Knock Knock, Who's There?
Membership Inference on Aggregate Location Data

Apostolos Pyrgelis
University College London
apostolos.pyrgelis.14@ucl.ac.uk

Carmela Troncoso
IMDEA Software Institute
carmela.troncoso@imdea.org

Emiliano De Cristofaro
University College London
e.decristofaro@ucl.ac.uk

***Abstract*—Aggregate location data is often used to support smart services and applications, e.g., generating live traffic maps or predicting visits to businesses. In this paper, we present the first study on the feasibility of membership inference attacks on aggregate location time-series. We introduce a game-based definition of the adversarial task, and cast it as a classification problem where machine learning can be used to distinguish whether or not a target user is part of the aggregates.**

We empirically evaluate the power of these attacks on both raw and differentially private aggregates using two mobility datasets. We find that membership inference is a serious privacy threat, and show how its effectiveness depends on the adversary's prior knowledge, the characteristics of the underlying location data, as well as the number of users and the timeframe on which aggregation is performed. Although differentially private mechanisms can indeed reduce the extent of the attacks, they also yield a significant loss in utility. Moreover, a strategic adversary mimicking the behavior of the defense mechanism can greatly limit the protection they provide. Overall, our work presents a novel methodology geared to evaluate membership inference on aggregate location data in real-world settings and can be used by providers to assess the quality of privacy protection before data release or by regulators to detect violations.

I. Introduction

The ability to model the context in which users and applications operate enables the development of intelligent applications and pervasive personalized services. Naturally, location information plays a crucial role in shaping such context, motivating the continuous collection of users' location data by applications and service providers. In some cases, entities may be interested in only collecting or releasing *aggregate location statistics*, which, for instance, can be used to calculate the average speed along a road and generate live traffic maps [3, 34], or to estimate the number of people at a restaurant and predict availability and waiting times [21]. Apple also lets iOS as well as third-party app developers collect differentially private aggregate statistics about emojis, deep links, as well as locations, via dedicated APIs [31]. Moreover, aggregate location information is relied upon by companies like factual.com to offer statistics to advertisers, or like Telefonica, which provides consultancy services around footfall measures calculated from network events [30].

Aggregation is often considered as a way to hinder the exposure of individual users' data [21]; however, access to, or release of, aggregate location statistics might ultimately violate privacy of the individuals that are part of the aggregates [22, 36]. In this paper, we focus on *membership inference attacks*, whereby an adversary attempts to determine whether or not location data of a target user is part of the aggregates.

Motivation. The ability of an adversary to ascertain the presence of an individual in aggregate location time-series constitutes an obvious privacy threat if the aggregates relate to a group of users that share a sensitive characteristic. For instance, learning that an individual is part of a dataset aggregating movements of Alzheimer's patients implies learning that she suffers from the disease. Similarly, inferring that statistics collected over a sensitive timeframe or sensitive locations include a particular user also harm the individual's privacy.

Recent work [22] also shows that an adversary with some prior knowledge about a user's mobility profile can exploit aggregate information to improve this knowledge, or even localize her. Also, users' "trajectories" can in some cases be extracted from aggregate mobility data, even without prior knowledge [36]. However, in order to mount these attacks, the adversary needs to know that the user is part of the aggregate dataset, which further motivates our research objectives.

Membership inference can also be leveraged by providers to evaluate the quality of privacy protection on the aggregates *before* releasing them, and by regulators, to support enforcement of individual's rights (e.g., the right to be forgotten) or detect violations. For instance, if a service provider is not allowed to release location data, or make it available to third-parties even in aggregate form, one can use membership inference attacks to verify possible misuse of the data.

Approach & Results. In this paper, we present the first formalization of membership inference in the context of location data. We model the problem as a game in which an adversary aims at distinguishing location aggregates that include data of a target user from those that do not. We instantiate the distinguishing task using a machine learning classifier trained on the prior knowledge of the adversary (e.g., past users' locations, aggregates of groups including and excluding the target user), and use it to infer the target's membership in unseen aggregate statistics.

Network and Distributed Systems Security (NDSS) Symposium 2018
18-21 February 2018, San Diego, CA, USA
ISBN 1-1891562-49-5
http://dx.doi.org/10.14722/ndss.2018.23183
www.ndss-symposium.org

We evaluate our approach on two mobility datasets with different characteristics, and find that releasing raw aggregates poses a significant privacy threat. In particular, our results show that membership inference is very successful when the adversary knows the locations of a small subset of users including her target – the classifier achieves up to 0.83 Area Under Curve (AUC) with 100 users per aggregation group – or when she has prior information for user groups on which she attempts to infer membership (up to 1.0 AUC even with 9,500 users per group). In weaker adversarial knowledge settings, membership inference is less effective but still yields non-negligible privacy leakage. Overall, we find that the number of users as well as the timeframe used to compute the aggregation have a profound effect on the accuracy of the attacks. Interestingly, certain characteristics of the data, like regularity and sparseness, also affect the power of the membership inference adversarial task.

We also study membership inference on statistics protected using defense mechanisms based on differential privacy. We find that they are generally effective at preventing inference, although at the cost of a non-negligible reduction in utility. Moreover, we show that a strategic adversary mimicking the behavior of the mechanisms – i.e., training the classifier on noisy aggregates – can reduce their protection (up to 83%).

Contributions. In summary, this paper makes the following contributions: (i) we introduce a generic methodology to study membership privacy in location aggregates that formalizes membership inference on aggregate location data as a distinguishability game and instantiates the distinguishing task with a machine learning classifier; (ii) we deploy our methods to quantify privacy leakage on raw aggregates, using two real-world mobility datasets; and (iii) we illustrate how our techniques can be used to study the effectiveness of defense mechanisms aimed at preventing these attacks.

Paper Organization. The rest of this paper is organized as follows. The next section reviews related work. Then, we formalize the problem of membership inference on aggregate location time-series in Section III and present the methodology used to evaluate it in Section IV. In Section V, we introduce our experimental setup and, in Section VI, present the results of our experiments on raw aggregates. After evaluating differentially private defense mechanisms in Section VII, the paper concludes in Section VIII.

II. Related Work

In this section, we review previous work on membership inference, differentially private release of location data, as well as location privacy.

Membership inference attacks. Such attacks aim to determine the presence of target individuals within a dataset. This is relevant in many settings, e.g., in the context of *genomic research*, where data inherently tied to sensitive information, such as health stats or physical traits, is commonly released in aggregate form for Genome Wide Association Studies (GWAS) [35]. Homer et al. [15] show that one can learn whether a target individual was part of a case-study group associated to a certain disease by comparing the target's profile against the aggregates of the case study and those of a reference population obtained from public sources. This attack has then been extended by Wang et al. [33] to use correlations within the human genome, reducing the need for prior knowledge about the target. Also, Backes et al. [2] show that membership inference can be mounted against individuals contributing their microRNA expressions to scientific studies.

Another line of work focuses on membership inference in *machine learning* models. Shokri et al. [26] show that such models may leak information about data records on which they were trained. Hitaj et al. [14] present active inference attacks on deep neural networks in collaborative settings, while Hayes et al. [13] focus on privacy leakage from generative models in Machine Learning as a Service applications. Moreover, Buscher et al. [4] recently evaluate membership inference in the context of data aggregation in *smart metering*, studying how many household readings need to be aggregated in order to protect privacy of individual profiles in a smart grid.

Overall, our research differs from these works in that we focus on membership inference over aggregate location time-series, which present different characteristics and challenges than the other domains. Despite the importance of location data in terms of its availability, frequency of collection, and the amount of sensitive information it carries [25], this problem, to the best of our knowledge, has not been examined before.

Differentially private mechanisms. Differential privacy (DP) [8] can be used to mitigate membership inference, as its indistinguishability-based definition guarantees that the presence or the absence of an individual does not significantly affect the output of the data release. Li et al. [18] introduce a framework geared to formalize the notion of Positive vs Negative Membership Privacy, considering an adversary parameterized by her prior knowledge. However, to the best of our knowledge, no specific technique has been presented to instantiate their framework in our setting. Common mechanisms to achieve DP include using noise from the Laplacian [8] or the Gaussian distribution [9] (see Section VII).

Specific to the context of spatio-temporal data are the techniques proposed by Machanavajjhala et al. [19], who use synthetic data generation to release differentially private mobility patterns of commuters in Minnesota. Also, Rastogi and Nath [23] propose an algorithm based on Discrete Fourier Transform to privately release aggregate time-series, while Acs and Castelluccia [1] improve on [23] and present a differentially private scheme tailored to the spatio-temporal density of Paris. Finally, To et al. [32] release the entropy of certain locations with DP guarantees, and show how to achieve better utility although with weaker privacy notions.

Location privacy. Previous location privacy research taking into account traces or profiles of *single users* [7, 12, 16, 24, 28, 37] does not apply to our problem, which focuses on aggregate location statistics. Closer to our work is our own PETS'17 paper [22], which shows that aggregate location time-series can be used by an adversary to improve her prior knowledge about users' location profiles. Also, Xu et al. [36] present an attack that exploits the uniqueness and the regularity of human mobility, and extracts location trajectories from aggregate mobility data. As opposed to these efforts, which attempt to learn data about individuals (e.g., mobility profiles, trajectories) from the aggregates, we focus on inferring their membership to datasets, which, to the best of our knowledge, has not been studied before.

Symbol	Description
Adv, Ch	Adversary, Challenger
$\mathcal{P}$	Adv's prior knowledge
U	Set of mobile users
S	Set of locations (ROIs)
T	Time period considered
T_O	Observation period
T_I	Inference period
L_u	User u's location time-series, where $L_u[s,t]$ = 1 if u is in s at time t, 0 otherwise
$\mathcal{L}$	Location time-series of all users in U
$\Upsilon \subset_\$ U$	Random subset $\Upsilon \subset U$
A_X	Aggregate location time-series of users in $X \subset U$ where $A_X[s,t] = \sum_{j \in X} L_j[s,t]$
m	Variable representing size of aggregation group

TABLE I: Notation.

III. Defining Membership Inference on Aggregate Locations

In this section, we formalize the problem of membership inference on aggregate location time-series. We consider the case where one or more entities periodically release the number of users in some Regions Of Interest (ROIs), within a given time interval (e.g., 99 taxis in Union Square on Fri, Aug 11th between 9–10am). By relying on this data, as well as some prior knowledge, an adversary tries to infer if a target individual contributed to the aggregates, i.e., whether or not she is a *member* of the group yielding the released aggregates.

