User-Level Differentially Private Mean Estimation for Real-World Datasets

V. Arvind Rameshwar, Anshoo Tandon, and Abhay Sharma

***Abstract*—This paper considers the problem of the private release of sample means of speed values from real-world traffic datasets. In earlier work, we developed user-level differentially private algorithms, with carefully chosen parameter values, which ensure low estimation errors on real-world Intelligent Traffic Management System (ITMS) data from an Indian city and on large synthetic datasets, while ensuring privacy. In the ITMS dataset, the speeds of different buses are drawn in a potentially non-i.i.d. manner from an unknown distribution. Moreover, in both the real-world and synthetic datasets, the number of speed samples contributed by different buses is potentially different. In this paper, we first provide theoretical justification for the performance trends we observe. We then present a novel procedure to improve performance based on the creation of pseudo-users, which optimizes the worst-case total estimation error.**

I. Introduction

It is now well-understood that the release of even seemingly innocuous functions of a dataset that is not publicly available can result in the reconstruction of the identities of individuals (or users) in the dataset with alarming levels of accuracy (see, e.g., [1]–[4]). To alleviate concerns over such attacks, the framework of differential privacy (DP) was introduced in [5], which guarantees the privacy of users when each user contributes *at most one* sample. However, most real-world datasets, such as traffic datasets, record multiple contributions from every user; naïvely applying standard DP techniques achieves poor estimation errors, owing to the addition of a large amount of noise to guarantee privacy.

In our recent work [6], we constructed algorithms, based on [7], for ensuring *user-level* privacy in the release of sample means of speed records in traffic datasets. We then emperically evaluated the performance of such algorithms, via extensive experiments, on real-world ITMS (Intelligent Traffic Management System) traffic data, supplied by IoT devices deployed in an Indian city, and on large synthetic datasets. Our main contributions there were carefully chosen, albeit heuristic, subroutines for clipping the number of samples contributed by each user and for clipping each speed value to lie in a high-probability interval, to achieve good estimation-privacy tradeoffs in practice.

In this work, we first provide theoretical proofs for the performance trends we observe in [6]. We then propose a novel procedure based on the creation of pseudo-users (see also [8]), which clips the number of samples contributed by a user in such a manner as to jointly optimize the *worst-case errors* due to clipping and due to privacy. As an important by-product, we obtain an upper bound on the total error incurred by the "best" pseudo-user creation-based algorithm on *any* dataset. The full version of this manuscript, which includes results from [6] and selected results in this work, can be found at [9].

The authors are with the India Urban Data Exchange Program Unit, Indian Institute of Science, Bengaluru, India, emails: {arvind.rameshwar, anshoo.tandon}@gmail.com, abhay.sharma@datakaveri.org.

II. Preliminaries

A. Notation

For a given $n \in \mathbb{N}$, the notation $[n]$ denotes the set $\{1, 2, \ldots, n\}$. We use the notation $\text{Lap}(b)$ to refer to the zero-mean Laplace distribution with standard deviation $\sqrt{2}b$ and $\mathcal{N}(\mu, \sigma^2)$ to denote the Gaussian distribution with mean μ and variance σ^2. For a random variable $X \sim \mathcal{N}(0, 1)$, we denote its complementary cumulative distribution function (c.c.d.f.) by Q, i.e., for $x \in \mathbb{R}$, $Q(x) := \Pr[X \geq x] = \int_x^\infty \frac{1}{\sqrt{2\pi}} e^{-z^2/2} \mathrm{d}z$.

B. Problem Setup

The ITMS dataset that we consider stores records of the data provided by IoT sensors deployed in an Indian city, containing, among other information, the license plate of the bus, the location at which the data was recorded, a timestamp, and the actual data value itself, which is the instantaneous speed of the bus. We process the dataset to extract the data records corresponding to that hexagonal "grid" in the city and that timeslot which has the highest traffic.

