Privacy-Utility Trades in Crowdsourced Mobile Network Data Obfuscation

Jiang Zhang, Lillian Clark, Matthew Clark, Konstantinos Psounis *Fellow, IEEE*, Peter Kairouz

Abstract—Cellular providers and data aggregating companies crowdsourced mobile network data from user devices to measure network performance. Recognizing that this data collection may be at odds with growing awareness of privacy concerns, we consider obfuscating such data before the data leaves the mobile device. The goal is to increase privacy such that it is difficult to recover sensitive features from the obfuscated data, e.g. user whereabouts, while still allowing network providers to use the data for improving network services, e.g. create accurate signal maps. To examine this privacy-utility tradeoff, we identify privacy and utility metrics and threat models suited to mobile network data. We then obfuscate data using several preeminent techniques, spanning differential privacy, generative adversarial privacy, and information-theoretic privacy techniques, in order to benchmark a variety of promising obfuscation approaches and provide guidance to real-world engineers who are tasked to build systems that protect privacy without hurting utility. Our evaluation results, based on multiple, diverse, real-world mobile network datasets, demonstrate the feasibility of concurrently achieving adequate privacy and utility, with obfuscation strategies which use the structure and intended use of datasets in their design, and target average-case, rather than worst-case, guarantees.

Index Terms—Crowdsourced mobile network data, data obfuscation, privacy-utility tradeoff

1 Introduction

Network providers and data aggregating companies crowdsource mobile user data for a variety of reasons. This data can reveal network performance, allow for the generation of signal strength maps, inform decisions on where to deploy cell towers or sensors, and provide insight on how to improve user experience. The measurements are collected directly from user devices, via standalone mobile apps [1], or measurement software development kits [2] integrated into popular partnering apps. Providers and aggregators then sell this data to network operators, regulators, and device and equipment manufacturers. For the operators, regulators, and manufacturers, this crowdsourced data offers clear value for network planning. For the user, contributing data can in turn be useful, given that it leads to better network performance. However, participation also raises legitimate privacy concerns.

For example, some cellular providers have allegedly been selling their users' real-time location data to credit agencies, bail bondsmen, and other third parties [3]. Furthermore, while these measurements are assumed to be sparse in space and time and over thousands of users, previous work has shown that identities are inferable from anonymized data [4].

In recent years, privacy issues have come to the front of news, politics, and public opinion [5], [6], [7] and pioneering privacy laws have been enacted [8], [9]. To protect user privacy, a plethora of data masking, or obfuscating, schemes have been proposed, see, for example, [10]. However, by obfuscating the original data for the sake of privacy, data can no longer provide the exact insights it once could, sacrificing data utility for privacy [11].

In this work we examine the privacy-utility tradeoff in the context of mobile network data measurements, focusing on device-level obfuscation where the data is obfuscated, or privatized, before it leaves the user's phone. The goal is to increase privacy such that it is difficult to recover sensitive features from the obfuscated data, e.g. user whereabouts, while still allowing network providers to use the data for improving network services, e.g. create accurate signal maps. To examine this privacy-utility tradeoff, we identify privacy and utility metrics and threat models suited to the application at hand. We then obfuscate data using using a number of promising approaches at the forefront of privacy research, in order to benchmark them and provide guidance to real world engineers who are tasked with building systems that provide (some) privacy while maintaining (adequate) utility. To evaluate the different approaches, we use multiple, diverse, real-world mobile network datasets to ensure real world applicability of our findings.

We implement four strategies for obfuscating data to assess and compare their application-specific performance, selecting preeminent methods from the literature that span a range of complexities and privacy guarantees. Specifically, the first is a noise-adding privatizer, which adds independent, identically distributed Gaussian noise across the features of the data. Albeit simple, this scheme provides intuition into the privacy-utility tradeoff via the choice of how much noise to add. The second is based on differential privacy (DP) [12], a leading approach to data obfuscation which provides probabilistic worst-case guarantees against any arbitrary adversary, including one with unlimited resources and access to side-information. In this work we apply the popular local Gaussian mechanism [12], as well as the recent Truncated Laplacian Mechanism [13]. The

Lillian Clark, Konstantinos Psounis and Jiang Zhang are with the University of Southern California.
Peter Kairouz is with Google.
Matthew Clark is with The Aerospace Corporation.

third leverages the idea of generative adversarial networks to allow a data-driven method of learning an obfuscation scheme. This method, which is referred to as generative adversarial privacy (GAP) [14], positions a privatizer and an adversary, both modeled as neural networks, against each other. The privatizer learns to obfuscate the data such that the adversary cannot infer sensitive features, and the adversary simultaneously learns to infer sensitive features. While this method cannot offer the formal worst-case guarantees of the differentially private methods, the learning approach offers the potential to leverage structure in the data set and take advantage of the specific utility objectives in the network. The fourth strategy is motivated by an information-theoretic treatment of the problem. Considering mutual information as a convex metric for privacy performance, we frame a formal optimization problem as finding the obfuscation strategy which maximizes privacy subject to a constraint on utility. This approach maximizes user privacy in an average sense, but sacrifices the worst-case guarantees offered by the deferentially private methods. Section 5 discusses these privatizers in more detail.

We analyze the performance of each of these privatizers using three, diverse, real-world mobile network datasets. The first one is collected from cellular users over a seven month period in the city of Chania, Greece [15]. The second one is collected over a period of four months by Android smartphones in the University of California Irvine campus [16]. The last one is sampled from the Radiocell dataset [17], one of the largest publicly available datasets with millions of measurements from nearly one million macrocells around the world. The sample we work with contains mobile network measurements from hundreds of users over a one year period in UK's countryside. Section 3.1 discusses these datasets in detail.

An important aspect of our study is to identify privacy and utility metrics (Section 4) as well as threat models (Section 3.2) suited to mobile network data. We assess our obfuscation schemes against specific adversaries, modeled as neural networks (Section 3.5 discusses adversary models in detail), which estimate private user information from observing obfuscated data, and we take the adversary's estimation performance as a practical, application-specific privacy metric. We also consider more robust privacy guarantees, such as DP, which is not dependent on any specific adversary implementation. With respect to utility, we consider two metrics. First, we consider a received signal strength (RSS) model which accurately predicts signal maps when trained with unobfuscated data. We train this model with the obfuscated data. Then, we use as an application-specific utility metric the L1 distance between the parameters of the RSS model trained with obfuscated versus unbofuscated data. As a general utility metric, we use the overall assessment of data distortion. This serves as a proxy for utility under a wide variety of other potential mobile data applications.

The main contributions of this work are: (1) a framework for formalizing privacy and utility for crowdsourced mobile network data, (2) a systematic exploration of the parameterized privacy-utility tradeoff, (3) an analysis and comparison of four obfuscation schemes, and (4) an evaluation of the feasibility of achieving different notions of privacy in the mobile network data setting. Our findings show that while local DP provides privacy guarantees under worst-case conditions, it comes with a substantial cost in utility. GAP and IT privacy can offer significantly improved privacy-utility tradeoff by sacrificing worst-case guarantees and incorporating application-specific context, such as structure in the datasets and network objectives.

In the next section, we briefly discuss relevant work in privacy, especially as it relates to mobile network data. Section 3 describes our system model, including the three real-world datasets that we use in our evaluation, the threat models we consider, the privatizer and adversary model we implement, and the service provider model we consider. Section 4 rigorously defines our privacy and utility metrics. Section 5 presents each of the four obfuscation schemes. In Section 6, we evaluate and compare these schemes, and analyze our results. We discuss the limitations and future works in Section 7, and present our conclusions in Section 8.

2 Related Work

2.1 Privacy mechanisms

Differential privacy (DP) [12], [18], [19] is a mathematically rigorous definition of privacy which is useful for quantifying privacy loss and designing randomized algorithms with privacy guarantees. Motivated by statistical disclosure control, or providing accurate statistics while protecting individual survey respondents, DP approaches the problem of releasing coarse-grained information while keeping fine-grained details private. This popular approach to data privatization is studied under the local [20] and global models. The global model assumes a trusted data analyst has access to the dataset and wants to release queries computed on it in a privacy-preserving fashion. The local model assumes the absence of a trusted server, thus the data is randomized prior to aggregation. This work applies the local model of differential privacy.

Generative Adversarial Privacy (GAP) [14], [21] offers an alternative to noise-adding mechanisms in that it is context-aware, meaning it takes the dataset statistics into account. GAP learns from the dataset without the need to explicitly model the underlying distribution. Leveraging recent advancements in generative adversarial networks (GANs) [22], [23], [24], GAP allows the privatizer to learn obfuscation schemes from the dataset itself. Like the generator and discriminator in a GAN, the privatizer and adversary optimize a minimax game.

Information-theoretic (IT) privacy [11], [25], [26] provides an alternative in which privacy metrics are motivated by concepts from information theory. For example, mutual information [27] is the measure of how much one random variable tells us about another. Obfuscation schemes which minimize mutual information intuitively provide privacy. Unlike DP which provides guarantees on worst-case privacy, mutual information is an expectation, i.e. provides guarantees on average privacy.

2.2 Theoretical studies of privacy-utility trades

Previous work has analyzed distortion in the context of DP [28] or attempted to minimize the utility loss incurred by

DP [29]. Previous work in GAP maximizes privacy subject to a constraint on distortion [14]. Additionally, previous IT privacy metrics have been considered in the context of theoretically motivated utility metrics [30]. In contrast, in this work we consider utility metrics beyond distortion which are specific to our application and are both more intuitive and relevant for mobile network data. We also formally compare the performance of context-free (Gaussian noise-adding, local DP) and context-aware (GAP, IT) approaches in the context of our application.

Prior theoretical studies on the privacy-utility tradeoff include [11], [31], [32]. The authors of [11] formally define an analytical model for trading equivocation (a privacy metric based on Shannon entropy) and distortion (a utility metric which could be Euclidean distance, Hamming distortion, Kullback-Leibler divergence, etc). This model is designed for "universal" metrics, but is not generalized for non-i.i.d. datasets or datasets lacking strong structural properties. A so called geometric mechanism is presented in [32] as a utility-maximizing alternative to the Laplace or Gaussian mechanisms typically used in differential privacy, where utility is the expected loss of any symmetric, monotonic loss function. In [31], the authors define a bound on the information-theoretic min-entropy leakage of ϵ-differential privacy, and a bound on utility (where utility is roughly the number of differing dataset entries). Our work uniquely examines this tradeoff for all of these approaches in the unifying context of a single application, allowing us to present additional insight.

