

Distributed versus Centralized Computing of Coverage in Mobile CrowdSensing

Michele Girolami¹, Alexander Kocian^{2*}, Stefano Chessa^{1,2}

¹ISTI-CNR, Institute of Information Science and Technologies, 56124 Pisa, Italy, Moruzzi, Pisa, 56124, Italy, Italy.

^{2*}Department of Computer Science, University of Pisa, Largo Bruno Pontecorvo, Pisa, 56127, Italy, Italy.

*Corresponding author(s). E-mail(s): alexander.kocian@unipi.it;
Contributing authors: michele.girolami@isti.cnr.it; stefano.chessa@unipi.it;

Abstract

The expected spatial coverage of a crowdsensing platform is an important parameter that derives from the mobility data of the crowdsensing platform users. We tackle the challenge of estimating the anticipated coverage while adhering to privacy constraints, where the platform is restricted from accessing detailed mobility data of individual users. Specifically, we model the coverage as the probability that a user detours to a point of interest if the user is present in a certain region around that point. Following this approach, we propose and evaluate a centralized as well as a distributed implementation model. We examine real-world mobility data employed for assessing the coverage performance of the two models, and we show that the two implementation models provide different privacy requirements but are equivalent in terms of their outputs.

Keywords: mobility, probability, privacy, point-of-interest

1 Introduction

Mobile CrowdSensing (MCS) [Capponi et al. \(2019\)](#); [Liu et al. \(2019\)](#); [Belli et al. \(2019\)](#) is a widely used computational paradigm designed to exploit data provided by the crowd, through their mobile devices, giving rise to a collaborative approach. More precisely, users, referred to as MCS subscribers, actively participate in the crowdsensing platform by installing a mobile app on their smartphones. Through this app, they can respond to specific tasks requested by the platform and receive services as a reward. A task involves activating sensors present in the smartphone, such as GPS, and can operate without

direct user intervention. In such cases, the app initiates data sampling from the sensors and sends the collected data to the MCS in accordance with the specified request, continuing until the task concludes or until the user manually stops it. Alternatively, certain tasks may necessitate explicit user engagement, requiring them to physically reach a specified location and perform a designated action, such as taking a picture at that particular place.

Given the widespread usage of smartphones, the potential for MCS to collect user-generated data on a broad and detailed scale, especially in urban areas, is very high [Cardone et al. \(2014\)](#); [Chessa et al. \(2017\)](#). This is coupled with the

advantage of requiring limited investments for the management and maintenance of the crowdsensing platform. Nevertheless, the effective coverage of the environment by such a platform remains constrained by the number of subscribed users along with their mobility. With the term *coverage*, we refer to the probability that a site is covered within a confidence region using sensed data. As a general observation, dense areas tend to have high coverage due to the presence of many users who can collect data. Conversely, peripheral regions might lack coverage, resulting in no expected data from such places. For these reasons, constructing a coverage map of the monitored environment is a promising approach to identify covered and uncovered regions. Subsequently, this can aid in planning a data-driven CrowdSensing data collection campaign.

In this study, we explore the methodology of constructing a coverage map for a crowdsensing platform by leveraging knowledge of user mobility. From a practical standpoint, the computation of the coverage map can be achieved through both a *centralized* and a *distributed* approach. In the former, the crowdsensing platform retrieves all mobility data from participating users, such as GPS traces, irrespective of their engagement in a task or the task’s requirement for mobility data. This approach, which was preliminarily investigated in our prior work [Girolami et al. \(2022a\)](#), albeit in a different context, namely Edge computing [Girolami et al. \(2022b\)](#); [Bellavista et al. \(2019\)](#), involves centrally analyzing the collected mobility data to derive the coverage map of the environment. Specifically, the crowdsensing platform designates a region as covered based on the frequency of user visits and the likelihood of users diverting toward that region. The higher the probability of users detouring towards a region, the greater the likelihood of collecting data from that specific area.

The centralized approach, however, raises privacy concerns for users, as it necessitates the disclosure of their mobility data to create a global coverage map. Therefore, in this study, we introduce an alternative distributed approach to construct a coverage map. In this latter approach, the crowdsensing platform only gathers aggregated data from users, comprising user-generated coverage maps. The individual users independently compute a coverage map by analyzing their

own trajectories. The coverage maps are then uploaded to the crowdsensing platform, which subsequently generates an aggregated-anonymous map. We analyze both models and provide analytical proof that the two models converge, meaning the aggregate map computed with the distributed approach is identical to the map computed with the centralized model.

We evaluate the performance of both models in an experimental setting utilizing real mobility data from a public dataset. Firstly, we define three experimental scenarios of increasing complexity, demonstrating the equivalence of the centralized and distributed models under similar conditions. Secondly, we examine the performance of the distributed model by varying the two key model parameters. Additionally, we analyze the dataset used for our experiments from a mobility perspective.

The remainder of this work is organized as follows. Section 2 frames the state-of-the-art of coverage computation with crowdsensing scenarios. Section 3 reports our reference scenarios and a first introduction to the centralized and distributed models. Section 4 formally describes the two models and Section 5 details our experimental settings.

2 Related Work

Network coverage, specifically the sensor set with the maximal residual energy to cover all points-of-interest, is a critical Quality-of-Service parameter, attracting significant research interest in recent years. Beginning with Wireless Sensor Networks (WSN), [Chen et al. \(2010\)](#) proposes sensor deployment strategies to maximize residual energy when covering all points-of-interests (POIs). The work by [Senouci et al. \(2012\)](#) introduces an evidence-based sensor coverage model that addresses deployment-related issues, including sensor reliability. [Yang et al. \(2019\)](#) suggests a model for coverage degree, representing the number of sensors required to cover a POI, specifically for visual WSNs deployed in line-of-sight communication [Yang et al. \(2019\)](#). [Akbarzadeh et al. \(2013\)](#) incorporates terrain information into visual WSN deployment, employing a probabilistic approach to model binary coverage. The coverage area can be significantly expanded when

end-devices operate in Multiple-Input Multiple-Output Space Division Multiplex (MIMO-SDMA) mode [Kocian et al. \(2017\)](#). However, mobility in WSNs is typically very limited.