A. Notation

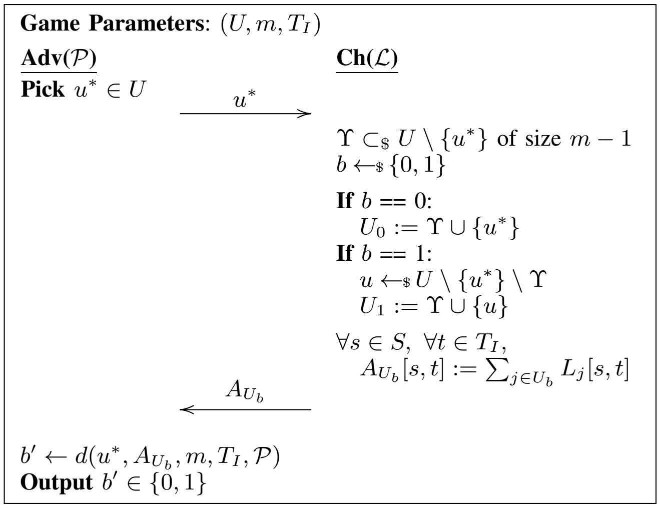
The notation used throughout the paper is summarized in Table I. We denote the set of all users as $U = \{u_1, u_2, \cdots, u_{|U|}\}$, and the set of regions of interest as $S = \{s_1, s_2, \cdots, s_{|S|}\}$. We also use $T = \{t_1, t_2, \cdots, t_{|T|}\}$ to denote the set of time intervals on which aggregate locations are collected, although, without loss of generality, the problem can be extended to infinite intervals. We model the location of a user $u \in U$ over time as a binary matrix L_u of size $|S| \times |T|$, where $L_u[s,t]$ is 1 if u is in location $s \in S$, at time $t \in T$, and 0 otherwise. That is, L_u contains the location time-series of u, while those of *all* users are stored in a matrix $\mathcal{L}$, which is of size $|U| \times |S| \times |T|$.

Also, A_X denotes the *aggregate* location time-series over the users in $X \subset U$. A_X is modeled as a matrix of size $|S| \times |T|$, where each element $A_X[s,t]$ represents the number of users in X that are in ROI s at time t.

Finally, we denote the *prior knowledge* of an adversary (Adv) about users as $\mathcal{P}$, which is built during an *observation period*, denoted as $T_O \subset T$ (see Section IV-A). The prior knowledge is used by Adv to perform membership inference during the *inference* period, denoted as $T_I \subset T$, for which aggregates are available.

B. Membership Inference as a Distinguishability Game

We model membership inference by means of a distinguishability game (DG), played by the adversary Adv and a challenger Ch, which generates the location aggregates over various user groups. The former, having some prior knowledge about the users ($\mathcal{P}$), tries to infer whether data of a particular user (u^*) is included in the aggregates. Naturally, Adv could be interested in multiple target users, however, to ease presentation, we describe the case of a single target user.



```
Game Parameters: $(U, m, T_I)$
Adv($\mathcal{P}$)                         Ch($\mathcal{L}$)
Pick $u^* \in U$        $u^*$ →
                                   $\Upsilon \subset_\$ U \setminus \{u^*\}$ of size $m-1$
                                   $b \leftarrow_\$ \{0,1\}$
                                   If $b == 0$:
                                       $U_0 := \Upsilon \cup \{u^*\}$
                                   If $b == 1$:
                                       $u \leftarrow_\$ U \setminus \{u^*\} \setminus \Upsilon$
                                       $U_1 := \Upsilon \cup \{u\}$
                                   $\forall s \in S,\ \forall t \in T_I$,
                                       $A_{U_b}[s,t] := \sum_{j \in U_b} L_j[s,t]$
                        ← $A_{U_b}$
$b' \leftarrow d(u^*, A_{U_b}, m, T_I, \mathcal{P})$
Output $b' \in \{0,1\}$
```

Fig. 1: Distinguishability Game (DG) between adversary Adv and challenger Ch, capturing membership inference over aggregate location time-series. The game is parameterized by the set of users (U), the aggregation group size (m) and the inference period (T_I).

The game is parameterized by the set of users U, the number of users included in the aggregation group (m), and the inference period T_I. Note that m and T_I inherently affect Adv's performance, as we discuss in our experimental evaluation (Section VI).

We present the game in Fig. 1. First, Adv selects the target user u^* and sends it to Ch. The latter randomly selects a subset $\Upsilon \subset U$ of size $m-1$, excluding u^*, and draws a random bit b. If $b = 0$, she aggregates the location matrices of all users in Υ along with that of u^*; whereas, if $b = 1$, she selects another random user $u \neq u^*$ not in Υ and adds her data to the aggregates instead. The resulting matrix A_{U_b}, computed over all timeslots of T_I, is sent back to Adv, which attempts to guess b. Adv wins if $b' = b$, i.e., she successfully distinguishes whether u^* is part of the aggregates or not; naturally, her goal is to win the game, over multiple iterations, with probability higher than 1/2 (i.e., a random guess).

We model Adv's guess as a *distinguishing function*, d, with input $(u^*, A_{U_b}, m, T_I, \mathcal{P})$. How to instantiate the function is discussed in Section IV-B. Observe that the parameters of the DG game include the set of users U, but this information is *not* used in the distinguishing function. In other words, we only assume that Adv knows that u^* is in the universe of possible users, but not that she knows all users in U.

IV. Methodology

We now introduce our methodology to evaluate membership inference on aggregate location time-series, modeled by the DG game in Fig. 1. Specifically, we discuss ways to build Adv's prior knowledge ($\mathcal{P}$) during the observation period T_O, how to instantiate the distinguishing function (i.e., deciding the bit b'), and measure the performance of the inference.

A. Adversarial Knowledge

Our generic game-based definition of the adversarial goal enables the consideration of adversaries of variable strength, modeled by their prior knowledge, $\mathcal{P}$. We consider two possible priors, discussed next.

(1) Subset of Locations. We start with a setting in which Adv knows the real locations of a subset of users $Y \subset U$, *including* the target user (i.e., $u^* \in Y$), during the inference period T_I. Thus, in this case observation and inference periods coincide (i.e., $T_O = T_I$). We consider $|Y| = \alpha \cdot |U|$, where $\alpha \in [0, 1]$ models the percentage of users for which Adv knows their actual location. Formally, we define it as:

$$\mathcal{P}: \; L_u[s,t] \;\; \forall u \in Y \; \forall s \in S \; \forall t \in T_I \tag{1}$$

This prior knowledge represents the case of an adversary that has access to location information of some users at a point in time, e.g., a telecommunications service provider getting locations from cell towers, or a mobile app provider collecting location data. Using this information she attempts to infer membership of her target to an aggregate dataset published by another entity.

(2) Participation in Past Groups. We then consider an adversary that knows aggregates computed during an observation period T_O, disjoint from the inference period T_I (i.e., $T_O \cap T_I = \emptyset$) for β groups W_i of size m, which may or may not include u^*. For each group W_i, we assume that Adv knows: (i) the aggregates of the observation period, i.e., $A_{W_i}[s,t], \forall s \in S$ and $t \in T_O$, and (ii) u^*'s membership to the group. More formally:

$$\mathcal{P}: A_{W_i} \;\wedge\; \mathbb{1}_{W_i}(u^*) \; \forall \; i \in \{1, \cdots, \beta\} \tag{2}$$

where $\mathbb{1}_{W_i}(u^*)$ is the indicator function modeling the membership of the target user to the group W_i. In our experiments, we consider two different "flavors" of this prior:

- **(2a)** ***Same Groups as Released***, Adv knows the target user's participation in past groups which *are* also used to compute the aggregates released by Ch during the inference period;
- **(2b)** ***Different Groups than Released***, Adv knows the user's participation in past groups that *are not* used to compute aggregates released in the inference period.

Observe that (2a) simulates the case of continuous data release related to particular groups, where users are stable over time (e.g., statistics about a neighborhood), and with the adversary (e.g., a group member) having observed the participation of the target user in past aggregates of the same groups. Prior (2b) is less restrictive, as it only assumes that the adversary has some aggregates of groups in which the target was previously included, but does not require these groups to be fixed over time – e.g., if the target user moves to a new neighborhood and her data is mixed with other users, Adv attempts to infer membership using past information.

B. Distinguishing Function

Recall from Section III that, in the DG game (Fig. 1), the adversary tries to guess whether or not the target user is part of the aggregates using a distinguishing function, which we denoted as d. This function takes as input the target user u^*, the "challenge" A_{U_b}, parameters of the game m and T_I, and the prior knowledge $\mathcal{P}$.

We opt to instantiate d with a *supervised machine learning classifier*, trained using data included in the adversarial prior knowledge. Our intuition is that the adversary's distinguishing goal can be modeled as a binary classification task, i.e., categorizing observations into two classes corresponding to whether or not the data of target user u^* is part of the location aggregates under examination.

C. Privacy Metric

Given our game-based definition, we reason about privacy leakage in terms of the adversarial performance in distinguishing whether or not u^*'s data is included in the aggregates. In particular, we introduce a *privacy loss* metric, capturing Adv's advantage in winning the DG game over a random guess (assuming that the adversary plays the distinguishability game for a specific user multiple times), while relying on the Area Under the Curve (AUC) to measure Adv's performance.

AUC Score. For a series of instances of the game for u^*, we count the Adv's guesses b' regarding the presence of u^*'s data in the released aggregate location time-series as:

- True Positive (TP) when $b = 0$ and $b' = 0$;
- True Negative (TN) when $b = 1$ and $b' = 1$;
- False Positive (FP) when $b = 1$ and $b' = 0$;
- False Negative (FN) when $b = 0$ and $b' = 1$.

We then calculate True Positive and False Positive Rates, as TPR = TP/(TP+FN) and FPR = FP/(FP+TN), respectively. From these, we derive the Receiver Operating Characteristic (ROC) curve, which represents the TPR and FPR obtained at various discrimination classification thresholds, and compute the Area Under Curve (AUC). The AUC captures a classifier's overall performance in the distinguishability game.

Privacy Loss (PL). As mentioned, we measure the privacy loss of u^* as the adversary's *improvement* over a random guess baseline (AUC = 0.5). Formally, we define PL as:

$$\text{PL} = \begin{cases} \frac{\text{AUC}-0.5}{0.5} & \text{if } \; \text{AUC} > 0.5 \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Hence, PL is a value between 0 and 1 that captures the adversary's advantage over random guessing when distinguishing whether the target user's data is part of the aggregates.

V. Experiment Design

In this section, we present our experimental setup as well as the datasets used in our evaluation. (Results are given later in Sections VI and VII).

A. Datasets

We use two real-world datasets that capture different mobility characteristics, obtained, respectively, from the Transport for London (TFL) authority and the San Francisco Cab (SFC) network. Both datasets contain about one month of location data, and have been used often in ubiquitous computing [5, 29] and location privacy [22, 27] research. We choose these datasets primarily because of their different characteristics:

data from public transport (TFL) includes more users and is more "predictable" (due to commuting patterns) than the SFC dataset, which, on the other hand, is less sparse (i.e., it involves more data points per user per day).