2.3 Prior work on mobile network data privacy

Previous work on privacy in mobile network data has considered strategic sampling, distribution modeling, and noise addition as obfuscation strategies. In [10], the authors exploit compressive sensing techniques to sample and compress received signal strength values in a privacy-preserving RSS map generation scheme. While privacy is gained in sampling and compression, the authors of [10] do not take a formal approach to quantifying privacy. In [33], distributed algorithms for Gaussian and exponential noise addition are explored in a crowdsourced data setting. Local differential privacy is applied to the user-tracking problem for indoor positioning systems in [34]. The authors of [35] present a relaxed version of differential privacy, probabilistic DP, which accounts for certain worst-case privacy scenarios being highly unlikely. They apply this to the generation of synthetic data which maps daily commutes. In [36], a novel privacy-preserving incentive mechanism is proposed for mobile crowd sensing, where the authors employed DP to perturb aggregated data. In each of [10], [33], [34], [35], [36], utility is not rigorously considered. Our work takes a formal approach to both privacy and utility.

In recent years, researchers have grown interested in studying the privacy-utility tradeoff in mobile network applications. Reza et al. in [37] propose an optimal strategy against location attack based on Stackelberg Bayesian game theory, which provide the best location privacy while satisfying the user's service quality requirements. Nicolás et al. in [38] formulate the tradeoff optimization problem between geo-indistinguishability and quality of service, and proposed a method based on linear optimization to solve this problem. In [39], the authors design novel privacy and utility metrics for location privacy, and perform large-scale evaluation and analysis of several existing location-privacy preserving mechanisms. However, these works only focus on location privacy without considering other privacy metrics for mobile user data. Moreover, the proposed approaches in [37] and [38] are based on linear programming and discrete locations, which cannot be easily applied under our threat models (continue locations and non-linear adversary), and, [39] is an empirical study with no formal analysis or obfuscation schemes that formally consider both privacy and utility in their design.

In [40], the authors propose a novel framework, DP-star, for publishing trajectory data with differential privacy guarantees, while preserving high utility. In [41] the authors present AdaTrace, a utility-aware location trace synthesizer which provides a differential privacy guarantee and inference attack resilience. This work is closely related to ours in that the authors employ both learning and noise-adding to generate datasets which they evaluate for statistical utility, and analyze how the choice of privacy parameter effects utility. In [42], the authors proposed a privacy-preserving and utility-aware mechanism based on mutual information optimization, with application to the data uploading phase in participatory sensing. However, these works only consider the database-level threat model during dataset publishing, which requires a trust-worthy third party to distort data before release, and they cannot be directly applied to the device-level obfuscation in our application.

In [43] and [44], GANs are leveraged to achieve utility-aware obfuscation of mobile sensor data. However, these works focus on obfuscating image sensor data to reduce sensitive information leakage in mobile apps, where both the dataset structure and threat model are different from ours. Moreover, [44] does not compare GANs with other formal obfuscation schemes, and [43] does not compare against obfuscation schemes that formally consider both privacy and utility in their design.

While the advantages and disadvantages of a range of obfuscation methods are to some extent known in principle from prior work, how they compare in any given mobile network data application is unclear. In this work, we implement representative obfuscation schemes based on preeminent approaches, apply them to the important real-world application of generating signal maps via crowd-sourcing, and compare their performance. Our performance results can serve as benchmarks, offering insights about how to design real-world systems to generate accurate signal maps while protecting user privacy.

3 System Model

Fig. 1 illustrates the system model we consider, which involves mobile users, a service provider or a third party, and an adversary. User devices record network measurement data and transmit it to a service-provider or third-party server. Since the reported data contains information that the users may deem private (e.g. user location, see Section 3.1), users apply device-level privatizers to obfuscate their data locally before uploading them to the server (see Section 3.3).

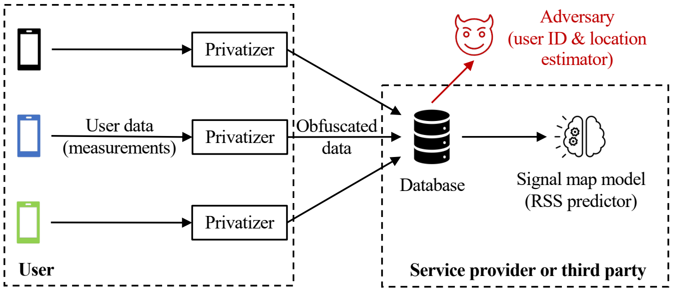


Fig. 1: Overview of system model. 1) Users collect measurement data and obfuscate it before uploading it to the service provider; 2) The service provider or third party aggregates obfuscated user data to train a signal map model, i.e. RSS prediction model; 3) The adversary has access to the obfuscated user data and uses it to estimate the user ID and locations.

The goal of the service provider is to train an RSS model based on the aggregated obfuscated user data, which can be used to generate signal maps and thus guide network planning and operation [45] (see Section 3.6). Finally, an adversary with access to the obfuscated data estimates the whereabouts of users, by estimating the user ID and location corresponding to the incoming measurements (see Section 3.5). Note that we assume the adversary has access to the obfuscated data as it arrives at the server, but no side information that directly reveals the identity of users (see Section 3.2 for a detailed description of the threat model).

3.1 User Data

We use three real-world datasets collected from different countries and over different time periods to evaluate the performance of our privatization schemes under different environments and user behaviors, and thus make our findings more conclusive.

The first dataset is taken from users in Chania, Greece, and will be referred to as the Chania dataset, which contains measurements from nine users over seven months in 2014. The nine users are mobile device owners who carry their devices with them throughout the day collecting measurements. Each measurement contains 24 features: device address, timestamp (to the second), received signal strength (RSS) in dBm, latitude, longitude, cellID identifying the base station, downlink carrier frequency, uplink carrier frequency, mobile network code, etc.

The second dataset contains measurements from seven users over four months in the University of California Irvine (UCI) campus in 2017, and will be referred to as the UCI dataset. Each measurement consists of 15 features including latitude, longitude, reference signal received power (RSRP) in dBm, reference signal reference quality (RSRQ) in dBm, timestamp. deviceID, cellID, etc.

The third dataset is collected by Radiocell.org [17], which has been crowdsourcing wireless network measurements from world-wide mobile users since 2009. It is the largest open-source mobile network dataset we can have access to. We sample about 0.5 million measurements from 219 mobile users in UK, 2019[1], and refer to it as Radiocell dataset. Each measurement has 23 features including latitude, longitude, altitude, speed, signal strength (SS) in dBm, country code, mobile network code, etc.

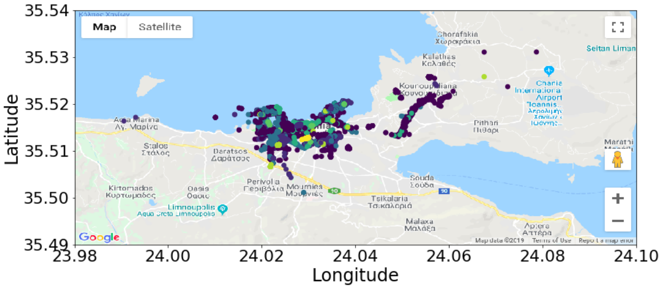


Fig. 2: Chania Dataset (colored by user)

The most relevant features to this paper are tabulated in Table 1 along with an indicator of their sensitivity. User ID and location are assumed sensitive features (private), whereas RSS/RSRP and others are not sensitive (public).

TABLE 1: Dataset Features

Features	User ID	Latitude	Longitude	RSS	Others
Sensitivity	Private	Private	Private	Public	Public
Variable	u	x_1	x_2	x_3	$x_{j,\ j>3}$

For visualization purposes, we have plotted the data of the first dataset over the geographic region in Fig. 2. The colors indicate user ID, and it is apparent that one cannot easily assume user ID based on location alone.

3.2 Threat Model

Our adversary has the goal of gathering private information that may be revealed by users operating in the mobile data network who are sending data reports to the service provider. The adversary may use this information for purposes not in the users' interest, or even to aid criminal attacks such as identity theft. To accomplish his/her goal, the adversary will seek to obtain access to as many user feature reports as possible, consisting of $(u, x_1, x_2, x_3, ...)$. Since the primary information sought by the adversary may not be explicitly present in the reports, e.g., if the reports are intentionally obfuscated, the adversary will perform inference attacks to estimate the private user information they desire. The nature of the threat may have some variation dependent on the specific mobile data application and the capabilities of the adversary. With this in mind, we consider the following properties as part of the definition of the threat model:

- Whether the adversary can access individual user reports directly, or whether their access is limited to the aggregated reports of all users,

[1] We choose UK since most of the collected measurements in 2019 come from mobile users in UK (10 million measurements in total). To limit computational complexity, we select three cells containing the largest amount of data.

- whether the adversary should be assumed to have bounded computational resources,
- whether the adversary has access to relevant side information, and
- whether users are primarily concerned with potential exposure of private information from their reports on average or in the worst-case.

Side information is any additional information that may be available to an adversary that could be used to supplement the information collected from the user reports to increase the efficacy of an inference attack. This could include public databases from organizations like the US Census Bureau or the Department of Transportation which allow an adversary to associate data features, e.g., addresses with names.

Typical mobile network data threat model: For most mobile network data applications and users, we apply the following threat model:

- The adversary can access individual user reports directly,
- the adversary's computational resources are bounded,
- the adversary has limited access to side information, and
- users are primarily concerned with privacy exposure on average.

We consider that many users are likely to have reservations about providing private data to a service provider, either because they do not trust the provider to adequately protect their data or they believe the service provider will themselves use the data in ways that do not align with the user's interests. For this reason, we assume a threat model where users must be able to protect their private information at the local level, e.g., at the user device. We also recognize that some users likely will not have such reservations, and thus a minority of users can be incentivized, e.g., through discounts, to trust a data aggregator with their data, allowing for the possibility of training or tuning privacy schemes based on real user data. Adversary computational resources are assumed to be bounded, recognizing that other methods outside the scope of the data network could be employed to reliably obtain the same private information if an adversary is assumed to have limitless resources. Adversary access to side information is assumed to be limited for the same reason. Finally, we assume users will typically be concerned with the exposure of their private information on average. For most mobile data applications, a user will likely operate with the network over a long period of time and will generate many feature reports as a result. Further, exposure of the private data of any one report will typically pose a much lower risk than exposure through the aggregation of many reports over a period of time. Thus, protecting against an adversary attack on any single report under worst-case conditions is unnecessary for typical applications.

Worst-case mobile network data threat model: Due to the wide variety of potential mobile network data applications and possible user privacy concerns, we acknowledge there may be some use cases where a worst-case threat model is appropriate. To account for this, we also treat such a model in our analysis. This adversary can access individual reports directly, but in contrast to the typical threat model we assume the adversary has unbounded computational resources and unlimited access to side information. Also, users are concerned with exposure of any single feature report, and their private information in each report must be protected from exposure under worst-case conditions.

