8-31-2021

Multi-Operator Cell Tower Locations Prediction from Crowdsourced Data

Mostafizur Rahman
The University of Texas Rio Grande Valley, mostafizur.rahman@utrgv.edu

Mohammad Arif Hossain
New Jersey Institute of Technology

Murat Yuksel

Recommended Citation

M. Rahman, M. A. Hossain and M. Yuksel, "Multi-Operator Cell Tower Locations Prediction from Crowdsourced Data," 2021 International Conference on Computer Communications and Networks (ICCCN), 2021, pp. 1-9, doi: 10.1109/ICCCN52240.2021.9522192.

Multi-Operator Cell Tower Locations Prediction from Crowdsourced Data

Mostafizur Rahman*, Mohammad Arif Hossain†, and Murat Yuksel‡
*Department of Electrical and Computer Engineering, University of Texas Rio Grande Valley, USA
†Department of Electrical and Computer Engineering, New Jersey Institute of Technology, USA
‡ Department of Electrical and Computer Engineering, University of Central Florida, USA
*mostafizur.rahman@utrgv.edu, † mh624@njit.edu, ‡murat.yuksel@ucf.edu

***Abstract*—Cell tower locations are not publicly available due to business interests of wireless providers. Very often wireless providers provide exaggerated coverage maps that may mislead the public. In addition to providing a neutral check on the coverage maps, prediction of cell tower locations hosting multiple operators' access nodes could also be helpful in disaster communications and public safety in general. The localization of the disaster-affected towers can be very conducive to respond and reach to the victims. Further, victims' devices could utilize this knowledge to initiate device-to-device (D2D) or unmanned aerial vehicular (UAV) communications as alternatives to the damaged cellular infrastructure. Publicly available crowdsourced cell (base station) locations and FCC's sites can be used to predict the cell tower/site locations in the United States. In this work, we utilized a weighted k-means algorithm to predict cell tower locations from OpenCellid crowdsourced dataset and implemented a mapping algorithm to locate nearest physical towers. We map the predicted towers to two different sources of physical towers. Our comparison shows a significant accuracy in predicting tower locations regardless of sources of physical towers. The technique can be used to predict the tower locations in other countries as well.**

***Index Terms*—Cell Sites, Disaster Management, Clustering Algorithm, Mapping Algorithm**

I. Introduction

Cell tower locations are crucial for the network coverage of a cellular provider. With the increase of small cell deployments, it is becoming challenging to identify and differentiate a provider's tower locations unless the provider exclusively provides the information. Unfortunately, cellular providers are reluctant to provide actual coverage maps due to business concerns [1]. In most cases, cellular providers provide cell tower locations that are above a certain height, e.g., towers above 200 feet are registered to the Federal Communications Commission (FCC) in the US [2]. Most of the cases, these towers host antennas of multiple cellular operators. The US government has already announced a subsidy for the providers to reveal their coverage maps so that the uncovered areas can be covered by government-subsidized infrastructure [3]. Still, cellular providers do not comply and reveal their actual coverage maps. Thus, a truthful coverage map generation requires truthful cell tower locations of the providers. As the providers do not provide actual information and a regulatory organization does not have enough data to validate, it is essential to find alternate solutions for identifying cell tower locations and generating coverage maps. Crowdsourced data can be a good source of such information. The benefit of the crowdsourced data is to have actual perceived wireless experience of cellular users. Thus, it makes a more reliable source of data to generate the actual coverage map.

Knowing the nature of the communication infrastructure can enhance alternative modalities of communication during disasters or major failures. Prompt response during a disaster is very crucial. People are trying and will try to make the emergency services and response more efficient and instantaneous. It is always troublesome to respond to all the victims rapidly. One of the greatest ways to minimize the loss of the victims is to keep communication among the victims and the first responders. However, the cellular infrastructure, i.e., cell towers and antennas, can be destroyed during a disaster. So, an alternative, that can be promising, is to continue communication among the victims and the first responders. In particular, emerging modalities such as device-to-device (D2D) communication and unmanned aerial vehicular (UAV) communications are capable of maintaining communication without cellular infrastructure, and hence, can be a good option for such alternative emergency and disaster communications.

Prioritizing a victim could also be very important during a disaster-affected scenario because there might be many requests from victims at a single time. Therefore, it will take time to prioritize. It would be very efficient if we can initiate the D2D communication in small scales in case of emergency. Prediction of cell tower locations can be one way to predict the areas where the D2D communication is needed and to help manage the overall communication in a disaster-affected scenario. Identification of vulnerable cell tower locations can give guidance in terms of the number of D2D hops needed to reach the closest active tower in case of a disaster. Further, by identifying the tower locations which can be damaged during a disaster, we can trigger alternative forms of providing wireless access such as portable base stations using UAVs or ground vehicles, which can save many invaluable lives. Yet, satisfying these needs require an accurate and fast measurement of where cell towers and base stations are.

Several Internet sites [4], [5] including FCC [2] show the locations and the number of cellular towers all over the world.

This work was supported in part by NSF award 16226110 and NIST PSIAP Program grant 70NANB17H188. Dr. Rahman and Mr. Hossain were with the Elec. & Comp. Eng. Dept. at the U. of Central Florida during this work.

FCC only contains registered tower locations which include already-dismantled towers' information. As a result, the number of cell towers provided by FCC may be different from the number of cell towers existing within an area. We have also found that the number of cell sites provided by FCC status reports [6] may be different from the number of cell towers shown at these sites. There are no such authentic sources that provide accurate locations of all antennas including towers, and the cellular providers do not share information about the location of towers. There are only few countries where the correct locations of towers are known [7]. The sites may provide the information based on war-driving data or from crowdsourced data. So, there is no entirely valid source of information to know and validate the exact locations of cell towers, which makes it even more important to design a system for predicting them given publicly available data.

To optimize the time and cost of finding cell towers' location, crowdsourcing can be one of the most effective ways. In this work, we utilized the crowdsourced data from OpenCelliD database [8] and processed it for using in prediction. OpenCelliD data contain access points or antennas information regardless of cellular operators. Also, it does not provide information on the types of antennas, e.g., directional or omnidirectional, operating radius, etc [9]. Our objective is to find cell towers hosting multiple operators access networks from a collection of towers and different providers' base stations (or antennas). By referring to a cell tower and a base station, we mean physical infrastructures containing multiple operators and single operator's antennas, respectively. Thus, predicting a cell tower's location information will help the emergency responders to identify the importance of towers considering population effects. To our best knowledge, it is the first work to identify and separate cell towers hosting multiple operator networks from crowdsourced data. We used weighted k-means clustering algorithm [10] to predict the locations of cell towers from processed OpenCelliD data. We also utilized tower locations data from *AntennaSearch.com* [11] and FCC [2] to compare our predicted locations of cell towers. For the comparison, we developed a mapping algorithm to show the distance between the cell towers from these two sources and our predicted towers. We also analyzed the cumulative distribution function (CDF) for the results found from the comparisons. We have performed the clustering analysis, mapping and CDF analysis for an urban and two rural counties of Florida, USA.

