Apostolos Pyrgelis*, Carmela Troncoso, and Emiliano De Cristofaro

What Does The Crowd Say About You? Evaluating Aggregation-based Location Privacy

Abstract: Information about people's movements and the locations they visit enables an increasing number of mobility analytics applications, e.g., in the context of urban and transportation planning, In this setting, rather than collecting or sharing raw data, entities often use aggregation as a privacy protection mechanism, aiming to hide *individual* users' location traces. Furthermore, to bound information leakage from the aggregates, they can perturb the input of the aggregation or its output to ensure that these are differentially private.

In this paper, we set to evaluate the impact of releasing aggregate location time-series on the privacy of individuals contributing to the aggregation. We introduce a framework allowing us to reason about privacy against an adversary attempting to predict users' locations or recover their mobility patterns. We formalize these attacks as inference problems, and discuss a few strategies to model the adversary's prior knowledge based on the information she may have access to. We then use the framework to quantify the privacy loss stemming from aggregate location data, *with* and *without* the protection of differential privacy, using two real-world mobility datasets. We find that aggregates do leak information about individuals' punctual locations and mobility profiles. The density of the observations, as well as timing, play important roles, e.g., regular patterns during peak hours are better protected than sporadic movements. Finally, our evaluation shows that both output and input perturbation offer little additional protection, unless they introduce large amounts of noise ultimately destroying the utility of the data.

Keywords: location privacy; privacy quantification; aggregate locations; inference attacks; differential privacy

DOI 10.1515/popets-2017-0028
Received 2017-02-28; revised 2017-06-01; accepted 2017-06-02.

1 Introduction

The availability of people's locations and movements supports progress in "mobility analytics" – e.g., applications geared to improve urban planning [4], study the effect of "shocks" on transport [44], predict events [22], detect traffic anomalies [32], generate real-time traffic statistics [1], etc. At the same time, however, large-scale collection of individuals' whereabouts prompts serious privacy concerns, as location data may reveal one's occupation, lifestyle, as well as political and religious beliefs [25, 33]. A possible approach toward mitigating these concerns is to anonymize location traces prior to releasing them. Alas, this is ineffective, as location data is inherently unique to the user, and the identities of the subjects generating the traces can often be recovered [11, 20, 52].

In some cases, mobility models can be trained using only aggregate statistics [10, 31, 37], e.g., how many people are in a certain location at a given time. Therefore, a common approach is to consider aggregation as a privacy defense, and, by using appropriate cryptographic protocols, the aggregation can take place in a privacy-preserving way, i.e., removing the need for a trusted aggregator [24, 36, 37]. Moreover, Differential Privacy (DP) [13] can be used to bound the privacy leakage from releasing aggregate statistics [2, 21], using output [8, 14, 39] or input [17, 38] perturbation. However, there is no sound method to reason about the privacy lost by single individuals from the release of raw aggregate time-series. Even when using DP, we only get privacy guarantees in terms of the theoretical upper-bounds provided by a generic indistinguishability parameter – ϵ. Existing location privacy quantification frameworks [41, 42] do not help either, as they typically focus on evaluating single user-centric privacy defense mechanisms (e.g., when one user accesses a location-based service).

In this paper, we present a framework geared to address this gap, and use it to provide a thorough evaluation of aggregation-based location privacy. We consider an adversary aiming to perform *localization* attacks, i.e., recovering users' punctual locations, as well as *profiling*, i.e., inferring their mobility patterns. We define appropriate metrics to express the privacy lost from the availability of the aggregates, with respect to the adversary's prior knowledge. We propose a few approaches to build such priors, parameterized by location and

***Corresponding Author: Apostolos Pyrgelis:** University College London, E-mail: apostolos.pyrgelis.14@ucl.ac.uk
Carmela Troncoso: IMDEA Software Institute, E-mail: carmela.troncoso@imdea.org
Emiliano De Cristofaro: University College London, E-mail: e.decristofaro@ucl.ac.uk

time observations available to her (e.g., users' frequent locations on a Monday morning, or observations from the previous week, etc.), and discuss inference strategies, which employ either Bayesian reasoning or greedy approaches to improve the knowledge of users' whereabouts, by using the aggregates.

We then utilize our framework to experimentally measure users' privacy loss when raw aggregate time-series are released. We use two mobility datasets obtained from Transport for London and the San Francisco Cab network. Our comparative analysis shows that, overall, aggregates do improve the adversary's prior knowledge about mobility patterns and help her localize users. Users' loss of privacy depends not only on the prior knowledge and the inference strategy of the adversary, but also on the density of her observations. Furthermore, the adversary's inference power is influenced by the nature of the patterns to infer, being regular movements (e.g., peak hours/weekdays) better protected than irregular ones (e.g., evenings/weekends).

Next, we study the privacy protection provided by DP mechanisms as compared to the release of raw aggregates, vis-à-vis the utility they provide. Although DP ensures an upper bound on the amount of leakage, determined by the ϵ parameter, it is often difficult to interpret its real-world meaning and to choose appropriate values for it, despite directly affecting the resulting utility of the data. Using our framework, we measure the privacy gain provided by using DP techniques, and find that, in our adversarial model, these mechanisms only provide meaningful additional privacy protection if the noise they introduce is so high that data utility is ultimately destroyed. This holds for both input and output perturbation techniques.

Our results demonstrate that, while differential privacy offers a promising privacy-enhancing solution to several analytics and data mining problems, its use in location-oriented applications (including those recently announced by Google [15] and Apple [23]) needs to be carefully evaluated with respect to the actual privacy it provides. Overall, our work highlights the need for novel defense mechanisms that can offer better privacy guarantees to individuals whose location data is part of aggregate time-series releases.

Paper Organization. The next section reviews some background information. Then, in Section 3, we formalize the problem of quantifying privacy leakage from aggregate location time-series. Section 4 presents an experimental evaluation on two real-world mobility datasets, while Section 5 analyzes DP techniques for protecting aggregates. After reviewing related work in Section 6, the paper concludes in Section 7.

2 Preliminaries

Kullback-Leibler (KL) Divergence [27]. Also known as discrimination information, the Kullback-Leibler (KL) divergence captures the "difference" between two probability distributions. Specifically, for two discrete probability distributions W and X, the KL-divergence from X to W is defined as:

$$\mathsf{D_{KL}}(\mathsf{W}||\mathsf{X}) = \sum_{\mathsf{i}} \mathsf{W}(\mathsf{i}) \cdot \log \frac{\mathsf{W}(\mathsf{i})}{\mathsf{X}(\mathsf{i})} \quad (1)$$

where W usually represents the *true* distribution of data and X an approximation of W. In other words, KL-divergence from X to W measures the information lost when X is used to approximate W. Note that KL is not a *metric* as it does not satisfy the triangle equality and in general not symmetric in W and X.

Jensen-Shannon (JS) Divergence [16, 29]. It is used to calculate the similarity between two probability distributions. It is based on KL-divergence but it is symmetric and always a finite value. The JS-divergence is a smoothed version of the KL-divergence $\mathsf{D_{KL}}(\mathsf{W}||\mathsf{X})$, defined by:

$$\mathsf{JS}(\mathsf{W}||\mathsf{X}) = \frac{1}{2} \cdot \mathsf{D_{KL}}(\mathsf{W}||\mathsf{Z}) + \frac{1}{2} \cdot \mathsf{D_{KL}}(\mathsf{X}||\mathsf{Z}) \quad (2)$$

where $\mathsf{Z} = \frac{1}{2} \cdot (\mathsf{W} + \mathsf{X})$. When employing the base 2 logarithm for calculating KL-divergence, the JS-divergence is bounded by 1, thus $0 \leq \mathsf{JS}(\mathsf{W}||\mathsf{X}) \leq 1$. Note that the square root of the JS-divergence is a *metric* denoted as Jensen-Shannon distance [16] (also bounded by 1). We use JS-distance to calculate the adversarial error in profiling users.

F1 Score. F1 is often used to evaluate the accuracy of classification/prediction tasks, as it captures overall performance by taking into account both precision and recall. Precision (aka positive predictive value, or PPV) and recall (aka true positive rate, or TPR) are defined, respectively, as $\mathsf{PPV} = \mathsf{TP}/(\mathsf{TP} + \mathsf{FP})$ and $\mathsf{TPR} = \mathsf{TP}/(\mathsf{TP} + \mathsf{FN})$, where TP, FP, and FN denote, respectively, true positives, false positives, and false negatives. The $\mathsf{F1}$ score is calculated as:

$$\mathsf{F1} = \frac{2 \cdot \mathsf{TPR} \cdot \mathsf{PPV}}{\mathsf{TPR} + \mathsf{PPV}} \quad (3)$$

We use it to quantify adversary's accuracy in localizing users.

3 Quantifying Aggregate Location Privacy

3.1 Problem Statement

In the rest of the paper, we use the notation summarized in Table 1. We consider a set of users U that move among a set S of regions of interest (ROIs) – e.g., landmarks, neighborhoods,

Symbol	Description
Adv	Adversary
U	Set of mobile users
S	Set of locations (ROIs)
T	Time period considered
T'	Inference period
$\tilde{T}$	Observation period
L	Ground truth
L^p	Ground truth mobility profile
A	Aggregate time-series
A^p	Aggregate mobility profile
A'	Perturbed aggregate time-series
P	Adv's prior knowledge
$\hat{P}$	Adv's inference output

Table 1. Notation.

stations – at time instances in the set T. This set represents the time frame in which locations are collected (e.g., 1 week, 1 month, 1 year), while locations can be aggregated in epochs of different granularity (e.g., 15 mins, 30 mins, 1 hour).

Ground truth. We model the actual locations S of a user $u \in U$, during T, using a *ground truth matrix* L of size $|S| \times |T|$, in which rows are ROIs and columns are epochs. L is a binary matrix s.t. $l_{s,t} \in L$ is 1 if the user was in $s \in S$ during epoch $t \in T$, and 0 otherwise. We note that depending on the time granularity of location reports users can be in more than one ROI in the same epoch, thus, there can be more than one 1 per column. We also define a *mobility profile*, L^p, where $l^p_{s,t} \in L^p$ represents the probability that a user is in location s at time slot t and is computed as $l_{s,t}/\sum_{j \in S} l_{j,t}$.

Aggregates. The aggregate location time-series is represented by the matrix A, of size $|S| \times |T'|$. We call T' the *inference period*, i.e., aggregation does not need to happen in the full collection period. Each item $a_{s,t'} \in A$ represents the number of users in s at epoch t', and is calculated as $a_{s,t'} = \sum_{u=1}^{|U|} l_{s,t'}$, where $l_{s,t'}$ are the entries of each user's L. The aggregation can be performed by a trusted aggregator or via cryptographic protocols [24, 36, 37]. We also define A^p, the *aggregate mobility profile*, as a probability distribution matrix whose entries $a^p_{s,t'} \in A^p$ are computed as $a_{s,t'}/\sum_{j \in S} a_{j,t'}$. This represents the probability of users being in a ROI at an epoch, while observing the aggregates. For instance, $a^p_{s,t'} = 0.1$ indicates that at time t', 10% of the user observations are in ROI s.

Prior knowledge. We model the prior knowledge the adversary, denoted as Adv, may have about a user $u \in U$ for the inference period T', using a matrix P, of size $|S| \times |T'|$. P can be probabilistic (i.e., describing how likely a user is to visit a ROI) or binary (i.e., indicating whether a user will visit a ROI or not). We discuss how to build these priors in Section 3.2.

Quantifying aggregate location privacy. Given the observation of the aggregates (A), and the prior knowledge about each user (P), Adv aims to infer information about individual users from the time-series. We model the output of this inference as a matrix $\hat{P}$, for each user, of size $|S| \times |T'|$. We do so to quantify the privacy loss for individual users given the adversary's prior knowledge and her capability to exploit the aggregates. Specifically, we measure the adversary's *error* vis-à-vis the ground truth L, after executing inference attacks, considering two goals: user profiling and user localization.

User Profiling: Adv aims to infer the mobility profile of the users. Given P and A (or A^p), Adv outputs a matrix $\hat{P}$. This matrix contains a probability distribution profile for each user reflecting the likelihood that the user is in each ROI at each epoch. To compute Adv's error, we compare the ground truth mobility profile L^p to Adv's inference $\hat{P}$ if we consider the result of the inference attack, or to the prior P if aggregate data is not available. For each user and each $t' \in T'$, we use the JS metric, reviewed in Section 2, to measure the distance between the probability distributions. For each user profile, we measure Adv's total error over the inference period T' as:

$$\mathsf{AdvErr_{JS}} = \frac{\sum_{t' \in T'} \mathsf{JS}(L^p_{t'} || P_{t'})}{|T'|} \tag{4}$$

Intuitively, at each time slot, JS computes the distance between the inferred and the ground truth profile, averaged over the ROIs. This captures the adversary's error regarding profile estimation. Eq. 4 averages the distance per slot over all time slots, i.e., it computes the adversary's mean error on the inference period.

User Localization: Adv aims to infer the punctual locations of the users over time. More formally, given P and A (or A^p), Adv outputs a binary matrix $\hat{P}$ for each user, with 1's for ROIs Adv predicts the user to be in, and 0's elsewhere. To measure Adv's performance we compare her predictive assignment matrix on each user (either prior P or posterior $\hat{P}$) against the ground truth L. Concretely, we use Adv's precision and recall when predicting users' locations to derive the F1 score, reviewed in Section 2, and measure the total adversarial error as:

$$\mathsf{AdvErr_{F1}} = 1 - \mathsf{F1} \tag{5}$$

The F1 score captures the distance between the inferred and the ground truth binary matrices, i.e., Eq. 5 reflects the adversarial error regarding localization over the inference period. Note that both adversarial goals have been considered in location privacy literature [12, 25, 42, 49], although in different contexts, namely, reconstructing traces or recovering a user's location from obfuscated individual data.

Privacy Loss (PL). For both adversarial goals, we measure the privacy loss for an individual user from the aggregate location time-series as the normalized difference between Adv's error using her prior knowledge (P), with and without A ($\mathsf{AdvErr_{P,A}}$ and $\mathsf{AdvErr_P}$ resp.). More specifically, for each user we define the privacy loss (PL) as:

$$\mathsf{PL} = \begin{cases} \frac{|\mathsf{AdvErr}_{\mathsf{P,A}} - \mathsf{AdvErr}_{\mathsf{P}}|}{\mathsf{AdvErr}_{\mathsf{P}}} & \text{if } \mathsf{AdvErr}_{\mathsf{P}} \neq 0 \wedge \\ & \quad \mathsf{AdvErr}_{\mathsf{P,A}} < \mathsf{AdvErr}_{\mathsf{P}} \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

PL is a value between 0 and 1 and captures Adv's improvement towards her goal (either profiling or localizing users).