3.3 Data Obfuscation and Privatizers

To protect against the adversary threat, privacy can be preserved through obfuscation of the feature data provided by individual users before being released to the service provider. At a minimum, the feature set is stripped of user ID. Remaining features are then obfuscated according to the selected privatization scheme, or ″privatizer″ for short. This is needed because the adversary may learn patterns in the data which associate public and private features, thus it is not sufficient to only obfuscate private features.

The privatizer will produce an obfuscated feature report $(u, x_1, x_2, x_3, \ldots) \rightarrow (y_1, y_2, y_3, \ldots)$, with y_i denoting the obfuscated version of x_i, where the mapping depends on the design of the privatizer. We will consider several privatizers, described fully in Section 5. Some privatizers leverage actual user data in their design. We assume such data is collected either through opt-in surveys and service provider incentives, or else collected by the provider through other means such as wardriving. In our analysis, we use 70% of our available dataset for training our adversary (see Section 3.5 for more details) as well as for training, fitting models, and/or choosing parameters of the privatizers (see Section 5 for more details). The remainder of the dataset will be used to test our privatizers against the adversary.

3.4 Context

User data is a type of application-specific context, and different privatizers may use the actual data, data distributions, or merely data moments like mean and variance. There are other types of application-specific context, e.g. privacy and utility metrics of interest, which privatizers may optimize over. Since mobile service providers know what they want to use the data for, and may ask their clients about privacy concerns, such metrics may indeed be available to be used in the design of privatizers.

Using context has implications to the threat model. For example, optimizing over a particular privacy metric guarantees protection against this privacy metric but not against any function of the data. As another example, if a privatizer optimizes its design under a known data distribution, or is trained under a given dataset, its performance is not guaranteed under different distributions and datasets.

Using context may also offer utility guarantees since optimizing over, or putting a constraint on a utility metric, restricts the privatizer from making obfuscation decisions that reduce utility below acceptable levels.

Table 2 compares different privatizers with respect to how much context they use and which threat model properties they can protect against. LDP offers stronger privacy protection than the rest as it provides worst-case privacy guarantees against any adversary with potentially unlimited resources and side information access. However, it does not have a formal mechanism to guarantee a minimum level of utility. In contrast, GAP and IT are aware of application

		LDP	*GAP*	*IT*	
Threat model	*Adversary computational resources*	Unlimited	Limited	Unlimited	
	Adversary side-info access	Unlimited	Limited	Unlimited	
	Type of privacy-loss guarantee	Worst-case	On-average	On-average	
	Provable adversary privacy protection	Against any adversary	Against trained adversary	Against any adversary	Context
	Privatizer access to data for training	Not necessary but helpful*	Yes	No	
	Privatizer access to data distribution	Not necessary but helpful*	No	Yes	
	Utility protection type	None/Some*	Maximize utility	Lower bound on utility	

* As discussed in detail in Section 5.2, LDP requires clipping. While clipping can be done in a manner which is agnostic to the data [46], [47], this may result in large utility loss. As a result, clipping is usually performed using information about the data to ensure the added noise is calibrated with the range of data values, see Eq. (16).

TABLE 2: Context used by privatizers (last 4 rows) and properties of threat models (first 6 rows).

specific utility metrics which they include in their optimization setups, and thus provide utility guarantees. The GAP privatizer in particular optimizes a multi objective function which considers both privacy and utility. That said, it is optimized and can offer formal guarantees only against the particular adversary in its training loop. These fundamental distinctions among the different obfuscation approaches are discussed in more detail in Section 5 and their implications to the privacy-utility tradeoff are presented and discussed in detail in Section 6.

3.5 Adversary Model

Depending on whether users upload measurements to the server one at a time or in batches, the adversary may or may not know whether a sequence of measurements originated from a single user or multiple users. Consider first the scenario where each user uploads one obfuscated measurement each time. Given that the user ID of each obfuscated measurement is unknown, the adversary takes as input one measurement from the obfuscated dataset $(y_1, y_2, y_3, \ldots)$ and predicts the user ID and true location (the unobfuscated latitude and longitude) from which the measurement originated $(\hat{u}, \hat{x}_1, \hat{x}_2, , \hat{x}_3 \ldots)$. Now consider the scenario where, for the sake of reduced system complexity, each user uploads a sequence of obfuscated measurements each time[2]. While the user ID of each obfuscated measurement in the database is unknown, the adversary knows that measurements in the same batch belong to the same user, and can take advantage of correlations across measurements to improve estimation. In this case the adversary takes as input a measurement sequence $\{(y_{1i}, y_{2i}, y_{3i}, \ldots)\}_{i=1}^{i=L}$ from a single user and predicts a single user ID $\hat{u}$ and the true locations $\{(x_{1i}, x_{2i})\}_{i=1}^{i=L}$ (i denotes the i^{th} measurement in this sequence, and L is the sequence length). In Section 6 we investigate the performance under both scenarios, see Section 6.2 for a direct comparison between the two.

The adversary estimation is a mapping from $(y_1, y_2, y_3, \ldots)$ to $(\hat{u}, \hat{x}_1, \hat{x}_2)$ and one may use a number of approaches to perform that mapping. In theory, one may discretize the continuous x_i's and y_i's and use empirical conditional probabilities and maximum likelihood estimation, but in practice the state space would explode. Given the availability of real world datasets, learning is a better choice. We experimented with linear and non-linear models for used ID estimation, and chose a deep neural network (DNN) to model our adversary (see Fig. 3), given the effectiveness of DNNs in approximating non-linear functions.

Specifically, our adversary is modeled as a fully-connected DNN containing two hidden layers with 256 neurons each. Between layers we employ Rectified Linear Unit (ReLU) activations, and our optimization relies on Adaptive Moment Estimate (Adam) stochastic gradient descent with a learning rate of 0.001. These values were empirically selected to maximize the adversary's performance when given the unobfuscated data as input.

Assume that the input measurement contains m features and there are k users (m and k depend on three datasets described in Section 3.1). Then each input batch has n measurements containing the m features. The output of the adversary neural net is a $n \times (k+2)$ matrix representing estimates of user ID and location ($n = 1024$ in our experiments). Each row in this matrix contains the likelihood that this measurement belongs to different users, and the estimated latitude and longitude of the original measurement. The loss function used to train the adversary is a weighted sum of the categorical cross entropy loss of the user ID estimate vector and the euclidean distance between the actual location and the location estimate. The user ID estimate error, location estimate error, and adversary loss functions are defined in Section 4.

We provide our adversary 70% of the obfuscated dataset to train on, for which it has access to the unobfuscated user IDs and locations, and test it on the remaining 30% of the data. Providing the adversary such a high portion of the data for training makes our privacy results conservative. In our threat model we have assumed some access to side information but comprehensively modeling access to possible forms of side information is intractable. The adversary's access to 70% of the dataset with obfuscated and true user ID and location labels serves as an approximation of some form of side information. Side information may include known user locations at certain times, or inputs from the adversary's own devices on the network to establish ground truth. The training set could also correspond to the adversary simulating published privatizers which may be revealed by the service provider to help convince users regarding their ability to preserve privacy.

[2] In practice, the service provider can require users to upload their data weekly or monthly.

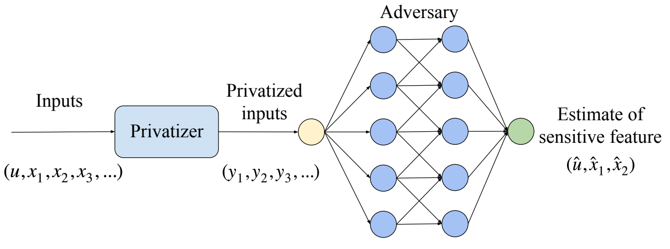

Adversary
Inputs
Privatizer
Privatized inputs
Estimate of sensitive feature
$(u, x_1, x_2, x_3, ...)$
$(y_1, y_2, y_3, ...)$
$(\hat{u}, \hat{x}_1, \hat{x}_2)$

Fig. 3: A diagram of adversary implementation.

3.6 Signal Map Model

The service provider trains an RSS predictor based on the the aggregated user data such that it can generate accurate signal maps. Specifically, the model input features include (obfuscated) latitude, longitude and other features (i.e. $(x_1, x_2, x_j, j > 3)$), and the model output is the RSS value x_3 in dBm.

There is a long line of research on RSS predictor models, see, for example [48], [49], [50]. We first consider a simple path loss model [51] but find its accuracy to be underwhelming. We also consider a linear and a neural network model and find that both have good comparable accuracy yet the former is easier to work with. Notably, its parameters can be estimated in one step which allows us to calculate application-specific utility metrics more efficiently (see Section 4.2). We thus select a linear RSS prediction model. Specifically, we use the following model:

$$x_3 = a_0 + \sum_{j=1, j\neq 3}^{j=m} a_{j-1} x_j, \tag{1}$$

where k is the total number of features in a measurement and $\alpha = [a_0, ..., a_{m-1}]^T$ is the parameter vector. Given a set of n measurements $X = [x_{ji}]$ where $j = 1, ..., m$ and $i = 1, ..., n$ (m is the number of features), the parameter vector of the RSS prediction model can be estimated via linear regression as follows:

$$\alpha_X = (X_{-3}^T X_{-3})^{-1} X_{-3}^T X_3, \tag{2}$$

where X_3 is the third column of X and X_{-3} is the remaining columns of X without the third column. Similarly, given a set of n obfuscated measurements $Y = [y_{ji}]$ where $j = 1, ..., m$ and $i = 1, ..., n$, the parameter vector of RSS prediction model can be estimated as α_Y.

4 Definition of Metrics

In this section we define the metrics used to evaluate privacy and utility.

4.1 Privacy

Let n denote the number of measurements per batch. $u = [u_i]$, $i = 1 \ldots n$ represent the user ID of each collected measurement and $\hat{u} = [\hat{u}_i]$ is the adversary's estimate of u. The adversary computes a probability distribution over the space of possible user IDs and selects for each measurement the user ID estimate with the maximum likelihood. We define the adversary estimate accuracy as the fraction of correct user ID estimates, that is,

$$acc(\hat{u}, u) = \frac{1}{n}\sum_{i=1}^{n} 1_{\hat{u}_i = u_i},$$

where the indicator function $1_{\hat{u}_i = u_i}$ is equal to 1 if the estimate is correct and 0 otherwise. Since high values of accuracy correspond to low values of privacy, we define the first privacy metric as

$$P_1(\hat{u}, u) = 1 - acc(\hat{u}, u). \tag{3}$$

$\hat{x}_1$ and $\hat{x}_2$ are the adversary's estimates of the true latitude x_1 and longitude x_2. While $\hat{u}$ represents a probability distribution, $\hat{x}_1$ and $\hat{x}_2$ specify an exact location. Our second privacy metric is the Euclidean distance between the true location and the adversary's estimate averaged over the batch, defined by

$$P_2(\hat{x}_1, \hat{x}_2, x_1, x_2) = \frac{1}{n}\sum_{i=1}^{n}\sqrt{(\hat{x}_{1i} - x_{1i})^2 + (\hat{x}_{2i} - x_{2i})^2}, \tag{4}$$

where the subscript $i = 1...n$ corresponds to the i^{th} measurement in the batch of size n. This metric defines how well the adversary is able to recover the original user location. High values of adversary location error correspond to high privacy.