The rest of this paper is organized as follows: Section II includes the related works. Section III presents the proposed methodology for predicting cell tower while Section IV shows a mapping algorithm to find out the accuracy of our proposed algorithm. Section V describes the results of our proposed method and mapping algorithm. Finally, Section VI draws the conclusion of the work and points out the future directions.

II. Related Works

The localization of cellular networks receives great attention from the research community. Many researchers have worked on the localization of mobile devices because many works consider the location of cell tower known by identifying the location of mobile devices [12]. However, localization of cell towers have not achieved much attention. In this section, we discuss only the works related to the localization of cell towers.

We found most of the cell tower localization works are based on wardriving [13], [14], a procedure of getting cellular network data with a mobile platform. Collecting such data requires development and systematic deployment of such platforms, and hence, the collected data costs money and is not available for public use. In our work, we use crowdsourced data by OpenCelliD, which contains random positions of volunteer users within a cell sector. The ad-hoc (or volunteer) and random collection of the data makes it harder to predict the tower locations from it as the users of a particular cell tower can be in the range of other towers and the ad-hoc measurements may not capture sufficient samples.

Received signal strength (RSS) based cell tower prediction is a well-known technique to predict cell towers [15]–[17]. War-driven cell tower location prediction where RSS is used as a weight for locating cell towers using weighted centroids was considered in [18]. A cell tower is more leaned towards a location with better signal strength. Though this work considered better signal strength, it misses the importance of daily use. How many customers are actually being benefited by a tower is lacking from this analysis where our sample count-based weighted cell tower location prediction considers the effect of usage count. More sample counts contribute more weight in the prediction. A seamless transition to the best network with the tower location prediction for a user of mobile virtual network operator (MVNO) was introduced in [19]. However, the technique only considers a single MVNO user to switch to the best network based on the tower location prediction from RSS.

Data-driven cell tower location estimation using crowdsourcing became essential given the lack of proper information from the governments as well as the wireless providers [20]–[22]. Data-driven cell tower location prediction considers all types of towers and access points [23], [24]. These do not differentiate among multi-operator antenna hosting cell towers to single provider access points or base stations.

OpenCelliD-based cell tower localization with cell tower direction from the RSS measurements was considered in [25]. The prediction method is applicable only to multi-sector cells and does not consider omni-directional cells. Further, it assumes availability of the correct tower locations. In our work, we tackle the problem while considering the inaccuracies in the tower locations as well as their count.

III. Prediction Methodology

For optimizing time and cost of finding the location of cell towers, crowdsourcing can be one of the most effective ways. We visualize tower locations prediction procedure from crowdsourced data in Fig. 1. We utilized the crowdsourced data from OpenCelliD database and processed the data for several counties of the US. We used county boundary information

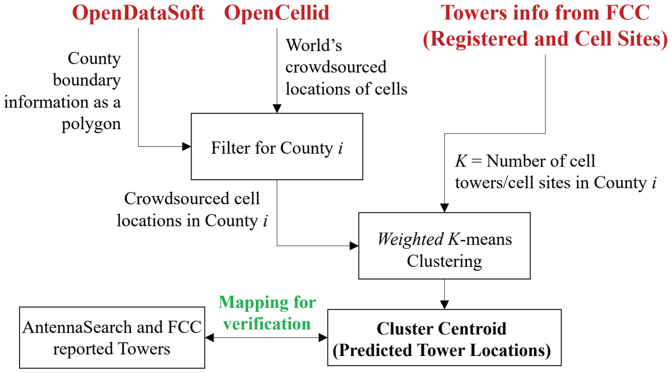
OpenDataSoft
OpenCellid
Towers info from FCC (Registered and Cell Sites)
County boundary information as a polygon
World's crowdsourced locations of cells
Filter for County i
K = Number of cell towers/cell sites in County i
Crowdsourced cell locations in County i
Weighted K-means Clustering
Mapping for verification
AntennaSearch and FCC reported Towers
Cluster Centroid (Predicted Tower Locations)

Fig. 1. Cell Tower Locations Prediction Framework

from OpenDataSoft [26], and filtered cell information within each county. Later, we used a weighted k-means clustering algorithm [10] to predict the locations of cell towers from the processed OpenCelliD data. In this work, we mostly used active towers count that are registered to FCC [2] to determine the value of k. We also used cell sites [6] count as the value of k for further analysis. We have collected tower locations data which we call physical towers from *AntennaSearch.com* [11] and FCC [2] to compare our predicted locations of cell towers. For the comparison, we have developed a mapping algorithm to show the difference of distances among the physical towers and the predicted towers. We have also analyzed the CDF for the results found from the comparison. We have performed the clustering analysis, mapping, and CDF analysis for an urban county (Orange) and two rural counties (Calhoun and Union) of Florida.

A. Data Collection and Processing

1) County Data: The first phase of our work is to collect the required data set for a county. In order to locate cell towers for a certain region, we need the information on boundaries of that region. We have chosen Florida to find out the predicted cell towers. We have collected the boundary information for the counties of Florida. The boundary information is available in the form of a polygon with corners defined by longitude-latitude pairs [26].

2) OpenCelliD Data: We utilized the crowdsourced data from *opencellid.org* in our prediction algorithm. The dataset has different fields that include information regarding a cell. The data are given under different fields such as *radio*, *mcc*, *net*, and *area*. The fields of *lon* and *lat* represent estimated longitude and estimated latitude of the cell's base station, respectively. The *samples* field show the number of measurements made to estimate the location of the cell. We used the information these three fields (i.e., *lon*, *lat*, and *samples*) to predict the tower locations. The dataset uses the longitude and latitude coordinates for a particular cell by measuring the mean of measurements of longitudes and latitudes done for that cell. Thus, the larger the *samples* field, the more reliable the data fields of that cell.

Since the OpenCelliD dataset is too large (i.e., it includes the estimated locations of base stations in the entire world), we filtered out entries with longitude and latitude falling outside of the boundary of Florida. To do so, we used Matlab's inpolygon algorithm to determine if a point is inside a given polygon or not. This enabled us to work with less data when applying the clustering algorithm, explained later in Section III-B. We further filtered the data of three counties of Florida and predicted the cell tower locations in those counties.

3) AntennaSearch.com Data: We explored different websites to find out the cell sites in particular and found *AntennaSearch.com* as one of the best sources for collecting cell tower locations. We downloaded cell site locations for the three counties from *AntennaSearch.com*. The interface at *AntennaSearch.com* only gives the cell sites within 2 miles radius from a particular point of a location. To get the locations of the cell sites around a particular place, the site requires street address, city and state information.