3.2 Adversary's Prior Knowledge

We now present a few different approaches to build the adversary's prior knowledge, which we divide in *probabilistic* priors, i.e., user profiles averaging location reports, and *assignment* ones, i.e., binary matrices representing users' location visits over a certain period of time. Essentially, they differ in how the P matrix is populated, depending on what information is assumed to be available to the adversary and the strategy used to extract prior knowledge about each user.

In real life, adversarial prior knowledge may originate from, e.g., social networks, data leaks, location traces released by providers, personal knowledge. Here we aim to describe a generic quantification framework, comparing different adversarial strategies, hence, we opt to construct priors from a subset of the users' ground truth matrices (L), including epochs in $\tilde{\mathsf{T}} \subseteq \mathsf{T}$, which we call the *observation period*. We follow intuitive strategies, based on a sensible threat model in which Adv obtains information about users' routines and punctual locations (e.g., where one works/lives) over a certain period of time. Nonetheless, our framework is generic enough so that new ways of building Adv's priors can be easily incorporated.

3.2.1 Probabilistic Priors

Probabilistic priors model prior information that represent knowledge of user profiles.

ROI Frequency. We start by considering that Adv knows the probability of a user visiting a given ROI during the full observation period. We assume that Adv has access to a vector of size $|\mathsf{S}|$, indicating how frequently the user visits each ROI during $\tilde{\mathsf{T}}$. Adv then populates P by: (i) transforming the vector into a probability distribution using the total number of user's observations, M, as normalizing factor, and (ii) copying the distribution onto P, for all time slots of the inference period T'. More specifically, using the entries $\mathsf{l}_{\mathsf{s,t}}$ in the ground truth L, we build P, $\forall \mathsf{s} \in \mathsf{S}, \mathsf{t}' \in \mathsf{T}'$, as:

$$\mathsf{P}_{\mathsf{FREQ\_ROI}}(\mathsf{s},\mathsf{t}') := \sum_{\mathsf{t} \in \tilde{\mathsf{T}}} \mathsf{l}_{\mathsf{s,t}}/\mathsf{M} \tag{7}$$

ROI Seasonality. This prior models the case that Adv knows the seasonal probability of a user visiting a ROI during the observation period $\tilde{\mathsf{T}}$, for a given seasonal time period SEAS. For instance, if SEAS corresponds to one day, and epochs are of one hour, we assume that Adv obtains a probability distribution over the ROIs for every hour in a day. If seasonality is on days of the week, the probability distribution over ROIs available to the adversary is for each hour, for each day of the week. More formally, if c denotes the seasonality cycle of SEAS (e.g., $\mathsf{c} = 24$ hours for daily or $\mathsf{c} = 7 \cdot 24$ hours for weekly seasonalities), then the seasonality profile is, $\forall \mathsf{s} \in \mathsf{S}, \forall \mathsf{i} \in \{1,\dots,\mathsf{c}\}$:

$$\mathsf{ROI\_SEAS}_{\mathsf{s,i}} = \frac{\sum_{\mathsf{k}=0}^{\tilde{\mathsf{T}}/\mathsf{c}-1} \mathsf{l}_{\mathsf{s,i+k\cdot c}}}{\sum_{\mathsf{j}\in\mathsf{S}} \sum_{\mathsf{k}=0}^{\tilde{\mathsf{T}}/\mathsf{c}-1} \mathsf{l}_{\mathsf{j,i+k\cdot c}}} \tag{8}$$

Then, we build P, $\forall \mathsf{s} \in \mathsf{S}, \mathsf{t}' \in \mathsf{T}'$ as:

$$\mathsf{P}_{\mathsf{ROI\_SEAS}}(\mathsf{s},\mathsf{t}') := \mathsf{ROI\_SEAS}_{\mathsf{s},\mathsf{t}' \bmod \mathsf{c}} \tag{9}$$

Time Seasonality. We assume Adv knows the seasonal probability of a user reporting her location (without any information about which concrete ROIs) during the observation period $\tilde{\mathsf{T}}$, for a given seasonal time period SEAS. For instance, if SEAS corresponds to one day, and the granularity is one hour, Adv learns which hours of a day a user is likely to report locations. More formally, if c denotes the seasonality cycle of SEAS, then the time seasonality profile is, $\forall \mathsf{i} \in \{1,\dots,\mathsf{c}\}$:

$$\mathsf{TIME\_SEAS}_{\mathsf{i}} = \frac{\sum_{\mathsf{j}\in\mathsf{S}} \sum_{\mathsf{k}=0}^{\tilde{\mathsf{T}}/\mathsf{c}-1} \mathsf{l}_{\mathsf{j,i+k\cdot c}}}{\mathsf{M}} \tag{10}$$

where M is the total number of user's observations within the period $\tilde{\mathsf{T}}$. Then, P is built, $\forall \mathsf{s} \in \mathsf{S}, \mathsf{t}' \in \mathsf{T}'$, as:

$$\mathsf{P}_{\mathsf{TIME\_SEAS}}(\mathsf{s},\mathsf{t}') := \begin{cases} 1/|\mathsf{S}| & \text{if } \mathsf{TIME\_SEAS}_{\mathsf{t}' \bmod \mathsf{c}} > 0 \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

i.e., it is a *uniform* probability distribution over ROIs for the time slots when the user is likely to report locations.

3.2.2 Assignment Priors

Next, we describe strategies to compute prior information that represents knowledge of users' punctual locations. An assignment prior is modeled as a binary matrix that predicts whether or not a user will be in a location $\mathsf{s} \in \mathsf{S}$, at time $\mathsf{t}' \in \mathsf{T}'$.

Most popular prior ROIs. We model the case that Adv only considers users' favorite locations (POP). Given a probabilistic prior knowledge P, and a threshold δ modeling what the adversary considers to be *favorite*, Adv builds a binary location matrix so that $\forall \mathsf{s} \in \mathsf{S}, \mathsf{t}' \in \mathsf{T}'$:

$$\mathsf{P}_{\mathsf{POP}}(\mathsf{s},\mathsf{t}') := \begin{cases} 1 & \text{if } \mathsf{P}_{\mathsf{s,t}'} \geq \delta \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

All prior ROIs. Next, we consider a scenario where Adv considers every location that users visit, but not the frequency

Prior	Description
FREQ_ROI	Frequent ROIs, over time
ROI_DAY	Most frequent ROIs, for each time instance of a day
ROI_DAY_WEEK	Most frequent ROIs, for each time instance of a week
TIME_DAY	Most frequent time instances of a day, reporting ROIs
TIME_DAY_WEEK	Most frequent time instances of a week, reporting ROIs
LAST_WEEK	Last week's ROIs
LAST_DAY	Last day's ROIs
LAST_HOUR	Last hour's ROIs

Table 2. Different ways to build adversarial prior knowledge.

(ALL). Given a probabilistic prior knowledge P, Adv builds a binary location matrix so that $\forall \mathsf{s} \in \mathsf{S}, \mathsf{t}' \in \mathsf{T}'$:

$$\mathsf{P}_{\mathsf{ALL}}(\mathsf{s}, \mathsf{t}') := \mathsf{CEIL}(\mathsf{P}_{\mathsf{s},\mathsf{t}'}) \tag{13}$$

where CEIL is the ceiling function, thus $\mathsf{P}_{\mathsf{ALL}}(\mathsf{s}, \mathsf{t}') = 1$ iff the probability of visiting s at t' is greater than 0 (i.e., the user has visited that location during time slots of $\tilde{\mathsf{T}}$).

Last Season. We assume that Adv has access to the last seasonal information for each user, i.e., the last season SEAS constitutes the observation period $\tilde{\mathsf{T}}$. For instance, if SEAS corresponds to 1 day and time granularity is 1 hour, Adv only knows the locations visited in each hour of the last day. Formally, if c denotes the seasonality cycle, e.g., $\mathsf{c} = 7 \cdot 24$ hours for weekly seasonality, P is built utilizing a sliding window as:

$$\forall \mathsf{s} \in \mathsf{S}, \mathsf{t}' \in \mathsf{T}' \; : \mathsf{P}_{\mathsf{LAST\_SEAS}}(\mathsf{s}, \mathsf{t}') := \mathsf{L}_{\mathsf{s},\mathsf{t}'-\mathsf{c}} \tag{14}$$

Summary. Table 2 summarizes our approaches to construct the adversarial prior knowledge. For priors taking *seasonality* into account, SEAS takes a value indicating the seasonal period we consider to build Adv's initial knowledge.

3.3 Location Inference Strategies

We now describe possible strategies that the adversary can follow to exploit aggregate locations in order to make inferences about individuals. We present algorithms that, taking as input Adv's prior knowledge about a given user (P) and the location aggregate time-series (A or A^{p}), output an updated matrix $\hat{\mathsf{P}}$. This matrix represents Adv's posterior knowledge about the user's whereabouts over the inference period T' by virtue of the availability of the aggregate time-series. The proposed strategies can be used for both profiling and localization attacks, the difference being the nature of the output matrix $\hat{\mathsf{P}}$, which is probabilistic in the former case and binary in the latter. In the following, we use $\Theta(:, \mathsf{x})$ or $\Theta(\mathsf{x}, :)$ to denote all the instances of a dimension in a matrix Θ.

Bayesian Updating. The first strategy, summarized in Algorithm 3.1, computes the posterior probability of $\mathsf{u} \in \mathsf{U}$ being in each ROI $\mathsf{s} \in \mathsf{S}$, during $\mathsf{t}' \in \mathsf{T}'$, given the adversarial prior knowledge (P) and the aggregate mobility profile A^{p}. Let $\mathsf{E}_{\mathsf{s},\mathsf{t}'}$ denote the event in which the user appears in location $\mathsf{s} \in \mathsf{S}$ at time $\mathsf{t}' \in \mathsf{T}'$. Given Adv's prior information about this event, $\Pr[\mathsf{E}_{\mathsf{s},\mathsf{t}'}] = \mathsf{P}_{\mathsf{s},\mathsf{t}'}$, and her observation O at time t', i.e., a probability distribution of users over all ROIs $\mathsf{s} \in \mathsf{S}$ at time t', we compute the posterior probability using the Bayes theorem as:

$$\hat{\mathsf{P}}_{\mathsf{s},\mathsf{t}'} = \Pr[\mathsf{E}_{\mathsf{s},\mathsf{t}'}|\mathsf{O}] = \frac{\Pr[\mathsf{O}|\mathsf{E}_{\mathsf{s},\mathsf{t}'}] \cdot \Pr[\mathsf{E}_{\mathsf{s},\mathsf{t}'}]}{\Pr[\mathsf{O}]} \tag{15}$$

where $\Pr[\mathsf{O}]$ represents the user's un-normalized distribution over ROIs at time t' and can be calculated using the law of total probability, i.e., $\Pr[\mathsf{O}] = \sum_{\mathsf{j} \in \mathsf{S}} \mathsf{A}^{\mathsf{p}}_{\mathsf{j},\mathsf{t}'} \cdot \mathsf{P}_{\mathsf{j},\mathsf{t}'}$, and $\Pr[\mathsf{O}|\mathsf{E}_{\mathsf{s},\mathsf{t}'}] = \mathsf{A}^{\mathsf{p}}_{\mathsf{s},\mathsf{t}'}$ is given by the released aggregate statistics.

Algorithm 3.1: BAYES

```
Input: $P, A^p$
for each $u \in U$ do
    for each $t' \in T'$ do
        $\hat{P}(:, t') = P(:, t') \times A^p(:, t')$
        $\hat{P}(:, t') = \hat{P}(:, t') / \sum_{j \in S} \hat{P}(j, t')$
    return $\hat{P}$;
```

Algorithm 3.2: MAX_ROI

```
   Input: $P, A$
 1 $P_U(:, :, :) = \emptyset$
 2 $LOC_U(:, :, :) = \emptyset$
 3 for each $u \in U$ do
 4     $P_U = P_U || P^{(u)}$
 5 for each $s \in S, t' \in T'$ do
 6     if $A(s, t') == 0$ then
 7         $LOC_U(:, s, t') = 0$
 8     else
 9         $X = P_U(:, s, t')$
10         $U^* = SORT(X, A(s, t'))$
11         for each $z \in U^*$ do
12             $LOC_U(z, s, t') = 1$
13 for each $u \in U$ do
14     $\hat{P} = LOC_U(u, :, :)$
15     return $\hat{P}$;
```

Max-ROI. The Bayesian approach is well-principled but considers users independently, thus losing information related to the fact that at most $\mathsf{A}_{\mathsf{s},\mathsf{t}'}$ users can be assigned to a location s, at time t'. We now describe a *greedy* alternative that accounts for this constraint. The algorithm aims at maximizing the total probability for each ROI by assigning the most probable users to each location. It is summarized in Algorithm 3.2, where $\mathsf{SORT}(\mathsf{V}, \mathsf{x})$ denotes a function that returns the indexes of the *top* x values, of a vector V. Specifically, Adv first concatenates the probabilistic prior P matrices of all users $\mathsf{u} \in \mathsf{U}$ and creates a 3-dimensional matrix of size $|\mathsf{U}| \times |\mathsf{S}| \times |\mathsf{T}'|$, which we denote as P_{U} (lines 3–4, Algorithm 3.2). Additionally, she creates a localization matrix of same size, denoted as $\mathsf{LOC}_{\mathsf{U}}$. Next, Adv selects all s, t' s.t. $\mathsf{A}_{\mathsf{s},\mathsf{t}'} = 0$ and sets the corresponding indexes of $\mathsf{LOC}_{\mathsf{U}}$ to zero, i.e., she discards locations where no users have been observed during the aggre-

Algorithm 3.3: MAX_USER

```
   Input: $P, A$
1  $LOC_U(:,:,:) = \emptyset$
2  for each $t' \in T'$ do
3      for each $u \in U$ do
4          $Idx = INDEX(P(:,t') > 0.0)$
5          for each $i \in Idx$ do
6              if $\sum_{v \in U} LOC_U(v,i,t') < A(i,t')$ then
7                  $LOC_U(u,i,t') = 1$
8          if $\sum_{w \in U, j \in S} LOC_U(w,j,t') == \sum_{j \in S} A(j,t')$
             then
9              $break$;
10 for each $u \in U$ do
11     $\hat{P} = LOC_U(u,:,:)$
12     return $\hat{P}$;
```

gation period (lines 6–7). Then, for all non-zero entries in A, she selects the $A_{s,t'}$ most probable users according to her prior, P_U, setting the corresponding indexes in LOC_U to 1 (lines 9–12). If there are users with equal probability, Adv can use any criterion, e.g., the total number of location reports (as we do in our experiments that are presented in Section 4) to make a decision. Finally, Adv outputs the location assignment profile of each user as her $\hat{P}$ matrix (lines 13–15).