C. Problem Formulation

Let L be the number of users (or distinct license plates) and for every user $\ell \in [L]$, let the number of records contributed by the user be m_ℓ. We set $m^\star := \max_{\ell \in [L]} m_\ell$ and $m_\star := \min_{\ell \in [L]} m_\ell$. We assume that L and the collection $\{m_\ell : \ell \in [L]\}$ are known to the client. Now, let the collection $\left\{S_j^{(\ell)} : \ell \in [L], j \in [m_\ell]\right\}$ denote the speed values present in the records chosen. We assume that each $S_j^{(\ell)} \in (0, U]$. For the real-world ITMS dataset that we work with, the speed samples are drawn according to some unknown distribution P that is potentially non-i.i.d. (independent and identically distributed) across samples and users. However, when we generate synthetic datasets, we draw each speed value $S_j^{(\ell)}$ i.i.d. according to some distribution P_s that is obtained by analyzing the statistics of the ITMS data.

Call the dataset consisting of the speed records of users as $\mathcal{D} = \left\{\left(u_\ell, S_j^{(\ell)}\right) : \ell \in [L], j \in [m_\ell]\right\}$, where the collection $\{u_\ell : \ell \in [L]\}$ denotes the set of users.

The function that we are interested in computing is the sample average $f(\mathcal{D}) := \frac{1}{\sum_{\ell=1}^{L} m_\ell} \cdot \sum_{\ell=1}^{L} \sum_{j=1}^{m_\ell} S_j^{(\ell)}$.

D. User-Level Differential Privacy

Consider datasets $\mathcal{D}_1 = \left\{\left(u_\ell, x_j^{(\ell)}\right) : \ell \in [L], j \in [m_\ell]\right\}$ and $\mathcal{D}_2 = \left\{\left(u_\ell, \overline{x}_j^{(\ell)}\right) : \ell \in [L], j \in [m_\ell]\right\}$ consisting of the same users, with each user contributing the same number of (potentially different) data values $\left\{x_j^{(\ell)}\right\}$. Let D denote a universal set of such datasets. We say that $\mathcal{D}_1$ and $\mathcal{D}_2$ are "user-level neighbours" if there exists $\ell_0 \in [L]$ such that $\left(x_1^{(\ell_0)}, \ldots, x_{m_{\ell_0}}^{(\ell_0)}\right) \neq \left(\overline{x}_1^{(\ell_0)}, \ldots, \overline{x}_{m_{\ell_0}}^{(\ell_0)}\right)$, with $\left(x_1^{(\ell)}, \ldots, x_{m_\ell}^{(\ell)}\right) = \left(\overline{x}_1^{(\ell)}, \ldots, \overline{x}_{m_\ell}^{(\ell)}\right)$, for all $\ell \neq \ell_0$.

Definition II.1. For a fixed $\epsilon > 0$, a mechanism $M : \mathsf{D} \to \mathbb{R}$ is said to be user-level ϵ-DP if for every pair of user-level neighbours $\mathcal{D}_1, \mathcal{D}_2$ and for every measurable subset $Y \subseteq \mathbb{R}$, we have that $\Pr[M(\mathcal{D}_1) \in Y] \leq e^\epsilon \Pr[M(\mathcal{D}_2) \in Y]$.

Definition II.2. Given a function $g : \mathsf{D} \to \mathbb{R}$, we define its user-level sensitivity Δ_g as $\Delta_g := \max_{\mathcal{D}_1, \mathcal{D}_2 \text{ u-l nbrs.}} |g(\mathcal{D}_1) - g(\mathcal{D}_2)|$, where the maximization is over datasets that are user-level neighbours.