The presence of mobile embedded sensors in smartphones and wearable devices enables the monitoring of environmental parameters on a large scale. User mobility, coupled with ubiquitous computing, enhances context awareness and understanding of user behavior compared to network-connected wireless sensor networks, thereby creating shared value. The success of MCS relies on the extensive participation of users. To effectively motivate users in sensing and reporting information, incentive mechanisms are employed [Capponi et al. \(2019\)](#). Various research works suggest three types of rewards that encourage users to share their data, including location tagging and activity tracking: i) gamification; ii) services; or iii) monetary rewards [Zhang et al. \(2016\)](#). Challenges in MCS encompass data quality in terms of accuracy, latency, and data security [Zhang et al. \(2021\)](#). A comprehensive review, focusing on taxonomy, applications, and challenges, can be found in [Boubiche et al. \(2019\)](#). Finally, an MCS campaign may deploy a combination of static and mobile sensors and utilize sensing interpolation strategies to ensure coverage in less frequented areas [Girolami et al. \(2017\)](#).

Focusing on MCS applications that explore network coverage, it is worth noting the work of [Wu et al. \(2021\)](#), which addresses fine-grained user profiling for personalized task matching. In contrast, [Xu et al. \(2022\)](#) employs a reinforcement-learning approach to maximize rewards in an online participant selection scheme, incorporating both area coverage ratio and degree. [Wang et al. \(2018\)](#) maximizes the future location coverage of the mobile crowd under a guaranteed location privacy protection scheme. The issue of Coverage-Aware Stable Task Assignment in Opportunistic MCS is addressed by [Yucel et al. \(2021\)](#). Finally, we want to highlight the coverage-guaranteed and energy-efficient participant selection strategy proposed by [Ko et al. \(2019\)](#).

Many current Mobile CrowdSensing (MCS) solutions heavily depend on some form of *centralized* data fusion. However, this approach encounters three issues: i) to guarantee spatial coverage of the targeted area, the MCS cloud requires location information of the assigned tasks, inevitably

revealing participants' locations; ii) raw sensor readings can be easily intercepted, leading to the disclosure of sensitive information to adversaries; iii) in scenarios where Internet bandwidth is low or the geographical environment causes deep channel fades, the number of active connections can overwhelm the network.

To address these challenges, network requirements are often better met by employing *distributed* or fog computing [Baresi et al. \(2016\)](#). The burden on back-end cloud servers can be reduced by providing distributed analysis for submitted data at the fog level [Jayaraman et al. \(2015\)](#). Based on the Quality-of-Service provided by participants, their selection can be offloaded to the mobile edge nodes [Lamaazi et al. \(2020\)](#). However, these frameworks assume trust among users, fog nodes, or cloud servers, which is not always the case. To address this issue, [Lamaazi et al. \(2022\)](#) proposes adopting a feedback mechanism to ensure cooperation between edge nodes, aiming to eliminate untrustworthy participants. To safeguard data in MCS, recent research has focused on credible and distributed incentive mechanisms based on blockchain technology [Kadadha et al. \(2020\)](#); [Chen et al. \(2022\)](#). Distributed Mobile CrowdSensing (MCS) has found numerous real-life applications. For instance, the application discussed in [Mowafi et al. \(2022\)](#) utilizes built-in cameras in smartphones to capture images of environmental disasters from various perspectives. Instead of transmitting all images to the server, they are exchanged with other nodes to avoid sharing redundant photos. This approach results in traffic reduction and energy savings exceeding 20% and 25%, respectively, compared to centralized fusion.

In the application highlighted in [Paricio and Lopez-Carmona \(2019\)](#), individual (static) traffic-weighted multimap creation is employed to precisely control vehicle routes. This multimap reflects link costs based on historic real-time data about the network, traffic status, and driver behavior obtained from distributed embedded measurement rootkits.

Another notable application of MCS involves influencing participant behavior. For example, in [Ji et al. \(2022\)](#), tools from the Mental Accounting Theory are deployed to create accounts for task execution profit and bonus, which are then used to motivate workers to alter their original

Table 1 An overview on recent distributed MCS: location of algorithm deployment and application.

Reference	architecture	application
Jayaraman et al. (2015)	fog	generic
Baresi et al. (2016)	fog	generic
Paricio and Lopez-Carmona (2019)	cloud	traffic control
Kadadha et al. (2020)	fog-cloud	trusted reputation
Qian et al. (2020)	edge	task allocation
Wang et al. (2020)	emulation	generic
Shao et al. (2021)	simulation	generic
Chen et al. (2022)	simulation	trusted reputation
Guo et al. (2022)	simulation	fake news
Ji et al. (2022)	simulation	generic
Lamaazi et al. (2022)	emulation	edge selection
Mowafi et al. (2022)	simulation	area coverage
Ning et al. (2022)	edge-cloud	traffic management
Zhou et al. (2022)	emulation	environmental data
Wu et al. (2023)	cloud	federated learning
our work	emulation	coverage map

travel schedules. In Guo et al. (2022), distributed MCS, combined with machine learning tools such as Bidirectional Encoder Representations from Transformers (BERT), is proposed for tracking down fake news.

To safeguard participant privacy, it is crucial to ensure the uncorrelation of their location data. In line with this approach, Qian et al. (2020) proposes obfuscating participants’ real locations as disturbed locations and achieving optimal task allocation based on these perturbed locations. Differential privacy and distributed user location obfuscation are suggested by Zhou et al. (2022). On the other hand, Shao et al. (2021); Wu et al. (2023) propose using k-anonymity to protect the privacy of datasets. Sparse Mobile CrowdSensing (MCS) with Differential and Distortion Location Privacy is addressed in Wang et al. (2020).

To simultaneously address user data security and system latency, the work presented in Ning et al. (2022) proposes a blockchain-enabled distributed MCS framework for traffic management.

3 System Model

In this study, we focus on a typical MCS scenario in which platform users are equipped with a mobile device, typically a smartphone. These devices are capable of performing various sensing tasks through an MCS app. The term *task* in this context refers to a specific action executed by a device, such as collecting sensing information using on-board sensors or capturing a picture with the embedded camera. Tasks can be autonomously performed by a device, or they may necessitate explicit user intervention. A MCS architecture also includes a back-end server in charge of:

- sending tasks to users: this action is required to start a data collection campaign. Some noteworthy examples include gathering information about the quality of WiFi networks in a particular area or collecting data on the noise intensity in a specific region;
- retrieving collected data: this action is necessary for retrieving and storing data collected by users. The data can be stored on the back-end and subsequently analyzed using post-processing analytics.

Typical MCS systems necessitate a mobile application to transfer the collected information from the user’s device to the back-end, similar to navigation, sports tracker, or recommendation apps. The amount of retrievable data and the regions from which data are collected depend on the number of participating users. Specifically, the higher the number of users in a MCS campaign, the greater the expected data and the broader the covered area. Refer to Table 2 for the adopted notation.

For the coverage, peripheral regions not visited by any user are likely to be uncovered, as no sensing information is collected from those areas using the user’s device. In contrast, popular areas are likely to be highly covered, as a significant amount of data is retrieved from user’s device. The concept of data coverage is prevalent in popular MCS applications like Google Maps. These applications are built on the idea of collecting and sharing user contributions, such as real-time traffic conditions, road accidents, or slowdowns.