We selected several locations for three counties to download cell site locations from *AntennaSearch.com*. We chose the locations in such a way that the distance between two neighboring locations would be less than 2 miles so that we can collect every cell site location within the county. We used the boundary data to download all the cell sites in a county. The downloaded cell site locations contain duplicate values as the locations' ranges were overlapping as shown in Fig. 2. Moreover, there will be some cell site locations which are outside of the boundary of the counties. So, we have de-duped and filtered the cell site locations to get the locations inside the county boundaries.

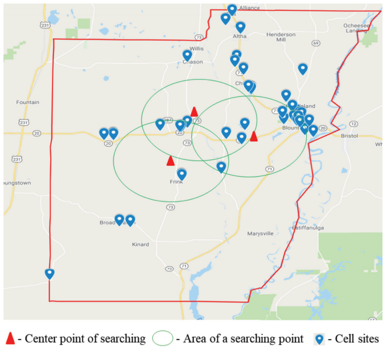

Fig. 2. Overlapping ranges of search for cell site locations in [11]

4) Registered Towers at FCC Data: In the USA, cellular providers have two types of towers: *Registered* or *unregistered* to FCC. Cellular providers are required to register tower location information to FCC for those at least 200 feet in height. We collected the registered towers data from FCC site [2]. We collected each tower's information manually which is not dismantled or under-construction. There were options to download FCC data all-together. However, it requires additional data cleaning to parse the necessary information. We were able to handle the data manually for the three counties we worked on, but a more systematic access to the FCC's data will be necessary to scale the system to large areas.

B. Clustering Algorithm to Predict Tower Locations

We use the locations (longitude and latitude) data reported in OpenCelliD, and apply a weighted k-means clustering algorithm [10] where the number of samples is used as a weight to predict the location of the cell towers in that county. Here, k is the number of predicted locations for a particular county. We use the number of active towers from the list of registered towers of a county given by FCC [2] as well as the cell sites count [6] as the values of k for clustering. In the future, to further improve the prediction, machine learning techniques can be used to improve tower count prediction by using trustworthy sources of tower/antenna locations as training data sets that contain both the registered and unregistered towers. However, since such data is unavailable we use the cleaned FCC's data to guide the clustering algorithm.

IV. Mapping Predicted and Physical Towers

According to FCC site [2], we counted the number of active cell towers in Orange, Calhoun, and Union counties as 326, 21, and 13, respectively. For these counties, *AntennaSearch.com* reports 1605, 52, and 22 cell towers, respectively. Also, the cell site counts given by FCC [6] are 1160, 15, and 13, respectively. The cell tower and cell site counts given in the FCC sites, and *AntennaSearch.com* are different. And, we do not have any ground truth information about the location of the cell towers. In order to understand how our tower location prediction is performing, we apply one-to-one mapping of cell towers based on the distance to find out how close our predicted towers are to the ones made available from *AntennaSearch.com*. We also apply similar mapping to FCC-reported tower locations to generate CDF for our predicted towers. The towers reported either in FCC site or *AntennaSearch.com*, we call them 'physical towers' or 'physical locations'. Note that this mapping, beyond allowing us to observe how well our prediction technique is performing, is a critical necessity in order to improve the tower location prediction system. Since ground truth is not available, techniques such as this mapping of the predicted and the physical towers are needed to guide machine learning techniques so as to attain better predictions.

In essence, we find the nearest cell tower among the physical locations for a particular predicted location in the mapping. This mapping, however, must be done so that the sum of the distances between the predicted and physical locations is minimized. To solve this mapping problem, we designed a randomized heuristic algorithm (Algo. 2). When designing the mapping algorithm, we faced two issues:

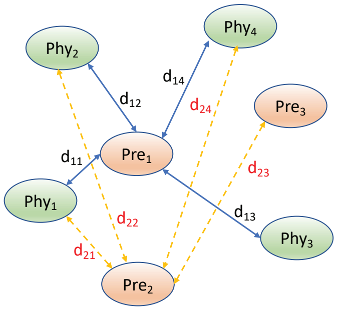

Fig. 3. Tower Mapping

- *Case I:* A single physical tower can be the nearest one for multiple predicted towers.
- *Case II:* Multiple predicted towers can have equal distance to a physical tower.

We visualize the mapping and both cases in Fig. 3. Assume a set of predicted towers: $\{Pre_1, Pre_2, Pre_3\}$, and a set of physical towers: $\{Phy_1, Phy_2, Phy_3, Phy_4\}$. We denote the distance from a predicted tower Pre_i to a physical tower Phy_j as d_{ij}. We find Case I, when Phy_1 is the nearest physical tower for both Pre_1 and Pre_2 predicted towers, given that $d_{11} < d_{12}$, $d_{11} < d_{13}$, $d_{11} < d_{14}$, $d_{21} < d_{22}$, $d_{21} < d_{23}$, and $d_{21} < d_{24}$. In this case, we map predicted tower that has the minimum distance to physical tower Phy_1. We find Case II, if both predicted towers have the same distance to physical tower Phy_1, i.e., $d_{11} = d_{21}$. In such scenario, we map randomly to any of them.

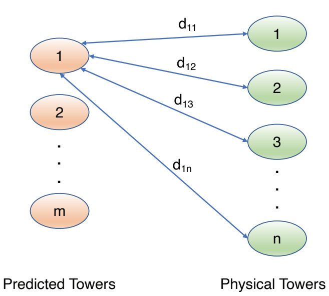

Fig. 4. NP-Hardness

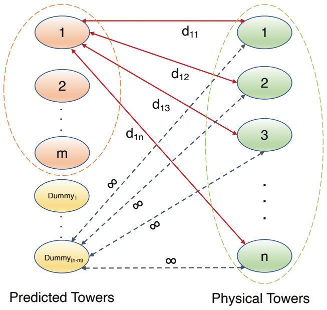

Fig. 5. Reduction to NP-Complete

In the rest of this section, we first show the NP hardness of the mapping problem. Then, we detail the algorithm, illustrate it on an example, and discuss its complexity.

A. Cell Tower Mapping is NP Complete

Our goal is to find the minimum distance mapping between all predicted and physical towers. This original problem is actually similar to the Bipartite Graphs Matching problem where cost is minimized. However, in our problem, we need to select a tower from a set of predicted towers, P, to a tower in a set of physical towers, W, such that the the sum of distances among mapped towers is minimized. Consider the mapping of predicted towers to physical towers in Fig. 4. Initially, we can choose a tower from M predicted towers and get the distance to all N physical towers. Now, we have MN selections to get a pair of mapped towers having minimum distance. Sequentially, we will continue until the set of predicted towers become empty as we assume $M \leq N$. We get $MN(M-1)(N-1)...1(N-M+1)$ selections to get the minimum distance mapping. Thus, the complexity of the search space becomes: $\mathcal{O}(M!N!/(N-M)!)$. We can reduce the problem to an NP Complete problem by adding necessary number of dummy towers to predicted towers set in Fig. 5. These dummy towers reduce the original problem to the Bipartite Minimum Weight Perfect Matching (BMWPM) [27], an NP Complete problem, where each edge is weighted with the geo-distance between the towers of the two towers sets, and the distance from a dummy tower to a physical tower is assigned as ∞. The reduction of our mapping problem has the polynomial complexity of $\mathcal{O}(N^2)$ which shows that our mapping problem is NP Complete. Once we have the minimum distance perfect matching, we can remove matches with distances ∞, and we will get our final mapping of predicted towers to physical towers.