Since both user IDs and locations are considered as private and sensitive information in our application, we further define the following composite privacy metric:

$$P(\hat{x}_1, \hat{x}_2, \hat{u}, x_1, x_2, u) = v_1 P_1(\hat{u}, u) + v_2 P_2(\hat{x}_1, \hat{x}_2, x_1, x_2), \tag{5}$$

where v_1 and v_2 are parameters controlling the weights of the two aforementioned privacy metrics. P_1, P_2 and P are the privacy metrics we use throughout the paper to compare the performance of different privatizers.

4.1.1 *Additional privacy metrics*

The composite privacy metric defined above is not differentiable because P_1 is not differentiable. This is a problem for adversary training. To handle this we use the cross entropy loss of the user ID estimate

$$P_1^{ce}(\hat{u}, u) = -\frac{1}{n}\sum_{i=1}^{n}\log p_i, \tag{6}$$

where p_i is the estimated likelihood of user ID u_i for measurement i, and define the loss function of the adversary as

$$L_a(\hat{x}_1, \hat{x}_2, \hat{u}, x_1, x_2, u) = v_1 P_1^{ce}(\hat{u}, u) + v_2 P_2(\hat{x}_1, \hat{x}_2, x_1, x_2), \tag{7}$$

which is used in the training of the adversary and of the GAP neural networks (the GAP privatizer and adversary used in the iterative training, see Section 5.3).

Our IT privacy approach, see Section 5.4, is motivated by the use of mutual information as a measure a privacy. The mutual information between two random variables X (e.g., our input) and Y (e.g., the obfuscated data) quantifies how much information about one random variable is obtained through observing the other. It is given by

$$I(X;Y) = \sum_{y\in\mathcal{Y}}\sum_{x\in\mathcal{X}} p_{X,Y}(x,y)\log\Big(\frac{p_{X,Y}(x,y)}{p_X(x)p_Y(y)}\Big), \tag{8}$$

where $p_{X,Y}$ is the joint probability mass function and p_X, p_Y are the marginal probability mass functions.

Last, the privacy metrics defined above are well suited for the typical threat model discussed in Section 3.2. However, for the worst-case threat model involving adversaries with unbounded computational resources and auxiliary information where users seek protection of any single report (see Section 3.2) we resort to differential privacy (DP) [19]. Specifically, let K be a randomized function applied to the input dataset. K gives ϵ,δ-differential privacy if for all datasets D_1 and D_2 which differ in at most one element and $\forall S \in \mathit{range}(K)$,

$$Pr[K(D_1) \in S] \leq e^{\epsilon} Pr[K(D_2) \in S] + \delta, \tag{9}$$

where the probability is taken over the randomness in K. ϵ and δ bound the difference between the output of K on D_1 and D_2 thus making it hard to guess the input (D_1 versus D_2) by observing the output. DP is a strong guarantee, since it doesn't make any assumptions about the computation power and auxiliary information available to the adversary, and ϵ and δ serve as metrics for privacy, see [12] for more details.

4.2 Utility

Let m be the number of features at each measurement excluding the user ID which is stripped from the input. The output of the privatizer $y = [y_j]$, $j = 1 \ldots m$ is the obfuscated data, e.g. y_1 and y_2 denote obfuscated latitude and longitude, respectively. Our utility metrics quantify the difference between the input data $x = [x_j]$ and the obfuscated data $y = [y_j]$. Recall that we consider n measurements per batch thus x_j and y_j are vectors of size n with elements x_{ji} and y_{ji}, $i = 1 \ldots n$, respectively. We consider several utility metrics motivated by real-world applications of crowdsourced network data.

The first metric quantifies the overall distortion of the dataset, considering all m features, by the L2 norm distance between input and obfuscated data, averaged over all n batch measurements:

$$U_1(x, y) = -\frac{1}{n} \sum_{i=1}^{n} \sqrt{\sum_{j=1}^{m} (y_{ji} - x_{ji})^2}. \tag{10}$$

Intuitively, high values of distortion correspond to low utility thus the minus sign in front of distortion in Eq. (10).

The second utility metric is related to the RSS prediction model described in Section 3.6. Recall that the goal of service provider is to estimate an accurate RSS prediction model based on the aggregated user data. However, with obfuscated user data, the estimated parameters of RSS prediction model differs from those estimated by unobfuscated user data (i.e. the estimated parameter vector changes from α_X to α_Y, see Eq. (2)). To minimize the difference between them, we define our second utility function as the opposite of L1-norm distance between α_X and α_Y as follows:

$$U_2(x, y) = -||\alpha_X - \alpha_Y||_1. \tag{11}$$

where α_X represents the RSS prediction model parameters estimated by unobfuscated user data (i.e. the privatizer's input) and α_Y represents the RSS prediction model parameters estimated by obfuscated user data (i.e. the privatizer's output). We refer to $||\alpha_X - \alpha_Y||_1$ as the generated map error, where higher values of map error corresponds to lower utility. While many metrics could be used to measure the distance between α_X and α_Y, comparing the fitted parameters over this bounded space provides a simple, effective loss function. Note that this map error does not capture how well a RSS prediction model generated by the obfuscated data could be used to predict RSS values at a new location, but rather captures the "distance" between maps generated before and after obfuscation.

Envisioning that the service provider may care for more than a single application-specific utility metric like U_2 in practice, we further define a composite utility metric $U(x, y)$ as

$$U(x, y) = w_1 U_1(x, y) + w_2 U_2(x, y), \tag{12}$$

where w_1 and w_2 are parameters adjusting the weights of each utility metric.

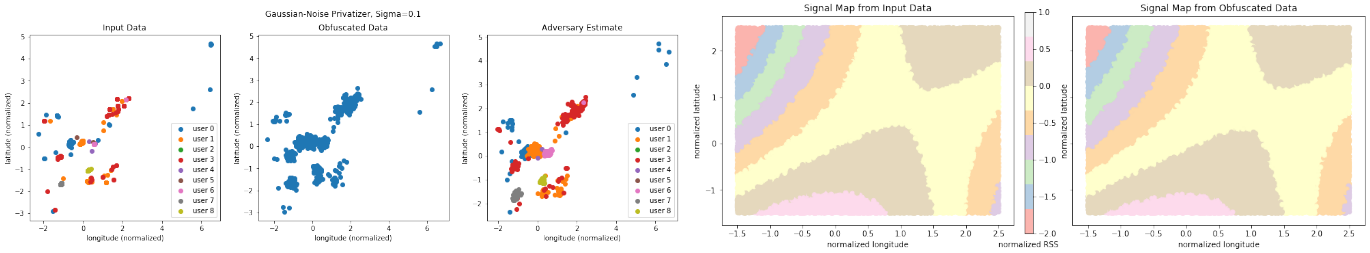
5 Privatizers

In this section we introduce in detail each of the four privatizers, which represent different types of obfuscation schemes. Specifically, we first select a Gaussian noise-adding privatizer for its simplicity and as a benchmark. We then select a locally differentially private privetizer (LDP) motivated by the well known strengths of Differential Privacy. We then select a privatizer based on GANs (referred to as the GAP privetizer), given the recent interest on how adversarial learning may be used to train privatizers by positioning them against adversities. Last, we select the so-called IT privatizer since it is a good representative of obfuscation schemes which use mutual information as a privacy metric and optimization to optimally design obfuscation.

5.1 Gaussian Noise-Adding Privacy

Our Gaussian noise-adding privatizer (Noise privatizer) takes the simplest approach to data obfuscation. For each input batch of size $n \times m$, where n is the number of points and m is the number of features, we add an $n \times m$ matrix of Guassian noise. Each element in this noise matrix is normally distributed with a mean of 0 and a standard deviation of σ. Since the data is also normalized such that each feature has a mean value of zero with a standard deviation of 1, values of σ close to 1 add a significant amount of noise and we choose to vary σ between 0 and 1. Fig. 4a provides a visualization of what the normalized input data, obfuscated data, and adversary's estimate look like side by side using the Noise privatizer for a low value of σ. Fig. 4b shows the signal maps generated before and after obfuscation. This shows that even in the presence of obfuscation, we can generate representative signal maps with the obfuscated data. Figures 5a and 5b show the same plots for a high value of σ. Note that while the privacy is improved, i.e. the adversary estimate is further from the input data, the signal maps differ significantly.

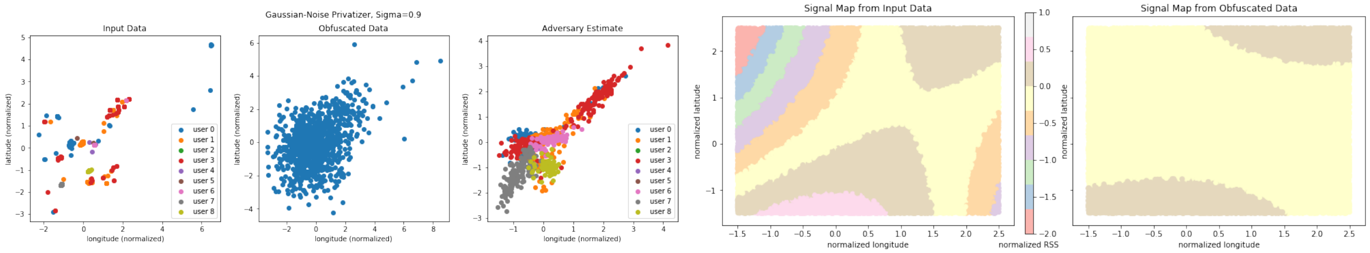
For reference, a privatizer which releases a completely random dataset (from a normal distribution with variance of 1.0) regardless of input data would observe the errors shown in Table 3.



(a) Input Data, Obfuscated Data, Adversary Estimate

(b) Signal Maps Before and After Obfuscation

Fig. 4: Noise privatizer when σ=0.1.



(a) Input Data, Obfuscated Data, Adversary Estimate

(b) Signal Maps Before and After Obfuscation

Fig. 5: Noise privatizer when σ=0.9.

TABLE 3: Utility Reference Values

Metric	No Obfuscation	Random Data
Distortion ($-U_1$)	0	5.74
Generated Map Error ($-U_2$)	0	2.41

5.2 Local Differential Privacy

We implement two local DP (LDP) privatizers which provide mathematical guarantees for privacy (see Eq. (9)) under the worst-case threat model discussed in Section 3.2.