We now turn our attention to the primary objective of this paper, which is to predict the coverage in an MCS system by leveraging user

mobility. We quantify the probability of collecting data from a specific set of locations $L = \{l_h : h \in [1, H]\}$, $|L| = H$ given users $U = \{u_k : k \in [1, K]\}$, $|U| = K$. Let the coverage $C^h \in [0, 1]$ be defined as the event where at least one user visits location $l_h \in L$. Moreover, let $\mathbb{P}(C^h)$ denote the probability of coverage at location l_h . The coverage map, designed to indicate the service area of all locations L , corresponds to the 2-dim. probability density function of the site coverage if a user is present in a region around the site. As an example, Figure 1 illustrates the coverage map for $H = 1954$ locations assuming that each site is equally likely visited. Each entry in the map is specified by the triple $(l_{h,x}, l_{h,y}, \frac{1}{H}\mathbb{P}(C^h))$.

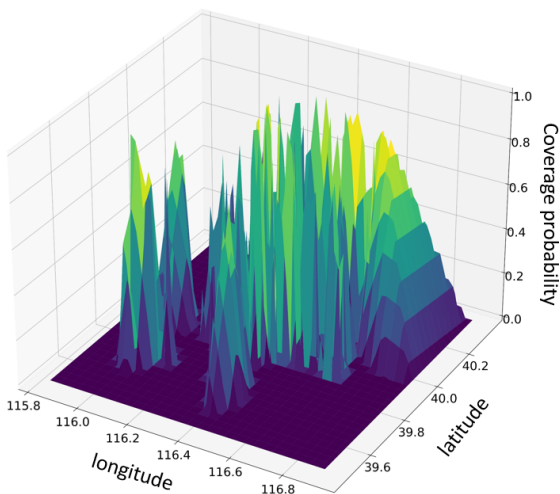


Fig. 1 Example of coverage map C for L locations, we report on x and y -axes the longitude and latitude and on the z-axis the coverage value for the corresponding location.

To achieve this, we propose a mobility-based approach to measure C^h by identifying highly visited or scarcely visited locations and, in turn, estimating the expected coverage. This approach assumes a positive correlation between a location’s visits and the expected coverage: the more likely a location is visited, the more likely a user will collect data from there.

As anticipated in the introduction, we consider two approaches for the prediction of the coverage that are based on *centralized* and on a *distributed* model, respectively. The centralized model requires that the back-end has the full

knowledge of the user’s mobility, e.g. the users’ trajectories as GPS coordinates. By analyzing the user’s mobility, it is possible to identify crowded areas and to assume that visited locations are also covered locations. The model also introduces the concept of *detour* as the possibility for a user to deviate from its original trajectory and to pass close to location $l_h \in L$. Detouring increases the possibility of collecting data from l_h , and hence of increasing the coverage C^h . MCS typically adopts this approach by providing reward to the participants Dasari et al. (2020); Hu et al. (2020). We report in Figure 2 a graphical representation of detour. In the figure we show 2 users: u_i, u_j and location l_h as a red box. Users travel along a different trajectory. We can identify for each user’s trajectory the nearest point j with respect to l_h , namely $d_{i,j}^h, d_{j,j'}^h$. Such points are those from which users will likely accept or decline a detour toward l_h . The advantage of the centralized model is the possibility of building an accurate coverage map, as full details of the user’s mobility are stored on the back-end. Nevertheless, the main drawback is that user’s mobility represents highly personal information, as it can reveal the identity of a subject Peng et al. (2017). Under this respect, it is worth to mention the strict rules applied with the GDPR framework.

The second model we study in this work is a distributed and privacy-preserving approach. The idea is to let users locally compute the coverage map for a given set of locations L and storing on the back-end only the coverage map for every user i . Differently from the previous model, the back-end does not require to store the user’s trajectories, rather the back-end only retrieves the coverage map from every user. It is worth to mention that the coverage map does not disclose detailed information about the user’s mobility, rather it only provides the probability for user i of visiting a specific location (locations L are the same for all the users). In turn, the back-end aggregates the retrieved coverage maps, building a final map. The two proposed models (centralized and distributed) provide the same result: a coverage map for the set L of locations. However, With the centralized model we assume to transfer personal data to the back-end, while with the distributed one we relax this assumption as users locally compute their partial coverage maps and

Table 2 The adopted notation for the coverage models.

Symbol	Description
U	User set
L	Location set
C	total coverage value
C^h	Coverage value of location l_h by any user
C_i	Coverage value of any locations by user i
$\mathbb{P}(C_i^h)$	Coverage map of location l_h by user i
$l_{h,x}$	x-coordinate of location l_h
$l_{h,y}$	y-coordinate of location l_h
$d_{i,j}^h$	Closest point of user i along trajectory j with respect to location l_h

then they transfer this aggregated information to the back-end.

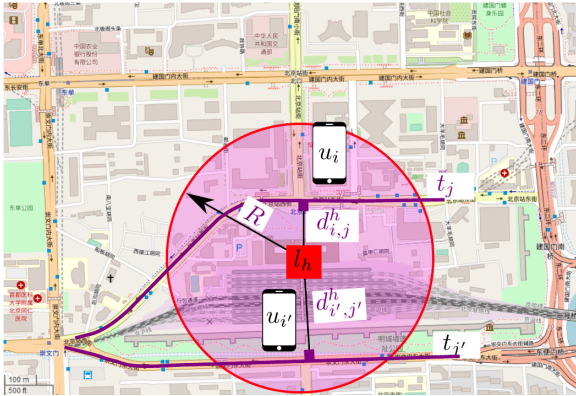


Fig. 2 Graphical representation of detour toward location l_h .

4 On the Computation of the Coverage Map

In this section we formalize the distributed coverage model introduced in Section 3, recap the centralized and demonstrate their equivalence.