Algo. 1 details how we reduce the original towers mapping problem to the Minimum Weight Perfect Matching problem and achieve a near-optimal polynomial solution. Initially, we add $(N-M)$ dummy towers to the predicted towers set P to match the number of towers of physical towers set W, and get

Algorithm 1 Near optimal solution of mapping predicted towers to physical towers

Input: Set of longitudes and latitudes for predicted towers, *P* with size *M*, and physical towers, *W* with size *N*
Output: A mapping of *P* to *W*

```
 1: procedure NearOptimalMapping($P$, $M$, $W$, $N$)
 2:     if $M < N$ then
 3:         Add $(N - M)$ dummy towers to the set of $P$.
 4:         Add edges from all towers in $P$ to all towers in $W$ with geo-distance as
            weight, set distance from dummy towers in $P$ to all towers in $W$ as $\infty$,
            and form edge set, $E$.
 5:     end if
 6:     if $M > N$ then
 7:         Return NearOptimalMapping($W$, $N$, $P$, $M$).
 8:     end if
 9:     Compute MIN-WEIGHT-PERFECT-MATCHING(($P \cup W$), $E$) [27] where edge
        set $E$ denotes the one-to-one mapping of $N$ predicted towers to $N$ physical
        towers.
10:     Remove towers from $P$ and $W$ those having edges with $\infty$ distance after
        mapping.
11:     Return mapped towers set $P$ and $W$ with modified edge set $E$.
12: end procedure
```

N towers for both sets of towers in line 3. Later, we add edges from all towers of P to all towers of W in line 4, where we use geo-distance between two towers as weight of each edge. We set edge weight from all dummy towers of P to all towers in W as ∞ in line 4. Next, we call the Minimum Weight Perfect Matching of bipartite graphs of towers at line 6. We get the final mapped bipartite graphs of towers after removing towers from both sets those have ∞ distance. This polynomial time reduction algorithm proves the NP completeness our original tower mapping problem.

B. Proposed Mapping Algorithm

Assume that there are two sets of data P and W of longitude and latitude pairs for predicted and physical locations, respectively. We also consider that there are M and N data points (each corresponding to a location) in P and W, respectively:

$$P = \begin{bmatrix} 1 \\ 2 \\ . \\ . \\ M \end{bmatrix} \quad W = \begin{bmatrix} 1 \\ 2 \\ . \\ . \\ N \end{bmatrix}$$

Now, we can get the distance for every data point of P to the all points of W which means every location in P will have N distance values according to line 4 of Algo. 2. We represent these distances as an M-by-N matrix, D_{MN} where d_{mn}, $m = 1..M$, $n = 1..N$ is the distance between mth point in P and nth point in W:

$$D_{MN} = \begin{bmatrix} d_{1,1} & d_{1,2} & . & d_{1,N} \\ d_{2,1} & d_{2,2} & . & d_{2,N} \\ . & . & . & . \\ d_{M,1} & d_{M,2} & . & d_{M,N} \end{bmatrix}$$

Our goal is to find out the minimum distance from N distance values for every location of P. We sort the N distance values in each row of D_{MN} to get the minimum distance for all location points of P. In this way, we get another matrix $D_{MN(sorted)}$ according to line 8 of Algo. 2 where the distance values are sorted in each row. An example $D_{MN(sorted)}$ may appear as:

$$D_{MN(sorted)} = \begin{bmatrix} d_{1,7} & d_{1,N} & . & d_{1,8} \\ d_{2,3} & d_{2,5} & . & d_{2,10} \\ . & . & . & . \\ d_{M,N} & d_{M,8} & . & d_{M,1} \end{bmatrix}$$

Then, we create another matrix $D_{MN(index)}$ called index matrix to show only the index for the sorted distance matrix where the row of the matrix stands for the locations of P and the column stands for the locations of W. It is obvious that the 1st value before comma in the subscript of every element in $D_{MN(sorted)}$ matrix represents the row number while the 2nd value after the comma represent the index value for the $D_{MN(index)}$ matrix. However, the first column of every row of the matrix $D_{MN(index)}$ represents the closest tower location. For instance, $D_{MN(index)}$ for the above $D_{MN(sorted)}$ will look like this:

$$D_{MN(index)} = \begin{bmatrix} 7 & N & . & 8 \\ 3 & 5 & . & 10 \\ . & . & . & . \\ N & 8 & . & 1 \end{bmatrix}$$

As the last step of the initialization of the algorithm, we create a binary matrix B of same size as the distance matrix to keep track of the mapping values. We initialize B to all zeros, i.e., $B(i,j)_{i=1..M,j=1..N} = 0$ at line 13 of Algo. 2.

Algorithm 2 Mapping between predicted and physical towers

Input: Set of longitudes and latitudes for predicted towers, *P* with size *M*, and physical towers, *W* with size *N*
Output: A mapping of locations of a predicted tower, *m* and a physical tower, *n* for the towers in *P* and *W*

```
 1: procedure Mapping($P$, $M$, $W$, $N$)
 2:     for $i = 1$ to $M$ do
 3:         for $j = 1$ to $N$ do
 4:             Find Haversine distance from $i$-th element to $j$-th element, and
                update distance matrix, $D_{MN}$.
 5:         end for
 6:     end for
 7:     for $i = 1$ to $M$ do
 8:         Sort the values of $i$-th row of $D_{MN}$ in ascending order and update
            $D_{MN(sorted)}$.
 9:     end for
10:     for $i = 1$ to $M$ do
11:         Find the indexes of the elements of $i$-th row in $D_{MN(sorted)}$ from $D_{MN}$
            and update $D_{MN(index)}$.
12:     end for
13:     Create a binary matrix, $B$ of size $M$-by-$N$ with all values are '0'.
14:     for $i = 1$ to $M$ do
15:         Randomly select row $r$ from $D_{MN(index)}$ and find $k =$
            $D_{MN(index)}(r, 1)$.
16:         Check the first column of every other rows if there are same index values,
            $k$ i.e., $D_{MN(index)}(j, 1) == k$ where $j \neq r$.
17:         if There are multiple rows with first columns having same index, $k$ then
18:             Form a set, $P_k$ of data points in $P$ such that $D_{MN(index)}(l, 1) ==$
                $k$ where $l \in P$ and $l$ includes $r$.
19:             Find out the data point $u \in P_k$ such that $D(u, k) == \min(d_{u,k})$
                where $u \in P$.
20:             if There are multiple $u$ then
21:                 Pick a $u \in P_k$ randomly.
22:                 Map $k \in W$ to $u \in P$ for $u \in P_k$ by putting $B(u, k) = 1$.
23:             else
24:                 Map $k \in W$ to $u \in P$ for $u \in P_k$ by putting $B(u, k) = 1$.
25:             end if
26:             Remove $u$-th row of $D_{MN(index)}$ and $k$ from every other row
                of the index matrix, $D_{MN(index)}$.
27:         else
28:             Map $k \in W$ to $r \in P$ by putting $B(r, k) = 1$.
29:         end if
30:         Remove $r$-th row of $D_{MN(index)}$ and $k$ from every other row of the
            index matrix, $D_{MN(index)}$.
31:     end for
32: end procedure
```