Max-User. Our final inference attack is similar in spirit to the previous *greedy* strategy but, rather than maximizing the probability over ROIs, it maximizes each user's probability over the ROIs by assigning them to their most likely locations. The algorithm is summarized in Algorithm 3.3, where $INDEX(V > x)$ denotes a function that returns the indexes of a vector V, whose values are larger than x. More precisely, Adv first sorts users by some criterion, e.g., the total number of locations that they report (as we will do in our experiments in Section 4). Then, at each time slot $t' \in T'$, Adv iterates over the users and assigns each of them to their most likely ROIs, provided that each ROI's aggregate $A(s,t')$ is still not consumed (lines 3–7, Algorithm 3.3). The procedure is repeated until the assignments cover all the revealed aggregate information (lines 8–9).

Note: Our strategies are suitable for both Adv's inference goals, i.e., profiling and localization. For instance, if Adv is given a probabilistic prior, she can follow MAX_ROI or MAX_USER strategies and transform their assignment outputs to probability distributions that can be used for her profiling goal. Similarly, she can run BAYES on the prior and evaluate POP and ALL on its output to localize users.

4 Privacy Evaluation of Raw Aggregates

We now use our framework to experimentally evaluate aggregate location privacy from raw aggregates release. We compare different approaches to build priors (Section 3.2) as well as strategies to perform inference attacks (Section 3.3), using two mobility datasets obtained from London's transportation authority and the San Francisco Cab network.

4.1 Datasets

Transport for London (TFL). We have obtained, from the TFL authority, all Oyster card trips on the TFL network from March 2010 (8GB uncompressed). The Oyster is a personal, pre-paid, RFID-enabled card, and the most common payment system on TFL-operated services. Each entry in the data describes a unique trip and consists of the following fields: (anonymized) oyster card id, start time, touch-in station id, end time, and touch-out station id. (Note that the same dataset has also been used in [7, 28, 37]).

Pre-processing & Sampling. We discard trips from March 29–31, 2010 to obtain exactly four weeks of data, i.e., from Monday March 1st to Sunday 28th. This yields 60 million trips, performed by 4 million unique oyster cards, covering 582 stations (ROIs). Next, we select the top 10,000 oyster ids per total number of trips: these account for about 6M trips (10%). Considering oyster trips start/end stations as ROIs, the top 10,000 users report, 171 ± 26 ROIs in total and 19 ± 9 unique ROIs. Setting the time granularity to one hour, the mean number of *active* time slots for the top 10,000 oysters is 115 ± 21 out of the 672 slots (28 days×24 hours).

Ground Truth. We use the trips performed by each Oyster card in the dataset to populate its ground truth matrix L. More specifically, $l_{s,t} \in L$ is 1 if the user touched-in or out at station s, during time slot $t \in T$, and 0 otherwise. When an Oyster card does not report any location at a particular time slot, we assign it to a special ROI denoted as *null*. Thus, the ground truth L is a matrix of size $|S| \times |T| = 583 \times 672$.

Prior Knowledge (Training data). We build the probabilistic adversarial prior knowledge using the *first 3 weeks* of L (i.e., 75% of data are used for training). Thus, the *observation* period $\tilde{T}$ consists of $21 \times 24 = 504$ hourly time slots. For the seasonal assignment priors, we utilize a *sliding window* on L, as described in Section 3.2.2.

Testing Data & Aggregates. We evaluate Adv's performance in profiling/localizing users against the *last week* of L (i.e., 25% of the data are used for testing). Thus, the *inference* period T'

consists of $7 \times 24 = 168$ hourly time slots. For each station $s \in S$, we count the number of users that report their presence in it (touch-in or touch-out) during each epoch $t' \in T'$, and create the aggregate time-series A (of size $|S| \times |T'| = 583 \times 168$) whose items $a_{s,t'}$ are computed as $\sum_{u=1}^{|U|} l_{s,t'}$ (remind that $l_{s,t'}$ are the entries of each oyster's L). During T', each station is reported $818 \pm 1,361$ times while stations have commuters touching in/out for 71 ± 54 out of the 168 hourly time slots.

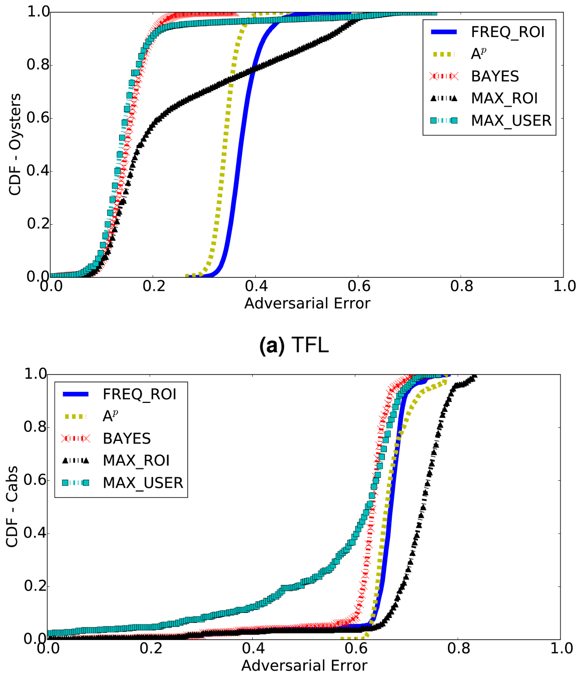
San Francisco Cabs (SFC). We also use the SFC dataset [34], with mobility traces recorded by cabs in the San Francisco area from May 17 to June 10, 2008. Each record includes: cab identifier, latitude, longitude and a time stamp.

Pre-processing & Sampling. The dataset consists of approximately 11 million GPS coordinates, generated by 536 taxis. To facilitate our experiments, we focus on exactly 3 weeks: Monday May 19 to Sunday June 8. We restrict to the downtown San Francisco area, dividing it into a grid of $10 \times 10 = 100$ regions (ROIs), each covering an area of 0.5×0.37 square miles. We group traces in one-hour epochs. We also remove duplicates (e.g., a taxi reporting the same ROI multiple times during a time slot). This yields a dataset of over 2 million ROIs reported by 534 taxis, reporting $3,663 \pm 1,116$ locations in total, covering 77 ± 6 unique ROIs (out of the 100 we consider). Unlike TFL data, the SFC data is less sparse, with cabs reporting more locations. On average, cabs are "inside the system" for 340 ± 94 out of the 504 (21 days×24 hours) time slots.

Ground Truth. For each cab, we build its L matrix, by setting $l_{s,t}$ to 1 if the cab was in the s "cell" during time slot $t \in T$, and 0 otherwise. As for TFL data, if a cab does not report any location at a time slot we assign it to a special ROI, which we denote as *null*, thus, the L matrix is of size $|S| \times |T| = 101 \times 504$.

Prior Knowledge (Training data). We build the probabilistic adversarial prior knowledge using the *first 2 weeks* of L (i.e., $\sim 66\%$ of the data), thus the *observation* period $\tilde{T}$ consists of $14 \times 24 = 336$ hourly time slots. For the LAST_SEAS assignment priors, we utilize a sliding window on L.

Testing Data & Aggregates. We quantify Adv's performance in profiling/localizing cabs against the *last week* of L (i.e., $\sim 33\%$ of the data are used for testing), thus, the inference period T' consists of $7 \times 24 = 168$ hourly time slots. For each $s \in S$, we count the number of taxis reporting it during epoch $t' \in T'$, and create the aggregate time-series A (of size $|S| \times |T'| = 101 \times 168$) whose items $a_{s,t'}$ are computed as $\sum_{u=1}^{|U|} l_{s,t'}$ (where $l_{s,t'}$ are the entries of each cab's L). During T', each ROI is reported $6,714 \pm 7,624$ times while ROIs have taxis in them for 135 ± 61 out of the 168 time slots.



(a) TFL

(b) SFC

Fig. 1. Adv's Profiling Error - FREQ_ROI prior.

4.2 User Profiling

We start our experimental evaluation by quantifying aggregate location privacy against *user profiling* (cf. Section 3.1). We study the impact of the information used to build Adv's prior vis-à-vis the strategy used to exploit aggregate data. Specifically, we measure Adv's performance using the JS distance from the ground truth (Eq. 4), and use this metric in our plots (Figures 1–3). During our analysis, we also discuss the privacy loss (PL, Eq. 6), allowing us to better understand the effect of aggregate data publication on privacy, independently of the prior mobility pattern of the user, as PL reflects how much the adversary has learned with respect to her initial knowledge.

4.2.1 Probabilistic Priors

FREQ_ROI. In Fig. 1, we plot the CDF (over the user population) of Adv's error over the testing week when building priors using FREQ_ROI, i.e., each user's frequent ROIs over time, for both datasets, using different inference strategies. Using the TFL dataset (Fig. 1a), a baseline attack where Adv uses only her prior (blue line) has an average error of 0.37, while A^p (i.e., the population profile extracted from the aggregates) reduces her error to 0.34. When Adv uses both her prior and the aggregates for the inference, the error is notably reduced, yielding average errors amounting to 0.15, 0.25 and 0.15, respectively, with BAYES, MAX_ROI and MAX_USER.

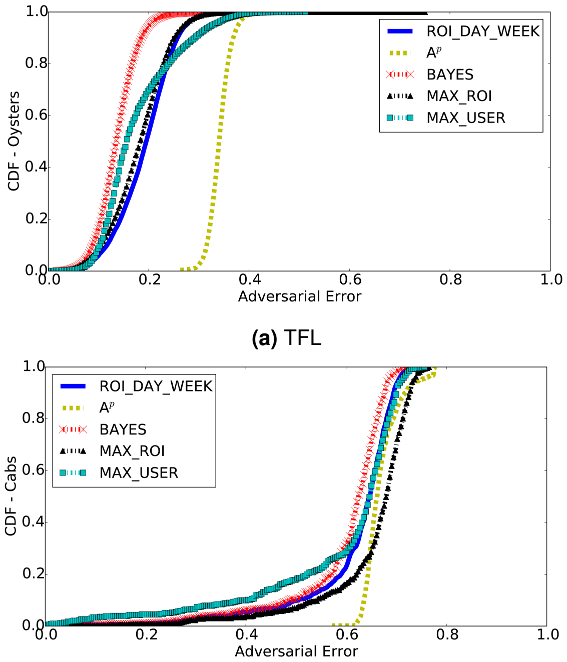
More specifically, this corresponds to an average privacy loss of, resp., 0.6, 0.41, and 0.59 for individuals whose locations are included in the aggregate time-series.

We also observe that inferences affect users in different ways, i.e., with BAYES, the adversarial error is reduced for all users, while with MAX_ROI and MAX_USER for 77% and 95% of all users, respectively. This confirms that MAX_ROI and MAX_USER are somewhat greedy strategies and may end up selecting users that either report few (MAX_ROI) or many (MAX_USER) ROIs overall, to "consume" the aggregates.

With the SFC data (cf. Fig. 1b), the adversarial error only relying on cabs' frequent ROIs prior (FREQ_ROI) is higher compared to that of TFL – 0.65 on average, and in this case it is quite similar to that owing to the aggregates (A^P). It drops to 0.62 with the Bayesian updating (corresponding to 0.06 privacy loss) and to 0.56 with MAX_USER (0.16 PL), indicating that taxis reporting the most locations are regular within them and end up losing more privacy. We also observe that, unlike in the TFL experiments, the greedy strategy MAX_ROI actually deteriorates Adv's mean error (0.71), owing to the bias introduced by taxis visiting few ROIs (i.e., cabs having high probability to appear in a ROI). Overall, we find that profiling commuters based on their frequent ROIs is more effective than profiling cabs, as cabs report more locations and follow variable routes during their shifts.

ROI_DAY_WEEK. Next, we report Adv's error when using the ROI_DAY_WEEK as her prior, i.e., a weekly profile that takes into account location frequency as well as time and day semantics (e.g., users' locations on Mondays, 3pm). The results are plotted in Fig. 2, for both datasets. We have also experimented with location frequency and time only (and not day) semantics to build the prior (ROI_DAY), which yields larger errors, as less information is considered. To ease presentation, we defer details to Appendix A.1.1.

With the TFL data (Fig. 2a), it is clear that commuters' most frequent ROIs for the time instances of a week (ROI_DAY_WEEK) are a more informative prior than their frequent ROIs (FREQ_ROI), with an average prior error as low as 0.19. This shows how time and day semantics help Adv profile tube commuters. MAX_ROI and MAX_USER strategies slightly enhance Adv's posterior knowledge and result in, resp., 0.08 and 0.14 mean privacy loss. Whereas, the Bayesian inference significantly improves Adv's performance towards her profiling goal, yielding an average of 0.27 privacy loss for the users. With the SFC dataset (Fig. 2b), the average prior error is lower than with FREQ_ROI as time and day semantics enhance Adv's performance, but it still remains relatively high (0.61). Two of the inference strategies reduce Adv's error, although not dramatically: BAYES and MAX_USER help Adv to profile cabs' mobility and yield, resp., 0.03 and 0.07 privacy loss. In contrast, MAX_ROI actually deteriorates Adv's performance and does not harm the cabs' privacy. Overall, we notice that profiling cabs using their weekly profiles as prior knowledge is more challenging than profiling commuters whose mobility patterns are more regular.