4.1 Distributed Model

To start out, we need to make a few definitions and assumptions. For the sake of simplicity, the coverage area is resembled as circle with radius R and centered around the respective POI located at $l_h \in L$. Let the random variable $D_{i,j}^h$ capture the distance between the trajectory t_j of u_i and the POI at location l_h . To model the probability distribution for $D_{i,j}^h$, consider the following

scenario. Suppose that user u_i , moving along trajectory t_j chooses to leave the trajectory, to visit location l_h . The shorter the detour distance $d_{i,j}^h$, the more likely the users leave their trajectories. The exponential distribution is commonly used to model this behavior, as preliminary proposed in [Girolami et al. \(2022a\)](#). However, beyond coverage area, the users are too far off the POI. Hence, a better approach is to truncate the distribution at the edge of the coverage area R according to

$$p(t) = \begin{cases} \frac{\lambda \exp\{-\lambda t\}}{1 - \exp\{-\lambda R\}} & 0 < t \leq R \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

$\lambda > 0$. Finally, let us define the deterministic variable $d_{i,j}^h$ denoting the minimum distance between trajectory t_j of user u_i and location l_h . With the definitions made above, we are now ready to compute the probability of the event as the tail probability of (1),

$$\begin{aligned} \mathbb{P}(D_{i,j}^h \geq d_{i,j}^h) &= \int_{d_{i,j}^h}^{\infty} \frac{\lambda \exp\{-\lambda t\}}{1 - \exp\{-\lambda R\}} dt \\ &= \frac{\exp\{-\lambda d_{i,j}^h\}}{1 - \exp\{-\lambda R\}} \end{aligned} \quad (2)$$

Note that the closer the user, the higher the probability to detour. Subsequently, this probability is dubbed *Coverage probability*, to match the scope of the paper.

In the distributed model, the MCS back-end collects the coverage maps from the individual users instead of their mobility data. This approach limits the amount of sensitive information transferred from users to the back-end. As user u_i does not know the location of $u_{i'}$, $i \neq i'$, we start with the probability that user u_i along trajectory t_j visits cite l_h in (2). Given the users are present in a region around the site l_h , the joint probability that l_h is *not* covered by neither of the user's trajectory is the product of

$$\mathbb{P}(\overline{C_i^h}) = \prod_{t_j \in T_i} (1 - \mathbb{P}(D_{i,j}^h \geq d_{i,j}^h)) \quad (3)$$

when the trajectories of u_i are independent. Therefore, the probability that l_h is covered by at

least one trajectory of u_i has the form

$$\mathbb{P}(C_i^h) = 1 - \prod_{t_j \in T_i} (1 - \mathbb{P}(D_{i,j}^h \geq d_{i,j}^h)). \quad (4)$$

The coverage map from point-of-view of u_i is the probability of the union of the (mutually exclusive) events located at l_h and weighted by the density that a user is actually present in a region around the POI:

$$p(C_i) = \sum_{l_h} \mathbb{P}(C_i^h) p(l_h). \quad (5)$$

With no prior information available, the a-priori density of visited sites, $p(l_h)$ is to be assumed uniform on the number of possible locations H . The resulting 2-dim. density has the form of a matrix. The entry in the latitude-th row and longitude-th column corresponds to the coverage probability for the respective location. The pseudo-code of this distributed fusion algorithm in (5) is outlined in Algorithm 1.

Algorithm 1 Distributed computation of coverage map

```

1: function DISTRIBUTEDFUSION(L,  $u_i$ , T,  $\lambda$ , R)
2:    $C \leftarrow \mathbf{0}$ 
3:   for all  $l_h \in L$  do
4:      $x \leftarrow l_{h,x}$ 
5:      $y \leftarrow l_{h,y}$ 
6:      $C[x, y] \leftarrow 1$ 
7:     for all  $t_j \in T_i$  do
8:        $d_{i,j}^h \leftarrow \text{MINDISTANCE}(l_h, t_j)$ 
9:       if  $d_{i,j}^h \leq R$  then
10:         $C[x, y] \leftarrow C[x, y] \times$ 
         $(1 - (\exp\{-\lambda d_{i,j}^h\} / (1 - \exp\{-\lambda R\})))$ 
11:       end if
12:     end for
13:      $C[x, y] \leftarrow 1 - C[x, y]$ 
14:   end for
15:    $C \leftarrow 1/H \times C$ 
16:   return C
17: end function

```

4.2 Centralized Model

For the centralized model, the MCS back-end collects mobility data from all the users, to extract the coverage map, revealing the users' identity. We preliminary recap the coverage map of the centralized model that is used to benchmark the distributed model formulated above.

When the trajectories of all users are independent, it can be readily seen from (2) that the joint probability for the location l_h to be uncovered is given by

$$\mathbb{P}(\overline{C^h}) = \prod_{u_i} \prod_{t_j \in T_i} (1 - \mathbb{P}(D_{i,j}^h \geq d_{i,j}^h)). \quad (6)$$

Ergo, the probability that l_h is covered at least by one user yields

$$\mathbb{P}(C^h) = 1 - \prod_{u_i} \prod_{t_j \in T_i} (1 - \mathbb{P}(D_{i,j}^h \geq d_{i,j}^h)). \quad (7)$$

The coverage map corresponds to the probability of the union of the events that *any* site is covered weighted by the density of the users' presence in the region around the site i.e.,

$$p(C) = \sum_{l_h} \mathbb{P}(C^h) p(l_h). \quad (8)$$

4.3 Proof of Model Equality

The two proposed models provide the same outcome, a coverage map for a set of L locations. In this section, we formally proof that the centralized model and the distributed model are equivalent in terms of the obtained results. We start with the distributed model seen from the point-of-view of user u_i . When the users roam around independently, the probability that a given POI l_h is not covered by any user has the form

$$\mathbb{P}\left(\bigcap_{u_i} \overline{C_i^h}\right) = \prod_{u_i} (1 - \mathbb{P}(C_i^h)) \quad (9)$$

Substituting (3) for (9), it follows that

$$\mathbb{P}\left(\bigcap_{u_i} \overline{C_i^h}\right) = \prod_{u_i} \prod_{t_j \in T_i} (1 - \mathbb{P}(D_{i,j}^h \geq d_{i,j}^h)) \quad (10)$$

The probability that at least one user covers this POI equals to one minus above result, yielding

$$\begin{aligned}
& 1 - \mathbb{P} \left(\bigcap_{u_i} \overline{C_i^h} \right) \\
&= 1 - \prod_{u_i} \prod_{t_j \in T_i} (1 - \mathbb{P}(D_{i,j}^h \geq d_{i,j}^h)) \quad (11) \\
&= \mathbb{P}(C^h)
\end{aligned}$$

which is equal to the corresponding joint probability for POI l_h in (7) of the centralized model.

5 Experimental Settings

We now detail the experimental settings for the tests executed with the centralized and distributed models. The objective of the experiments described in this section is twofold. On the one hand, we experimentally demonstrate the equivalence of the 2 proposed models. As demonstrated in Section 4.3, the centralized and the distributed models provide the same results, but with different privacy levels. On the other hand, we focus on the distribute model as it guarantees a higher privacy level with respect to the centralized model, thus it represents a valid candidate for a real deployment in MCS systems.