The first approach is the Gaussian Mechanism with parameters ϵ and δ, which we refer to as GLDP [12]. This mechanism adds zero-mean Guassian noise with variance b to each feature. This variance is defined by

$$b = \frac{\Delta^2}{\epsilon^2} 2 \ln(\frac{1.25}{\delta}), \tag{13}$$

where Δ is the L2-sensitivity of a function f given by

$$\Delta = \max_{D_1, D_2 \in D} ||f(D_1) - f(D_2)||_2, \tag{14}$$

where D_1 and D_2 are subsets of the dataset D differing only by one element. Generally, in the local DP model, one can think of D_1 and D_2 as datasets of size 1 (i.e. one data point) and f as an identity function. Therefore the sensitivity becomes the greatest L2-distance between any two data points. In practice, we use an analytically calibrated Gaussian Mechanism which is shown to extend better to the very high privacy regime ($\epsilon \longrightarrow 0$) and the low privacy regime ($\epsilon \longrightarrow \infty$), see Algorithm 1 in [52] for the exact calculation for the variance of the added noise b.

The second approach is the Truncated Laplacian Mechanism with parameters ϵ and δ, which we refer to as LLDP, recently proposed in [13]. This mechanism adds noise satisfying the truncated Laplacian distribution, with probability density function

$$f(x) = \begin{cases} Be^{-\frac{|x|}{\lambda}}, & for \quad x \in [-A, A] \\ 0, & otherwise \end{cases} \tag{15}$$

where

$$\lambda = \frac{\Delta}{\epsilon}, \quad A = \frac{\Delta}{\epsilon} \log(1 + \frac{e^\epsilon - 1}{2\delta}), \quad B = \frac{1}{2\lambda(1 - e^{-\frac{A}{\lambda}})},$$

and Δ is defined in Eq. (14).

For both approaches, we follow standard practice and use $\delta = 0.00001$ (δ should be much smaller than $1/n$ [53]) and ϵ between 1 and 10 (larger ϵ values yield a very loose bound, see Eq. (9)), where low values of epsilon guarantee better privacy.

Moreover, following standard practice again, we clip each data point to have L2-norm $\leq \frac{\Delta}{2}$. Then, by invoking the triangle inequality, we can ensure that sensitivity is no greater than Δ. Specifically, for both the Gaussian mechanism and Laplacian mechanisms, we clip each data point according to the following function

$$x_{new} = \frac{x}{||x||_2} \min{(\frac{\Delta}{2}, ||x||_2)}. \tag{16}$$

To choose $\frac{\Delta}{2}$, we use the rule of thumb that clipping should occur 5% of the time. Using the pilot dataset to approximate how much of the data would be clipped for a given value, we choose $\frac{\Delta}{2} = 7.154$ and use this parameter during testing.

TABLE 4 compares the GLDP and LLDP privatizers with respect to our privacy and utility metrics on Chania dataset. We notice that both GLDP and LLDP privatizers yield quite large utility losses. From this table, it is evident that GLDP achieves sizably higher privacy than LLDP w.r.t. P_1 and P_2, especially for larger ϵ values. Although GLDP privatizer has larger loss in utility, both GLDP and LLDP privatizers can not offer any utility protection. Hence, we use GLDP with higher privacy in the rest of the paper when comparing LDP with other approaches under the typical threat model, see Section 6.

Note that while our Noise and LDP privatizers both add normally distributed noise, the key difference between the two is the noise clipping step. Intuitively, this ensures that no two data points are too different. This gives an anonymity to each measurement that is crucial to privacy under a worst-case threat model.

TABLE 4: Comparison of GLDP and LLDP privatizer on Chania dataset.

ϵ	Mechanism	P_1	P_2	U_1	U_2
1	GLDP	0.68	0.94	113.76	2.96
	LLDP	0.68	0.94	84.60	2.95
10	GLDP	0.63	0.91	16.46	2.39
	LLDP	0.49	0.69	8.50	2.49
100	GLDP	0.32	0.36	4.21	2.44
	LLDP	0.05	0.10	0.93	2.20

5.3 Generative Adversarial Privacy

Generative Adversarial Privacy is a data-driven approach to obfuscation which learns a privatization strategy by positioning the privatizer and adversary against each other in a minimax game [14], [21]. Our privatizer is a fully-connected feedforward neural network with a similar structure to our adversary. It has two hiddens layers of 256 units each. Between layers we employ Rectified Linear Unit (ReLU) activations, and our optimization relies on Adaptive Moment Estimate (Adam) stochastic gradient descent with a learning rate of 0.001. Our privatizer, which takes an input batch of size $n \times m$, outputs an $n \times m$ batch of obfuscated data, where each measurement has been obfuscated independently. (We treat the case where measurements are grouped into batches and then jointly obfuscated in Section 6.2.)

Our privatizer wants to minimize the following loss function

$$\begin{aligned} L_p(x, y, \hat{u}, \hat{x}_1, \hat{x}_2) &= -\rho U(x, y) \\ &- (1-\rho) L_a(\hat{x}_1, \hat{x}_2, \hat{u}, x_1, x_2, u), \end{aligned} \tag{17}$$

where U is the composite utility metric defined in Eq. (12) and L_a is the adversary loss function defined in Eq. (7) which is a differentiable version of the composite privacy metric and depends on the adversary estimate error of the user ID and location.

Notice that as the adversary's loss decreases (implying less privacy), the privatizer's loss increases. ρ quantifies the penalty on utility loss, as opposed to privacy loss. Utility losses have a large effect on the privatizer when $\rho \longrightarrow 1$ and privacy losses have a large affect on the privatizer when $\rho \longrightarrow 0$.

We take an iterative approach to training the two neural networks. We first train the adversary, specifically, we fix the neural network (NN) weights of the privatizer and perturb the NN weights of the adversary along the negative gradient of L_a for k epochs. We then train the privatizer, that is, we perturb the NN weights of the privatizer along the negative gradient of L_p for k epochs, and so on and so forth. When both have converged, we have found the equilibrium of our minimax game. We then fix the weights of both NNs during testing.

The GAP privatizer incorporates the privacy and utility metrics in its loss function L_p and trains against an adversary with the same loss function L_a as the one used to evaluate privatizers. While it is advantageous to incorporate specific privacy metrics, for generality we evaluate the GAP privatizer's performance against other loss functions too, see Section 6.5.

5.4 Information-theoretic Privacy

For this approach, we consider the privacy-utility tradeoff in an analytically tractable fashion under a formal optimization framework.

Considering $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$ as random variables describing our input and obfuscated data respectively, our IT privatizer tries to minimize the mutual information $I(X;Y)$, see Eq. (8), subject to a constraint on utility. The privatizer is specified by the conditional probability distribution $p_{Y|X}$, the probability of releasing Y given input data X. Without the utility constraint, the optimal solution is to release Y independent of X.

Formally, the problem becomes:

$$\min_{p_{Y|X}} I(X;Y) \tag{18a}$$

$$\text{s.t.} \sum_{y \in Y} p_{Y|X}(y|x) U(x,y) \geq U_c, \forall x \in \mathcal{X} \tag{18b}$$

$$p_{Y|X}(y|x) \geq 0, \forall x \in \mathcal{X}, \forall y \in \mathcal{Y} \tag{18c}$$

$$\sum_{y \in \mathcal{Y}} p_{Y|X}(y|x) = 1, \forall x \in \mathcal{X}, \tag{18d}$$

where (18b) is a constraint on the composite utility $U(x,y)$ defined in Eq. (12), and constraints (18c) and (18d) ensure that $p_{Y|X}$ is a valid probability distribution.

We approach this constrained minimization problem by rewriting it as a Lagrange function whose optimal point is a global minimum over the domain of the choice variables and a global maximum over the Karush-Kuhn-Tucker (KKT) multipliers [54]. We analyze the KKT conditions below to derive key observations on the optimal solution:

$$p_X(x) log(\frac{p^*_{Y|X}(y|x)}{p_Y(y)}) - \mu_1^* U(x,y) - \mu_2^* + \lambda = 0 \tag{19a}$$

$$\mu_1^*(U_c - \sum_{y \in \mathcal{Y}} p^*_{Y|X}(y|x) U(x,y)) = 0, \forall x \in X \tag{19b}$$

$$\mu_2^* p^*_{Y|X}(y|x) = 0, \forall x \in \mathcal{X}, \forall y \in \mathcal{Y} \tag{19c}$$

$$\mu_1^*, \mu_2^* >= 0. \tag{19d}$$

Solving this for the optimal conditional probability distribution, we see

$$p^*_{Y|X}(y|x) = p^*_Y(y) \exp\left(\frac{\mu_1^* U(x,y) + \mu_2^* - \lambda^*}{p_X(x)}\right). \tag{20}$$

We take the sum of both sides,

$$\sum_{y \in \mathcal{Y}} p^*_Y(y) \exp\left(\frac{\mu_1^* U(x,y) + \mu_2^* - \lambda^*}{p_X(x)}\right) = 1. \tag{21}$$

We then manipulate this to get an expression in terms of λ^*, which we substitute back into Equation (20) to get the following:

$$p^*_{Y|X}(y|x) = \frac{1}{\eta} p^*_Y(y) \exp\left(\frac{\mu_1^* U(x,y)}{p_X(x)}\right), \tag{22}$$

where η is a normalization term over $y \in \mathcal{Y}$. From this formal treatment, and reminiscent of our previous work [55], [56], we derive two important characteristics of the optimal solution: (i) $p_{Y|X}$ should exponentially increase with utility, and (ii) $p_{Y|X}$ should linearly increase with p_Y, the probability that y is reported for any x, i.e. we should reuse released datasets to the extent practical.

Given the above qualities of an optimal solution, we design the following heuristic approach. We use the pilot dataset to empirically determine the distribution p_X using multi-variate Gaussian kernel density estimation. We then sample from this distribution N_s times to create a "codebook" which approximates the sample space $\mathcal{Y}$. Limiting N_s allows us to reuse released datasets, as mentioned above.

The weight of each "code" or possible y value is given by

$$w(y) = \exp\left(\mu_1^* U(x,y)\right), \tag{23}$$

where μ_1^* is our KKT multiplier. Given an input data x, our information theoretic privatizer selects a y from the codebook with probability $w(y)/\sum_{y \in codebook} w(y)$. This ensures the likelihood of reporting a y increases exponentially with utility. As μ_1^* increases, the IT privatizer offers higher utility but lower privacy. By contrast, as μ_1^* approaches zero, the IT privatizer achieves lower utility while higher privacy.

In implementation, we use a codebook with size of 51. This codebook size was empirically determined to be large enough that one or more codes would provide good utility, yet small enough that codes are reused to the extent practical. Note that we bias the codebook by including a copy of the unobfuscated data (i.e. 50 obfuscated codes + 1 unobfuscated code). This ensures at least one y has very high utility even for relatively small codebooks. Also, to reduce computational complexity, we split the n measurements into batches and for each batch x we select a batch y from the codebook.