For example, the user-level sensitivity of f is $\Delta_f = \frac{U m^\star}{\sum_\ell m_\ell}$. We use the terms "sensitivity" and "user-level sensitivity" interchangeably. The next result is well-known [5, Prop. 1]:

Theorem II.1. *For any $g : \mathsf{D} \to \mathbb{R}$, the mechanism $M_g^{Lap} : \mathsf{D} \to \mathbb{R}$ defined by $M_g^{Lap}(\mathcal{D}_1) = g(\mathcal{D}_1) + Z$, where $Z \sim Lap(\Delta_g/\epsilon)$ is user-level ϵ-DP.*

III. Outline of Algorithms Used

For the datasets we work with, we reindex the users so that $m_1 \geq m_2 \geq \ldots \geq m_L$. We first briefly recall the algorithms introduced in [6]: Baseline, Array-Averaging, Levy, and Quantile, for private mean estimation with user-level privacy. These algorithms are based on a strategy for creating *pseudo-users* (which we also call "arrays"), called BestFit, described in Appendix A. Each algorithm then modifies the function f to be estimated, to attempt to reduce the noise added, in one of two ways: (i) the number of samples contributed by any user ℓ is clipped to $\min\{m_\ell, m_{\text{UB}}\}$, where $m_\star \leq m_{\text{UB}} \leq m^\star$ depends on $\{m_\ell\}_{\ell \geq 1}$, and (ii) the speed samples are clipped to lie in a high-probability interval.

A. Baseline

The Baseline algorithm computes $M_{\text{Baseline}}(\mathcal{D}) = f(\mathcal{D}) + \text{Lap}(\Delta_f/\epsilon)$, where $\Delta_f = \frac{U m^\star}{\sum_\ell m_\ell}$ is the sensitivity of the function f. Clearly, a large amount of noise needs to be added for privacy when either U or $m^\star$ is large, thereby increasing the error in estimation.

B. Array-Averaging

For a fixed m_{UB}, consider the application of BestFit on the dataset $\mathcal{D}$. In particular, we set $f_{\text{arr}}(\mathcal{D}) := \frac{1}{\overline{K}} \cdot \sum_{i=1}^{\overline{K}} \overline{A_i}$, with $\overline{A_i} := \frac{1}{w(A_i)} \sum_{j=1}^{w(A_i)} A_i(j)$ being the mean of the samples contributed by array $i \in [\overline{K}]$; here, $w(A)$ denotes the number of filled locations in array A. Note that the sensitivity in this case is $\Delta_{f_{\text{arr}}} = \frac{U}{\overline{K}}$.

The Array-Averaging algorithm with BestFit grouping computes $M_{\text{ArrayAvg}}(\mathcal{D}) = f_{\text{arr}}(\mathcal{D}) + \text{Lap}(\Delta_{f_{\text{arr}}}/\epsilon)$. Clearly, both these algorithms are ϵ-DP, from Theorem II.1.

We now define $\tilde{\Delta}_{f_{\text{arr}}} := \frac{U m_{\text{UB}}}{\sum_{\ell=1}^{L} \min\{m_\ell, m_{\text{UB}}\}}$ as a proxy for an upper bound on $\Delta_{f_{\text{arr}}}$ (see Appendix A for a simple lower bound on $\overline{K}$). If $\tilde{\Delta}_{f_{\text{arr}}}$ is small, then so is $\Delta_{f_{\text{arr}}}$.

The following lemma, whose proof is in [9], then holds:

Lemma III.1. *For any $m_{UB} \leq m^\star$, we have that $\Delta_f \geq \tilde{\Delta}_{f_{arr}}$.*

Next, we shall embark on choosing a "good" m_{UB}, which provides a large "gain" $\frac{\Delta_f}{\tilde{\Delta}_{f_{\text{arr}}}}$. To this end, call $\alpha := \frac{m^\star}{m_{\text{UB}}}$; in our setting, we have $\alpha \geq 1$. For fixed $\{m_\ell\}_{\ell \geq 1}$,

$$\frac{\Delta_f}{\tilde{\Delta}_{f_{\text{arr}}}} = \frac{\alpha}{\sum_\ell m_\ell} \cdot \sum_\ell \min\{m_\ell, m_{\text{UB}}\} \leq \frac{\alpha}{\sum_\ell m_\ell} \cdot \min\left\{\sum_\ell m_\ell, m^\star L/\alpha\right\} = \min\left\{\alpha, \frac{m^\star L}{\sum_\ell m_\ell}\right\}. \quad (1)$$