5.1 The Experimental Mobility Dataset

Our experiments are based on a real-world experimental mobility dataset, namely GeoLife [Zheng et al. \(2009, 2008\)](#). The dataset has been collected by Microsoft Research Asia, and it involves about 182 participants recruited on a volunteer basis, as shown in Figure 3.

The collected data include GPS coordinates of participants obtained with GPS trackers or, in some cases, with the user’s device. The dataset covers an extended period from April 2007 to August 2012. Some of the user’s trajectories are labelled with the adopted transportation mean, e.g., car, bus, metro etc. Data comes from Beijing area; the dataset is publicly available and widely adopted in the current literature.

We show in Figure 4 how the number of GPS points vary along the time. The graph reports the variation of the number of GPS traces aggregated on a monthly basis as reported in Figure 5.

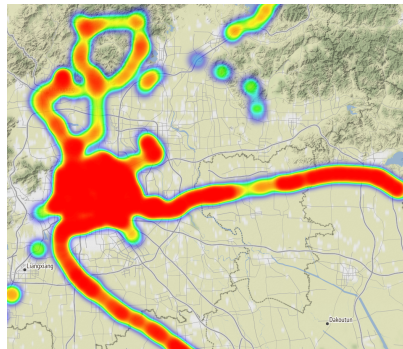


Fig. 3 Overview of the GeoLife dataset (Map data copyrighted OpenStreetMap contributors and available from <https://www.openstreetmap.org>).

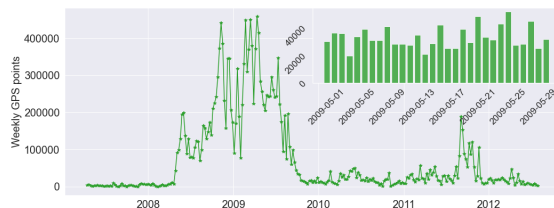


Fig. 4 Weekly number of GPS points.

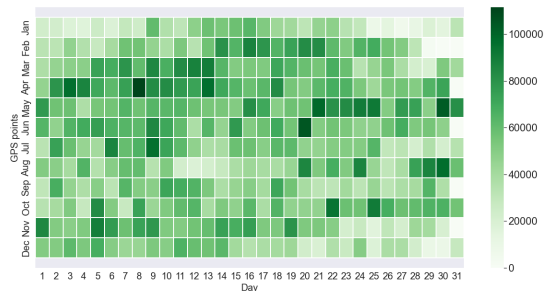


Fig. 5 Number of GPS data aggregated per month.

From Figure 4 we can observe that the time windows starting from late 2008 to early 2010 represents the densest period in terms of available data. For the purpose of this work, we restrict the analysis to a sub-period, and we also restrict the geographic extension of the dataset. In particular, we focus on 2009 and we crop the data to Beijing city ¹. The inset in Figure 4, shows the variation

¹min lat = 39.54, max lat = 40.3, min lon = 115.75, max lon = 117.13, EPSG 4326 reference system

of GPS collected data only in May 2009. During this limited period, the dataset provides 982,304 GPS points for a total of 737 distinct trajectories.

5.2 Evaluation Metrics

To quantitatively measure the similarity between the centralized and the distribute models (see Section 4.3), we empirically generate coverage maps and compare their distributions. There are several statistical tests available in the literature such as the two-sample Anderson-Darlington and the two-sample Kolomogorov-Smirnov (KS) tests. Both tests are non-parametric, and they do not require normality either. We have chosen to rely on the latter option, though, as it is simpler to implement. Subsequently, we briefly review the KS test Wang and Wang (2010).

To get started, the two empirical coverage maps are transformed to cumulative density functions (CDFs). The KS test accepts two empirical CDFs and returns two parameters, namely the so-called KS statistic and the p-value. The former statistic corresponds to the maximum absolute deviation between the two empirical CDFs. To compute the latter value, suppose that the null hypothesis is that the two sample vectors are drawn from populations with same distribution. Then, the p-value is the probability that the KS statistics is greater than the observed value under the null hypothesis of no difference between class levels or samples. When the two datasets arise from the same (continuous) distribution, the KS statistics and the p-value should converge to 0 and 1, respectively.

5.3 Experimental Results

We first assess the equivalence of the two models. To this purpose, we consider three scenarios of increasing complexity and of increasing duration. The goal is to compute the coverage maps resulting from the centralized and distributed models, and to execute the KS test, as detailed in Section 5.2. The result of the KS test provides a reliable indication of the model’s equality.

The considered scenarios are all based on data extracted from the GeoLife dataset (see Section 5.1, as reported in Figure 4. The scenarios we consider are the following:

- Scenario 1: May 1, 2009 to May 7, 2009;

Table 3 Results of the KS test for the three test scenarios.

	Scenario 1	Scenario 2	Scenario 3
KS	0.0009	0.0009	0.0009
p-value	0.99	0.99	0.99

- Scenario 2: May 1, 2009 to May 15, 2009;
- Scenario 3: May 1, 2009 to May 31, 2009;

Concerning the locations of interest L , we extract from the Beijing area a collection of places labeled with the following tags: **square**, **monument**, **mall**, **station** and **bus**, extended with a set of random points from the city center. As a result, we obtain 1954 locations that are exploited to compute the coverage map with the two models. For optimized visualization, both models are configured with detour radius $R = 800$ and a scale parameter $1/\lambda = 50$. Table 3 lists the result of the KS test applied to the three scenarios. In all cases the p-values are close to one and hence, the two models are equivalent as anticipated in Section 4.

We also show a graphical representation of the coverage maps obtained with the centralized and distributed models. Figure 6 shows a 3D representation of the models for the 3 scenarios. The graphs report for each location $l_h \in L$, the corresponding coverage value (z-axis).

Now, our focus shifts to the distributed model, which serves as a privacy-preserving approach for computing coverage. In particular, we examine two orthogonal aspects: the impact of the scale $1/\lambda$ and the detour radius R on the resulting coverage maps. To achieve this, we vary the scale parameter within the range: [50, 100, 500, 800] and R in the range: [100, 200, 500, 800]. Results have been obtained by considering Scenario 2, ranging from May 1, 2009 to May 15, 2009 but similar results also apply to the other scenarios. The impact of the scale parameter is to modify the truncated exponential distribution reported in Equation 1 and shown in Figure 7. In particular, the scale parameter affects the dispersion of the exponential distribution that can be used to model the *preferential* distance at which a user will likely accept a detour toward a location. We observe that, the smaller the scale, the more users will accept a detour only at short distances. Differently, the higher the scale, the more likely users will accept a detour also at high distances from the target location.