6 Performance evaluation

In this section we compare the performance of the privatizers against different adversaries when users upload a single or a batch of measurements, and evaluate where they sit in the privacy-utility design trade space. All performance comparisons in this section are under the typical threat model (bounded adversary). We use the three real-world traces introduced in Section 3.1 in our evaluation. Unless otherwise stated, the default trace is the Chania dataset.

6.1 Comparison of Privatizers

Consider the scenario where users upload a single measurement at a time. Fig. 6a/Fig. 6b show the adversary estimate user ID error/adversary location error respectively against each privatizer (its privacy), and Fig. 6c/Fig. 6d show the distortion/generated map error of each privatizer(its utility). The x-axis in these and the following plots represents the parameterization of each privatizer, i.e. σ, ϵ, ρ, and μ_1^*.

As expected, for the noise privatizer, as σ increases from 0 to 1 the adversary's user ID and location estimate errors increase, demonstrating higher privacy (larger P_1 and P_2). At the same time, both the distortion and generated map errors increase, demonstrating lower utility (smaller U_1 and U_2). For the GLDP/GAP/IT privatizers, decreasing $\epsilon/\rho/\mu_1^*$ leads to higher privacy (i.e. an increase in the adversary's user ID and location estimate error rate) and lower utility (i.e. an increase in distortion and generated map error).

Among these privatizers, the GLDP privatizer consistently achieves high privacy for typical values of ϵ. Specifically, against the GLDP privatizer with $1 \leq \epsilon \leq 10$, the adversary's user ID estimation error is around 70% and the adversary's location estimate error is close to 1. These numbers can be explained as follows. In the absence of any intelligible patterns due to obfuscation, the adversary learns to assume all measurements came from the geographic center of the dataset, thus its error is on the same order as the spread of input data, i.e. roughly 1. Both the IT and GAP privatizers can approach this privacy performance as μ_1^* and ρ get close to 0.1 or smaller. As for the user ID estimation error, the user with the most measurements contributes roug hly 30% of them, thus a simple adversary assigning this user's ID to all measurements would have 70% user ID error, hence this can be considered as the upper bound of P_1.

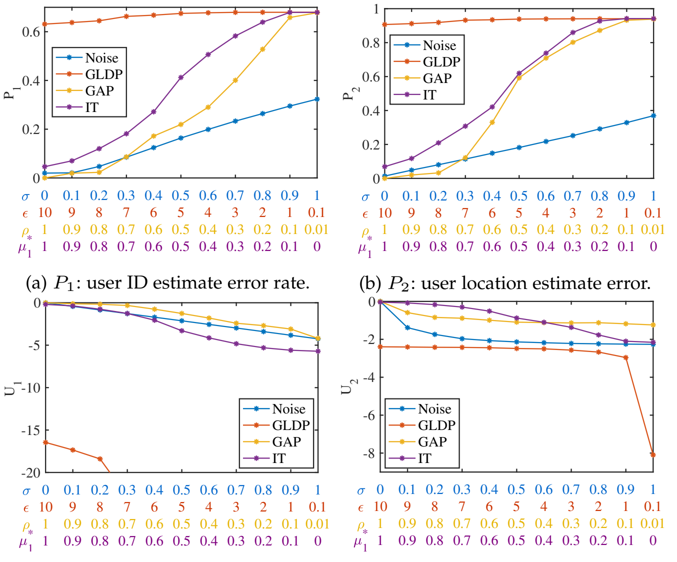
With respect to the utility, the GLDP privatizer offers the worst performance. The GAP privatizer outperforms the others for ρ in the range [0.0,0.4] (i.e. high privacy region), while the IT privatizer achieves the best utility for μ_1^* in the range [0.4,1.0] (i.e. low privacy region). As it will become clear in the next couple of sections, a major reason why GLDP has the best privacy and worst utility is that for the range of ϵ values considered, it distorts the data to a larger extent than the rest of the approaches. We discuss in more detail the differences between the 4 privatizers in the Section 6.3.

6.2 Leveraging Measurement Sequences

To directly investigate the effect of correlations and predictable patterns when considering mobile measurements as a time sequence, we consider an adversary which takes measurement sequences as input, i.e. time sequences of lengths 1, 5, 10, and 20 which belong to a single user, and estimates the (common) user ID of all these measurements, taking advantage of correlations across data of the same user. In practice, the adversary can do this when users upload measurements in batches.

The adversary we consider is trained via supervised learning with the final output of the converged GAP privatizer. The GAP privatizer is a good choice to study sequences of data as it can be trained to consider correlations of sequences and privatize batches of data in one shot as well.

The results shown in Fig. 7 consider three cases: only the adversary, only the privatizer, and both of them consider sequences of data. Fig. 7a shows results when only the adversary considers measurement sequences. It shows that the



(a) P_1: user ID estimate error rate.

(b) P_2: user location estimate error.

(c) U_1: $-$distortion.

(d) U_2: $-$generated map error.

Fig. 6: Privacy and utility of different privatizers. Note that Noise, GLDP, GAP, and IT refer to the Gaussian noise-adding, local Gaussian Mechanism DP, GAP, and the information-theoretic privatizers, respectively.

longer the sequence the better the adversary performance, as the adversary achieves smaller error for the same data distortion. Fig. 7b shows results when only the privatizer considers measurement sequences. It shows that the longer the sequence the better the privatizer performance, as the privatizer forces the adversary to achieve higher error for the same data distortion. Thus, sequences of measurements help both the adversary and the privatizer, which is expected in the presence of inter-measurement correlations. That said, the tradeoff in both cases above is the additional computational and memory resources required to handle input sequences as opposed to single measurements. Lastly, Fig. 7c shows results when both the adversary and privatizer consider sequences of the same length. We observe that longer sequences result in better privacy, as the adversary's user ID estimate error increases.

6.3 Analysis of Privacy-Utility Trade Space

Fig. 8 and Fig. 9 illustrate where each privatizer sits in the privacy-utility tradeoff space under real-world datasets with different metrics. Specifically, in Fig. 8, we consider the composite privacy and utility metrics on Chania, UCI, and Radiocell datasets, where the x axis shows the composite privacy P defined in Eq. (5) with weights $v_1 = v_2 = 1$ and the y axis shows the composite utility U defined in Eq. (12) with weights $w_1 = w_2 = 1$. Note that we consider such composite privacy and utility metrics since in practice a service provider may care about both P_1 and P_2 and about both U_1 and U_2. In Fig. 9, we further consider four different combinations of non-composite privacy and utility metrics, and we use the Chania dataset as an example to illustrate how the privacy-utility tradeoff curves of different privatizers change with different non-composite metrics.

In all these plots, the ideal privatizer should sit in the top right corner implying high privacy and high utility. While the three traces are collected in different countries, areas, and years, the results are qualitatively the same. From both plots we conclude that GAP and the IT privatizer outperform the Noise and GLDP privatizers. It is important to remind the reader that the above comparison is under the typical threat model where the adversary is bounded, whereas GLDP privatizer is the only privatizer that provides privacy guarantees under the worst-case threat model. As discussed in detail in Section 3.2, we focus on the typical threat model as it is more relevant to our context/application.

A major reason why GAP and the IT privatizer perform well is that they rely on the notion of *context*, as we have already discussed in Section 3.4. The GAP privatizer gains some insight about the structure of the dataset through data-driven learning. It also tries to minimize the difference between the true and obfuscated data while achieving privacy, as encoded in its loss function. In summary, GAP uses P, U and the data. The IT privatizer gains some insight about the structure of the databset through Gaussian kernel density estimation. It does well because it releases obfuscated datasets which inherently mirror the true dataset's structure, thanks to a constraint on utility. In summary, IT uses U and the data distribution. In contrast to GAP and IT, GLDP only needs information about the data to perform clipping without hurting utility too much (in our implementation we used the data directly for this purpose, see Eq. (16)), and Noise only needs the variance of the data to normalize the amount of Gaussian noise that it adds.

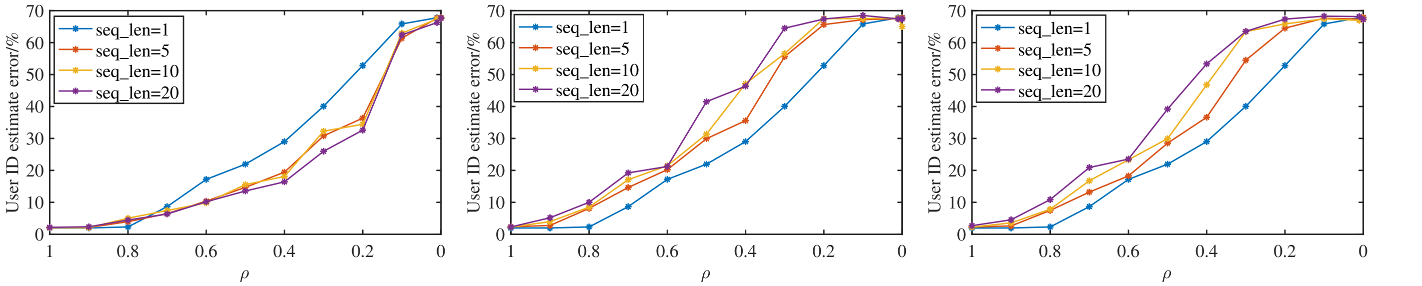
Comparing GAP with IT, because GAP tries to prevent an adversary from estimating features of x given y, this strategy can be thought of as a data-driven approach to what IT does, i.e. minimizing mutual information. Yet while the IT strategy adds privacy by choosing y randomly (with appropriate weights), the GAP privatizer maintains a model of a rational adversary which it intentionally tries to deceive. Training against an adversary with the same loss function as the adversary used to test the performance of the privatizers, might be perceived as unfair. To address this, in Section 6.5 we test privatizers against adversaries with different loss functions.

The GLDP privacy-utility curve shown in Fig. 8 shows values of ϵ up to 100. Note that this is an order of magnitude greater than the values of ϵ shown in Fig. 6c/Fig. 6d and such high values yield a very loose bound on Eq. (9), yet we do so to show that the Noise and GLDP privatizer meet when noise levels are similar. Note that values of $\epsilon \leq 10$ lie along the asymptotic behavior around the $P = 1.6/1.5/2.0$ line for the Chania/UCI/Radiocell dataset, respectively.

Finally, note that when we train the GAP privatizer and compute the codebook of the IT privatizer to generate the results of Fig. 9, we use the composite privacy and utility metrics to avoid retraining/recomputing them for each case. Interestingly, this doesn't deteriorate their performance in a visible manner. While real-world engineers could retrain/recompute the GAP/IT privatizers for the specific privacy and utility metrics they care about, in practice this may be cumbersome.