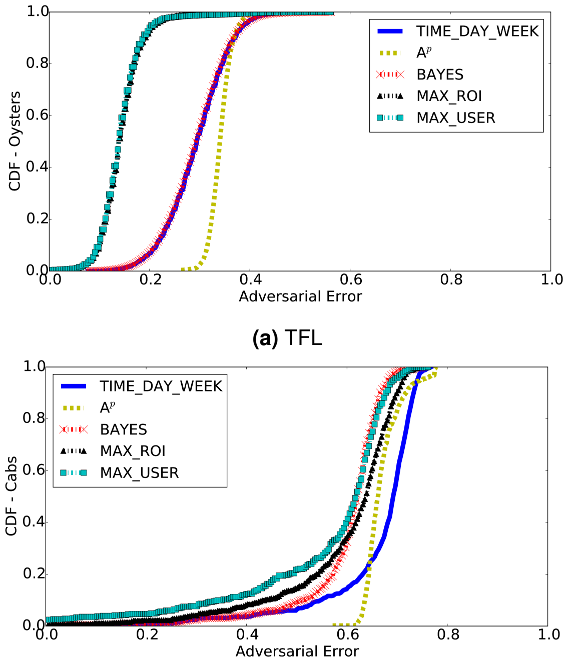
(a) TFL

(b) SFC

Fig. 2. Adv's Profiling Error - ROI_DAY_WEEK Prior.

TIME_DAY_WEEK. Our last experiments with probabilistic priors measure Adv's error (see Fig. 3) when her prior knowledge consists only of time information for the users, i.e., she knows which time slots of the inference week a user is likely to report ROIs, but not which ROIs. Similar experiments in which Adv knows which time slots of *any* day a user reports ROIs (TIME_DAY) result in larger error and are discussed in Appendix A.1.1. With the TFL data (Fig. 3a), the error based on this prior is larger than with ROI_DAY_WEEK, namely, 0.3. "Greedy" strategies (MAX_ROI and MAX_USER) remarkably improve Adv's performance (i.e., they result in 0.5 privacy loss on average), as in this case the users reporting the most ROIs are chosen to consume the aggregates (due to the prior, users have equal probability to appear in ROIs). On the other hand, the Bayesian inference only slightly decreases the adversarial error, due to the small probabilities of her prior, which consists of a uniform distribution over the tube stations for the users' most frequent time slots of a week.

With the SFC data (Fig. 3b), when Adv knows the cabs' most frequent time slots reporting ROIs, her prior error is larger compared to that of cabs' frequent ROIs over the time instances of a week (ROI_DAY_WEEK), i.e., 0.66. However, exploiting the aggregate knowledge the error is reduced



(a) TFL

(b) SFC

Fig. 3. Adv's Profiling Error - TIME_DAY_WEEK Prior.

and BAYES, MAX_ROI and MAX_USER inferences yield $0.09, 0.1$ and 0.2 mean privacy loss, respectively. Overall, we point out that due to the different nature of the datasets (sparse TFL vs. dense SFC), user profiling with time information as prior knowledge yields different amounts of privacy leakage.

4.2.2 Assignment Priors

Next, we evaluate Adv's performance with assignment priors, i.e., when she obtains a historical location profile as her prior knowledge for the users. We experiment with LAST_WEEK, LAST_DAY and LAST_HOUR, described in Section 3.2.2. Unlike probabilistic ones, the privacy loss from aggregates with assignment priors is very small, as the sliding window on the ground truth of commuters/cabs already yields highly informative priors. Since the CDF plots are less illustrative in this setting, we defer them to Appendix A.1.2 due to space limitations (Figures 12–13).

With TFL, when Adv knows users' last week's whereabouts (LAST_WEEK), her baseline mean error is 0.17, indicating that commuters are fairly regular in their weekly patterns. BAYES, MAX_ROI and MAX_USER inferences somewhat reduce Adv's error and achieve only little privacy loss (0.01, 0.03 and 0.05 resp.). When the users' last day's ROIs are available to Adv (LAST_DAY), her initial error is comparable to LAST_WEEK but smaller (0.15 on average). BAYES and MAX_ROI only slightly reduce the adversarial error, causing, resp., 0.02 and 0.05 privacy loss. On the contrary, MAX_USER does not harm commuters' privacy as it actually increases Adv's error, indicating the most mobile users might not follow the patterns of their previous day. LAST_HOUR generates larger error compared to the previous ones (0.19), as passengers do not exhibit as strong hourly seasonality. Once again, all inferences yield negligible privacy loss. In general for TFL, we remark that seasonal historic profiles are more instructive priors than probabilistic ones (e.g., FREQ_ROI or TIME_DAY_WEEK), thus, the privacy loss for individuals from the aggregate time-series is actually small compared to that of probabilistic priors.

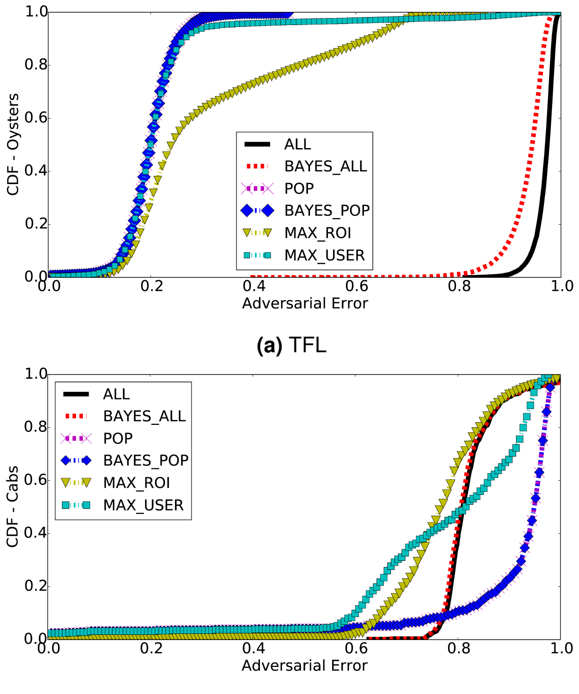
Our experiments on the SFC data show that, unlike TFL, LAST_HOUR is the most "revealing" among the assignment priors, with a mean error of 0.53 (vs. 0.63 for LAST_DAY and 0.67 for LAST_WEEK). Interestingly, Adv profiles cabs more efficiently knowing their last hour's ROIs than with probabilistic priors, e.g., their most frequent ROIs (FREQ_ROI) or their most frequent ROIs for the time slots of a week (ROI_DAY_WEEK). That is, cabs of San Francisco are more likely to appear in those ROIs they visited during the last hour, while their daily/weekly patterns are less regular. In all assignment prior cases, BAYES and MAX_USER reduce Adv's error by little, while MAX_ROI increases it, thus, the privacy loss from the aggregates is again quite low.

4.2.3 Take Aways

Overall, our experiments show that aggregates do help the adversary on the profiling inference goal. The actual degree of privacy loss for the users depends on the prior: assignment ones yield smaller privacy leakages, as they are already quite informative for the adversary compared to probabilistic ones. We also observe that inferring the mobility profiles of commuters from aggregates is significantly easier than profiling cabs. In other words, cabs' patterns are not as regular as those of tube passengers, who exhibit high seasonality. As a consequence, commuters lose much more privacy than cabs from aggregate locations.

4.3 User Localization

We now measure privacy loss in the context of *localization* attacks, i.e., as Adv attempts to predict users' future locations. Our experimental setup is the same as with profiling. However, Adv's output is not a probability distribution, but a binary localization matrix, and Adv's main performance metric (error) is now computed as $1 - \text{F1}$ (see Eq. 5).



(a) TFL

(b) SFC

Fig. 4. Adv's Localization Error - FREQ_ROI Prior.

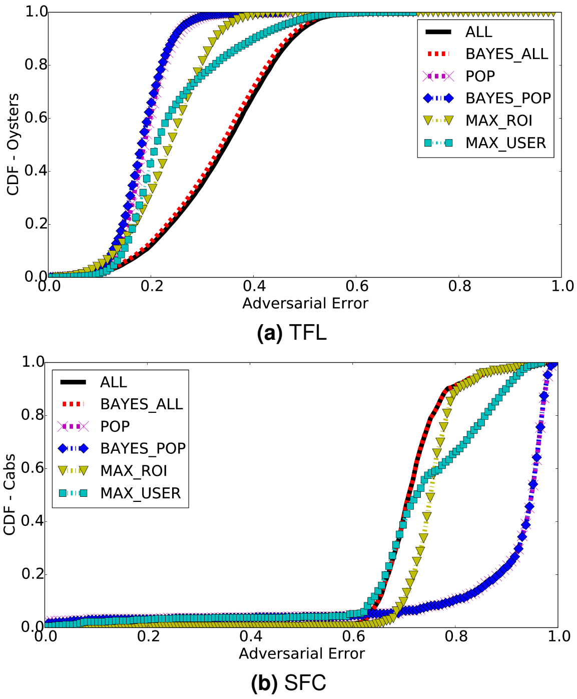
4.3.1 Probabilistic Priors

We quantify Adv's error in localizing users when the prior knowledge matrix P is built according to users' most frequent ROIs over time (FREQ_ROI) and their most frequent ROIs for each time slot of a week (ROI_DAY_WEEK). Since her prior is a probability distribution over ROIs for each time slot of the inference period, Adv's baseline prediction is to extract the users' most popular prior ROIs (POP) or all prior ROIs (ALL) (cf. Section 3.2.2). For POP, we set the threshold δ to 0.5, i.e., we consider users' favorite ROIs those with more than 50% chance of visiting. As part of her inference strategy, Adv (i) applies BAYES and evaluates POP and ALL on its output, and (ii) employs MAX_ROI and MAX_USER. Figures 4–5 plot the corresponding results, while additional experiments with Adv knowing users' most frequent time slots of a week reporting ROIs are deferred to Appendix A.2.1.

FREQ_ROI. Fig. 4a plots the CDF of Adv's error in localizing TFL passengers with their frequent ROIs over time as prior knowledge. Using only the prior, i.e., predicting that users will appear in all their frequent ROIS (ALL), we get a very large average error (0.97); evaluating ALL after applying the Bayesian inference slightly reduces the adversarial error (0.93) and yields very small privacy loss (on average, 0.04). When predicting that commuters will appear in their most popular ROIs (POP), Adv's mean error drops to 0.21. Again, BAYES does not improve Adv's performance, as the prior probabilities are so small that, after updating, they do not exceed δ. We observe that with POP, Adv predicts users to be *out* of the transportation system during the time slots of T′. Interestingly, such a conservative strategy yields a small adversarial error overall, however, this occurs due to the fact that the TFL dataset is relatively *sparse*. With the greedy inference strategies (MAX_ROI and MAX_USER), Adv's mean error is much smaller than ALL, respectively, 0.32 and 0.23. Their error patterns are different as they select different sets of users to cover the aggregates. MAX_ROI achieves an error of 0.5 or less for 70% of the users, while MAX_USER for 90%. In both cases, Adv's error is reduced notably in comparison with the ALL baseline strategy, and we find that the aggregates do indeed yield substantial privacy loss (resp., 0.66 and 0.77).

In Fig. 4b, we plot the CDF of Adv's error while attempting to localize SFC cabs over T′, again given their most frequent ROIs as prior. Similar to TFL experiments, when Adv extracts cabs' most popular prior ROIs (POP), she predicts all of them to be *outside* the network, since the prior probabilities are smaller than the threshold ($\delta = 0.5$), and the Bayesian inference updates them negligibly. However, unlike TFL, Adv's error with POP is 0.9 on average, proving it to be a bad strategy for localizing cabs. Predicting that cabs will show up in all their prior ROIs (ALL) slightly improves her predictive power as the mean error drops to 0.83, while BAYES negligibly reduces it further. Both MAX_ROI and MAX_USER inferences improve Adv's predictions compared to the ALL baseline, and they yield, resp., 0.08 and 0.11 privacy loss. However, we observe that MAX_ROI behaves more consistently than MAX_USER (which reduces Adv's error only for 50% of the cabs), indicating ROI regularity. Overall, it is clear that localization strategies behave quite differently on datasets of dissimilar characteristics.

ROI_DAY_WEEK. Fig. 5 displays the CDF of Adv's error localizing users with their most frequent ROIs for each time slot of a week (ROI_DAY_WEEK) as prior knowledge. For TFL, we notice that all prior ROIs yield a mean adversarial error of 0.34 and Bayesian updating slightly reduces it and yields insignificant privacy loss (0.03). In this case, users' most popular prior ROIs (POP) reduce Adv's error to 0.19. The BAYES and POP inference results in a negligible mean privacy loss (0.06). In contrast, compared to ALL, MAX_ROI and MAX_USER generate a notable privacy loss (0.29 and 0.26 on average). MAX_USER yields larger errors for users that are selected to cover the aggregates, while the error gets smaller for those users that were not (because the aggregates were consumed). On the other hand, MAX_ROI predicts better than MAX_USER for 25% of users, who are highly regular in the ROIs they visit. In comparison to FREQ_ROI,



(a) TFL

(b) SFC

Fig. 5. Adv's Localization Error - ROI_DAY_WEEK Prior.

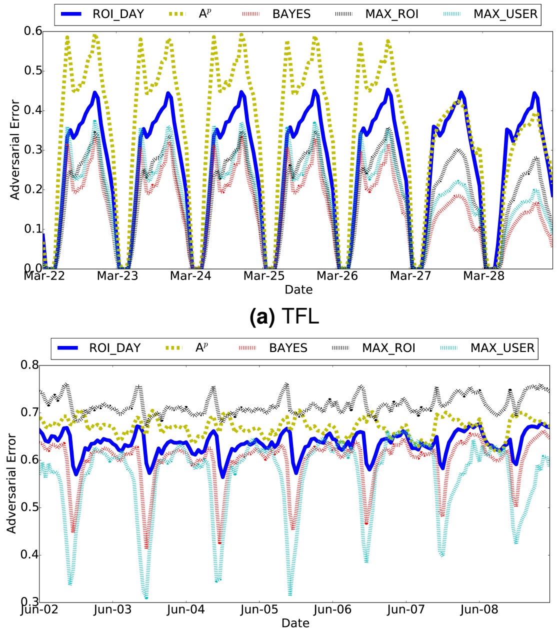
ROI_DAY_WEEK enables Adv to localize commuters more efficiently, proving it to be a more informative prior.

With the SFC data (Fig. 5b), localizing cabs via all their prior ROIs (ALL) yields a mean error of 0.71, while BAYES reduces it insignificantly. Interestingly, with ROI_DAY_WEEK being an instructive prior, ALL proves to be the best strategy. Extracting the cabs most popular ROIs (POP) results in an average error of 0.9 confirming once again that this strategy does not perform well on the dense cab data. Furthermore, MAX_ROI and MAX_USER yield significant privacy loss (resp., 0.17 and 0.18), compared to the baseline POP.