Then, we iteratively map a data point in P to W one by one starting from line 14 of Algo. 2. We randomly pick a data point $r \in P$ and try to map the closest point in W to it. This means we need to mark the corresponding element of B with it. Let $k = D_{MN(index)}(r, 1)$ be the data point in W that is closest to r. The operation we would like to do is $B(r, k) = 1$. However,

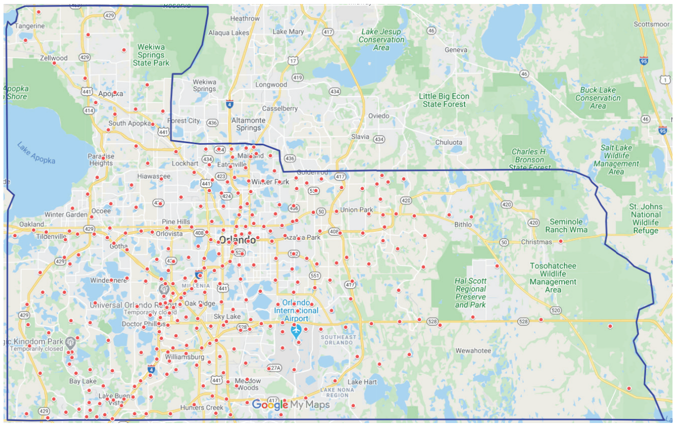

Fig. 6. Orange County

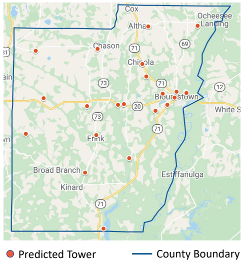

Fig. 7. Calhoun County

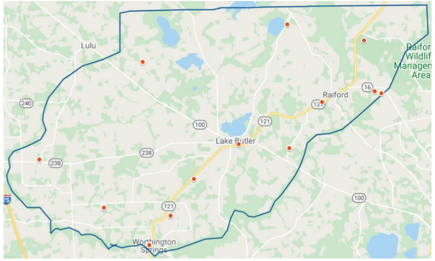

Fig. 8. Union County

it is possible that $k \in W$ is the closest data point to data points in P other than r, i.e., $D_{MN(index)}(j,1) == k$ where $j \in P$ and $j \neq r$. This is Case I of our earlier discussion on tentative issues during mapping. To resolve this case, for those data points in P closest to k in W, we compare their distances to k and pick the one with the shortest distance to k. At this iteration of the algorithm, let P_k be the set of all data points in P such that $D_{MN(index)}(j,1) == k$ where $j \in P$. We choose the data point $u \in P_k$ such that $D(u,k) == \min(d_{u,k})$ where $u \in P$. Then, map $k \in W$ to $u \in P$, i.e., $B(u,k) = 1$.

It is also possible that there are multiple u satisfying $D(u,k) == \min(d_{u,k}), u \in P$. This is Case II and, it means that the minimum of the closest distances to $k \in W$ happens for multiple data points in P. To resolve this tie, we randomly pick a $u \in P_k$ such that $D(u,k) == \min(d_{u,k}), u \in P$. Then, we map $k \in W$ to $u \in P$, i.e., $B(u,k) = 1$.

At the end of the above steps, let $r \in P$ be mapped to $k \in W$. We remove kth row of $D_{MN(index)}$ and k from every other row of the index matrix $D_{MN(index)}$ so that it does not get mapped to another data point in P in the subsequent iterations. This operation decreases the size of $D_{MN(index)}$ by 1 in both dimensions.

The steps above complete the mapping of one data point in P to another one in W. We continue to these mappings iteratively based on the updated $D_{MN(index)}$ and B. At the end, B holds the resulting one-to-one mapping of all data point in P to W, i.e., $\sum_{n=1..N} B(m,n) = 1, \forall m \in P$ and $\sum_{m=1..M} B(m,n) \leq 1, \forall n \in W$. Note that the algorithm assumes $M \leq N$, which requires that the data set with smaller size be given as P to the algorithm. This design of the algorithm is a minimalist approach as it may not necessarily inspect all possibilities and it just attempts to make a mapping to every data point in the smaller input set. We consider this strategy because of the number of active towers in FCC record [2] is lower than the total reported tower locations. It is also much fewer than the towers reported in *Antennasearch.com*.

C. Example

Consider two data sets P and W of Cartesian coordinates shown below, with M = 3 and N = 4 coordinates, respectively.

$$P = \begin{bmatrix} (0,0) \\ (1,0) \\ (2,0) \end{bmatrix} \quad W = \begin{bmatrix} (0,1) \\ (0,2) \\ (1,1) \\ (1,2) \end{bmatrix}$$

We initialize the distance matrix D_{MN}, a 3×4 matrix, as below:

$$D_{MN} = \begin{bmatrix} d_{1,1} & d_{1,2} & d_{1,3} & d_{1,4} \\ d_{2,1} & d_{2,2} & d_{2,3} & d_{2,4} \\ d_{3,1} & d_{3,2} & d_{3,3} & d_{3,4} \end{bmatrix} = \begin{bmatrix} 1 & 2 & \sqrt{2} & \sqrt{5} \\ \sqrt{2} & \sqrt{5} & 1 & 2 \\ \sqrt{5} & \sqrt{8} & \sqrt{2} & \sqrt{5} \end{bmatrix}$$

So, the $D_{MN(sorted)}$ and $D_{MN(index)}$ matrices will be:

$$D_{MN(sorted)} = \begin{bmatrix} 1 & \sqrt{2} & 2 & \sqrt{5} \\ 1 & \sqrt{2} & 2 & \sqrt{5} \\ \sqrt{2} & \sqrt{5} & \sqrt{5} & \sqrt{8} \end{bmatrix}$$

$$D_{MN(index)} = \begin{bmatrix} 1 & 3 & 2 & 4 \\ 3 & 1 & 4 & 2 \\ 3 & 1 & 4 & 2 \end{bmatrix}$$

We also initialize B to all zeros.