6.4 Constraining Distortion Levels

Previously we have considered privacy and utility as two components of our objective. Suppose instead we wish to

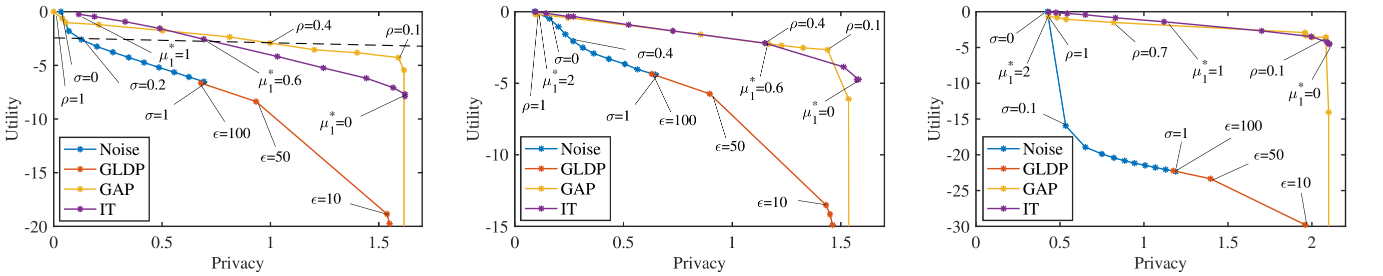


(a) Only adversary leverages sequences.

(b) Only privatizer leverages sequences.

(c) Both leverage sequences.

Fig. 7: Effect of leveraging measurement sequences on adversary user ID estimate accuracy.

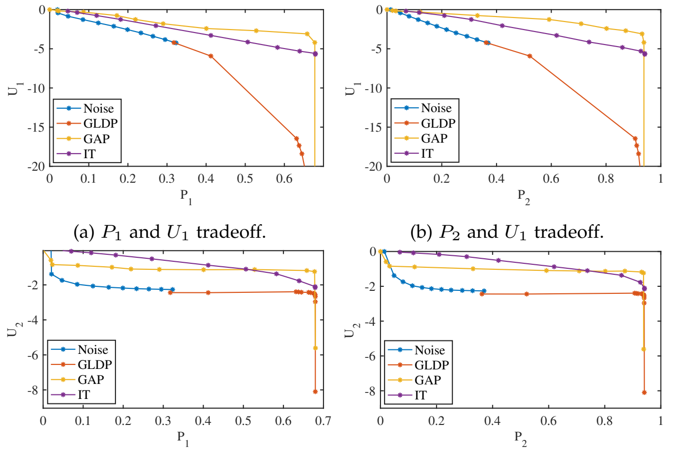


(a) P vs U on Chania dataset.

(b) P vs U on UCI dataset.

(c) P vs U on Radiocell dataset.

Fig. 8: Privacy-utility tradeoff of different privatizers under the Chania, UCI, and Radiocell datasets with composite metrics. Note that Noise, GLDP, GAP, and IT refer to the Gaussian noise-adding, local Gaussian Mechanism DP, GAP, and the information-theoretic privatizers, respectively.



(a) P_1 and U_1 tradeoff.

(b) P_2 and U_1 tradeoff.

(c) P_1 and U_2 tradeoff.

(d) P_2 and U_2 tradeoff.

Fig. 9: Privacy-utility tradeoff of different privatizers under the Chania dataset with non-composite metrics.

maximize privacy subject to a constraint on utility. In Fig. 10 we re-frame previous results to demonstrate choosing the appropriate parameters to meet a constraint on distortion ($-U_1$), which can act as an empirical measure of how different the obfuscated data is compared to the original data. Fig. 10 presents a plot which can be interpreted as a continuous lookup table. For example, to meet the constraint $-U_1 \leq 3$, we could choose $\sigma = 0.2$ or $\mu_1^* = 0.6$ or $\rho = 0.4$. This plot also offers a sense of which range of distortion each approach may achieve for its selected range of parameter.

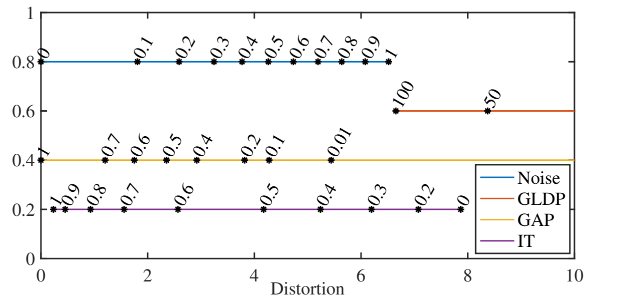


Fig. 10: Choosing parameters under a constraint on distortion.

6.5 Performance against different adversaries

So far we have tested each privatizer against an adversary trained on the obfuscated data generated by the privatizer. We refer to this as the privatizer's "own" adversary. What is more, the GAP privatizer is explicitly trained to beat its own adversary and it would be informative to investigate its performance against other adversaries.

Motivated by the above, we investigate how the four privatizers perform against the following three adversaries: (i) "Unobfuscated" adversary which is trained with the unobfuscated data via supervised learning (rather than the obfuscated data that we have used so far), (ii) "Aggregate" adversary which has access to all obfuscated data generated by all privatizers, and is trained with the aggregated obfuscated data, and (iii) "Alternative" adversary trained with a different loss function than the one used so far, which has also been used for training the GAP adversary inside the

TABLE 5: Evaluation results against common adversaries. Baseline reports the privacy of a privatizer against the adversary trained with its own obfuscated data. Unobfuscated is trained against unobfuscated data. Aggregate is trained against aggregate obfuscated data. Alternative is trained using a different loss function.

Privatizer	Parameter	Utility U	Privacy P against different adversaries				
			Baseline	Unobfuscated	Aggregate	Alternative (different L_a)	
						$v_1 = 0.8, v_2 = 0.2$	$v_1 = 0.2, v_2 = 0.8$
Noise	$\sigma = 0.2$	-2.5	0.13	0.30	0.80	0.13	0.14
GAP	$\rho = 0.4$	-2.5	0.95	1.54	1.33	1.05	0.88
IT	$\mu_1^* = 0.6$	-2.5	0.70	1.26	1.18	0.68	0.70

iterative GAP loop. Specifically, alternative adversaries use different weights v_1 and v_2 in the loss function L_a.

Recall that for each privatizer, we have different parameter settings to trade privacy and utility. For a fair comparison, we first set a target utility value and use for each privatizer its parameter value that achieves this utility. Table 5 shows the corresponding parameter values for a composite target utility of -2.5. This value is motivated by Table 3 and Fig. 8, as the former shows the (negative) utility of a random dataset and the latter shows the entire privacy-utility spectrum considered. (Notice that for GLDP to achieve a -2.5 utility value it would use too large of an ϵ value (>100) thus we omit this line from the table.) We report the privacy achieved by each privatizer against its own adversary (Baseline) and the three adversaries introduced above.

Interestingly, the GAP privatizer outperforms all the other privatizers not only when privatizers are positioned against their own adversaries (see also Section 6.1) but also against the other adversaries, namely Unobfuscated, Aggregate, and Alternative. That said, the performance gap does reduce, which can be explained by the fact that the GAP privatizer is trained against an adversary with a loss function which is now different from that of the adversary used to test the privatizers.

As expected, all privatizers achieve the lowest privacy against their own adversary (baseline), since the latter is trained with the obfuscated data of each privatizer. Also, all privatizers achieve the highest privacy against the Unobfuscated adversary. This is also expected as the Unobfuscated adversary is trained using unobfuscated data thus it is weaker than the others.

7 Limitations and Future work

Points of interest: The adversary we consider predicts user IDs and all locations from where measurements are collected. However, an adversary may be particularly interested to learn specific users' points of interest (POIs). For instance, the adversary may want to predict the target user's home or work location. We do not consider this in the paper since users can choose to not collect measurements around POIs as a defense mechanism.

Side information: We assume the adversary has access only to the obfuscated user data shared with the service provider, which does not contain user ID information. A stronger adversary might leverage side information to estimate the user ID of each measurement. For example, the adversary might be able to monitor the network connection between the service provider and mobile users, such that it knows from which device each obfuscated measurement comes from and thus the user ID. This adversary may then build a user whereabouts model. Since it is much harder for an adversary to have access to such information than to merely access database updates, we do not consider this threat model.

Federated learning: Mobile crowdsourcing applications lend themselves to a federated learning implementation [57], [58], which can provide some privacy for mobile users. Recent works show that federated learning could also leak user privacy [59], [60], [61]. However, it would be a reasonable solution for opt-in mobile users used to collect training data for the GAP privatizer and to estimate data distributions for the IT privatizer.

Another avenue for future work is to investigate how federated learning can be applied, with additional privacy mechanisms, to achieve privacy-preserving training of an RSS predictor. For instance, one may add noise to local model updates or carefully select the measurements used for local model training epochs, to weaken data reconstruction attacks (see, for example, the DLG attack proposed in [62]).

8 Conclusions

In this work, we have systematically examined the privacy-utility tradeoff which exists in crowdsourced mobile network data obfuscation. We have considered four preeminent privatizers employing different obfuscation strategies. To compare them, we have identified several privacy and utility metrics as well as a number of adversaries under two different threat models suited to crowdsourced mobile network data, and evaluate the privacy-utility tradeoff performance of different privatizers on three diverse real-world mobile network datasets. The main takeaway is that under a typical threat model with a bounded adversary, which is of more practical interest in the context of our application, incorporating the structure and intended use of datasets in obfuscation can provide privacy gains without significant utility losses.

Acknowledgments

We would like to thank Athina Markopoulou and Emmanouil Alimpertis from the University of California Irvine for access to the dataset used in this work [15]. This material is based upon work supported by the National Science Foundation under grants CNS-1618450 and CNS-1901488.

REFERENCES

[1] O. S. Inc., "3g and 4g lte cell coverage map."

[2] T. Technologies, "Manage your mobile experience."

[3] J. Cox, "Eff hits at&t with class action lawsuit for selling customers' location to bounty hunters," Jul 2019.

[4] E. Agapie, G. Chen, D. Houston, E. Howard, J. Kim, M. Y. Mun, A. Mondschein, S. Reddy, R. Rosario, J. Ryder *et al.*, "Seeing our signals: Combining location traces and web-based models for personal discovery," in *Proceedings of the 9th workshop on Mobile computing systems and applications*. ACM, 2008, pp. 6–10.

[5] Y.-A. de Montjoye, S. Gambs, V. Blondel, G. Canright, N. De Cordes, S. Deletaille, K. Engø-Monsen, M. Garcia-Herranz, J. Kendall, C. Kerry *et al.*, "On the privacy-conscientious use of mobile phone data," *Scientific data*, vol. 5, 2018.