Overall, our experiments demonstrate that Adv is more effective in localizing commuters/cabs with ROI_DAY_WEEK than FREQ_ROI, however, the privacy loss for individuals is smaller due to the more revealing prior knowledge.

4.3.2 Assignment Priors

Finally, we assume Adv obtains a historical assignment prior for the users, i.e., we experiment with LAST_WEEK, LAST_DAY, and LAST_HOUR priors in the context of the localization inference task. Due to space limitations, we defer the details of the corresponding results (and plots) to Appendix A.2.2. We find that TFL commuters are best localized with their last week's ROIs (average error is 0.24 with LAST_WEEK, 0.27 with LAST_DAY, and 0.31 with LAST_HOUR), whereas, SFC cabs with their last hour's ROIs (average error is 0.73 with LAST_WEEK, 0.71 with LAST_DAY and 0.64 with LAST_HOUR). Moreover, as in the profiling case, the availability of aggregates yields limited privacy loss when the adversarial prior knowledge is built via assignments. This indicates that, since assignment priors are already quite instructive, the aggregates do not significantly improve Adv's knowledge of individual users' whereabouts.



(a) TFL

(b) SFC

Fig. 6. Adv's Hourly Profiling Error - ROI_DAY Prior.

4.3.3 Take Aways

Similar to profiling, localization inferences performed using the aggregates yield different degrees of loss in privacy for individual users, depending on Adv's prior knowledge. Assignment priors are more revealing than probabilistic ones, thus aggregates end up leaking less privacy overall. We also observe that commuters are best localized via their popular ROIs (POP), while cabs by their most recent ROIs (LAST_HOUR). Once again, localizing commuters is easier than localizing cabs as the former ones exhibit seasonality, while the latter ones have irregular patterns.

4.4 Privacy Implications of Regular Mobility Patterns

Experimenting with our framework also provides some interesting considerations about Adv's error over the time slots of the inference week. One would expect the leakage to vary according to time of the day (e.g., peak hours vs. night) or days of the week (e.g., weekdays vs. weekends) since the number of users in the system, and their concentration, varies significantly. In this case, users would have variable levels of privacy protection over time. In order to validate this intuition, we pick a case-study out of our experimental setup and examine the patterns in Adv's mean error during the hourly time slots of the inference week. Fig. 6 plots the evolution of Adv's mean error over time for tube passengers (TFL) and taxis (SFC), when Adv obtains their most frequent ROIs for the time slots of day (ROI_DAY) as prior knowledge.

For TFL (Fig. 6a), we observe different patterns w.r.t. hours of the day and weekdays, as expected. Only considering the prior (ROI_DAY), Adv's error is smaller in the morning hours than mid-day or evening hours, likely because tube passengers are regular in their commuting routines to work, while in the evening they might go to the gym, meet friends, or go shopping before traveling back home. As the aggregate time-series is available to Adv, her error is reduced during morning hours not nearly as much as in mid-day and evening hours. In other words, commuters lose more privacy if they travel during mid-day, as there are fewer users in the transportation system, or in the evening hours, because the aggregates reflect their irregular mobility pattern. Similarly, we observe that the aggregates give Adv a much more significant advantage during the weekends than on weekdays, as commuters more likely follow variable routes.

Likewise, for SFC (Fig. 6b), we observe distinct patterns in Adv's error w.r.t hours of the day and weekdays. With the prior (ROI_DAY), Adv's error has a spike in the morning peak hours of weekdays indicating that cabs follow variable routes at these times and are not highly predictable. We find that Adv's prior error is smaller (0.57) during mid-day hours (i.e., 12pm–4pm) as cabs might be parked waiting for clients, or fewer routes might be performed during that shift. Indeed, the availability of the aggregate time-series harms cabs' privacy more during mid-day time slots as BAYES and MAX_USER reduce Adv's error significantly (higher privacy loss). Finally, we note that, among the inference strategies, MAX_USER gives Adv remarkable advantage in profiling cabs during weekends as the cabs reporting the most ROIs are likely to follow routes that are reflected by the aggregates.

5 Privacy Evaluation of Defense Mechanisms

In the previous section, we have shown that aggregate location time-series leak information about individuals' whereabouts, and have evaluated how, based on different priors and inferences. Next, we study whether mechanisms supporting the release of aggregate information in a privacy-respecting manner are effective at avoiding such privacy leakage, and to what extent. Specifically, we focus on the protection offered by Differential Privacy (DP) [13], using either output or input perturbation techniques. The former add noise to the output of the aggregation process, whereas, with the latter, noise is added to users' inputs before aggregation. We do not consider other defense mechanisms, e.g., based on k-anonymity, as they have already been shown to be ineffective [43].

In theory, one can assess the level of privacy provided by DP mechanisms as it is configured by the parameter ϵ, which determines the privacy risk incurred when releasing statistics computed on sensitive data (providing an upper bound). While ϵ expresses the relation between the level of privacy before and after the release, and provides protection against arbitrary risks, it is not an absolute measure of privacy and it is often not clear how to interpret, in practice, the actual level of privacy enjoyed by individuals in the dataset, nor is how to choose the value of ϵ to obtain the desired protection.

In the rest of this section, we use our framework to measure to which extent DP mechanisms reduce the privacy leakage compared to the release of raw aggregates, vis-à-vis the resulting utility of the data. That is, we quantify the protection that these mechanisms provide to users in presence of an adversary that, as in Section 4, has access to the aggregates (now perturbed via a DP mechanism) and uses that information to improve her prior knowledge about users' whereabouts.

5.1 Metrics

Privacy Gain. We quantify the protection provided by DP techniques in terms of the "privacy gain" they yield, which we define to denote the difference in Adv's error when using her prior (P) with the noisy aggregates A′ ($\mathsf{AdvErr}_{\mathsf{P,A'}}$) minus that with the raw aggregates A ($\mathsf{AdvErr}_{\mathsf{P,A}}$), normalized by the maximum gain the mechanism can provide. That is, we measure privacy gain (PG) as:

$$\mathrm{PG} = \begin{cases} \frac{\mathsf{AdvErr}_{\mathsf{P,A'}} - \mathsf{AdvErr}_{\mathsf{P,A}}}{1 - \mathsf{AdvErr}_{\mathsf{P,A}}} & \text{if } \mathsf{AdvErr}_{\mathsf{P,A}} \neq 1 \wedge \\ & \quad \mathsf{AdvErr}_{\mathsf{P,A'}} > \mathsf{AdvErr}_{\mathsf{P,A}} \\ 0 & \text{otherwise} \end{cases} \tag{16}$$

PG is a value between 0 and 1 capturing Adv's deterioration towards her goal (e.g., profiling users) owing to the noise added by the DP techniques.

Mean Relative Error (MRE). We also use the MRE to measure utility, specifically, to capture the error between an original time-series Y of n time points and its *noisy* version Y′, which comes as the result of perturbation. More precisely:

$$\mathsf{MRE}(\mathsf{Y}, \mathsf{Y'}) = (1/\mathsf{n}) \sum_{\mathsf{i}=0}^{\mathsf{n}} \frac{|\mathsf{Y}'_{\mathsf{i}} - \mathsf{Y}_{\mathsf{i}}|}{\mathsf{max}(\beta, \mathsf{Y}_{\mathsf{i}})} \tag{17}$$

where β is a sanity bound mitigating the effects of very small counts. As done in previous work [2], we use MRE to measure the utility loss when a privacy mechanism is applied to an aggregate time-series, and we adjust β to 0.1% of $\sum_{i=0}^{n} Y_i$.

5.2 Output Perturbation

We first evaluate differentially private mechanisms based on output perturbation, in which an entity adds noise to the statistics prior to their release. This entity can be trusted with the individual users' data [2, 18] or only be allowed to compute aggregate statistics, e.g., using cryptographic protocols for private aggregation [5, 36, 37]. We evaluate two specific approaches: the Simple Counter Mechanism [8, 14] and the Fourier Perturbation Algorithm [39].

Simple Counter Mechanism (SCM) [8, 14]. SCM is a straightforward extension of the Laplace mechanism proposed by Dwork et al. [13] for time-series. It answers a new query at each time slot (e.g., how many users are in a ROI at that time) and randomizes the answer with fresh independent noise. Given ϵ, for a ROI $s \in S$, for each time slot $t' \in T'$, SCM samples a fresh random value from the Laplace distribution $\gamma_{t'} \sim \mathsf{Lap}(1/\epsilon)$ (recall that each user is counted at most *once* in $A_{st'}$) and releases the perturbed aggregate $A'_{st'} = A_{st'} + \gamma_{t'}$, where $A_{st'}$ is the true aggregate value.

Due to the composition theorem [14], and given that the number of locations and time slots in the inference period for which data is released, are $|S|$ and $|T'|$ respectively, the mechanism is overall $O(|S| \cdot |T'| \cdot \epsilon)$ differentially private. Thus, the privacy leakage increases linearly with the number of ROIs and the length of the inference period. This version of SCM only guarantees *event-level* privacy [8, 14] for the users, i.e., it protects whether or not a user was in a ROI at a specific time slot. If one desires to achieve stronger privacy guarantees with SCM, then the noise can be distributed according to $\mathsf{Lap}(|T'|/\epsilon)$ (i.e., users are protected within the aggregates of a region, during the whole period T') and SCM becomes $O(|S| \cdot \epsilon)$ differentially private. Alternatively, to guarantee ϵ-DP (i.e., users are protected within the aggregates of all regions, during T') the noise must be distributed according to $\mathsf{Lap}(|S| \cdot |T'|/\epsilon)$, increasing privacy at the cost of utility.

Fourier Perturbation Algorithm (FPA) [39]. FPA improves the privacy/utility trade-off offered by SCM by reducing the amount of noise needed to obtain the same level of privacy. This reduction is based on performing the noise addition in the compressed domain as follows. First the time-series is compressed using the Discrete Fourier Transform (DFT) and the first k Fourier coefficients, F_k, are kept. Then, F_k is perturbed with noise distributed according to $\mathsf{Lap}(\sqrt{k \cdot |T'|}/\epsilon)$, and padded with zeros to the size of the original time-series. Finally, the inverse DFT is applied to obtain the perturbed aggregates to be released. This version of FPA guarantees ϵ-DP for each ROI (thus, overall it's $O(|S| \cdot \epsilon)$ differentially private) with better utility than SCM. Note that a mechanism similar to FPA has also been applied in [2].

Evaluation. We present the results of our evaluation on two case-studies: (i) user profiling on the TFL dataset with Adv obtaining FREQ_ROI as her prior knowledge and following the greedy MAX_ROI strategy, and (ii) user profiling on the SFC data when Adv knows FREQ_ROI and employs MAX_USER. Although we restrict to two cases, due to space limitations, their choice is reasonable as our analysis in Section 4 shows that, in these settings, the aggregates yield significant privacy loss for individual users.

We parameterize SCM and FPA perturbation mechanisms with $\epsilon \in \{0.001, 0.01, 0.1, 1.0\}$. For SCM, we experiment with variable magnitude of Laplacian noise to demonstrate the actual protection it offers with respect to its theoritical privacy guarantees. Since SCM with $\mathsf{Lap}(|S| \cdot |T'|/\epsilon)$ is expected to yield unnecessarily huge error in the aggregates (i.e., it is practically impossible for commuters/cabs to appear in *all* ROIs in *every* time slot of the inference period), we also report SCM with noise distributed according to $\mathsf{Lap}(\Delta/\epsilon)$, where Δ is the sensitivity of users within the aggregates A, i.e., the maximum number of location reports by a user/cab during T' in the TFL and SFC datasets (224 and 2,687 resp.). Furthermore, for FPA, as done in [39], we experiment with the parameter k to minimize its total error, finding that $k = 25$ yields the best results on TFL and $k = 20$ on the SFC data.

Utility. Tables 3 and 4 report the utility for both datasets, in terms of MRE, of the mechanisms for different values of ϵ. Overall, as expected, for all mechanisms the higher the privacy (i.e., lower ϵ values), the lower the utility (i.e., bigger MRE).

In our first case study (Table 3), SCM-$\mathsf{Lap}(|S| \cdot |T'|/\epsilon)$ yields the worse utility, with perturbed aggregates being about 700 times worse estimates than raw ones, for all values of ϵ. Moreover, SCM-$\mathsf{Lap}(\Delta/\epsilon)$ and SCM-$\mathsf{Lap}(|T'|/\epsilon)$ still yield very high errors, even for a mild level of privacy ($\epsilon = 0.01$). The highest utility is provided by SCM-$\mathsf{Lap}(1/\epsilon)$, followed by FPA. Nonetheless, with the former, the utility is at least 8 times worse than the raw aggregates (MRE=7.8) for small ϵ values (0.01 or less). In our second case study (Table 4), we observe that SCM-$\mathsf{Lap}(|S| \cdot |T'|/\epsilon)$ and SCM-$\mathsf{Lap}(\Delta/\epsilon)$ result in very large errors (MRE$\geq$ 24), while SCM-$\mathsf{Lap}(|T'|/\epsilon)$ follows closely. FPA and SCM-$\mathsf{Lap}(1/\epsilon)$ yield the best utility, although for sensible levels of (expected) privacy (i.e., $\epsilon = 0.01$) the perturbed aggregates are about 8 and 5 times worse estimates than the raw ones, respectively.

ϵ	0.001	0.01	0.1	1.0
SCM - Lap($\|S\| \cdot \|T'\|/\epsilon$)	739.9	743.2	735.8	709.4
SCM - Lap(Δ/ϵ)	720.1	605.1	168.9	16.7
SCM - Lap($\|T'\|/\epsilon$)	719.8	549.6	123.5	12.8
FPA	117.1	11.7	1.3	0.3
SCM - Lap($1/\epsilon$)	74.4	7.8	0.9	0.1

Table 3. TFL: MRE (Utility) of output perturbation mechanisms.

ϵ	0.001	0.01	0.1	1.0
SCM - Lap($\|S\| \cdot \|T'\|/\epsilon$)	26.8	26.3	26.2	26.2
SCM - Lap(Δ/ϵ)	26.9	26.5	25.9	24.3
SCM - Lap($\|T'\|/\epsilon$)	26.1	25.9	22.4	8.3
FPA	24.1	8.8	1.1	0.3
SCM - Lap($1/\epsilon$)	19.9	5.1	0.6	0.1

Table 4. SFC: MRE (Utility) of output perturbation mechanisms.