Transport For London (TFL). The TFL dataset consists of trips made by passengers on the TFL network in March 2010 using the Oyster Card – an RFID pre-paid card. Each record in the data describes a unique trip and includes the (anonymized) oyster card id, start time, touch-in station id, end time, and touch-out station id. We discard trips from March 29–31 to have exactly four weeks of data, which contain 60M trips made by 4M unique Oyster cards, visiting 582 train/tube stations. We select the top 10K oyster ids per total number of trips, which account for about 6M trips. Considering oyster trips start/end stations as ROIs, the top 10K users report, on average, 728 ± 16 ROIs in total, out of which 20 ± 9 are unique. Setting the time granularity to *one hour*, the top 10K oysters are in the "system" for 115 ± 21 out of the 672 slots (28 days). We consider each Oyster card as a user u, and compute the matrix L_u setting $L_u[s,t]$ to 1 if the user u touched-in or out at station s, during time slot $t \in T$, and 0 otherwise. When a card does not report any station at a particular time slot, we assign it to a special ROI denoted as *null*. For this dataset, L_u is a matrix of size $|S| \times |T| = 583 \times 672$.

San Francisco Cabs (SFC). This dataset includes mobility traces recorded by San Francisco taxis from May 17 to June 10, 2008 [20]. Each record consists of a cab identifier, latitude, longitude, and a time stamp. The dataset includes approximately 11 million GPS coordinates generated by 536 taxis. We select 3 weeks (Monday May 19 to Sunday June 8) worth of data and discard traces outside downtown San Francisco (i.e., those of 2 taxis). To generate ROIs, we split the city into a 10 × 10 grid, whose cells are 0.18 sq. miles. Setting the time granularity to one hour, the 534 cabs report over 2M ROIs, on average 3,827 ± 1,069 locations per taxi, out of which 78 ± 6 ROIs are unique. The SFC data is less sparse than the TFL one, as cabs report locations more frequently – specifically 340 ± 94 out of the 504 time slots in the 21 considered days. For each cab u, we populate L_u by setting $L_u[s,t]$ to 1 if u was in the s cell at time $t \in T$, and 0 otherwise. If a cab does not report any location, we assign it to the special *null* ROI. For this dataset L_u is a matrix of size $|S| \times |T| = 101 \times 504$.

Sampling Users. For both datasets, we perform a basic analysis of the number of ROIs reported by their users. We observe that for TFL (resp., SFC), the median is 727 (resp., 4,111), with a maximum of 881 (resp., 8,136) and a minimum of 673 (resp., 504). We sort the users in each dataset per total number of ROI reports and split them in 3 groups of equal size, capturing their mobility patterns as: *highly*, *mildly*, and *somewhat* mobile. To avoid bias, to select target users, we sample 50 users from each mobility group *at random*. Thus, we run membership inference attacks against a total 150 users for each dataset.

B. *Experimental Setup*

Our experiments aim to evaluate the effectiveness of the distinguishing function d, used in the DG game, to guess whether the target user u^* is in the aggregates or not. As mentioned, we instantiate d using a machine learning classifier.

We train the classifier on a *balanced* dataset of *labeled* aggregates over user groups that include and groups that exclude u^*, so that it learns patterns that distinguish its participation in the aggregates. The training dataset is generated using data from the prior knowledge $\mathcal{P}$. We then play the game, i.e., we use the trained classifier to infer membership on a *balanced* testing set of aggregates previously *unseen*.

More specifically, we go through three phases: aggregation, feature extraction, and classification, which we describe in high-level. The concrete details of each phase depend on the adversarial prior knowledge, as we discuss later in Section VI (where we evaluate membership inference attacks with different priors). The three phases are discussed next.

Aggregation. We create a dataset D by repeating these steps:

1) Randomly generate a group U_0 of m users, which *includes* u^*;
2) Aggregate the location matrices of users in U_0, for $|T_I|$ intervals;
3) Append a row with the aggregates A_{U_0} to dataset D, and attach the label *in*;
4) Randomly generate a group U_1 of m users, which *excludes* u^*;
5) Aggregate the location matrices of users in U_1, for $|T_I|$ intervals;
6) Append a row with the aggregates A_{U_1} to the dataset D, and attach the label *out*.

Feature Extraction. For each row of the dataset, corresponding to the aggregates of a group with/without u^*, we extract statistics that are given as input to the classifier. Such statistics are calculated per location (ROI) and include variance, minimum, maximum, median, mean, standard deviation, as well as the sum of values of each location's time-series.

Classification. We first split the dataset D into the non-overlapping balanced training and testing sets mentioned above. We then train the classifier on the features extracted from the *training* set. Finally, we play the distinguishability game on the aggregates of the *testing* set (data previously unseen by the classifier), classifying them as including or excluding u^*.

Implementation. Our experiments are implemented in Python using the *scikit-learn* machine learning suite.[1] Source code is available upon request. We instantiate the following classifiers: i) Logistic Regression (LR), for which we employ a linear solver using a coordinate descent optimization algorithm suitable for binary classification [11]; ii) Nearest Neighbors (k-NN), configured to use Euclidean distance, with k set to 5, i.e., to predict the output class based on the votes of the 5 nearest samples; iii) Random Forests (RF), set up to train 30 decision trees and to consider all the features during the node splits using the Gini criterion to measure their quality; and iv) Multi-Layer Perceptron (MLP), consisting of 1 hidden layer with 200 nodes, whose weights are calculated via a stochastic gradient-based optimizer. For more details about the classifiers, we refer to Appendix A.

[1] http://scikit-learn.org/stable/

For the feature extraction, we use the *tsfresh* Python package.[2] For both datasets, and for all groups, we extract the 7 statistical features mentioned above, for each ROI. We obtain 4081 features for TFL (583 ROIs) and 707 features for SFC (101 ROIs). To avoid overfitting, we use Recursive Feature Elimination (RFE) to reduce the number of features to the number of samples we create for each user's dataset D. We then feed the features in their original form to all classifiers, except for MLP where we standardize them to have mean of 0 and variance 1.

VI. Evaluating Membership Inference on Raw Aggregate Locations

We now present the results of our experimental evaluation, measuring the performance of different classifiers in instantiating the distinguishing function (i.e., performing membership inference) on raw aggregates. We do so vis-à-vis the different priors discussed in Section IV-A, using the experimental methodology and the datasets described above. Recall that, in each experiment, we perform attacks against 150 users sampled from high, mild, and somewhat mobility profiles (50 each).

A. Subset of Locations

We start with the setting where the observation and inference periods coincide, and the adversary knows the time-series for a subset of users, including the target, during this period. This information can then be used by Adv to create groups, with and without her target, and train a classifier. We consider, as the observation/inference period, the first week of both TFL and SFC datasets – that is, $|T_O| = |T_I| = 24 \cdot 7 = 168$ hourly timeslots. We build Adv's prior by setting $\alpha = 0.11$ for TFL and $\alpha = 0.2$ for SFC, i.e, we randomly choose 1,100 out of 10,000 TFL users and 106 out of 534 SFC cabs. This represents a setting where Adv (e.g., a telecommunications service provider) knows location information for a small subset of users, including her target.

We then generate (i) a *balanced* training dataset by randomly sampling 400 *unique* user groups from Adv's prior knowledge, whereby half include the target user and half exclude it; and (ii) a *balanced* testing set by sampling 100 *unique* user groups from the set of users *not* in the prior knowledge (apart from the target user). Our choice for training/testing sizes (400 and 100, resp.) is so that the datasets are large enough to enable learning and evaluation, and experiments run in reasonable time. Finally, we extract features from the aggregates of both training and testing groups, labeling them as per the participation of the target in the group, and perform experiments with different values of m in order to evaluate the effect of aggregation group size.

TFL Dataset. Fig. 2 plots the CDF, computed over the 150 target TFL users, of the AUC score achieved by the classifiers for different values of m. Limited by the adversarial knowledge (1,100 users), we examine aggregation group sizes up to 1,000. Note that the orange line labeled as 'BEST' represents a hypothetical best case in which Adv chooses the classifier that yields the highest AUC score for each target user.

When groups are small, i.e., $m = 5$ or 10, all classifiers achieve very high AUC scores. For instance, with $m = 10$, Linear Regression (LR) and Random Forest (RF) achieve a mean AUC score of 0.97 and 0.99, respectively. This indicates that for such small groups, where users' contribution to the aggregates is very significant, membership inference is very effective. As the size of the aggregation groups increases to $m = 50$ or 100, the performance only slightly decreases, with RF outperforming LR, Nearest Neighbors (k-NN), and Multi-Layer Perceptron (MLP), yielding 0.94 mean AUC for groups of 50 users, and 0.83 for 100. With larger aggregation sizes, $m = 500$ or 1,000, performance drops closer to the random guess baseline (AUC = 0.5). Nonetheless, even for groups of 1,000 users, Adv can still infer membership of 60% of the target population with an AUC score higher than 0.6.

We also measure the impact of the effectiveness of the distinguishing function on privacy using the Privacy Loss metric (PL, cf. Eq. 3). More specifically, in Fig. 3a, we report a box-plot with the PL for different aggregation group sizes, when the adversary picks the best classifier for each target user (orange line in Fig. 2). For small groups, mean PL is very large, e.g., 0.99 for $m = 10$, 0.89 for 50, and 0.68 for 100. Unsurprisingly, PL decreases as the group size increases, i.e., users enjoy better privacy when their data is aggregated in larger groups. Yet, even then they experience a 25% reduction of privacy vs a random guess ($m = 1{,}000$).

SFC Dataset. In Fig. 4, we plot the classifiers' performance on the SFC dataset for groups of up to 100 users, as we are limited by the adversarial knowledge (106 cabs). As in the previous case, for small groups ($m = 5$, 10) Adv can infer membership with high accuracy. For instance, for groups of 10 users, LR and MLP achieve mean AUC of 0.9, followed by RF (0.84) and k-NN (0.7). Again, performance decreases as group size increases: for groups of 50 cabs (resp., 100) MLP and LR yield mean AUC scores of 0.72 (resp., 0.68) and 0.7 (resp., 0.67). Nonetheless, when Adv picks the best classifier for each cab (orange line), mean AUC score is still 0.72 even with 100 users per group.