In the above, the equality $\frac{\Delta_f}{\tilde{\Delta}_{f_{\text{arr}}}} = \alpha$ holds only if $m_{\text{UB}} = m^\star$; note that in this case $\alpha = \frac{\Delta_f}{\tilde{\Delta}_{f_{\text{arr}}}}$ in fact equals 1. On the other hand, the equality $\frac{\Delta_f}{\tilde{\Delta}_{f_{\text{arr}}}} = \frac{m^\star L}{\sum_\ell m_\ell} \geq 1$ holds only if $m_{\text{UB}} = m_\star$. We hence designate $\mathsf{OPT} := \frac{m^\star L}{\sum_\ell m_\ell}$.

While the choice $m_{\text{UB}} = m_\star$ results in a high gain, it could potentially cause poor accuracy in estimation of the true value $f(\mathcal{D})$, since each array contains only very few samples. In Appendix B, we show that choosing m_{UB} to be the sample median results in only a small loss in gain, compared to OPT.

In our experiments, we hence employ $m_{\text{UB}} = \text{med}(m_1, \ldots, m_L)$ for Array-Averaging. In Section VI, we introduce a novel method of choosing m_{UB} to jointly optimize the errors due to clipping and noise.

C. Levy

Levy puts together Array-Averaging and Algorithm 1 in [7]. Broadly speaking, Levy first obtains the array means $\overline{A}_i$, $1 \leq i \leq \overline{K}$, using a fixed m_{UB}. It then clips these array means to lie in a high probability interval, which is privately estimated with privacy loss $\epsilon/2$. The algorithm then adds a suitable amount to Laplace noise to the clipped array means to guarantee ϵ-DP overall. Detailed descriptions of Levy and the sensitivity $\Delta_{f_{\text{Levy}}}$ are provided in Appendix C.

D. Quantile

The Quantile algorithm sets m_{UB} to be the same value $m_{\text{UB}}^{(L)}$ chosen by Levy (see Appendix C for details on the choice of $m_{\text{UB}}^{(L)}$) and creates arrays (or pseudo-users) containing the speed samples. As in Levy, the Quantile algorithm then clips the array means to lie in a high-probability interval, which is chosen differently from Levy; the interval, as before, is privately estimated with probability $\epsilon/2$. More details on the Quantile algorithm and the sensitivity $\Delta_{f_{\text{Quantile}}}$ are given in Appendix D.

IV. Results

A. Setup

In [6], we evaluated our algorithms on two types of datasets: 1) a real-world ITMS dataset C containing non-i.i.d. speed values and 2) a synthetic dataset $\mathcal{D}$ containing i.i.d. samples drawn using insights from the ITMS data. We ran each private mean estimation algorithm 10^4 times and tested the accuracy of our algorithms for privacy loss $\epsilon \in [0.5, 2]$.

We used the mean absolute error (or MAE) metric, $E_{\text{MAE}}(C) = 10^{-4} \cdot \sum_{i=0}^{10^4} \left| M^{(i)}(C) - \mu(C) \right|$, to evaluate the performance of the algorithms. Here, for $i \in [10^4]$, $M^{(i)}(C)$ is the result of running each algorithm M in Section III on C at iteration i, and $\mu := \mu(C)$ is the true sample mean. Since, for $Z \sim \text{Lap}(b)$, we have $\mathbb{E}[|Z|] = b$, we simply used $E_{\text{MAE}}(C) = \Delta_f/\varepsilon$ for Baseline.

B. Experimental Results

We refer the reader to Appendix E for details on the ITMS dataset C and for plots of the performance of our algorithms on the ITMS data, which show that Baseline performs uniformly poorer than all other algorithms.