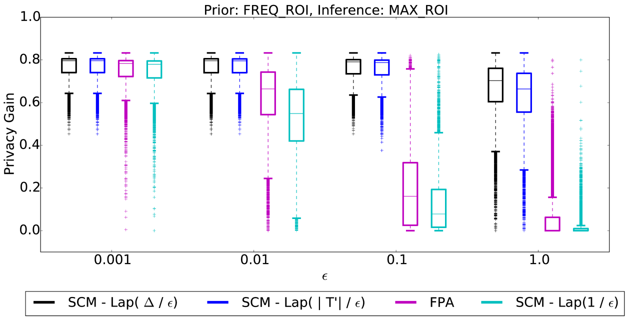


Fig. 7. TFL: Privacy gain for output perturbation DP mechanisms.

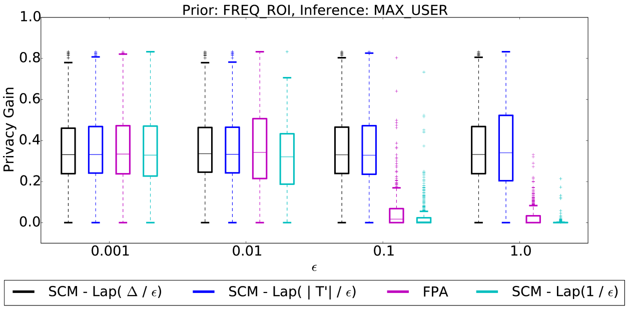


Fig. 8. SFC: Privacy gain for output perturbation DP mechanisms.

Privacy Quantification. Figs. 7 and 8 display box-plots of the privacy gain (PG, see Eq. 16) enjoyed by individual users in both datasets thanks to the perturbation mechanisms, for increasing values of ϵ. To ease presentation, the plots do not include SCM-Lap($|S| \cdot |T'|/\epsilon$), which, as discussed earlier, completely destroys utility. In the TFL case study (Fig. 7), the four mechanisms exhibit very different behaviors. SCM-Lap(Δ/ϵ) and SCM-Lap($|T'|/\epsilon$) offer the best privacy protection, with an average privacy gain as high as 0.77 for $\epsilon \leq 0.1$, and 0.65 and 0.62, resp., for $\epsilon = 1.0$. However, as discussed above (and shown in Table 3), this protection comes with very poor utility. We also find that SCM-Lap($1/\epsilon$) and FPA offer similar protection (PG=0.74 on average) for $\epsilon = 0.001$, while, as ϵ grows, the gain drops significantly, being negligible when $\epsilon = 1.0$. While this is somewhat expected for SCM-Lap($1/\epsilon$), it is quite surprising for FPA, which in theory should provide as much protection as SCM-Lap($|T'|/\epsilon$).

In the SFC case (Fig. 8), we observe that SCM-Lap(Δ/ϵ) and SCM-Lap($|T'|/\epsilon$) provide the best privacy gain (0.36 on avg.) for all values of ϵ. FPA and SCM-Lap($1/\epsilon$) behave similarly to the previous two for $\epsilon \leq 0.01$, however, as ϵ increases the privacy gain approaches zero.

Remarks. Our evaluation not only highlights a possible gap between theory and practice w.r.t. privacy guarantees offered by DP mechanisms, but also shows that these struggle to offer strong privacy under continual observation (as in the case of aggregate location time-series) without destroying utility. For instance, FPA with $\epsilon = 0.01$ provides reasonably high gain in privacy (PG=0.62) for TFL commuters, however, the MRE of the published aggregates is approximately 11. For instance, if there are 100 people in an underground station, the system will report that there are instead 1,200. Similarly, on the SFC dataset, when FPA provides good level of privacy for cabs (i.e., PG $= 0.36$ with $\epsilon = 0.01$), the MRE is almost 9.

5.3 Input Perturbation

We now look at input perturbation-based DP techniques, whereby users add noise to their inputs prior to the aggregation process. In particular, we focus on Randomized Response (RR) [17, 38, 47]. We do not consider geo-indistinguishability [3], a mechanism to provide individual users with differential privacy guarantees while using location-based services, since there is no scheme that uses such approach to collect or release aggregate locations. (In fact, we consider this as an interesting open problem for future work.)

Randomized Response (RR) can be used to privately collect statistics from users participating in surveys [47], crowdsourcing statistics from client software [17], sharing historical traffic data [15], as well as privately aggregating user locations in real-time [38]. In particular, the SpotMe system [38] lets users perturb their location at each time instance $t' \in T'$ by claiming to be in a ROI $s \in S$ (a "yes" response) with some probability p, or report the truth (i.e., whether they are or not in location s) with probability $1 - p$. The aggregator collects the perturbed user inputs and computes the aggregation estimating the number of individuals in each location $s \in S$ and every time slot $t' \in T'$, via $A_{s,t'} = \mathrm{total}_{s,t'} \cdot \frac{\mathrm{Pyes}_{s,t'} - p}{1-p}$, where $\mathrm{total}_{s,t'}$ is the total number of responses received for ROI s at time t' and $\mathrm{Pyes}_{s,t'} = \frac{\mathrm{yes}_{s,t'}}{\mathrm{total}_{s,t'}}$ depicts the proportion of "yes" responses. This mechanism is $\ln \frac{|S|-(|S|-1)\cdot p}{p}$-DP at each time slot [48],

p	0.1	0.3	0.5	0.7	0.9
TFL - MRE	2.1	3.9	6.1	9.3	17.6
SFC - MRE	0.4	0.7	1.1	1.6	2.9

Table 5. SpotMe [38]: MRE (Utility) for increasing values of p, on TFL and SFC datasets.

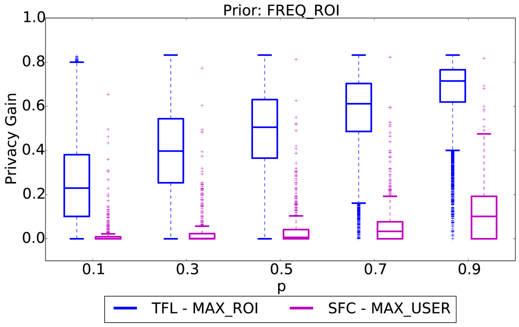


Fig. 9. SpotMe [38]: Privacy gain for increasing values of p, on TFL and SFC datasets.

thus, overall it guarantees $O(|T'| \cdot \ln \frac{|S|-(|S|-1)\cdot p}{p})$ differential privacy due to the composition theorem.

Evaluation. We evaluate SpotMe [38], as a representative for RR input perturbation mechanisms, using our framework. In this context, Adv is assumed to obtain the estimated perturbed aggregates A' that result as users apply the RR mechanism on their inputs. As in the output perturbation case, we focus on two user profiling case-studies: (i) TFL data with FREQ_ROI adversarial prior knowledge and MAX_ROI inference, and (ii) SFC dataset with FREQ_ROI prior and MAX_USER strategy.

Utility. Table 5 shows the MRE of the perturbed aggregates, highlighting that, as p grows (i.e., as commuters/cabs perturb their inputs with higher probability) the utility of the aggregates declines. For TFL, with $p = 0.1$, the MRE over all stations is 2.1, and 17.6 with $p = 0.9$. For SFC, the MRE over all ROIs is 0.4 for $p = 0.1$, while for $p = 0.9$ the perturbed aggregates are approximately 3 times worse than the raw ones.

Privacy Quantification. Fig. 9 plots the privacy gain provided by the RR mechanism w.r.t. the parameter p, for users in both TFL and SFC datasets. For TFL, we observe that, as p increases, PG also increments, reaching up to 0.6 with the most conservative parameterization ($p = 0.9$). In comparison to the output perturbation mechanisms applied on TFL data, SpotMe yields smaller privacy gains while keeping the utility levels higher. Interestingly, for SFC, we observe that, as p grows, the privacy gain only increases negligibly. For $p = 0.5$, the average PG is 0.04, while it's only 0.1 when $p = 0.9$. Recall that, with output perturbation mechanisms on the SFC data, privacy gain reaches 0.36, although yielding lower utility. This highlights the challenges of using RR mechanisms, such as SpotMe, on dense datasets with few users.

5.4 Discussion

Our evaluation of defense mechanisms based on differential privacy (DP) highlights the difficulty to fine-tune the trade-off between privacy and utility. More specifically, our case studies show that using existing DP mechanisms in a straightforward manner yields poor utility in the context of aggregate location time-series in the settings considered in this paper, i.e., mobility analytics over transport data. As expected, we observe that the performance of DP mechanisms in terms of privacy and utility is highly dependent on the intrinsic characteristics of the datasets used in our experiments. For instance, in the sparse TFL dataset containing thousands of users moving among a relatively large number of ROIs (583), output and input perturbation achieve reasonable levels of privacy, with the latter performing better than the former in terms of utility. On the other hand, on the denser SFC dataset, which includes fewer users and ROIs (101), output perturbation does not yield significant privacy protection, and input perturbation only a negligible one. Moreover, our analysis shows that it is challenging to achieve good utility while applying DP on continuous data, such as aggregate location time-series, and mechanisms that reduce the required amount of noise (e.g., FPA) still do not provide acceptable privacy guarantees.

Moreover, data pre-processing techniques [2], like subsampling and clustering, could theoretically be used to improve the utility of DP mechanisms (e.g., by reducing the number of locations reported by the users or merging sparse ROIs together), however, such an approach is application dependent and cannot be considered a generalizable solution.

Finally, although the generic framework of differential privacy abstracts from adversarial prior knowledge, our analysis indicates that the concrete nature of this prior should be taken into account when evaluating defense mechanisms. While some priors may not help the adversary, our experiments show that realistic approaches of building adversarial prior knowledge, for example considering users' frequent locations, can help an adversary when performing inference attacks to extract knowledge, even from aggregates perturbed with DP.

6 Related Work

Attacks on Location Privacy. Prior work presenting attacks on location privacy mostly focuses on inferring users' whereabouts from access to individuals' location data, whether obfuscated or not. Some show that both anonymization and k-anonymity-based mechanisms are ineffective at protecting privacy [20, 33, 42, 43, 52]. (Also see surveys by Krumm [26] and Ghinita [19]). More recently, researchers analyzed the protection provided by location proximity schemes adopted by social networks [35, 46, 51], confirming that mechanisms like cloaking or naive perturbation are also unsuccessful.

Independently of our work, Xu et al. [50] have recently presented an attack that recovers individual users' trajectories from aggregate mobility data, by exploiting the uniqueness and the regularity of human mobility. Although their setting is somewhat similar to ours, the adversarial task they consider is quite different. Moreover, our work introduces a methodology to reason about the effect of releasing location aggregates on individuals' privacy—with and without DP protection.

Privacy-Preserving Aggregation. There are two main privacy-enhancing strategies to collect location data and compute aggregate time-series. (1) Cryptographic protocols for private aggregation can let a server obtain aggregates without learning users' individual records [31, 36, 37], but make no consideration about the privacy loss from learning and/or releasing exact statistics. We have evaluated this scenario in Section 4. (2) Perturbation techniques can be used to hide individual inputs rather than encrypting them. Ho et al. [21] use quadtree spatial decomposition and density based clustering for privately mining location databases, while Kopp et al. [24]'s framework enables the collection of quantitative visits to sets of locations following a distributed approach. Chen et al. [9] focus on spatial data aggregation in the local setting and propose a framework that allows an untrusted server to learn the user distribution over a spatial domain relying on a personalized count estimation protocol and clustering. As discussed earlier, SpotMe [38] uses an algorithm based on Randomized Response [47] to estimate the number of people in geographic locations. We have evaluated this kind of solutions, specifically, SpotMe [38], in Section 5.3.

Private Location Data Publishing. Machanavjjhala et al. [30] use synthetic data generation techniques to publish commuting patterns in a differentially private way, while Acs and Castelluccia [2] describe a differentially private scheme to release the spatio-temporal density of Paris regions using records provided by a telco operator. To et al. [45] focus on releasing location entropy for ROIs under differential privacy guarantees: they study the bounds of location entropy and show that ϵ-differential privacy requires an excessive amount of noise, so they use weaker notions achieving better utility. Besides specific location-oriented private publishing, differential privacy has been proposed as a solution for releasing generic time-series of aggregate statistics. Examples are the various differentially private counting mechanisms by Chan et al. [8], or Fan et al.'s adaptive system [18] that uses a combination of filtering and sampling to increase the utility of differentially private aggregates. Rastogi and Nath [39] use an algorithm based on Discrete Fourier Transform to privately release aggregate time-series, while Shi et al. [40] combine encryption with data randomization to achieve differential privacy for time-series data. We have evaluated the privacy provided by this approach in Section 5.2, using the schemes in [8, 39].

Quantifying Location Privacy. Previous work on privacy quantification has studied the privacy loss incurred when disclosing obfuscated traces of individual users, e.g., when using location-based services. The main work in this area is the quantification framework by Shokri et al. [41, 42], which considers a strategic adversary that has prior information about users' mobility patterns, knows the location privacy-protection mechanism they use, and deploys inference attacks based on this information and the observation of the obfuscated traces.

This framework is conceived for evaluating privacy-preserving mechanisms applied to individuals' traces, therefore, the techniques used in their work are not applicable in the context of location privacy-preserving mechanisms based on aggregation. Nonetheless, if we were to compare our framework to Shokri et al.'s, we would observe that it does not only differ in the modeling of the adversary's prior knowledge, observation, and goal, but it is also driven by the definition of new metrics to model the adversary's error in this scenario. Moreover, we introduce new inference attacks tailored to the aggregate scenario and evaluate the impact on privacy of: (i) priors of different nature – specifically, both assignment and probabilistic, while only probabilistic are considered in [41, 42], (ii) priors based on more or less complete information, and (iii) sparsity of the location data that should be protected.

7 Conclusion

Publishing aggregate location information is often considered a privacy-friendly strategy to support mobility analytics applications, especially if the aggregation itself is performed in a privacy-preserving way [24, 36, 37] (i.e., without the need for trusted aggregators), and/or Differential Privacy (DP) is used to perturb aggregates [8, 14, 38, 39]. However, as opposed to privacy-preserving mechanisms for single users' traces [41, 42], there has been very little work on understanding the privacy threat that releasing aggregate location time-series poses on individuals whose locations are part of such aggregates.