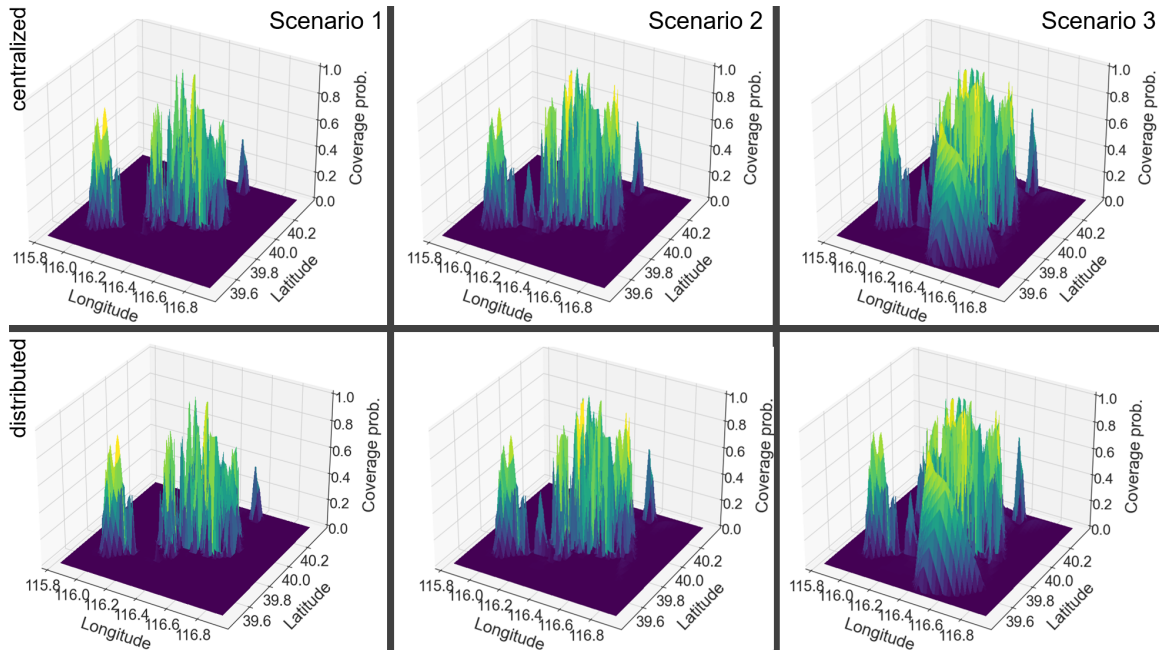


Fig. 6 3D representation of the coverage map with the centralized and distributed models and for the 3 testing scenarios.

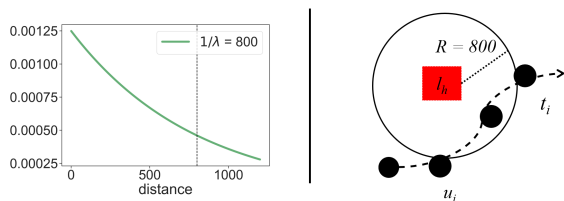


Fig. 7 Impact of the scale parameter and detour radius to the distribute model.

The detour radius R is employed to determine which points along the user’s trajectory are included or excluded, as illustrated in Figure 7. When R increases, more points can be included in the coverage computation for a given user, while a contraction in R results in a reduction of the included points. The results of the variations in scale and R are depicted in Figure 8: the rows display the scale variations in the interval 50 to 800, while the columns show the variations in R from 100m to 800m. From the figure, we observe two distinct trends. On one hand, increasing the scale does not significantly alter the resulting distributions; they still exhibit the same trend. On the other hand, increasing R has the effect of modifying the distributions. As R increases, the peak

of the distribution shifts from left to right, indicating that locations become highly covered with higher values of R .

6 Discussion and Conclusions

The adoption of the MCS paradigm has enabled the collection of representative datasets by leveraging sensing units on user devices. The volume of data and the set L of regions from which data are collected are closely tied to user mobility. Typically, regions that are frequently visited are considered *covered* in terms of the data collected from those areas. In contrast, peripheral regions with low visitation rates are anticipated to be uncovered. However, a crowdsensing platform may incorporate incentive mechanisms to motivate users to detour towards specific regions.

One approach to evaluate the coverage of a MCS system over a set of L locations involves collecting all data related to users’ mobility, such as GPS trajectories, at the centralized server of the MCS. This information is then utilized to identify highly/scarcely visited regions and apply a detour probability. However, this model raises inherent privacy concerns, as it necessitates all users to disclose their personal mobility, potentially serving

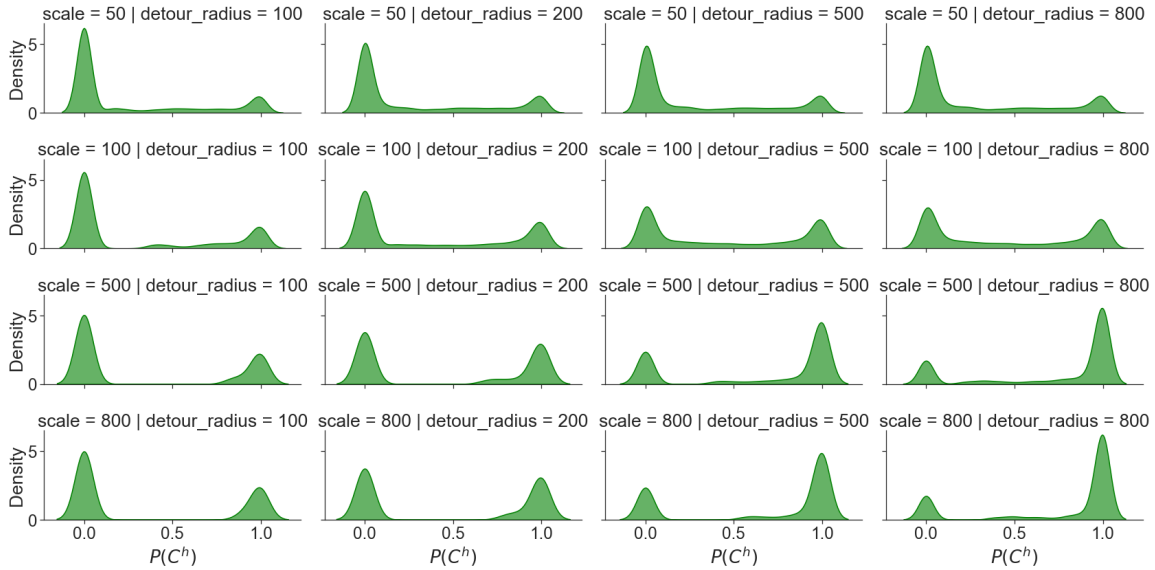


Fig. 8 Analysis of the impact of λ and of R to the distributed model.

as a deterrent to user participation in the MCS platform.