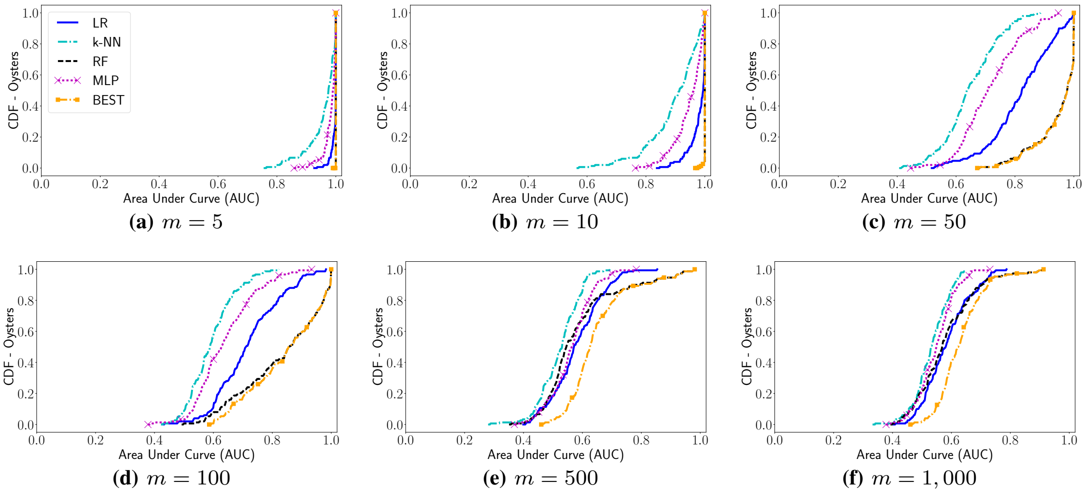
PL over the different values of m is explored in Fig. 3b, using the best classifier for each target. Similar to the TFL case, the loss is very large for small groups (e.g., PL = 0.86 when $m = 10$), and remains significant in larger ones (e.g., PL = 0.44 when $m = 100$). Interestingly, for groups up to 100 users, PL is larger on TFL than on SFC data (e.g., PL = 0.68 on TFL vs 0.44 on SFC, for $m = 100$), indicating that membership inference is easier on sparse data.

B. Participation in Past Groups

Next, we simulate the setting where Adv's prior knowledge consists of aggregates of groups released in a past observation period, labeled as including data from the target user or not. As discussed in Section IV-A, we consider two variants: when Adv's prior knowledge is built on either (a) the *same* groups as or (b) *different* groups than those used to compute the inference period aggregates.

***Same* Groups as Released.** In this setting, we generate D by computing the aggregates of $\beta = 150$ randomly sampled unique user groups – 75 that include and 75 that exclude the target – and set the corresponding label of participation. We

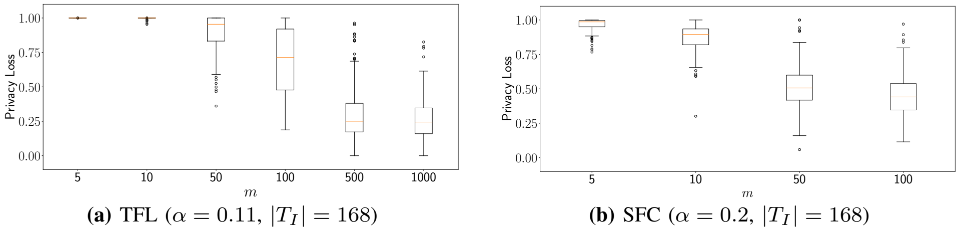
[2] http://tsfresh.readthedocs.io/en/latest/



(a) $m = 5$ **(b)** $m = 10$ **(c)** $m = 50$

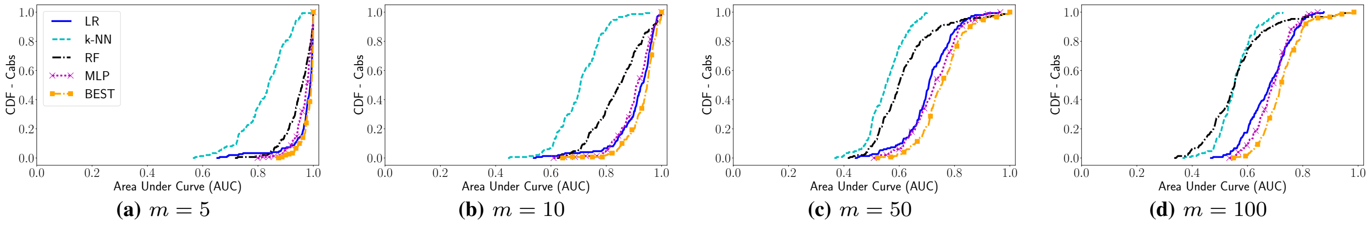
(d) $m = 100$ **(e)** $m = 500$ **(f)** $m = 1,000$

Fig. 2: *Subset of Locations* prior (TFL, $\alpha = 0.11$, $|T_I| = 168$) – Adv's performance for different values of m.



(a) TFL ($\alpha = 0.11$, $|T_I| = 168$) **(b)** SFC ($\alpha = 0.2$, $|T_I| = 168$)

Fig. 3: *Subset of Locations* prior - Privacy Loss (PL) for different values of m.



(a) $m = 5$ **(b)** $m = 10$ **(c)** $m = 50$ **(d)** $m = 100$

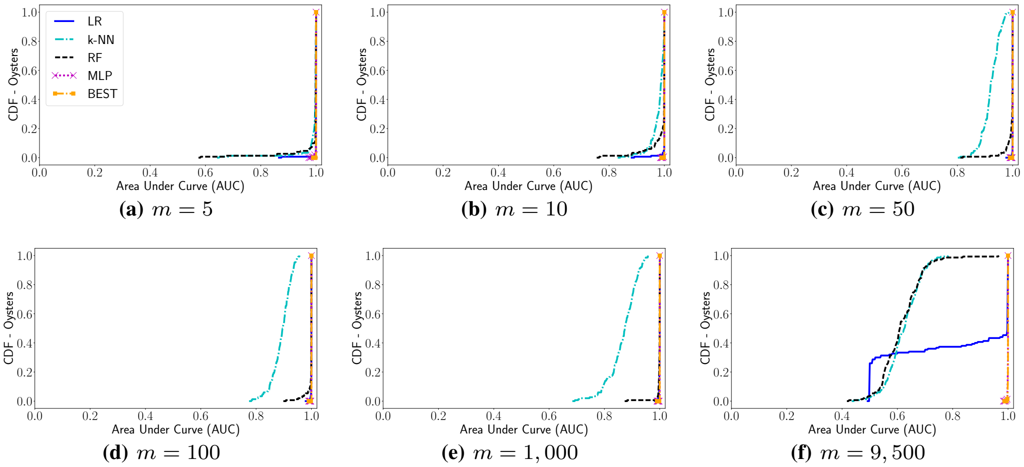
Fig. 4: *Subset of Locations* prior (SFC, $\alpha = 0.2$, $|T_I| = 168$) – Adv's performance for different values of m.

split D over time to obtain the training and testing sets. As observation period, i.e., where Adv builds her prior knowledge, we consider the first 3 weeks for TFL (i.e., $|T_O| = 3 \cdot 168 = 504$ hourly timeslots) and the first 2 weeks for SFC ($|T_O| = 336$). In both cases, the inference period is the last week of data, thus $|T_I| = 168$ hourly timeslots, yielding a 75%-25% split for TFL, and a 67%-33% split for SFC. Finally, we train the classifiers with features extracted from the aggregates of *each week* in the training set, and test them on those extracted from the aggregates of each group in the test set.

TFL Dataset. Fig. 5 shows the classifiers' performance for different aggregation group sizes (m). In this experiment, there is no limitation from the prior, thus we can consider groups as large as the dataset. As expected, we observe that for group sizes up to 100 (Figs. 5a–5d), membership inference is very accurate (all classifiers yield mean AUC scores over 0.9). Interestingly, as the groups grow to 1,000 commuters (Fig. 5e), LR, RF and MLP still yield very high AUC scores (0.99 on average), while k-NN slightly decreases (0.86). For groups of 9,500 commuters (Fig. 5f), MLP clearly outperforms the other classifiers yielding an AUC score of 0.99 compared to 0.81 for LR, 0.62 for k-NN and 0.61 for RF. Overall, this experiment indicates that when mobility patterns are regular, as the ones of commuters, an adversary with prior knowledge about specific groups can successfully infer membership in the future if groups are maintained, even if they are large.

Fig. 6a reports the privacy loss (PL) when the adversary picks the best classifier for each user. We see that, *independently* of the group size, commuters lose a huge amount of privacy when they are aggregated in groups for which the adversary has prior knowledge. The results reinforce the previous intuition: the effect of regularity on aggregates is very strong, and makes commuters very susceptible to membership inference attacks.

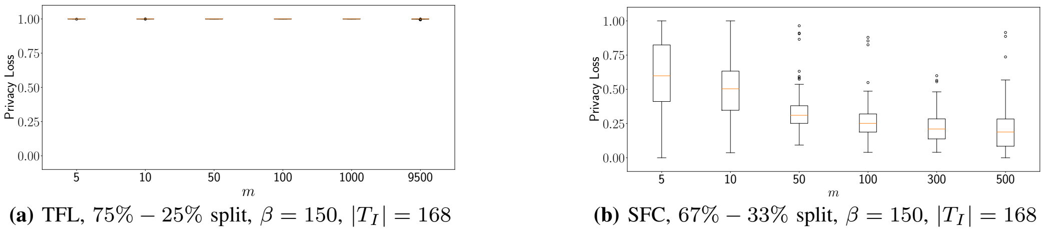
SFC Dataset. Fig. 7 illustrates the performance of the classifiers for variable aggregation group size on the SFC dataset. Recall that this is smaller than TFL, as it only contains 534 cabs. We observe that the lack of regularity in cabs movement has a great impact on the ability of an adversary to infer membership, even when the groups are maintained



(a) $m = 5$ **(b)** $m = 10$ **(c)** $m = 50$

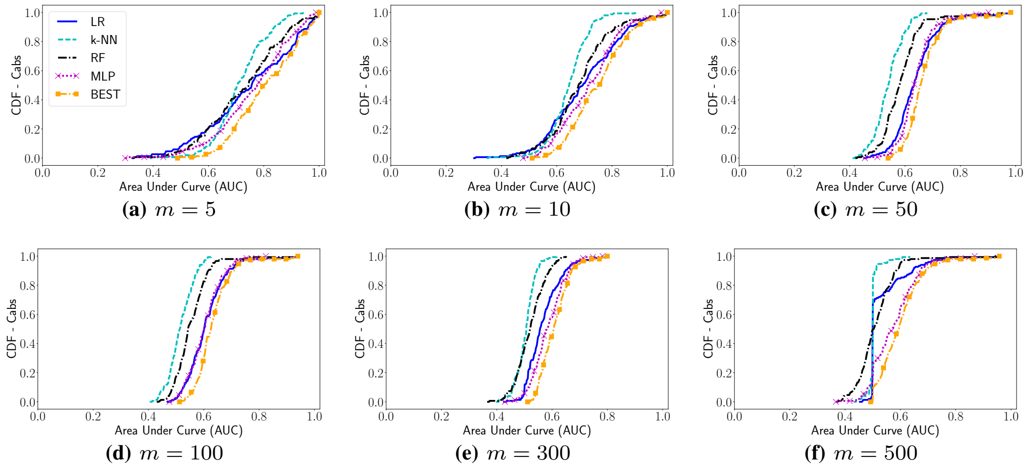
(d) $m = 100$ **(e)** $m = 1,000$ **(f)** $m = 9,500$

Fig. 5: *Same Groups as Released* prior (TFL, 75%-25% split, $\beta = 150$, $|T_I| = 168$) – Adv's performance for different values of m.