First Iteration: We see that the first elements of both rows 2 and 3 of $D_{MN}(index)$ are the same and equal to 3. So, this is a Case I and we choose the minimum of $d_{2,3}$ and $d_{3,3}$ in the D_{MN} matrix. We find that $d_{2,3} = 1$ and $d_{3,3} = \sqrt{2}$. So, $d_{2,3}$ is the one with minimum value. So, we mark 1 in the 3rd column of 2nd row in the binary matrix B and remove the row 2 and all the elements of $D_{MN}(index)$ which are 3 from other rows. So, the updated $D_{MN}(index)$ and B will be as below:

$$D_{MN}(index) = \begin{bmatrix} 1 & 2 & 4 \\ 1 & 4 & 2 \end{bmatrix} \; B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Second Iteration: Now, we see that the first elements of the row 1 and row 3 in $D_{MN}(index)$ are same and equal to 1. So, this is another Case I. We compare the distances $d_{1,1} = 1$ and $d_{3,1} = \sqrt{5}$. Here, $d_{1,1}$ is minimum and we remove row 1. We also remove '1' from the remaining row. Hence, the updated $D_{MN}(index)$ and B will be:

$$D_{MN}(index) = \begin{bmatrix} 4 & 2 \end{bmatrix} \; B = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Third Iteration: There is only one row remaining in $D_{MN}(index)$ which also means there is only one data point in P left to be mapped. We find that row 3 has 4 in the 1st column. So, we map the 3rd predicted location to the 4th web location. And, the final binary matrix is:

$$B = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

TABLE I
NUMBER OF PHYSICAL TOWERS IN THE STUDIED COUNTIES

Source	Orange	Calhoun	Union
AntennaSearch.com [11]	1,605	52	22
FCC [2]	430	26	15
FCC [2] after cleaning	326	21	13

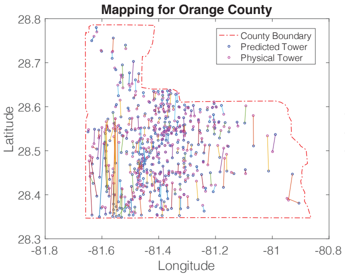

Fig. 9. Mapping in Orange County

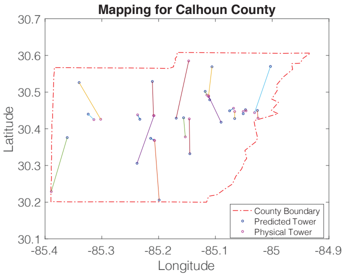

Fig. 10. Mapping in Calhoun County

So, we can find the mapped towers and the respective distances from matrix, D_{mapped}, as below. The matrix states that the 1st, 2nd, and 3rd locations (assume as predicted ones) are mapped the 1st, 3rd and 4th locations (physical towers).

$$D_{mapped} = \begin{bmatrix} d_{1,1} \\ d_{2,3} \\ d_{3,4} \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ \sqrt{5} \end{bmatrix}$$

D. Complexity of the Mapping Algorithm

If there is a huge amount of data points within a certain boundary, then the proposed mapping algorithm could take significant time to accomplish. Fortunately, we do not need to run the algorithm for huge geo-areas in real-time. Instead, we can divide areas and make run for each area, e.g., single county/state instead of an entire country at a time. So, the complexity of the algorithm would not be so problematic, but it is still desirable to reduce the complexity for potential close-to-real-time use in future. The computational complexity of the algorithm involves the following three major steps:

- Computation of the distance matrix, i.e., $\mathcal{O}(MN)$.
- Computation of the sorted and index matrices, i.e., $\mathcal{O}(MN\log N + MN)$.
- Computation of the final binary matrix after completing M iterations, i.e., $\mathcal{O}(M^2N)$.

Thus, the total computational complexity is $\mathcal{O}(MN + MN\log N + MN + M^2N)$ which can be simplified to $\mathcal{O}(MN(M + \log N))$.

V. RESULTS

We found the count of physical towers in the counties Orange, Calhoun, and Union, from two sources, i.e., *AntennaSearch.com* [11] and FCC site [2]. Table I shows the tower counts in these counties according to the two sources. The third row shows the counts after we remove dismantled or under-construction cell towers from FCC's count. We call this count as active tower count that we use for clustering.

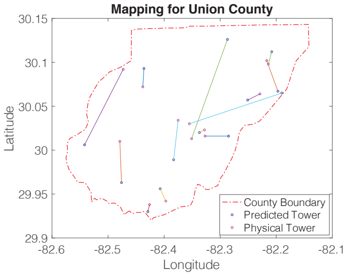

Fig. 11. Mapping in Union County

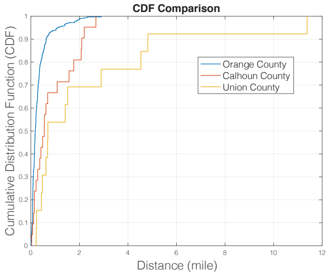

Fig. 12. Antennasearch towers

Initially, we worked with Matlab's default k-means clustering algorithm. However, it does not differentiate sample counts of the cell towers which inherently excludes population effects in predicting cell tower locations. It is expected to get more sample counts for a cell tower where more people live or use cellular services. Thus, high sample counts interpret high population counts. To include the population effect in the prediction, we utilize weighted k-means algorithm [10]. The advantage of using this algorithm is that it moves the centroid of a cluster towards high utilized towers or the towers around those more people live or use compared to other towers.

A. Predicted Towers on Google Maps

We apply weighted k-means algorithm on OpenCelliD data for predicting tower locations in Orange, Calhoun, and Union counties of Florida. The predicted towers on the maps of the counties are shown in Figs. 6, 7, and 8, respectively. Consider the predicted towers of Orange County on Google Maps. We choose it to analyze because of the high number of cell towers within the county. We can see from the map that predicted towers are mostly near the lines of the roads or the densely populated areas. We can also see that none of the predicted towers are in the deep forest or within a water body, seen in green and blue colors of the maps, respectively. We also observe the same pattern with the predicted towers of Calhoun and Union counties. The insights from the maps indicate the feasibility of our cell tower localization scheme and demonstrate a sign of predicting towers accurately.