In an effort to provide stronger privacy guarantees for users, we propose an alternative distributed model for computing coverage. This distributed model is based on the concept of collecting aggregated and anonymized data to construct the coverage map for the set of L regions. In this case, users only upload their coverage maps without including any information about personal mobility. Subsequently, the obtained coverage maps are merged in the cloud to determine the aggregated map.

To evaluate the performance of the two models, we compare their outputs using a publicly available mobility dataset, namely GeoLife. Specifically, we define three experimental scenarios of increasing complexity and apply the KS test to assess their equivalence. Our tests reveal that the two models perform similarly but with different privacy requirements. Furthermore, we analyze the performance of the distributed model by varying two core settings: the detour radius and the scale parameter.

It should be noted, however, that while the distributed model enhances user privacy, it introduces a potential vulnerability due to malicious or unfaithful users who may inject poisoned location data into the system. The purpose of such actions could be to alter the resulting coverage

map and launch attacks on the Mobile CrowdSensing (MCS) platform. This vulnerability exists in both the centralized and distributed models. However, in the centralized model, the server possesses the GPS locations of individual users, providing more opportunities to filter out poisoned data. In contrast, the distributed model involves the exchange of poisoned data with other honest users, potentially affecting the local coverage maps computed by numerous users over time. This is a limitation of the current work.

Therefore, in our ongoing and future work, we aim to delve deeper into understanding the implications of malicious users participating in an MCS initiative. Specifically, we are investigating scenarios where the intentional dissemination of inaccurate coverage maps by malicious users, which do not reflect the actual mobility patterns, could adversely impact the aggregated coverage map computed by the back-end server. In addressing this concern, our plan involves the identification of attacks initiated by users through data poisoning. We intend not only to develop measures for identifying these malicious users but also to implement countermeasures for purging the collected data tainted by false coverage maps. Additionally, future efforts will also focus on leveraging local, short-range communication technologies such as Bluetooth or WiFi to facilitate the exchange and

merging of coverage maps among users in proximity.

Declarations

Conflicts of interest

The authors declare that they have no conflict of interest.

Data availability

The datasets generated during and/or analyzed during the current study are available in the GeoLife repository, <https://www.microsoft.com/en-us/research/publication/geolife-gps-trajectory-dataset-user-guide/>

References

- Akbarzadeh V, Gagne C, Parizeau M, et al. (2013) Probabilistic sensing model for sensor placement optimization based on line-of-sight coverage. *IEEE Trans Ins and Meas* 62(2):293–303. <https://doi.org/10.1109/TIM.2012.2214952>
- Baresi L, Guinea S, Mendonca DF (2016) A3droid: A framework for developing distributed crowdsensing. In: 2016 IEEE Int. Conf. Pervasive Computing and Communication Workshops (PerCom Workshops), pp 1–6, <https://doi.org/10.1109/PERCOMW.2016.7457103>
- Bellavista P, Belli D, Chessa S, et al. (2019) A social-driven edge computing architecture for mobile crowd sensing management. *IEEE Comm Mag* 57(4):68–73
- Belli D, Chessa S, Kantarci B, et al. (2019) Toward fog-based mobile crowdsensing systems: State of the art and opportunities. *IEEE Comm Mag* 57(12):78–83
- Boubiche DE, Imran M, Maqsood A, et al. (2019) Mobile crowd sensing – taxonomy, applications, challenges, and solutions. *Computers in Human Behavior* 101:352–370. <https://doi.org/https://doi.org/10.1016/j.chb.2018.10.028>
- Capponi A, Fiandrino C, Kantarci B, et al. (2019) A survey on mobile crowdsensing systems: Challenges, solutions, and opportunities. *IEEE Comm Surv & Tut* 21(3):2419–2465. <https://doi.org/10.1109/COMST.2019.2914030>
- Cardone G, Cirri A, Corradi A, et al. (2014) The participact mobile crowd sensing living lab: The testbed for smart cities. *IEEE Comm Mag* 52(10):78–85
- Chen J, Li J, He S, et al. (2010) Energy-efficient coverage based on probabilistic sensing model in wireless sensor networks. *IEEE Comm Letters* 14(9):833–835. <https://doi.org/10.1109/LCOMM.2010.080210.100770>
- Chen S, Li B, Rui L, et al. (2022) A blockchain-based creditable and distributed incentive mechanism for participant mobile crowdsensing in edge computing. *Mathematical Biosciences and Engineering* 19(4):3285 – 3312. <https://doi.org/10.3934/mbe.2022152>
- Chessa S, Girolami M, Foschini L, et al. (2017) Mobile crowd sensing management with the participact living lab. *Perv and Mob Comp* 38:200–214
- Dasari VS, Kantarci B, Pouryazdan M, et al. (2020) Game theory in mobile crowdsensing: A comprehensive survey. *Sensors* 20(7). <https://doi.org/10.3390/s20072055>
- Girolami M, Chessa S, Adami G, et al. (2017) Sensing interpolation strategies for a mobile crowdsensing platform. In: 5th IEEE Int. Conf. Mobile Cloud Comp., Service, and Eng. (MobileCloud), pp 102–108, <https://doi.org/10.1109/MobileCloud.2017.8>
- Girolami M, Pacini T, Chessa S (2022a) Evaluation of a location coverage model for mobile edge computing. In: ICC 2022 - IEEE Int. Conf. Comm., pp 5011–5016, <https://doi.org/10.1109/ICC45855.2022.9838963>
- Girolami M, Vitello P, Capponi A, et al. (2022b) A mobility-based deployment strategy for edge data centers. *Journal of Par and Dist Comp* 164:133–141. <https://doi.org/https://doi.org/10.1016/j.jpdc.2022.03.007>
- Guo Y, Lamaazi H, Mizouni R (2022) Smart edge-based fake news detection using pre-trained bert