(a) TFL, $75\% - 25\%$ split, $\beta = 150$, $|T_I| = 168$ **(b)** SFC, $67\% - 33\%$ split, $\beta = 150$, $|T_I| = 168$

Fig. 6: *Same Groups as Released* prior - Privacy Loss (PL) for different values of m.



(a) $m = 5$ **(b)** $m = 10$ **(c)** $m = 50$

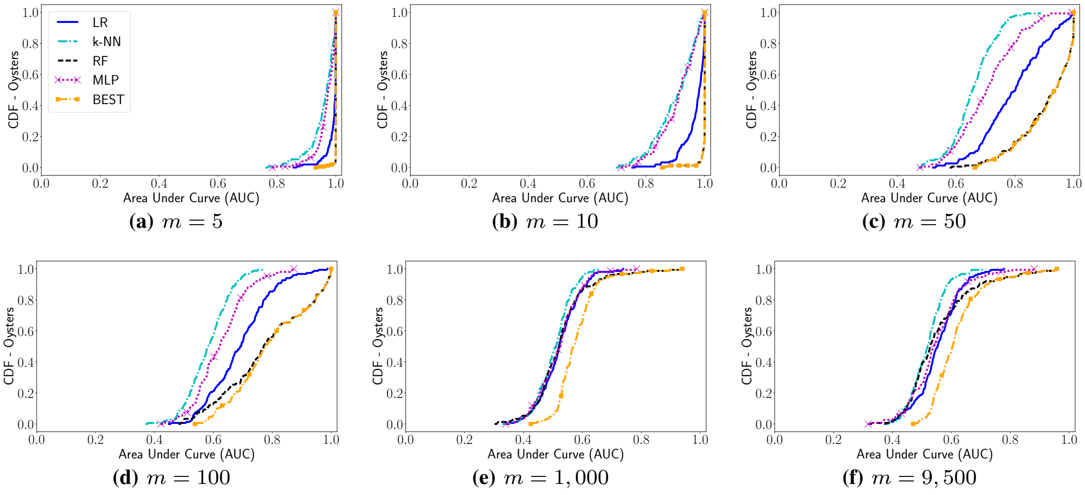
(d) $m = 100$ **(e)** $m = 300$ **(f)** $m = 500$

Fig. 7: *Same Groups as Released* prior (SFC, 67%-33% split, $\beta = 150$, $|T_I| = 168$) – Adv's performance for different values of m.

over time. For small groups ($m = 5$ or 10), the classifiers' AUC ranges between 0.76 and 0.64, as opposed to 0.9 or more in TFL, with MLP now yielding the best results. As groups become larger (Figs. 7c–7e), irregularity has a bigger effect and, unexpectedly, performance drops further. Already for $m = 100$, RF and k-NN perform similar to the random guess baseline, and LR's AUC drops to 0.52 when group size reaches $m = 500$. MLP, however, is still somewhat better than random (0.57 mean AUC). Overall, if the adversary picks the best classifier for each cab (orange line), she can infer membership for half the cabs with AUC score larger than 0.6.

In terms of PL, Fig. 6b shows that cabs lose privacy when they are aggregated in small groups. However, since cabs, as well as the groups they are aggregated in, are not as regular as TFL commuters, the loss drops drastically with larger groups (e.g., mean PL is 0.2 for groups of 500 cabs). In other words, irregularity makes inferring membership harder for the adversary. However, even though on average PL decreases, we observe that, for $m = 500$, some instances of our experiment exhibit larger privacy loss than for $m = 300$. This stems from the small size of the cab population. As there are only 534 cabs, when grouping them in batches of 500 elements, there inevitably is a big overlap across groups, which effectively creates a somewhat "artificial" regularity that increases the performance of the attack.

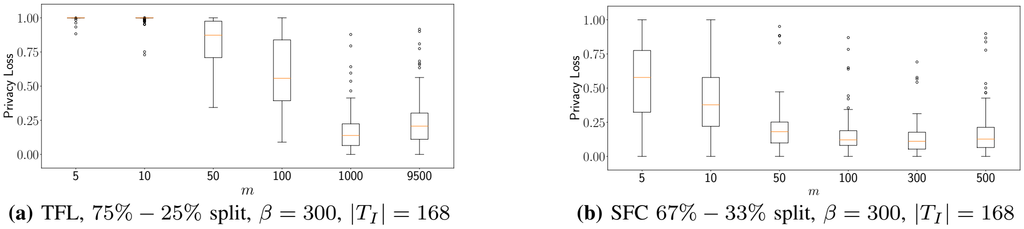
***Different* Groups than Released.** In this setting, for each



(a) $m = 5$ **(b)** $m = 10$ **(c)** $m = 50$

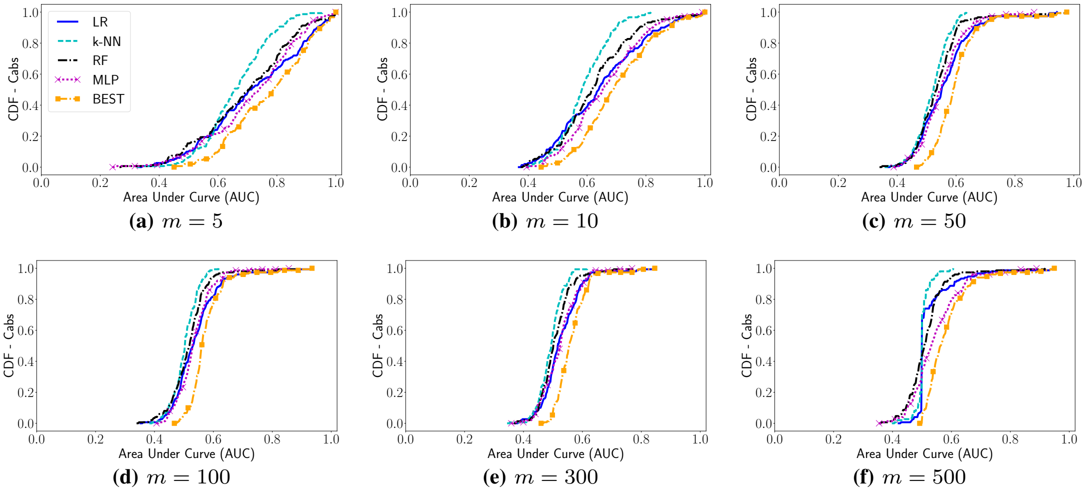
(d) $m = 100$ **(e)** $m = 1,000$ **(f)** $m = 9,500$

Fig. 8: *Different Groups than Released* prior (TFL, 75%-25% split, β=300, $|T_I|$=168) – Adv's performance for different values of m.



(a) TFL, $75\% - 25\%$ split, $\beta = 300$, $|T_I| = 168$ **(b)** SFC $67\% - 33\%$ split, $\beta = 300$, $|T_I| = 168$

Fig. 9: *Different Groups than Released* prior - Privacy Loss (PL) for different values of m.

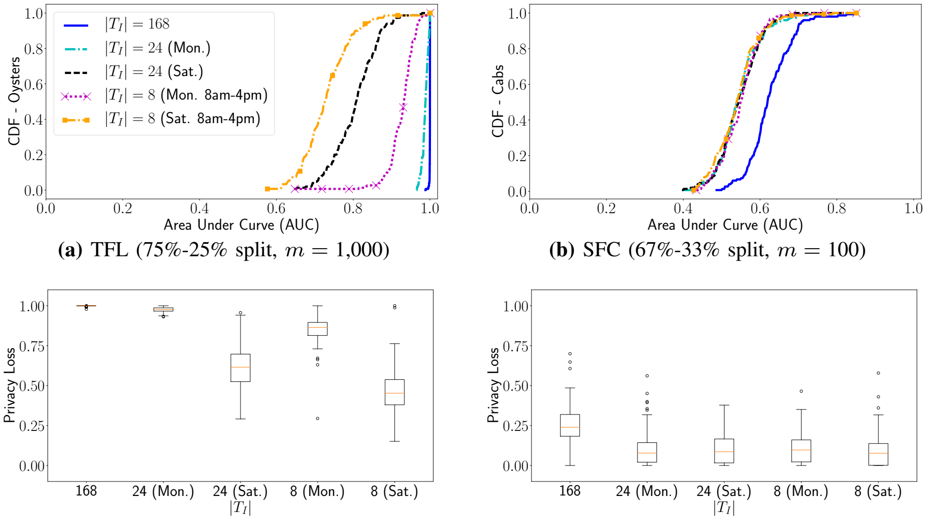


(a) $m = 5$ **(b)** $m = 10$ **(c)** $m = 50$

(d) $m = 100$ **(e)** $m = 300$ **(f)** $m = 500$

Fig. 10: *Different Groups than Released* prior (SFC 67%-33% split, β=300, $|T_I|$=168) – Adv's performance for different values of m.

target user, we design a balanced experiment by generating a dataset D with the aggregates of 400 *unique* randomly sampled groups – half including the target and half not – and set the corresponding participation label (in/out). Once again, the experiment size is chosen to provide enough data for our classifiers to learn patterns, while keeping the computation time reasonable on commodity hardware. To simulate the difference in groups between observation and inference period, we first perform a 75%-25% stratified random split of D, whereby we keep 300 groups for training and 100 for testing. Then, for TFL, we define the observation period to be the first 3 weeks of data (i.e., $|T_O| = 504$) while for SFC the first 2 weeks ($|T_O| = 336$) and in both cases, the inference period is the last week (i.e., $|T_I| = 168$). We then split the training and testing sets according to time: from the training set, we keep *only* the aggregates of the observation period, while, from the testing set, *only* those from the inference period (i.e., overall, we perform a 75%-25% split for TFL and 67%-33% for SFC). That is, we let the adversary obtain knowledge for 300 user groups (i.e., $\beta = 300$), half including her target, whose aggregates are generated during the observation period. Finally, we train the classifiers on the features extracted from



(a) TFL (75%-25% split, m = 1,000)

(b) SFC (67%-33% split, m = 100)

(c) TFL (75%-25% split, m = 1,000)

(d) SFC (67%-33% split, m = 100)

Fig. 11: *Same Groups as Released* prior (β=150) - Adv's performance for variable inference period length ($|T_I|$), on (a) TFL and (b) SFC, and Privacy Loss on (c) TFL and (d) SFC.

the aggregates of the groups in the *training* set for each week of the *observation period*, and test them against those extracted from the groups in the *testing* set (during the *inference* period).