This paper presented a first-of-its-kind analysis of aggregate location privacy. We introduced appropriate metrics to reason about privacy in the presence of an adversary aiming to localize and/or profile individual users, and proposed strategies to model the adversary's prior knowledge as well as to exploit aggregate information to perform inference attacks.

We used two real-world mobility datasets with different mobility characteristics to evaluate both the case in which raw aggregates are released, and when aggregates are perturbed to achieve Differential Privacy (DP) guarantees. Our experiments show that aggregates do help the adversary uncover mobility patterns and localize users, and that DP only improves privacy when adding so much noise that the utility of the time-series is destroyed.

We believe that our work will encourage further research on inference attacks as well as on the adversary's capability to obtain useful priors, also aiming to gain a better understanding of their dependence. Moreover, our results motivate future work towards the design of new differential privacy techniques that take into account temporal as well as spatial correlations, such as those discussed in [3, 6] which may provide a promising direction. Overall, we highlight the need for novel defense mechanisms that can offer better privacy guarantees to individuals whose location data is part of aggregate time-series releases, including in the context of "privacy-friendly" applications recently announced by Google [15] and Apple [23].

Acknowledgments. We wish to thank Mirco Musolesi and Gordon Ross for useful feedback and comments, as well as Rinku Dewri for shepherding the paper. This research is partially supported by a Xerox University Affairs Committee grant on "Secure Collaborative Analytics."

References

[1] Waze. https://www.waze.com, 2016.

[2] G. Acs and C. Castelluccia. A case study: privacy preserving release of spatio-temporal density in paris. In *KDD*, 2014.

[3] M. E. Andrés, N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi. Geo-indistinguishability: Differential privacy for location-based systems. In *CCS*, 2013.

[4] S. Bocconi, A. Bozzon, A. Psyllidis, C. Titos Bolivar, and G.-J. Houben. Social glass: A platform for urban analytics and decision-making through heterogeneous social data. In *WWW*, 2015.

[5] J. W. Brown, O. Ohrimenko, and R. Tamassia. Haze: privacy-preserving real-time traffic statistics. In *SIGSPATIAL*, 2013.

[6] Y. Cao, M. Yoshikawa, Y. Xiao, and L. Xiong. Quantifying differential privacy under temporal correlations. In *ICDE*, 2017.

[7] I. Ceapa, C. Smith, and L. Capra. Avoiding the crowds: understanding tube station congestion patterns from trip data. In *International Workshop on Urban Computing*, 2012.

[8] T.-H. H. Chan, E. Shi, and D. Song. Private and continual release of statistics. *ACM TISSEC*, 14(3), 2011.

[9] R. Chen, H. Li, A. Qin, S. P. Kasiviswanathan, and H. Jin. Private spatial data aggregation in the local setting. In *ICDE*, 2016.

[10] R. Chen, A. Reznichenko, P. Francis, and J. Gehrke. Towards Statistical Queries over Distributed Private User Data. In *NSDI*, volume 12, 2012.

[11] Y.-A. De Montjoye, C. A. Hidalgo, M. Verleysen, and V. D. Blondel. Unique in the crowd: The privacy bounds of human mobility. *Scientific reports*, 2013.

[12] Y. De Mulder, G. Danezis, L. Batina, and B. Preneel. Identification via location-profiling in GSM networks. In *WPES*, 2008.

[13] C. Dwork. Differential privacy: A survey of results. In *TAMC*, 2008.

[14] C. Dwork, M. Naor, T. Pitassi, and G. N. Rothblum. Differential privacy under continual observation. In *STOC*, 2010.

[15] A. Eland. Tackling urban mobility with technology. https://europe.googleblog.com/2015/11/tackling-urban-mobility-with-technology.html, 2015.

[16] D. M. Endres and J. E. Schindelin. A new metric for probability distributions. *IEEE Transactions on Information theory*, 2003.

[17] Ú. Erlingsson, V. Pihur, and A. Korolova. RAPPOR: Randomized Aggregatable Privacy-Preserving Ordinal Response. In *CCS*, 2014.

[18] L. Fan and L. Xiong. Real-time aggregate monitoring with differential privacy. In *CIKM*, 2012.

[19] G. Ghinita. Privacy for location-based services. *Synthesis Lectures on Information Security, Privacy, & Trust*, 4(1), 2013.

[20] P. Golle and K. Partridge. On the anonymity of home/work location pairs. In *Pervasive Computing*, 2009.

[21] S.-S. Ho and S. Ruan. Differential privacy for location pattern mining. In *Workshop on Security and Privacy in GIS and LBS*, 2011.

[22] E. J. Horvitz, J. Apacible, R. Sarin, and L. Liao. Prediction, expectation, and surprise: Methods, designs, and study of a deployed traffic forecasting service. *arXiv preprint arXiv:1207.1352*, 2012.

[23] J. Kaneps. Apple's 'differential privacy' is about collecting your data—but not your data. https://www.wired.com/2016/06/apples-differential-privacy-collecting-data/, 2016.

[24] C. Kopp, M. Mock, and M. May. Privacy-preserving distributed monitoring of visit quantities. In *SIGSPATIAL*, 2012.

[25] J. Krumm. Inference attacks on location tracks. In *Pervasive Computing*, 2007.

[26] J. Krumm. A survey of computational location privacy. *Personal and Ubiquitous Computing*, 13(6), 2009.

[27] S. Kullback and R. A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1), 1951.

[28] N. Lathia, C. Smith, J. Froehlich, and L. Capra. Individuals among commuters: Building personalised transport information services from fare collection systems. *Pervasive and Mobile Computing*, 9(5), 2013.

[29] J. Lin. Divergence measures based on the shannon entropy. *IEEE Transactions on Information theory*, 1991.

[30] A. Machanavajjhala, D. Kifer, J. Abowd, J. Gehrke, and L. Vilhuber. Privacy: Theory meets practice on the map. In *ICDE*, 2008.

[31] L. Melis, G. Danezis, and E. De Cristofaro. Efficient private statistics with succinct sketches. In *NDSS*, 2016.

[32] B. Pan, Y. Zheng, D. Wilkie, and C. Shahabi. Crowd sensing of traffic anomalies based on human mobility and social media. In *SIGSPATIAL*, 2013.

[33] V. Pandurangan. On Taxis and Rainbows. https://tech.vijayp.ca/of-taxis-and-rainbows-f6bc289679a1, 2014.

[34] M. Piorkowski, N. Sarafijanovic-Djukic, and M. Grossglauser. CRAWDAD Dataset. http://crawdad.org/epfl/mobility/20090224, 2009.

[35] I. Polakis, G. Argyros, T. Petsios, S. Sivakorn, and A. D. Keromytis. Where's wally?: Precise user discovery attacks in location proximity services. In *CCS*, 2015.

[36] R. A. Popa, A. J. Blumberg, H. Balakrishnan, and F. H. Li. Privacy and accountability for location-based aggregate statistics. In *CCS*, 2011.

[37] A. Pyrgelis, E. De Cristofaro, and G. Ross. Privacy-Friendly Mobility Analytics using Aggregate Location Data. In *SIGSPATIAL*, 2016.

[38] D. Quercia, I. Leontiadis, L. McNamara, C. Mascolo, and J. Crowcroft. Spotme if you can: Randomized responses for location obfuscation on mobile phones. In *ICDCS*, 2011.

[39] V. Rastogi and S. Nath. Differentially private aggregation of distributed time-series with transformation and encryption. In *SIGMOD*, 2010.

[40] E. Shi, H. Chan, E. Rieffel, R. Chow, and D. Song. Privacy-preserving aggregation of time-series data. In *NDSS*, 2011.

[41] R. Shokri, G. Theodorakopoulos, G. Danezis, J.-P. Hubaux, and J.-Y. Le Boudec. Quantifying location privacy: the case of sporadic location exposure. In *PETS*, 2011.

[42] R. Shokri, G. Theodorakopoulos, J.-Y. Le Boudec, and J.-P. Hubaux. Quantifying location privacy. In *IEEE Symposium on Security and Privacy*, 2011.

[43] R. Shokri, C. Troncoso, C. Diaz, J. Freudiger, and J.-P. Hubaux. Unraveling an old cloak: k-anonymity for location privacy. In *WPES*, 2010.

[44] R. Silva, S. M. Kang, and E. M. Airoldi. Predicting traffic volumes and estimating the effects of shocks in massive transportation systems. *Proceedings of the National Academy of Sciences*, 112(18), 2015.

[45] H. To, K. Nguyen, and C. Shahabi. Differentially Private Publication of Location Entropy. In *SIGSPATIAL*, 2016.

[46] G. Wang, B. Wang, T. Wang, A. Nika, H. Zheng, and B. Y. Zhao. Whispers in the dark: analysis of an anonymous social network. In *IMC*, 2014.

[47] S. L. Warner. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association*, 60(309), 1965.

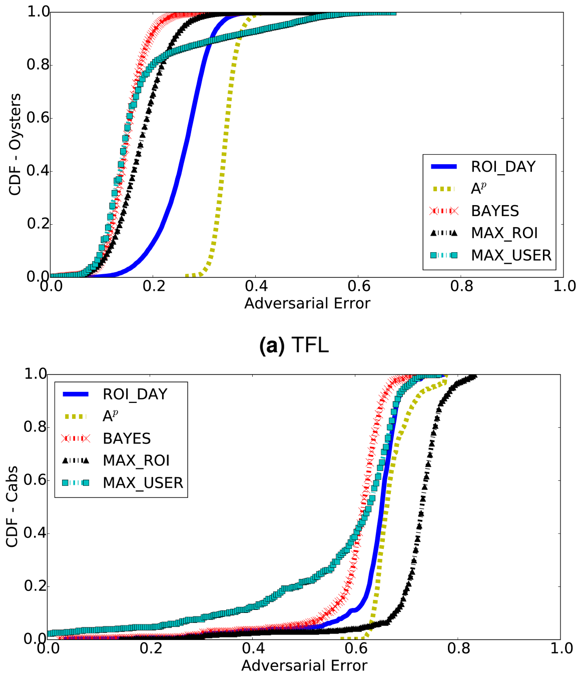
[48] A. Waseda and R. Nojima. Analyzing randomized response mechanisms under differential privacy. In *ICIS*, 2016.

[49] M. Wernke, P. Skvortsov, F. Dürr, and K. Rothermel. A classification of location privacy attacks and approaches. *Personal and Ubiquitous Computing*, 18(1), 2014.

[50] F. Xu, Z. Tu, Y. Li, P. Zhang, X. Fu, and D. Jin. Trajectory Recovery From Ash: User Privacy Is NOT Preserved in Aggregated Mobility Data. In *WWW*, 2017.

[51] M. Xue, C. L. Ballard, K. Liu, C. L. Nemelka, Y. Wu, K. W. Ross, and H. Qian. You can yak but you can't hide: Localizing anonymous social network users. In *IMC*, 2016.

[52] H. Zang and J. Bolot. Anonymization of location data does not work: A large-scale measurement study. In *MobiCom*, 2011.



(a) TFL

(b) SFC

Fig. 10. Adv's Profiling Error - ROI_DAY Prior.

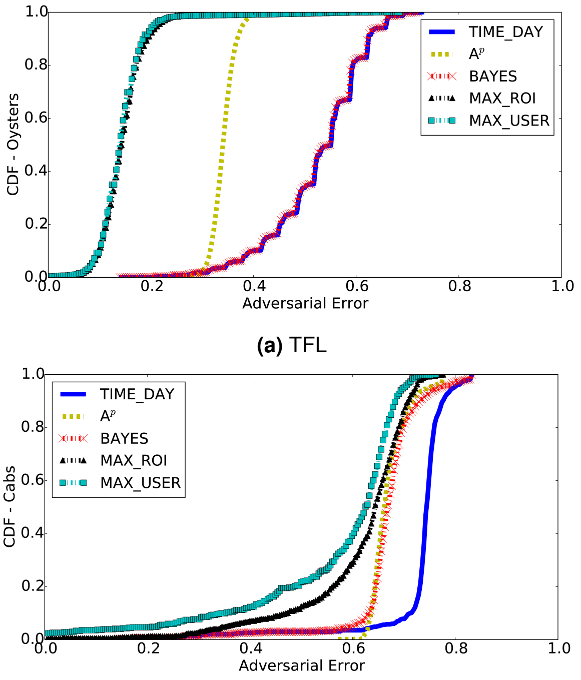
A Additional Experiments

We now report additional details about experimental results on Adv's inference tasks based on other approaches of prior knowledge.

A.1 User Profiling

A.1.1 Probabilistic Priors

ROI_DAY. Recall that, with ROI_DAY, Adv knows for the users, a profile for each hour of *any* day (e.g., user's frequent locations at 4pm). For TFL (Fig. 10a), we observe that this is a more instructive prior than commuters' frequent ROIs (FREQ_ROI), with an average prior error of 0.25. Moreover, we note that BAYES and MAX_ROI inferences remarkably improve Adv's profiling accomplishment for all users, yielding 0.41 and 0.31 average privacy loss, respectively. MAX_USER improves Adv's predictions for $\sim 80\%$ of the users and achieves 0.37 average loss in privacy. Similarly for SFC (Fig. 10b), ROI_DAY (0.63 avg. error) is a more revealing prior knowledge than cabs' frequent ROIs (FREQ_ROI – 0.65) for Adv. BAYES and MAX_USER give advantage to Adv in profiling users (resulting in 0.06 and 0.13 privacy loss, resp.) while MAX_ROI does not, once again, indicating the *bias* of this strategy towards less active cabs.



(a) TFL

(b) SFC

Fig. 11. Adv's Profiling Error - TIME_DAY Prior.

TIME_DAY. Fig. 11a plots the CDF of Adv's total error in profiling TFL commuters, with the TIME_DAY prior knowledge, i.e., a time profile indicating which hours of day a user is likely to report ROIs. We observe that Adv's performance is worse (0.52 mean error) compared to priors containing location information (i.e., FREQ_ROI, ROI_DAY or ROI_DAY_WEEK). This is expected, since TIME_DAY prior contains only time information for the users, and it is a *uniform* distribution over all ROIs, for the time slots that they are likely to be inside the transportation system. Indeed, Fig. 11a shows that profiling only with the aggregate profile (A^P), Adv achieves smaller error (0.34). Among the inference strategies, we note that BAYES negligibly improves Adv's error in profiling users due to the very small prior probabilities. MAX_ROI and MAX_USER attacks exhibit similar performance, as in both cases the users who are more likely to be *inside* the system, are selected to cover the aggregate values (in this case both strategies pick users based on their total number of ROIs). With these strategies, Adv's performance increases significantly and there is notable privacy loss for the users (0.72 on average).