[6] C. Warzel, G. S. Gerstell, S. Aziza, T. Klosowski, F. M. Bremer, Nadieh, S. Jeong, A. Foster, D. M. Primo, J. Rich, A. F. Cahn, and et al., "The privacy project," Sep 2019. [Online]. Available: https://www.nytimes.com/series/new-york-times-privacy-project

[7] "New survey finds deep consumer anxiety over data privacy and security," Apr 2018. [Online]. Available: https://newsroom.ibm.com/2018-04-15-New-Survey-Finds-Deep-Consumer-Anxiety-over-Data-Privacy-and-Security

[8] "General Data Protection Regulation (GDPR)." https://gdpr-info.eu/.

[9] "California Consumer Privacy Act (CCPA)," https://oag.ca.gov/privacy/ccpa.

[10] X. Wu, P. Yang, S. Tang, X. Zheng, and Y. Xiong, "Privacy preserving rss map generation for a crowdsensing network," *IEEE Wireless Communications*, vol. 22, no. 4, pp. 42–48, August 2015.

[11] L. Sankar, S. R. Rajagopalan, and H. V. Poor, "Utility-privacy tradeoffs in databases: An information-theoretic approach," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 6, pp. 838–852, 2013.

[12] C. Dwork, A. Roth *et al.*, "The algorithmic foundations of differential privacy," *Foundations and Trends® in Theoretical Computer Science*, vol. 9, no. 3–4, pp. 211–407, 2014.

[13] Q. Geng, W. Ding, R. Guo, and S. Kumar, "Privacy and utility tradeoff in approximate differential privacy," *arXiv preprint arXiv:1810.00877*, 2018.

[14] C. Huang, P. Kairouz, X. Chen, L. Sankar, and R. Rajagopal, "Generative adversarial privacy," *CoRR*, vol. abs/1807.05306, 2018. [Online]. Available: http://arxiv.org/abs/1807.05306

[15] E. Alimpertis, N. Fasarakis-Hilliard, and A. Bletsas, "Community rf sensing for source localization," *IEEE Wireless Communications Letters*, vol. 3, no. 4, pp. 393–396, Aug 2014.

[16] E. Alimpertis and A. Markopoulou, "Using antmonitor for crowdsourcing passive mobile network measurements," *NSDI'17 Poster Session*, 2017.

[17] "Radiocell dataset." https://www.radiocells.org/.

[18] C. Dwork, "Differential privacy," *Encyclopedia of Cryptography and Security*, pp. 338–340, 2011.

[19] ——, "Differential privacy: A survey of results," in *International Conference on Theory and Applications of Models of Computation*. Springer, 2008, pp. 1–19.

[20] P. Kairouz, S. Oh, and P. Viswanath, "Extremal mechanisms for local differential privacy," in *Advances in neural information processing systems*, 2014, pp. 2879–2887.

[21] C. Huang, P. Kairouz, X. Chen, L. Sankar, and R. Rajagopal, "Context-aware generative adversarial privacy," *Entropy*, vol. 19, no. 12, p. 656, 2017.

[22] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[23] R. Camino, C. Hammerschmidt, and R. State, "Generating multi-categorical samples with generative adversarial networks," *arXiv preprint arXiv:1807.01202*, 2018.

[24] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *CoRR*, vol. abs/1411.1784, 2014. [Online]. Available: http://arxiv.org/abs/1411.1784

[25] M. Askari, R. Safavi-Naini, and K. Barker, "An information theoretic privacy and utility measure for data sanitization mechanisms," in *Proceedings of the second ACM conference on Data and Application Security and Privacy*. ACM, 2012, pp. 283–294.

[26] K. Kalantari, L. Sankar, and O. Kosut, "On information-theoretic privacy with general distortion cost functions," in *2017 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2017, pp. 2865–2869.

[27] T. M. Cover and J. A. Thomas, *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. New York, NY, USA: Wiley-Interscience, 2006.

[28] A. D. Sarwate and L. Sankar, "A rate-disortion perspective on local differential privacy," in *2014 52nd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2014, pp. 903–908.

[29] X. Ren, C.-M. Yu, W. Yu, S. Yang, X. Yang, J. A. McCann, and S. Y. Philip, "Lopub: High-dimensional crowdsourced data publication with local differential privacy," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 9, pp. 2151–2166, 2018.

[30] W. Wang, L. Ying, and J. Zhang, "On the relation between identifiability, differential privacy, and mutual-information privacy," *IEEE Transactions on Information Theory*, vol. 62, no. 9, pp. 5018–5029, 2016.

[31] M. S. Alvim, M. E. Andrés, K. Chatzikokolakis, P. Degano, and C. Palamidessi, "Differential privacy: on the trade-off between utility and information leakage," in *International Workshop on Formal Aspects in Security and Trust*. Springer, 2011, pp. 39–54.

[32] A. Ghosh, T. Roughgarden, and M. Sundararajan, "Universally utility-maximizing privacy mechanisms," *SIAM Journal on Computing*, vol. 41, no. 6, pp. 1673–1693, 2012.

[33] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor, "Our data, ourselves: Privacy via distributed noise generation," in *Advances in Cryptology - EUROCRYPT 2006*, S. Vaudenay, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 486–503.

[34] J. W. Kim, D.-H. Kim, and B. Jang, "Application of local differential privacy to collection of indoor positioning data," *IEEE Access*, vol. 6, pp. 4276–4286, 2018.

[35] A. Machanavajjhala, D. Kifer, J. Abowd, J. Gehrke, and L. Vilhuber, "Privacy: Theory meets practice on the map," in *Proceedings of the 2008 IEEE 24th International Conference on Data Engineering*, ser. ICDE '08. Washington, DC, USA: IEEE Computer Society, 2008, pp. 277–286. [Online]. Available: https://doi.org/10.1109/ICDE.2008.4497436

[36] H. Jin, L. Su, H. Xiao, and K. Nahrstedt, "Inception: Incentivizing privacy-preserving data aggregation for mobile crowd sensing systems," in *Proceedings of the 17th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, 2016, pp. 341–350.

[37] R. Shokri, G. Theodorakopoulos, C. Troncoso, J.-P. Hubaux, and J.-Y. Le Boudec, "Protecting location privacy: optimal strategy against localization attacks," in *Proceedings of the 2012 ACM conference on Computer and communications security*, 2012, pp. 617–627.

[38] N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, "Optimal geo-indistinguishable mechanisms for location privacy," in *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, 2014, pp. 251–262.

[39] S. Boukoros, M. Humbert, S. Katzenbeisser, and C. Troncoso, "On (the lack of) location privacy in crowdsourcing applications," in *28th {USENIX} Security Symposium ({USENIX} Security 19)*, 2019, pp. 1859–1876.

[40] M. E. Gursoy, L. Liu, S. Truex, and L. Yu, "Differentially private and utility preserving publication of trajectory data," *IEEE Transactions on Mobile Computing*, vol. 18, no. 10, pp. 2315–2329, 2018.

[41] M. E. Gursoy, L. Liu, S. Truex, L. Yu, and W. Wei, "Utility-aware synthesis of differentially private and attack-resilient location traces," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2018, pp. 196–211.

[42] R. B. Messaoud, N. Sghaier, M. A. Moussa, and Y. Ghamri-Doudane, "Privacy preserving utility-aware mechanism for data uploading phase in participatory sensing," *IEEE Transactions on Mobile Computing*, vol. 18, no. 9, pp. 2160–2173, 2018.

[43] S. Liu, J. Du, A. Shrivastava, and L. Zhong, "Privacy adversarial network: representation learning for mobile data privacy," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 3, no. 4, pp. 1–18, 2019.

[44] N. Raval, A. Machanavajjhala, and J. Pan, "Olympus: Sensor privacy through utility aware obfuscation," *Proceedings on Privacy Enhancing Technologies*, vol. 2019, pp. 25 – 5, 2019.

[45] A. Taufique, M. Jaber, A. Imran, Z. Dawy, and E. Yacoub, "Planning wireless cellular networks of future: Outlook, challenges and opportunities," *IEEE Access*, vol. 5, pp. 4821–4845, 2017.

[46] O. Thakkar, G. Andrew, and H. B. McMahan, "Differentially private learning with adaptive clipping," *arXiv preprint arXiv:1905.03871*, 2019.

[47] H. B. McMahan, G. Andrew, U. Erlingsson, S. Chien, I. Mironov, N. Papernot, and P. Kairouz, "A general approach to adding differential privacy to iterative training procedures," *arXiv preprint arXiv:1812.06210*, 2018.

[48] V. Erceg, L. J. Greenstein, S. Y. Tjandra, S. R. Parkoff, A. Gupta, B. Kulic, A. A. Julius, and R. Bianchi, "An empirically based path loss model for wireless channels in suburban environments," *IEEE Journal on selected areas in communications*, vol. 17, no. 7, pp. 1205–1211, 1999.

[49] S. Y. Han, N. B. Abu-Ghazaleh, and D. Lee, "Double regression: Efficient spatially correlated path loss model for wireless network simulation," in *2013 Proceedings IEEE INFOCOM*. IEEE, 2013, pp. 1824–1832.

[50] E. Alimpertis, A. Markopoulou, C. Butts, and K. Psounis, "City-wide signal strength maps: Prediction with random forests," in *The World Wide Web Conference*, 2019, pp. 2536–2542.

[51] A. F. Molisch, *Wireless communications*. John Wiley & Sons, 2012, vol. 34.

[52] B. Balle and Y.-X. Wang, "Improving the gaussian mechanism for differential privacy: Analytical calibration and optimal denoising," *arXiv preprint arXiv:1805.06530*, 2018.

[53] B. Balle, "A short tutorial on differential privacy," Jan 2018.

[54] H.-C. Wu, "The karush–kuhn–tucker optimality conditions in an optimization problem with interval-valued objective function," *European Journal of Operational Research*, vol. 176, no. 1, pp. 46–59, 2007.

[55] M. A. Clark and K. Psounis, "Trading utility for privacy in shared spectrum access systems," *IEEE/ACM Transactions on Networking (TON)*, vol. 26, no. 1, pp. 259–273, 2018.

[56] L. Clark, M. Clark, K. Psounis, and P. Kairouz, "Privacy-utility trades in wireless data via optimization and learning," *Proceedings of Information Theory and Applications Workshop (ITA)*, Feb 2019.

[57] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *arXiv preprint arXiv:1912.04977*, 2019.

[58] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, 2020.

[59] Z. Wang, M. Song, Z. Zhang, Y. Song, Q. Wang, and H. Qi, "Beyond inferring class representatives: User-level privacy leakage from federated learning," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2019, pp. 2512–2520.

[60] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning," in *2019 IEEE symposium on security and privacy (SP)*. IEEE, 2019, pp. 739–753.

[61] L. Lyu, H. Yu, and Q. Yang, "Threats to federated learning: A survey," *arXiv preprint arXiv:2003.02133*, 2020.

[62] L. Zhu and S. Han, "Deep leakage from gradients," in *Federated Learning*. Springer, 2020, pp. 17–31.