- model. p 437 – 442. <https://doi.org/10.1109/WiMob55322.2022.9941689>
- Hu J, Yang K, Wang K, et al. (2020) A blockchain-based reward mechanism for mobile crowdsensing. *IEEE Trans Computational Social Systems* 7(1):178–191. <https://doi.org/10.1109/TCSS.2019.2956629>
- Jayaraman PP, Bártolo Gomes J, Nguyen HL, et al. (2015) Scalable energy-efficient distributed data analytics for crowdsensing applications in mobile environments. *IEEE Trans Computational Social Systems* 2(3):109–123. <https://doi.org/10.1109/TCSS.2016.2519462>
- Ji G, Yao Z, Zhang B, et al. (2022) Quality-driven online task-bundling-based incentive mechanism for mobile crowdsensing. *IEEE Transactions on Vehicular Technology* 71(7):7876 – 7889. <https://doi.org/10.1109/TVT.2022.3170505>
- Kadadha M, Otrok H, Mizouni R, et al. (2020) Sensechain: A blockchain-based crowdsensing framework for multiple requesters and multiple workers. *Future Generation Computer Systems* 105:650–664. <https://doi.org/https://doi.org/10.1016/j.future.2019.12.007>
- Ko H, Pack S, Leung VCM (2019) Coverage-guaranteed and energy-efficient participant selection strategy in mobile crowdsensing. *IEEE IoT Journal* 6(2):3202–3211. <https://doi.org/10.1109/JIOT.2018.2880463>
- Kocian A, Badiu MA, Fleury BH, et al. (2017) A unified message-passing algorithm for MIMO-SDMA in software-defined radio. *EURASIP Journal on W Comm and Networking* 2017(1):4. <https://doi.org/10.1186/s13638-016-0786-y>
- Lamaazi H, Mizouni R, Singh S, et al. (2020) A mobile edge-based crowdsensing framework for heterogeneous iot. *IEEE Access* 8:207524 – 207536. <https://doi.org/10.1109/ACCESS.2020.3038249>
- Lamaazi H, Mizouni R, Otrok H, et al. (2022) Trust-3dm: Trustworthiness-based data-driven decision-making framework using smart edge computing for continuous sensing. *IEEE Access* 10:133095 – 133108. <https://doi.org/10.1109/ACCESS.2022.3231549>
- Liu Y, Kong L, Chen G (2019) Data-oriented mobile crowdsensing: A comprehensive survey. *IEEE Comm Surv & Tut* 21(3):2849–2885. <https://doi.org/10.1109/COMST.2019.2910855>
- Mowafi M, Awad F, Al-Quran F (2022) Distributed visual crowdsensing framework for area coverage in resource constrained environments. *Sensors* 22(15). <https://doi.org/10.3390/s22155467>
- Ning Z, Sun S, Wang X, et al. (2022) Blockchain-enabled intelligent transportation systems: A distributed crowdsensing framework. *IEEE Trans Mob Comp* 21(12):4201–4217. <https://doi.org/10.1109/TMC.2021.3079984>
- Paricio A, Lopez-Carmona MA (2019) Urban traffic routing using weighted multi-map strategies. *IEEE Access* 7:153086–153101. <https://doi.org/10.1109/ACCESS.2019.2947699>
- Peng T, Liu Q, Wang G (2017) Enhanced location privacy preserving scheme in location-based services. *IEEE Sys Journal* 11(1):219–230. <https://doi.org/10.1109/JSYST.2014.2354235>
- Qian Y, Jiang Y, Hossain MS, et al. (2020) Privacy-preserving based task allocation with mobile edge clouds. *Information Sciences* 507:288 – 297. <https://doi.org/10.1016/j.ins.2019.07.092>
- Senouci MR, Mellouk A, Oukhellou L, et al. (2012) An evidence-based sensor coverage model. *IEEE Comm Letters* 16(9):1462–1465. <https://doi.org/10.1109/LCOMM.2012.070512.120999>
- Shao Z, Wang H, Zou Y, et al. (2021) From centralized protection to distributed edge collaboration: A location difference-based privacy-preserving framework for mobile crowdsensing. *Security and Communication Networks* 2021. <https://doi.org/10.1155/2021/5855745>
- Wang F, Wang X (2010) Fast and robust modulation classification via kolmogorov-smirnov test. *IEEE Trans Comm* 58(8):2324–2332. <https://doi.org/10.1109/TCOMM.2010.08.090481>

- Wang L, Qin G, Yang D, et al. (2018) Geographic differential privacy for mobile crowd coverage maximization. AAAI'18/IAAI'18/EAAI'18
- Wang L, Zhang D, Yang D, et al. (2020) Sparse mobile crowdsensing with differential and distortion location privacy. *IEEE Transactions on Information Forensics and Security* 15:2735 – 2749. <https://doi.org/10.1109/TIFS.2020.2975925>
- Wu F, Yang S, Zheng Z, et al. (2021) Fine-grained user profiling for personalized task matching in mobile crowdsensing. *IEEE Trans Mob Comp* 20(10):2961–2976. <https://doi.org/10.1109/TMC.2020.2993963>
- Wu H, Zeng H, Zhou T, et al. (2023) From eye to brain: A proactive and distributed crowdsensing framework for federated learning. *IEEE Internet of Things Journal* 10(9):8202 – 8214. <https://doi.org/10.1109/JIOT.2022.3230050>
- Xu Y, Wang Y, Ma J, et al. (2022) Psare: A RL-based online participant selection scheme incorporating area coverage ratio and degree in mobile crowdsensing. *IEEE Trans Vehicular Technology* 71(10):10923–10933. <https://doi.org/10.1109/TVT.2022.3183607>
- Yang X, Wen Y, Yuan D, et al. (2019) Coverage degree-coverage model in wireless visual sensor networks. *IEEE W Comm Letters* 8(3):817–820. <https://doi.org/10.1109/LWC.2019.2894667>
- Yucel F, Yuksel M, Bulut E (2021) Coverage-aware stable task assignment in opportunistic mobile crowdsensing. *IEEE Trans Veh Tech* 70(4):3831–3845. <https://doi.org/10.1109/TVT.2021.3065688>
- Zhang S, Ray S, Lu R, et al. (2021) Preserving location privacy for outsourced most-frequent item query in mobile crowdsensing. *IEEE Internet of Things Journal* 8(11):9139 – 9150. <https://doi.org/10.1109/JIOT.2021.3056442>
- Zhang X, Yang Z, Sun W, et al. (2016) Incentives for mobile crowd sensing: A survey. *IEEE Comm Surv & Tut* 18(1):54–67. <https://doi.org/10.1109/COMST.2015.2415528>
- Zheng Y, Li Q, Chen Y, et al. (2008) Understanding mobility based on GPS data. In: Proc. 10th Int. Conf. Ubiquitous Computing. ACM, UbiComp '08, p 312–321, <https://doi.org/10.1145/1409635.1409677>
- Zheng Y, Zhang L, Xie X, et al. (2009) Mining interesting locations and travel sequences from GPS trajectories. In: Proc. 18th Int. Conf. WWW. ACM, p 791–800, <https://doi.org/10.1145/1526709.1526816>
- Zhou T, Cai Z, Su J (2022) Discovering truth in mobile crowdsensing with differential location privacy. p 903 – 908, <https://doi.org/10.1109/GLOBECOM48099.2022.10001292>