B. Distance Between Predicted and Physical

Once we have the predicted tower locations for the three counties, we run our mapping algorithm Algo. 2 to map the towers with the physical towers reported in *AntennaSearch.com*. We repeat the procedure as many times as the predicted towers count in any county. We choose the mapping that has the minimum average distance between predicted towers to physical towers. The mapping of the towers for the Orange, Calhoun, and Union counties are shown in Figs. 9, 10, and 11, respectively. We observe that the more towers within a county, the better the prediction of their locations. A high number of predicted towers in Orange County is located within close proximity of the physical towers (CDF analysis in the following paragraph provides specific numbers). In Calhoun and Union counties, some of the predicted towers are close proximate of physical towers but some are far from their physical towers. This is likely mostly due to the data collecting procedures associated with OpenCelliD. As OpenCellid gets these data from the users voluntarily, the

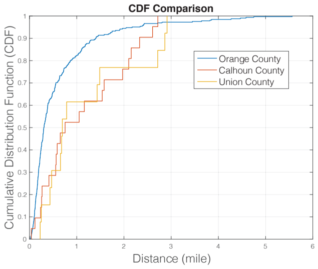

Fig. 13. CDF for FCC-reported towers

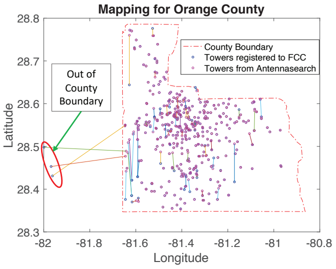

Fig. 14. *AntennaSearch.com* and FCC towers

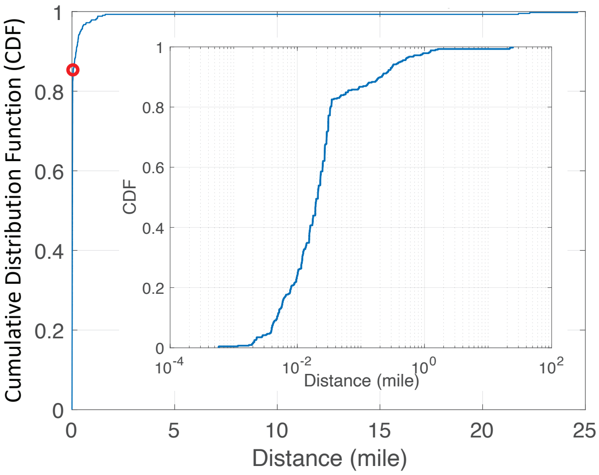

Fig. 15. Orange County: CDF for FCC and *AntennaSearch.com* towers

users are not interested to send data very often as they do not have any incentive for sharing their experience. In most cases, OpenCelliD contains a single-digit sample count which is not sufficient to identify an access point/antenna location properly. Hence, the locations reported in OpenCelliD can be already significantly away from an antenna if very few samples are taken for that particular antenna/cell.

The CDF of the distances between our predicted tower locations and their mapped physical tower locations from *Antennasearch.com* shows a better representation of the performance of our mapping and prediction. The CDF comparison for the three counties is shown in Fig. 12. Our approach attained notably better performance in predicting the tower locations for the average case in the urban county of Orange because of the high number of sample counts. We can see, almost 95% of physical towers are less than 1 mile away from their predicted locations, and 98% of physical towers are less than 2 miles away. We also generate CDF for the mapping of predicted towers to the physical towers reported in the FCC site in Fig. 13. It shows a similar trend of mapping in comparison to the CDF of *Antennasearch.com* towers. In Orange County, approximately 83% of the towers are less than 1 mile away from their predicted locations and 95% of the physical towers are less than 2 miles away. This illustrates the accuracy of our prediction and mapping process regardless of the sources of physical tower locations. Our algorithm performs better in mapping predicted towers to FCC registered towers compared to *Antennasearch.com* towers. It happens because *Antennasearch.com* approximates all access

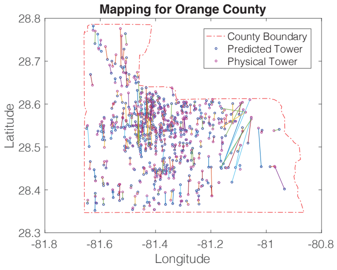

Fig. 16. Mapping among FCC-registered and predicted towers where k is cell sites count

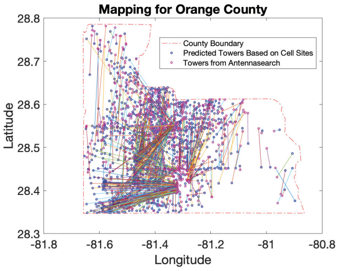
Mapping for Orange County
Latitude
Longitude
County Boundary
Predicted Towers Based on Cell Sites
Towers from Antennasearch

Fig. 17. Mapping among Antennasearch and predicted towers where k is cell sites count

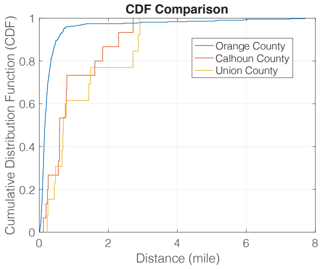

Fig. 18. CDF for FCC-registered towers where k is cell sites count

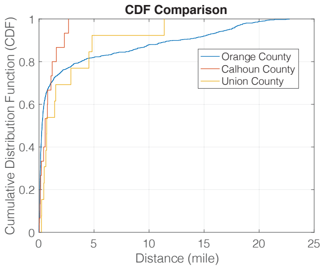

Fig. 19. CDF for Antennasearch towers where k is cell sites count

point locations where FCC provides accurate registered tower locations. As expected, due to the use of the crowdsourced data, our approach performs better when the mapping takes place with the high sample counts, i.e., in urban areas. For this reason, in the rural Calhoun and Union counties, we observe a lower percentage of predicted towers being within 2 miles of their corresponding physical towers.

C. *Lack of Ground-truth Tower Locations Data*

A crowdsourced data from actual users can represent better cellular coverage as well as help to predict cell tower locations. Though we have different sources of cell tower, still the reported data can be misled. As a regulatory body, FCC is expected to have updated data. Unfortunately, it is not always true. To analyze the locations of two trusted sources of towers (FCC and *AntennaSearch.com*), we map towers of both sources using our mapping algorithm and calculate the CDF of the mapping. To visualize the mapping, we consider all FCC reported towers (430 in total) for Orange county without discriminating their status, and perform one-to-one mapping with the *AntennaSearch.com* reported towers (1,605 in total), Fig. 14, and get the CDF in Fig. 15. We observe some of the FCC reported tower locations are actually outside of Orange county boundary in Fig. 14. It hints us on the reliability of entire data reported by each provider. However, this mapping also provides the accuracy on our mapping algorithm. We can see, almost all FCC towers are mapped identically with corresponding *AntennaSearch.com* towers. Fig. 15 shows a better representation of this claim. We observe, around 86% of FCC towers have exact mapping with the towers reported in *AntennaSearch.com*, and the rest have different mapping distances. In one side, FCC contains obsolete data and on the other side *AntennaSearch.com* data does not completely map with FCC data. Such unreliability motivates us to search actual perceived service by cellular users, utilize crowdsourced data, and construct predicted tower locations.

