. Accuracy characterization for metropolitan-scale wi-fi localization. In *Proc. of MobiSys* (2005).

[16] Chernoff, H., and Lehmann, E. The use of maximum likelihood estimates in χ^2 tests for goodness-to-fit. *The Annals of Mathematical Statistics* (1954).

[17] Chintalapudi, K., Padmanabha Iyer, A., and Padmanabhan, V. N. Indoor localization without the pain. In *Proc. of MobiCom* (2010).

[18] Cox, D., and Lewis, P. The statistical analysis of series of events.

[19] Gember, A., et al. Obtaining in-context measurements of cellular network performance. In *Proc. of IMC* (2012).

[20] Griffith, D. A. *Spatial autocorrelation and spatial filtering: gaining understanding through theory and scientific visualization*. 2013.

[21] Hall, M., et al. The weka data mining software: an update. *SIGKDD explorations* (2009).

[22] Hall, M. A. *Correlation-based feature selection for machine learning*. PhD thesis, University of Waikato, 1999.

[23] Han, D., and Seshan, S. a. o. Access point localization using local signal strength gradient. In *Proc. of PAM* (2009).

[24] Johnson, S. C. Hierarchical clustering schemes. *Psychometrika* (1967).

[25] Karypis, G., and Kumar, V. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing* (1998).

[26] Kim, M., Fielding, J. J., and Kotz, D. Risks of using ap locations discovered through war driving. In *Proc. of Pervasive* (2006).

[27] Li, L., et al. Experiencing and handling the diversity in data density and environmental locality in an indoor positioning service. In *Proc. of MobiCom* (2014).

[28] Lim, H., Kung, L.-C., Hou, J. C., and Luo, H. Zero-configuration, robust indoor localization: Theory and experimentation. In *INFOCOM* (2006).

[29] Littman, L., and Revare, B. New times, new methods: Upgrading spectrum enforcement. *Silicon Flatirons Center* (2014).

[30] Marina, M. K., Radu, V., and Balampekos, K. Impact of indoor-outdoor context on crowdsourcing based mobile coverage analysis. In *Proceedings of Workshop on All Things Cellular: Operations, Applications and Challenges* (2015).

[31] Megiddo, N. Linear-time algorithms for linear programming in r^3 and related problems. *SIAM journal on computing* (1983).

[32] Meng, C., et al. Truth discovery on crowd sensing of correlated entities. In *SenSys* (2015).

[33] Molinari, M., Fida, M.-R., Marina, M. K., and Pescape, A. Spatial interpolation based cellular coverage prediction with crowdsourced measurements. In *Proc. of SIGCOMM Workshop on Crowdsourcing and Crowdsharing of Big (Internet) Data* (2015).

[34] Neidhardt, E., Uzun, A., Bareth, U., and Kupper, A. Estimating locations and coverage areas of mobile network cells based on crowdsourced data. In *Proc. of WMNC* (2013).

[35] NURMI, P., BHATTACHARYA, S., AND KUKKONEN, J. A grid-based algorithm for on-device gsm positioning. In *Proc. of UbiComp* (2010).

[36] PFAMMATTER, D., GIUSTINIANO, D., AND LENDERS, V. A software-defined sensor architecture for large-scale wideband spectrum monitoring. In *IPSN* (2015).

[37] RAI, A., CHINTALAPUDI, K. K., PADMANABHAN, V. N., AND SEN, R. Zee: zero-effort crowdsourcing for indoor localization. In *Proc. of MobiCom* (2012).

[38] RUSVIK, J. A. N. Localizing cell towers from crowdsourced measurements. Master's thesis, 2015.

[39] SEN, S., ET AL. Can they hear me now?: a case for a client-assisted approach to monitoring wide-area wireless networks. In *IMC* (2011).

[40] SHEKHAR, S., AND CHAWLA, S. *Spatial databases: a tour. Introduction to Spatial Data Mining (Chapter 7)*. 2003.

[41] SHEN, G., CHEN, Z., ZHANG, P., MOSCIBRODA, T., AND ZHANG, Y. Walkie-markie: indoor pathway mapping made easy. In *Proc. of NSDI* (2013).

[42] SOLTANOLKOTABI, M., AND CANDES, E. J. A geometric analysis of subspace clustering with outliers. *The Annals of Statistics* (2012).

[43] SUBRAMANIAN, A. P., DESHPANDE, P., GAOJGAO, J., AND DAS, S. R. Drive-by localization of roadside wifi networks. In *Proc. of INFOCOM* (2008).

[44] ULM, M., WIDHALM, P., AND BRANDLE, N. Characterization of mobile phone localization errors with opencellid data. In *Proc. of ICALT* (2015).

[45] VARSHAVSKY, A., PANKRATOV, D., KRUMM, J., AND LARA, E. Calibree: Calibration-free localization using relative distance estimations. In *Proc. of Pervasive* (2008).

[46] XIONG, H., PANDEY, G., STEINBACH, M., AND KUMAR, V. Enhancing data analysis with noise removal. *IEEE Transactions on Knowledge and Data Engineering* (2006).

[47] YANG, J., ET AL. Accuracy characterization of cell tower localization. In *Proc. of Ubicomp* (2010).

[48] YANG, Z., WU, C., AND LIU, Y. Locating in fingerprint space: wireless indoor localization with little human intervention. In *Proc. of MobiCom* (2012).

[49] YEDAVALLI, K., KRISHNAMACHARI, B., RAVULA, S., AND SRINIVASAN, B. Ecolocation: a sequence based technique for rf localization in wireless sensor networks. In *Proc. of IPSN* (2005).

[50] YI, L., LIU, B., AND LI, X. Eliminating noisy information in web pages for data mining. In *Proc. of kdd* (2003).

[51] YUILL, R. S. The standard deviational ellipse; an updated tool for spatial description. *Geografiska Annaler* (1971).