TFL Dataset. Fig. 8 illustrates the classifiers' performance for different aggregation group sizes (up to 9,500 commuters, since there are no restrictions). Once again, for small groups (m = 5 or 10) membership can be easily inferred (AUC $>$ 0.89 for all classifiers). As m starts increasing, we first observe a small drop in the adversarial performance, with RF achieving mean AUC of 0.89 and 0.78 for groups of 50 and 100 commuters, resp. This indicates that regularity still helps membership inference in small groups even when these groups change. However, when m reaches 1,000 all the classifiers perform, on average, similar to the baseline indicating that the effect of regularity dilutes. Interestingly, for m = 9,500, we note a small increase in the classifiers' AUC scores due to the big user overlap across training and testing groups, i.e., the different-groups prior becomes more similar to the same-groups prior.

This effect can also be observed in terms of PL (Fig. 9a). Membership inference is quite effective for groups of size up to 100, where commuters suffer a privacy loss of at least 0.59. However, when data of more commuters is aggregated, mean PL decreases to 0.17 for groups of 1,000, and it slightly increases to 0.22 when m = 9,500. Overall, we note that the privacy loss is smaller in this setting, however, this is not surprising, since this is a much weaker adversarial setting than the previous ones.

SFC Dataset. Similar to the experiment with the same groups prior, we observe in Fig. 10 that the classifiers perform worse for SFC than TFL, due to the lack of regularity. Already for small groups (m = 5) the mean AUC drops to 0.71 for the best classifiers, LR and MLP. With larger groups, the performance is significantly lower, and all classifiers converge towards the random guess baseline. When m = 500, MLP and LR yield slightly better results and membership inference can be achieved with AUC larger than 0.6 for only a small percentage of cabs (about 20%).

From Fig. 9b, we see that, due to the weaker prior, PL values are smaller across the board compared to the previous setting. Overall, PL decreases with increasing aggregation group size, ranging from mean PL of 0.54 with $m = 5$ to 0.12 for m = 300. Similar to the TFL case, we observe a small increase for groups of 500 cabs. The reason is the same, i.e., the user overlaps between training and testing groups slightly improve the effectiveness of the membership inference attack.

C. Length of Inference Period

In the previous experiments, we have studied the effect of the size of the aggregation groups (m) on the success of membership inference, for various types of adversarial prior knowledge. In this section, we examine the *effect of the inference period length*, i.e., $|T_I|$. We consider lengths of 1 week (168 hourly timeslots), 1 day (24 timeslots), and 8 hours (8 timeslots). In particular, for the last two, we also consider working vs weekend days to account for the difference in mobility behavior.

Due to space limitations, we only report experiments in the setting where Adv has prior knowledge about the exact groups that are released by Ch – i.e., prior (2a) in Section IV-A – and fix the group size to 1,000 commuters for TFL and to 100 cabs for SFC. For each target user, we create a dataset of β = 150 random unique groups, half of which include the user and half of which do not, and split their aggregates in training and testing sets according to time following a 75%-25% split for TFL and a 67%-33% for SFC. We choose RF as classifier for TFL, and MLP for SFC, since they yield the best AUC scores in this setting, as shown in Figs. 5e and 7d. For each $|T_I| \in \{8, 24, 168\}$, we train the classifiers on aggregates of that length, for *each week* in the training set (observation period), and evaluate them against the corresponding aggregates in the test set (inference period).

Fig. 11a reports the results on the TFL dataset: as the number of points in the inference period $|T_I|$ decreases, the adversarial performance degrades as there is less information about mobility patterns to be exploited. Also, there is indeed a difference between working days and weekends. Mean AUC is 0.97 when training and testing on a Monday, and 0.8 on a Saturday. This seems to be due to regularity, as commuters' regular patterns during the week make them more susceptible to membership inference than sporadic/leisure activities over weekends. This is confirmed by the classifier's performance for $|T_I| = 8$, as we obtain much better results when the inference period is set to Monday 8am–4pm (AUC = 0.91) than on Saturday during the same timeframe (AUC = 0.72).

Once again, the lack of regularity affects negatively Adv's performance when attacking the SFC dataset (Fig. 11b). As for the length of the inference period, our results confirm that the inference task becomes harder with fewer points in time: mean AUC drops from 0.62 to 0.54 when $|T_I|$ goes from 1 week to 1 day. However, as cabs are never regular (their movements are mandated by client demand), we do not observe significant difference between working days and weekends, nor when considering full days vs 8h slots.

The Privacy Loss (PL) exhibits similar trends. For TFL (see Fig. 11c), the highest loss is observed when more points are available (0.98 on average when $|T_I|$ is 1 week), while the loss is reduced as the length of the inference period decreases and the adversary has less information. Also, we see how regularity in working days results in better membership inference attacks than during weekends, i.e., mean PL is 0.96 and 0.85 on Mondays vs 0.61 and 0.46 on Saturdays for $|T_I|$ set at 24 and 8 hours, respectively. Finally, Fig. 11d highlights smaller PL for the SFC cabs, for all period lengths, with a maximum mean PL of 0.25 when $|T_I|$ is 1 week, down to 0.1 and 0.09 for 1 day and 8 hours, respectively. There is no significant difference between Mondays and Saturdays, confirming that regularity has a strong influence on the problem.

D. Raw Aggregates Evaluation – Take-Aways

Overall, our evaluation showcases the effectiveness of modeling membership inference attacks on aggregate location time-series as a classification task, vis-à-vis different datasets and priors. We show that an adversary can build a machine learning model by extracting features from known aggregate location time-series and use it to guess whether a target user has contributed to a set of previously unseen aggregates. Our results evidence that the risks stemming from such attacks are significant, with the actual level of privacy leakage depending on the adversary's prior knowledge, the characteristics of the data, as well as the group size and timeframe on which aggregation is performed.

We find that, up to certain aggregation group sizes, membership inference is very successful when the adversary knows the actual locations of a subset of users (including her target), or when she knows past aggregates for the same groups on which she tries to perform the inference. In the least restrictive setting, where the past groups known to the adversary are different than those whose statistics are released, privacy leakage is relatively small, but still non-negligible.

Moreover, the characteristics of the data used for aggregation also influence the adversarial performance: in general, privacy leakage on the dataset containing mobility of commuters (TFL) is larger than on the one including cab traces (SFC). This highlights that regularity in users' movements, as well as sparseness of the location signal, significantly ease the membership inference task.

Finally, the number of users that contribute to aggregation also has a profound effect on the adversarial performance. Unsurprisingly, membership inference is very successful when aggregation is performed over small groups, while users generally enjoy more privacy in larger groups. A notable exception is the TFL case where, due to the regularity of commuters, membership inference attacks are still very effective even for large groups. Also, the length, as well as the time semantics, of the inference period play a very important role. Inference is easier if the aggregates of longer periods are released (i.e., more information is available to extract patterns), and at times when mobility patterns are likely to be more regular (e.g., workdays or mornings).

VII. Evaluating DP Defenses

In this section, we evaluate the effectiveness of available defense mechanisms to prevent privacy leakage from membership inference attacks on aggregate location time-series.

A. Differential Privacy (DP)

The established framework to define private functions that are free from inferences is Differential Privacy (DP) [8]. Applying differentially private mechanisms to a dataset ensures that only a bounded amount of information is disclosed upon its release. This can mitigate membership inference attacks, as DP's indistinguishability-based definition guarantees that the outcome of any computation on a dataset is insensitive to the inclusion of any data record (user) in the dataset.

Formally, we say that a randomized algorithm $\mathcal{M}$ is (ϵ, δ)-*differentially private* if for all datasets D_1 and D_2 that differ on at most one element, and all $\mathcal{S} \subseteq \text{Range}(\mathcal{M})$:

$$\Pr[\mathcal{M}(D_1) \in \mathcal{S}] \leq e^{\epsilon} \times \Pr[\mathcal{M}(D_2) \in \mathcal{S}] + \delta \qquad (4)$$

where the probability is calculated over the coin tosses of $\mathcal{M}$ and $\text{Range}(\mathcal{M})$ denotes the set of possible outcomes of $\mathcal{M}$. In other words, the outcome of $\mathcal{M}$ has a very small dependence on the members of the dataset. If $\delta = 0$, we say that $\mathcal{M}$ is ϵ-differentially private [8].

We also review the notion of *sensitivity*, which captures how much one record affects the output of a function. Formally, for any function $f : \mathcal{D} \to \mathbb{R}^d$, the sensitivity of f is:

$$\Delta f_p = \max_{D_1, D_2} \parallel f(D_1) - f(D_2) \parallel_p \qquad (5)$$

for all datasets D_1, D_2 differing on at most one element, with $\| \cdot \|_p$ indicating the ℓ_p norm.

Laplacian Mechanism (LPA): The most common used mechanism to achieve DP is to randomize the aggregate statistics to be released using random noise independently drawn from the Laplace distribution. For a function $f : \mathcal{D} \to \mathbb{R}^d$, a randomized mechanism $\mathcal{M}$ defined as $\mathcal{M}(X) = f(X) + \text{Lap}(0, \sigma)^d$ guarantees ϵ-DP if $\sigma \geq \frac{\Delta f_1}{\epsilon}$. A *weaker* version of LPA has been proposed for time-series, which perturbs the counts of a time-series with noise distributed according to Lap$(1/\epsilon)$ [6]. However, this mechanism only provides *event-level privacy* [10], i.e., in our setting, it can only protect single location visits. We include it as a *baseline* for our evaluation, since we do not expect it to perform well for full time-series.

Gaussian Mechanism (GSM): Another mechanism consists in perturbing the statistics with random noise drawn from the Gaussian distribution $\mathcal{N}$. Given $f : \mathcal{D} \rightarrow \mathbb{R}^d$, a randomized mechanism $\mathcal{M}$ defined as $\mathcal{M}(X) = f(X) + \mathcal{N}(0, \sigma^2)^d$ provides (ϵ, δ)-DP when $\sigma \geq \frac{\sqrt{2 \cdot ln(2/\delta)}}{\epsilon} \cdot \Delta f_2$ [9]. Note that this is a weaker privacy guarantee than the one offered by LPA.

Fourier Perturbation Algorithm (FPA): We then consider differentially private mechanisms proposed specifically for time-series settings. One is FPA [23], which performs the noise addition on the compressed frequency domain: a time-series is compressed using the Discrete Fourier Transform (DFT) and the first κ Fourier coefficients F_κ are kept. Then F_κ is perturbed with noise distributed according to Lap$(\sqrt{\kappa} \cdot \Delta f_2/\epsilon)$ and padded with zeros to the size of the original time-series. Finally, the inverse DFT is applied to obtain the perturbed time-series. As per [23], FPA provably guarantees ϵ-DP.