D. Predicted Towers Based on Cell Site Counts

We predict cell tower locations by using cell site counts [6] as k during clustering. We also map and compute CDF of the predicted tower locations to both registered towers at FCC and *AntennaSeach* data. The mapping of predicted towers for both cases in Orange county is shown in Fig. 16 and Fig. 17, respectively. We observe that the mapping of predicted towers to FCC registered towers offers better matching than the mapping of predicted towers to *AntennaSearch* reported towers. The CDF analysis of the mapping distances also supports our observation. FCC registered tower locations provide the actual physical locations of towers. However, for any third-party sources like *AntennaSearch*, the accuracy of reported tower locations may deteriorate with the increase of k. As a result, we find a significant deviation of CDF in Fig. 18 and Fig. 19. In Orange County, we find almost 80% of FCC registered towers are within 0.33 miles of our predicted towers. However, we get only 50% of the *AntennaSearch* towers within 0.33 miles of our predicted tower locations.

VI. Conclusion and Future Works

In this work, we have utilized location and sample counts of the OpenCelliD data, and used weighted k-means algorithm to predict cell tower locations. Later, we proposed a mapping algorithm to match our predicted tower locations to physical tower locations from two sources providing a limited number of tower locations. We demonstrated the accuracy of the prediction by comparing to the two different sources of tower locations. Our prediction shows a significant matching in both cases. We also demonstrated how these two sources of towers data can be unreliable, and why we need tower locations prediction from crowdsourced data.

Since exact locations of physical cell towers are either proprietary or partially available, there exists no ground truth. This necessitates mapping of predicted tower locations to an approximate/partial locations. We showed that this mapping of cell tower locations is a NP Complete. We reduced the mapping problem to another NP-Complete problem in polynomial time, and developed a heuristic for the mapping problem. In future, we plan to find better heuristic/approximation algorithms of this problem. It will give us more accuracy on predicting towers from crowdsourced data, and removes dependency on wireless provider-reported data to FCC or any third-party, e.g., *AntennaSearch.com*, reported data.

A good prediction of a cell tower location can provide a good idea of surrounding cellular coverage. It can be helpful to deploy network nodes in case of tower failure during disaster situations. To obtain each cellular provider's actual coverage map, crowdsourced data can be a truthful and alternative approach instead of wireless provider-reported coverage map which is mostly exaggerated. Crowdsourced database OpenCelliD contains a range value for each reported cell. Utilizing the range value with the predicted towers can give an excellent and reliable source of coverage map to generate in future.

References

[1] "Mobility Fund Phase II Coverage Maps Investigation Staff Report," https://docs.fcc.gov/public/attachments/DOC-361165A1.pdf, 2019.

[2] "Antenna Structure Registration," https://wireless2.fcc.gov/UlsApp/AsrSearch/asrRegistrationSearch.jsp.

[3] "FCC unveils $9 billion 5G Fund for rural America," https://www.rcrwireless.com/20191205/policy/fcc-unveils-9-billion-5g-fund-for-rural-america, 2019.

[4] "Cellmapper," https://www.cellmapper.net/.

[5] "OpenSignal," https://www.opensignal.com.

[6] "FCC Status Report," https://www.fcc.gov/document/hurricane-irma-communications-status-report-sept-18.

[7] J. A. Nordstrand, "Localizing Cell Towers from Crowdsourced Measurements," http://www.sciencedaily.com/releases/2016/04/160418092043.htm.

[8] "OpenCelliD," https://opencellid.org/.

[9] C.-L. Leca, I. Nicolaescu, and C.-I. Rîncu, "Significant location detection & prediction in cellular networks using artificial neural networks," *Computer Science and Information Technology*, vol. 3, pp. 81–89, 2015.

[10] T. Benham, "Fast K-means," https://www.mathworks.com/matlabcentral/fileexchange/31274-fast-k-means, 2020.

[11] "AntennaSearch.com," http://www.antennasearch.com/default.asp.

[12] A. Varshavsky, D. Pankratov, J. Krumm, and E. d. Lara, "Calibree: Calibration-free localization using relative distance estimations," in *Proc. of International Conference on Pervasive Computing*, May 2008.

[13] R. Cowell, "War Dialing and War Driving: An Overview," Global Information Assurance Certification, http://www.giac.org/paper/gsec/863/war-dialing-war-driving-overview/101791.

[14] Z. Li, "Towards practical data-driven network design," Ph.D. dissertation, UC Santa Barbara, 2019.

[15] A. Shokry, M. Torki, and M. Youssef, "DeepLoc: A ubiquitous accurate and low-overhead outdoor cellular localization system," in *Proc. of ACM SIGSPATIAL*, 2018, pp. 339–348.

[16] K. Chen, N. Pissinou, and K. Makki, "Cellular network location estimation via RSS-based data clean enhanced scheme," in *Proc. of IEEE Symp. on Computers and Communications*, 2011, pp. 924–930.

[17] A. A. Abdallah, S. S. Saab, and Z. M. Kassas, "A machine learning approach for localization in cellular environments," in *Proc. of IEEE/ION Position, Location and Navigation Symposium*, 2018, pp. 1223–1227.

[18] J. Yang, A. Varshavsky, H. Liu, Y. Chen, and M. Gruteser, "Accuracy characterization of cell tower localization," in *Proc. of ACM International Conference on Ubiquitous Computing*, 2010, pp. 223–226.

[19] A. V. Deshpande, M. H. Khandekar, C. G. Pethe, and J. Abraham, "Network recommendation based on route prediction and mobile tower localisation," in *Proc. of IEEE CCNC*, 2018, pp. 1–4.

[20] H. Wang, S. Xie, K. Li, and M. O. Ahmad, "Big data-driven cellular information detection and coverage identification," *Sensors*, vol. 19, no. 4, p. 937, 2019.

[21] S. Sundberg, "Localization of eNodeBs with a Large Set of Measurements from Train Routers," 2019.

[22] M.-R. Fida and M. K. Marina, "Uncovering mobile infrastructure in developing countries with crowdsourced measurements," in *Proceedings of Int. Conf. on Inf. & Comm. Technologies & Development*, 2019, pp. 1–11.

[23] Z. Li, A. Nika, X. Zhang, Y. Zhu, Y. Yao, B. Y. Zhao, and H. Zheng, "Identifying value in crowdsourced wireless signal measurements," in *Proc. of International Conference on World Wide Web*, 2017, p. 607–616.

[24] E. Alimpertis, N. Fasarakis-Hilliard, and A. Bletsas, "Community RF Sensing for Source Localization," *IEEE Wireless Communications Letters*, vol. 3, pp. 393–396, 2014.

[25] J. A. N. Rusvik, "Localizing cell towers from crowdsourced measurements," Master's thesis, The University of Bergen, 2015.

[26] "US County Boundaries," https://public.opendatasoft.com/explore/dataset/us-county-boundaries/table/.

[27] W. Cook and A. Rohe, "Computing minimum-weight perfect matchings," *INFORMS J. on Computing*, vol. 11, no. 2, pp. 138–148, 1999.