On the SFC dataset (Fig. 11b), we observe that when Adv knows the cabs' most frequent time slots of day (TIME_DAY), she obtains a worse prior (0.73 mean error) compared to cabs' most frequent ROIs (FREQ_ROI – 0.65) or cabs' most frequent ROIs with time and day semantics (ROI_DAY_WEEK – 0.61). Unlike TFL, Bayesian updating yields a 0.1 privacy loss (as with fewer ROIs in the SFC data, BAYES affects significantly the posterior probabilities), while the greedy strategies perform even better. More precisely, with MAX_ROI the mean privacy loss is 0.16 and with MAX_USER 0.22.

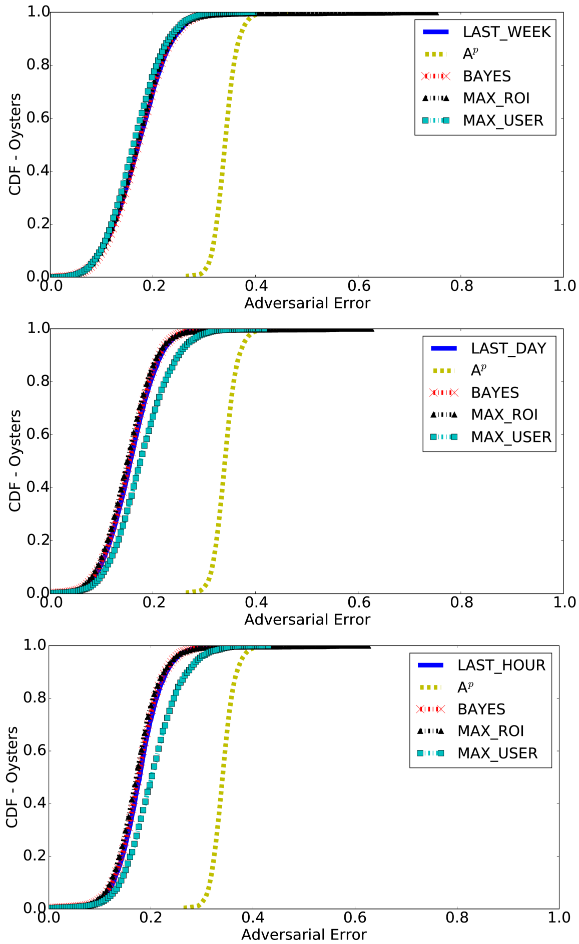


Fig. 12. Adv's Profiling Error - LAST_WEEK, LAST_DAY and LAST_HOUR Priors - TFL.

A.1.2 Assignment Priors

Due to space limitations, the figures that show the results of user profiling based on assignment priors (i.e., LAST_WEEK, LAST_DAY and LAST_HOUR) are presented here. Figures 12 and 13 plot the results that are discussed in Section 4.2.2.

A.2 User Localization

A.2.1 Probabilistic Priors

TIME_DAY_WEEK. Fig. 16 displays Adv's error when localizing users with the TIME_DAY_WEEK prior. For TFL (Fig. 16a), we observe that ALL results in a very large error (0.99 on average). This is not surprising since

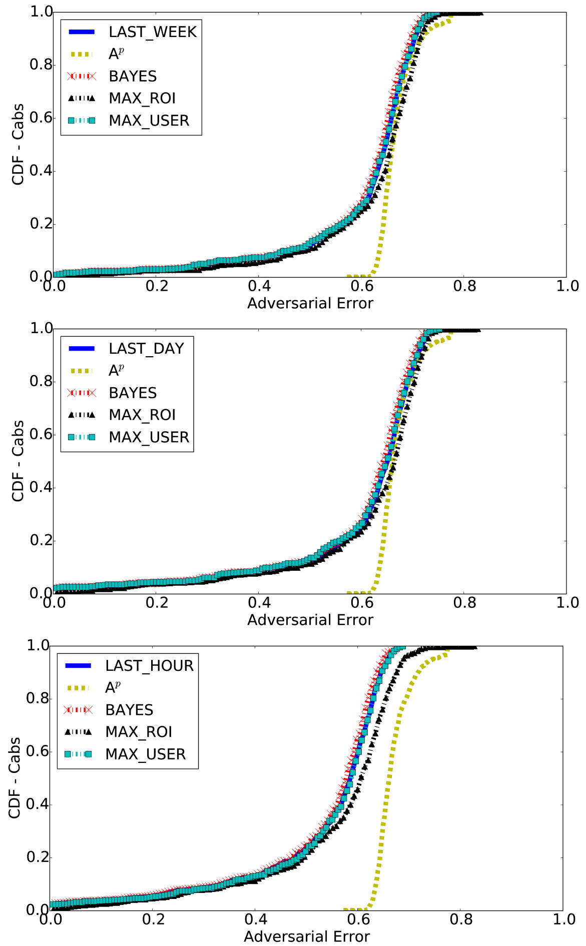


Fig. 13. Adv's Profiling Error - LAST_WEEK, LAST_DAY and LAST_HOUR Priors - SFC.

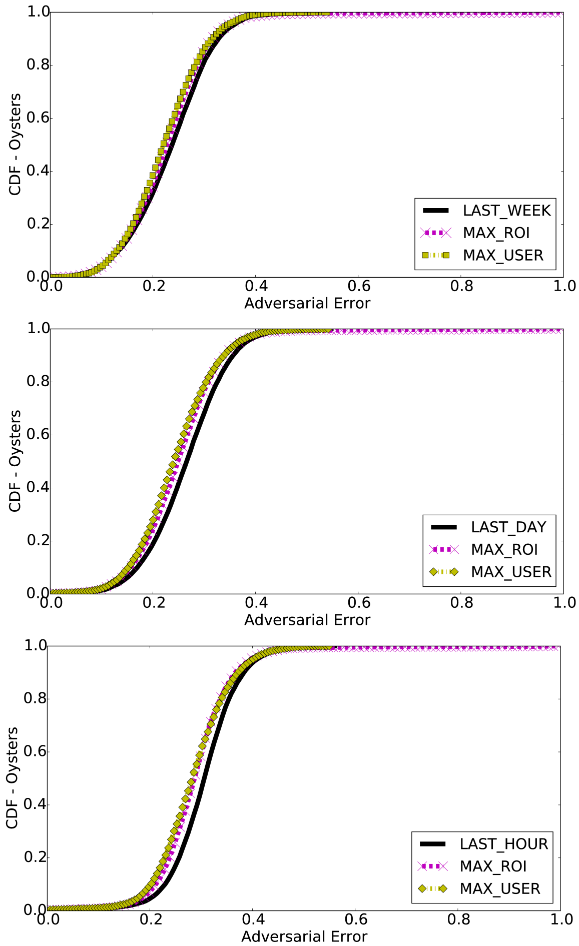


Fig. 14. Adv's Localization Error - LAST_WEEK, LAST_DAY and LAST_HOUR Priors - TFL.

TIME_DAY_WEEK is a uniform distribution over ROIs, for the time slots that users are likely to be *in* the transportation system. ALL after BAYES achieves negligible privacy loss (0.03), while we observe no adversarial advantage between POP and POP after BAYES due to the very small prior probabilities. Furthermore, both MAX_USER and MAX_ROI improve remarkably Adv's performance compared to ALL and they yield 0.79 and 0.77 average privacy loss respectively. We note that MAX_ROI achieves error larger than 0.25 for 20% of the users while MAX_USER yields error larger than 0.25 for only 5% of the users, i.e., those users that report the most locations and always get assigned to locations to consume the aggregates.

For the SFC data (Fig. 16b) we observe that ALL yields 0.84 average error, while ALL after BAYES results in small privacy loss (0.04). Once again, POP is the worst inference strategy as the small probabilities of the prior do not exceed the threshold δ and cabs are predicted to be outside the network. Moreover, unlike the case of ROI_DAY_WEEK, MAX_ROI now improves Adv's performance for localizing all the cabs, yielding 0.09 privacy loss. MAX_USER achieves a similar mean loss in privacy, however, Adv's knowledge is only improved for 60% of the cabs compared to the baseline ALL. Once again, we remark how localization strategies result to different amount of privacy leakage on sparse (TFL) and dense (SFC) datasets.

A.2.2 Assignment Priors

We evaluate Adv's performance against user localization, i.e., predicting users' future locations with a seasonal part of their ground truth as prior knowledge. In particular, we experiment with LAST_WEEK, LAST_DAY and LAST_HOUR and focus on the MAX_ROI and MAX_USER inference attacks. Adv's *baseline* prediction is to *replicate* the prior, as described in Section 3.2.2. Figs 14–15 plot the CDF of Adv's error in localizing commuters and cabs, over the inference week.

LAST_WEEK. Adv's average error localizing tube passengers with LAST_WEEK is 0.24. Both MAX_ROI and MAX_USER inference strategies vaguely improve her performance and yield small privacy loss (0.02). This indicates

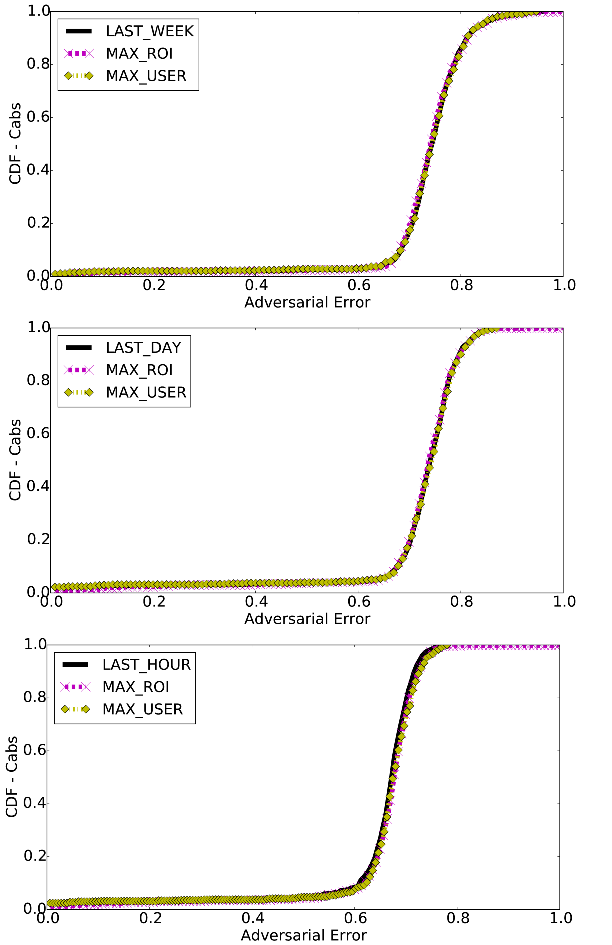
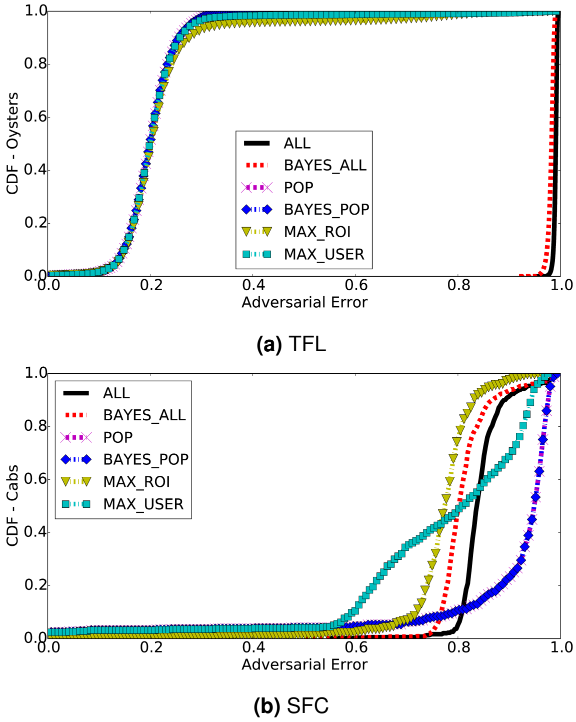


Fig. 15. Adv's Localization Error - LAST_WEEK, LAST_DAY and LAST_HOUR Priors - SFC.

that users reporting lots of ROIs and ROIs themselves show regularity within weeks. Furthermore, MAX_USER attack is more consistent in improving Adv's localization success than MAX_ROI, which increases Adv's error (compared to the prior) for 5% of the users. For SFC cabs, we observe that Adv's avg. localization error is 0.73, while both MAX_ROI and MAX_USER do not reduce it further. Unlike TFL, we observe that the aggregates do not give any advantage to Adv in localizing taxis and there is no privacy loss.

LAST_DAY. With LAST_DAY, Adv's mean error in predicting TFL passengers' locations during the inference week is 0.27, thus, this prior is less revealing than LAST_WEEK (0.24). This indicates that commuters show stronger weekly seasonality in their journeys. MAX_ROI and MAX_USER achieve very small privacy loss (0.01 and 0.02 resp.), thus, aggregate time-series enhance insignificantly Adv's inference goal. MAX_USER constantly reduces Adv's error over the prior, while MAX_ROI increases it for a small percentage of users (5%). For SFC, Adv's localization error with LAST_DAY is 0.71 indicating that cabs are a bit more likely to appear in the ROIs of last day rather than those of last week (0.73 error). Once again, the greedy inference strategies do not help Adv improve her predictions and there is negligible privacy loss.



(a) TFL

(b) SFC

Fig. 16. Adv's Localization Error - TIME_DAY_WEEK Prior.

LAST_HOUR. Finally, we plot Adv's error while localizing users with the LAST_HOUR prior. For TFL, we observe that her error is now larger (0.31) compared to the two previous cases (LAST_WEEK, LAST_DAY) indicating that, in general, commuters do not show up in the ROIs of their last hour. The knowledge of the aggregate time-series enables Adv to improve her localization performance insignificantly and the greedy strategies MAX_ROI and MAX_USER yield negligible amount of privacy loss (0.01 and 0.02 resp.). When localizing SFC cabs with LAST_HOUR, Adv's mean error is 0.64. Thus, as this assignment prior helps Adv localize cabs better than LAST_DAY (0.71) or LAST_WEEK (0.73) and unlike tube commuters, taxis are more likely to appear in the locations they have recently reported. Both inference strategies lead to very small privacy loss.