Enhanced Fourier Perturbation Algorithm with Gaussian Noise (EFPAG): EFPAG [1] improves FPA by choosing the number of coefficients (κ) to be perturbed probabilistically, and using the exponential mechanism to assign larger probability to values that minimize the root-sum-squared error between the input time-series and its noisy version. Then, rather than DFT, it uses the Discrete Cosine Transform (DCT) and employs Gaussian noise instead of Laplacian to achieve better accuracy. As a result, EFPAG guarantees (ϵ, δ)-DP [1].

B. Experimental Design

Intuition. Our evaluation of membership inference on raw aggregate location time-series showed that releasing them poses a significant privacy threat for users whose times-series are aggregated, and more so in settings where aggregation is performed over small groups. In this section, we present experiments aiming to evaluate the effectiveness of differentially private mechanisms in defending against such inferences. Note that we opt to evaluate them over large groups, since we expect (and have also verified experimentally) that, for small groups, the loss of utility incurred by DP-based mechanisms is prohibitively high. This is because the *sensitivity* of the location aggregation function, which directly affects the amount of noise to be employed, does not depend on the group size (m). As such, the aggregate time-series of groups with few users, which naturally have small counts, are affected more by the noise required by the DP-based mechanism.

As the large group size gives the defense mechanism an "advantage" in terms of utility, and since DP provides protection against arbitrary risks [8], we set to run our experiments considering a worst-case adversary that obtains *perfect* prior knowledge for the users, i.e., she knows the inference period aggregates of the groups that are released by the challenger as well as the target user's membership in these groups. With this knowledge, Adv is able to train an accurate machine learning classifier that, upon release of raw statistics, always guesses correctly the target user's membership – i.e., achieves an AUC score of 1. In other words, we evaluate the privacy/utility trade-off of differentially private mechanisms considering the best setting for utility and the worst one for privacy.

Experiments. We slightly modify the DG game in Fig. 1, such that Ch applies a differentially private mechanism on the aggregates, before sending her *challenge* to Adv. We evaluate the gain in privacy offered by the mechanisms on two cases, depending on whether Adv's classifier (which, once again, instantiates the distinguishing function) is trained on (1) the raw aggregates of the groups to be released; or (2) noisy aggregates of the groups to be released using the defense mechanism under examination.

In both cases, testing is done on the aggregates of the released groups, perturbed with the defense mechanism (using *fresh* noise). The first scenario represents a *passive* adversary that attempts to infer user membership on the noisy aggregates, exploiting only the raw aggregate information from her prior knowledge. The second one, represents a strategic *active* adversary that tries to mimic the behavior of the defender during training, knowing the parameters of the defense mechanism (i.e., ϵ and the sensitivity of the aggregation function denoted as Δ), but not knowing the concrete values used in the defender's perturbation. We follow the same procedure as in Section V, i.e., we extract features from the aggregate location time-series of the user groups, and use RFE to reduce the number of features to the number of samples.

Settings. We run membership inference attacks against the 150 users sampled from each dataset (cf. Section V-A). We take as observation/inference period the first week in each dataset, i.e. $|T_O| = |T_I| = 168$. Aiming to examine a favorable setting for the utility of DP-based mechanisms, we construct large user groups: we set m = 9,500 for TFL, and m = 500 for SFC. Then, we generate the dataset D by randomly sampling 200 and 400 elements for TFL and SFC, respectively, half including the target and half not. We pick a different number of groups for practical reasons: the TFL dataset has six times more ROIs than the SFC one, and this makes feature extraction significantly more expensive. As classifier, we employ MLP, which performed well overall in the previous experiments.

To configure the perturbation mechanisms, we calculate the sensitivity Δ for the users in each dataset (i.e., the maximum number of ROIs reported by an oyster/cab in the inference week), obtaining Δ = 207 for TFL and Δ = 2,685 for SFC. We consider ϵ values of DP in the range $\{0.01, 0.1, 1.0, 10.0\}$, and set δ = 0.1 for GSM and EFPAG. For FPA, we empirically find the best value for κ in terms of utility, setting κ = 25 for TFL, and κ = 20 for SFC.

Metrics. Our evaluation uses the following privacy and utility metrics to capture the amount of privacy gained compared to a setting where the DG game is played on raw aggregates, and the utility lost due to the noise addition.

Privacy Gain (PG): We define PG as the relative decrease in a classifier's AUC score when tested on perturbed aggregates (AUC$_{A'}$) versus raw aggregates (AUC$_A$):

$$\text{PG} = \begin{cases} \frac{\text{AUC}_A - \text{AUC}_{A'}}{\text{AUC}_A - 0.5} & \text{if AUC}_A > \text{AUC}_{A'} \geq 0.5 \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

ϵ	0.01	0.1	1.0	10
LPA(Δ/ϵ)	3056.1	812.6	81.7	8.2
GSM	753.2	75.8	7.4	0.75
FPA	67.2	6.1	0.7	0.03
EFPAG	36.8	3.6	0.4	0.03
LPA(1 / ϵ)	38.5	3.7	0.3	0.002

TABLE II: MRE of aggregate location time-series with different differentially private mechanisms and parameter ϵ (TFL).

ϵ	0.01	0.1	1.0	10
LPA(Δ/ϵ)	131.9	129.3	114.4	41.9
GSM	129.6	94.7	14.1	1.4
FPA	85.9	11.3	1.1	0.11
EFPAG	57.9	6.1	0.6	0.04
LPA(1 / ϵ)	24.7	2.5	0.2	0.001

TABLE III: MRE of aggregate location time-series with different differentially private mechanisms and parameter ϵ (SFC).

PG is a value between 0 and 1, which captures the decrease in adversarial performance, i.e., how much the adversary's inference power deteriorates towards the random guess baseline, when a defense mechanism is implemented.

Mean Relative Error (MRE): We evaluate the utility loss as a result of using DP-based defense mechanisms by means of the standard MRE metric, computed between the raw aggregate time series Y, of length n, and its perturbed version Y':

$$\text{MRE}(Y, Y') = \frac{1}{n}\sum_{i=1}^{n}\frac{|Y'_i - Y_i|}{\max(\gamma, Y_i)} \tag{7}$$

where γ is a sanity bound mitigating the effect of very small counts. Following previous work [1], we set γ to 0.1% of $\sum_{i=1}^{n} Y_i$. We compute the MRE over the aggregate time-series for all ROIs ($s \in S$) in our datasets and report the mean value.

C. Results

Utility. We first report the utility measured as per the Mean Relative Error (MRE) of the TFL and SFC perturbed aggregates, for each mechanism and different values of ϵ, in Tables II and III. Naturally, as ϵ increases, so does utility. Note that LPA(Δ/ϵ) incurs the highest MRE, with the noisy aggregate values being 8 (resp., 41) times less accurate than raw aggregates on TFL (resp., SFC) data, in the most relaxed privacy setting ($\epsilon = 10$). With GSM, utility does not improve much. On the other hand, FPA and EFPAG achieve better results, e.g., MRE is under 1.1 for values of $\epsilon \geq 1$. Finally, note that LPA($1/\epsilon$) achieves the best utility, but it provides poor privacy protection against membership inference attacks, as shown below.

Privacy. We now evaluate the Privacy Gain (PG) provided by the different DP-based mechanisms, distinguishing between the two settings introduced in Section VII-B.

Train on Raw / Test on Noisy Aggregates. Fig. 12 plots the PG achieved by various mechanisms against a MLP classifier trained on raw aggregates and tested on perturbed ones. For TFL (Fig. 12a), we observe that for low ϵ values (up to 0.1) all mechanisms provide excellent privacy protection, achieving Privacy Gain (PG) close to 1. However, this protection comes with poor utility, as shown in Table II. As ϵ increases to 1, LPA(Δ/ϵ) and GSM still provide good protection, while we observe a small drop in PG for the mechanisms that achieve MRE < 1. In particular, FPA now yields a mean PG of 0.9, while EFPAG and LPA($1/\epsilon$) 0.75 and 0.38, resp. When $\epsilon = 10$ and the utility of FPA and EFPAG is good, the decrease in PG is bigger (0.45 and 0.3, resp.), as expected.

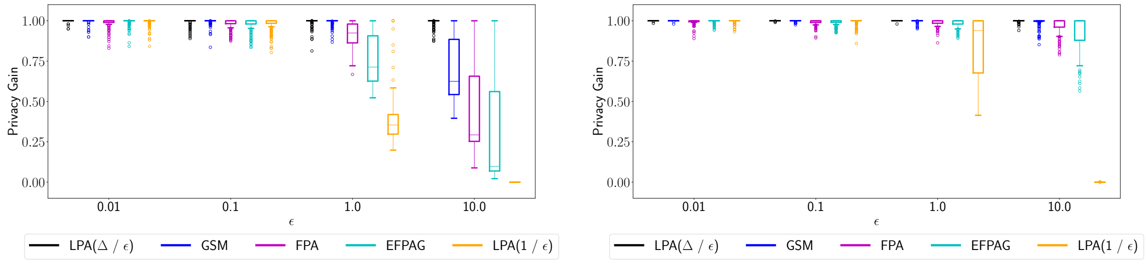
With SFC data (Fig. 12b), we find that PG for all the mechanisms stays high for values of ϵ up to 1. This is reasonable, since the sensitivity, and thus, the magnitude of noise required, is much larger on SFC compared to TFL (Δ = 2,685 vs 207). However, as seen from Table III, utility is quite poor in these settings. With $\epsilon = 10$, mean PG is almost 1 for LPA(Δ/ϵ) and GSM, i.e., users are well protected against membership inference attacks. Meanwhile, PG slightly drops for FPA and EFPAG (0.96 and 0.92 on average) while their utility is higher. Unsurprisingly, LPA($1/\epsilon$) achieves negligible privacy gain in this setting.

Train on Noisy / Test on Noisy Aggregates. Fig. 13 reports the PG results when the MLP classifier is trained on noisy aggregates. Interestingly, the protection of the mechanisms decreases much faster for increasing values of ϵ. For TFL (Fig. 13a), we observe that for values of $\epsilon \leq 1$, the PG decreases only slightly compared to the previous setting, where training was done on raw aggregates (Fig. 12a). That is, the DP-based mechanisms still provide good protection against membership inference. However, when $\epsilon = 10$, we notice a notable decrease in PG, with FPA and EFPAG. More precisely, FPA now achieves 0.2 mean PG (vs 0.45 in the previous setting) and EFPAG provides negligible protection against membership inference (compared to 0.3 in Fig. 12a, $\epsilon = 10$).

Similarly, with SFC (Fig. 13b), mean PG remains high for $\epsilon \leq 1$ for all mechanisms, except for LPA($1/\epsilon$). For $\epsilon = 10$, there is a significant decline in PG with GSM, FPA, and EFPAG. In particular, GSM now yields 0.8 mean PG, while FPA and EFPAG 0.32 and 0.15, respectively. This corresponds to a significant drop in privacy protection (20%, 66% and 83% for GSM, FPA, and EFPAG) compared to the setting where training was done on raw aggregates (cf. Fig. 12b).

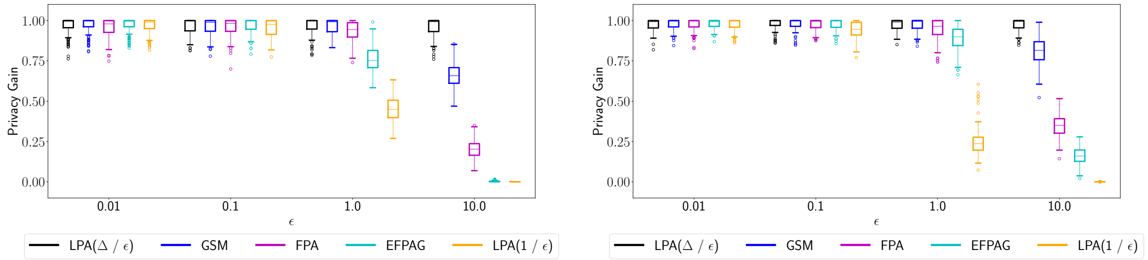
D. DP Evaluation – Take-Aways

The experiments presented in this section evaluate the performance of differentially private mechanisms against membership inference on aggregate location time-series, both in terms of privacy and utility. Considering an advantageous setting for utility, but worst-case for privacy, we find that differentially private mechanisms can be effective at preventing inferences, but with some important caveats. In particular, our results show that a *passive* adversary who trains a classifier on raw aggregate location data is not very successful at inferring membership on noisy aggregates. However, when we consider a *strategic* adversary that mimics the behavior of the defender, and trains a classifier on noisy aggregates, we find that the actual privacy gain offered from the DP-based mechanisms is significantly reduced, and also decreases much faster with increasing ϵ values. This should draw the attention of the research community as advances in deep learning, might lead to stronger attacks against defense mechanisms based on perturbation (e.g., by achieving noise filtering).



(a) TFL ($m = 9,500$, $|T_I| = 168$) **(b)** SFC ($m = 500$, $|T_I| = 168$)

Fig. 12: Privacy Gain (PG) achieved by differentially private mechanisms with different values of ϵ, against a MLP classifier trained on raw aggregates and tested on noisy aggregates.



(a) TFL ($m = 9,500$, $|T_I| = 168$) **(b)** SFC ($m = 500$, $|T_I| = 168$)

Fig. 13: Privacy Gain (PG) achieved by differentially private mechanisms with different values of ϵ, against a MLP classifier trained and tested on noisy aggregates.

Among the defense mechanisms considered, we observe that the straightforward application of LPA and GSM yields very poor utility. This is not surprising, as previous work highlights the difficulty of releasing private statistics under continual observation [6, 10, 17]. Mechanisms specifically proposed for time-series settings (i.e., FPA and EFPAG) yield much better utility, at the cost of reduced privacy. This shows that achieving an optimal trade-off between privacy and utility in the settings we consider is still a challenging task.

Finally, our analysis also shows how dataset characteristics affect the performance of differentially private mechanisms too. Specifically, the privacy gain on a sparser dataset (TFL) decreases faster with growing ϵ, compared to a denser one (SFC). This is not surprising, taking into account the scale difference between the sensitivity of the aggregation in each case (recall that $\Delta = 207$ on TFL and 2,685 on SFC).

VIII. Conclusion

Location privacy has been a prolific research area over the past few years, with a number of attacks and defenses having been proposed on mobility profiles and users' locations. Although this line of research has improved our understanding about protecting users against the disclosure of sensitive information, to the best of our knowledge, little work has focused on the privacy threats that the availability of aggregate location time-series may pose for individuals contributing to the aggregates.

This paper presented the first evaluation of membership inference in the context of location data. We formalize this inference as a distinguishability game in which an adversary has to guess whether or not a target user's location data has been used to compute a given set of aggregates. Instantiating the distinguishing function as a machine learning classifier, we quantify the inference power of an adversary with various types of prior knowledge on two real datasets with different characteristics. We show that, membership inference is very accurate when groups are small, and that users that have regular habits are easier to classify correctly than those performing sporadic movements.

We also evaluate the extent to which defense mechanisms based on differential privacy can prevent membership inference. We find that they are quite effective if the adversary trains the classifier on raw aggregates, though they entail a significant loss in utility. However, they are much less effective if the adversary mimics the behavior of the perturbation mechanism by training her classifier on noisy aggregates.

We remark that our methodology can be used to evaluate membership inference attacks, as well as defenses, in real-world settings. We also hope that our techniques can be leveraged by providers to test the quality of privacy protection before data release or by regulators aiming to detect violations.

Acknowledgments. We wish to thank Mirco Musolesi, Gordon Ross, and Claude Castelluccia for useful feedback and comments on our work. This research is partially supported by a Xerox University Affairs Committee grant on "Secure Collaborative Analytics" and the EU H2020 project "NEXTLEAP" (Grant Agreement No. 688722).

References

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

[2] M. Backes, P. Berrang, M. Humbert, and P. Manoharan. Membership privacy in MicroRNA-based studies. In *CCS*, 2016.

[3] D. Barth. The bright side of sitting in traffic: Crowdsourcing road congestion data. https://googleblog.blogspot.com/2009/08/bright-side-of-sitting-in-traffic.html, 2009.

[4] N. Buscher, S. Boukoros, S. Bauregger, and S. Katzenbeisser. Two Is Not Enough: Privacy Assessment of Aggregation Schemes in Smart Metering. In *PETS*, 2017.

[5] I. Ceapa, C. Smith, and L. Capra. Avoiding the crowds: understanding tube station congestion patterns from trip data. In *UrbComp*, 2012.

[6] T.-H. H. Chan, E. Shi, and D. Song. Private and continual release of statistics. *TISSEC*, 2011.

[7] Y.-A. de Montjoye, C. A. Hidalgo, M. Verleysen, and V. D. Blondel. Unique in the Crowd: The privacy bounds of human mobility. *Nature Scientific Reports*, 2013.

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

[9] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor. Our data, ourselves: Privacy via distributed noise generation. In *EUROCRYPT*, 2006.

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

[11] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. Liblinear: A library for large linear classification. *JMLR*, 2008.

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

[13] J. Hayes, L. Melis, G. Danezis, and E. De Cristofaro. LOGAN: Evaluating Privacy Leakage of Generative Models Using Generative Adversarial Networks. *arXiv preprint 1705.07663*, 2017.

[14] B. Hitaj, G. Ateniese, and F. Perez-Cruz. Deep Models Under the GAN: Information Leakage from Collaborative Deep Learning. In *CCS*, 2017.

[15] N. Homer, S. Szelinger, M. Redman, D. Duggan, W. Tembe, J. Muehling, J. V. Pearson, D. A. Stephan, S. F. Nelson, and D. W. Craig. Resolving individuals contributing trace amounts of DNA to highly complex mixtures using high-density SNP genotyping microarrays. *PLoS Genetics*, 2008.

[16] S. Ji, W. Li, N. Z. Gong, P. Mittal, and R. A. Beyah. On Your Social Network De-anonymizablity: Quantification and Large Scale Evaluation with Seed Knowledge. In *NDSS*, 2015.

[17] G. Kellaris, S. Papadopoulos, X. Xiao, and D. Papadias. Differentially private event sequences over infinite streams. In *VLDB*, 2014.

[18] N. Li, W. Qardaji, D. Su, Y. Wu, and W. Yang. Membership privacy: a unifying framework for privacy definitions. In *CCS*, 2013.

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

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

[21] R. A. Popa, A. J. Blumberg, H. Balakrishnan, and F. H. Li. Privacy and Accountability for Location-based Aggregate Statistics. In *CCS*, 2011.

[22] A. Pyrgelis, C. Troncoso, and E. De Cristofaro. What Does The Crowd Say About You? Evaluating Aggregation-based Location Privacy. In *PETS*, 2017.

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

[24] L. Rossi and M. Musolesi. It's the way you check-in: Identifying users in Location-based Social Networks. In *COSN*, 2014.

[25] R. Shokri. *Quantifying and protecting location privacy*. PhD thesis, École Polytechnique Féderale de Lausanne, 2012.

[26] R. Shokri, M. Stronati, C. Song, and V. Shmatikov. Membership inference attacks against machine learning models. In *S&P*, 2017.

[27] 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.

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

[29] R. Silva, S. M. Kang, and E. M. Airoldi. Predicting traffic volumes and estimating the effects of shocks in massive transportation systems. *PNAS*, 2015.

[30] Telefonica Smart Steps. https://www.business-solutions.telefonica.com/en/enterprise/solutions/smarter-selling/big-data-insights/, 2017.

[31] A. G. Thakurta, A. H. Vyrros, U. S. Vaishampayan, G. Kapoor, J. Freudinger, V. V. Prakash, A. Legendre, and S. Duplinsky. Emoji frequency detection and deep link frequency. US Patent 9,705,908, 2017.

[32] H. To, K. Nguyen, and C. Shahabi. Differentially private publication of location entropy. In *SIGSPATIAL*, 2016.

[33] R. Wang, Y. F. Li, X. Wang, H. Tang, and X. Zhou. Learning your identity and disease from research papers: information leaks in genome wide association study. In *CCS*, 2009.

[34] Waze. https://www.waze.com, 2017.

[35] D. Welter, J. MacArthur, J. Morales, T. Burdett, P. Hall, H. Junkins, A. Klemm, P. Flicek, T. Manolio, L. Hindorff, et al. The NHGRI GWAS Catalog, a Curated Resource of SNP-Trait Associations. *Nucleic Acids Research*, 2013.

[36] 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.

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

Appendix A
Machine Learning Classifiers

We now briefly review the machine learning classifiers used throughout the paper.

Logistic Regression (LR). LR is a linear model where the probabilities describing the possible outcomes of a single trial are modeled via a logistic (logit) function. The parameters of the model are estimated with maximum likelihood estimation, using an iterative algorithm.

Nearest Neighbors (k-NN). k-NN performs classification with a simple majority vote of the nearest neighbors of each data point: a query point is assigned the data class which has the most representatives within the nearest neighbors of the point.

Random Forest (RF). RF is an ensemble learning method which constructs a number of decision trees during training and outputs the majority class voted by the individual trees during testing. With RF, each tree in the ensemble is built from a sample drawn with replacement from the training set. When splitting a node during the construction of the tree, the split that is picked is the best split among a random subset of the features. As a result of this randomness, the bias of the forest usually slightly increases but, due to averaging, its variance also decreases.

Multi-Layer Perceptron (MLP). MLP is a kind of artificial neural network, consisting of at least three layers of nodes: the input, the hidden and the output layers. Except for the input nodes, each other node is a neuron that uses a non-linear activation function. MLP utilizes a supervised learning technique called back propagation for training and its multiple layers along with the non-linear activation allow it to distinguish data that is not linearly separable.