Next, we artificially generated[1] a large dataset $\mathcal{D}$ using insights from the ITMS dataset. Let the number of users in $\mathcal{D}$ be $\widehat{L}$ and let the number of samples per user be $\{\widehat{m}_\ell\}_{\ell \in [\widehat{L}]}$. We generated i.i.d. speed samples $\left\{ \widehat{S}_j^{(\ell)} : \ell \in [\widehat{L}], j \in [\widehat{m_\ell}] \right\}$, where for any ℓ, j, we have $\widehat{S}_j^{(\ell)} \sim \Pi_{[0,U]}(X)$, where $X \sim \mathcal{N}(\mu, \sigma^2)$. Here, $\sigma^2 = \sigma^2(C) = \frac{1}{\sum_{\ell=1}^{L} m_\ell} \cdot \sum_{\ell=1}^{L} \sum_{j=1}^{m_\ell} \left(S_j^{(\ell)} - \mu \right)^2$ is the true variance of the ITMS dataset.

We consider two settings, for a fixed positive integer λ.

1) **Sample scaling**: Here, we set $\widehat{L} = L$ and $\widehat{m}_\ell = \lambda \cdot m_\ell$, for all $\ell \in [\widehat{L}]$. In this case, $\widehat{m}^\star := \max_{\ell \in [\widehat{L}]} \widehat{m}_\ell = \lambda \cdot m^\star$.
2) **User scaling**: Here, we set $\widehat{L} = \lambda \cdot L$ and for each $i \in [\lambda]$, we let $\widehat{m}_{\lambda \cdot (\ell-1)+i} = m_\ell$, for all $\ell \in [L]$. Here, $\widehat{m}^\star = m^\star$.

Plots for the performance of our algorithms under **sample scaling** and **user scaling** can be found in Appendix E, with λ set to 10. Figure 2a there for **sample scaling** shows that Baseline performs worse than the other algorithms, as expected. However, interestingly, the Levy algorithm (with $\gamma = 0.2$) outperforms than all other algorithms.

Figure 2b in Appendix E for **user scaling** shows that the FixedQuantile subroutine outperforms all the other algorithms. Furthermore, Array-Averaging performs second-best. We reiterate that we used $m_{\text{UB}} = m_{\text{UB}}^{(L)}$ (see (7)) for Array-Averaging as well, to maintain uniformity.

In the next section, we provide theoretical justification for the performance trends we observe on large datasets.

V. Justification of Performance Trends On Synthetic Datasets

Let $\widehat{m}_{\text{UB}}$ denote the new lengths of the arrays used for the synthetic dataset $\mathcal{D}$, with $\widehat{\overline{K}}$ denoting the resultant number of arrays under the BestFit grouping strategy. Further, for each algorithm, we denote its sensitivity under sample scaling (resp. user scaling) by using an additional superscript '(s)' (resp. '(u)') over the existing notation.

A. Sample Scaling

We first consider the setting of **sample scaling**. The following simple lemma holds.

Lemma V.1. *Under sample scaling, we have that*

$$\widehat{m}_{UB} = \lambda \cdot m_{UB}, \ \widehat{K} = K, \ \text{and} \ \widehat{\overline{K}} = \overline{K},$$

for m_{UB} chosen either as the sample median or as in (7) *in Appendix C.*

The proof of the proposition below follows directly from the definitions of the sensitivities and from Lemma V.1.

Proposition V.1. *We have that*

$$\Delta^{(s)}_{f_{Baseline}} = \Delta_{f_{Baseline}}, \ \Delta^{(s)}_{f_{arr,\,wrap}} = \Delta_{f_{arr,\,wrap}}, \ \Delta^{(s)}_{f_{arr,\,best}} = \Delta_{f_{arr,\,best}},$$

and $\Delta^{(s)}_{f_{Levy}} = \frac{1}{\sqrt{\lambda}} \cdot \Delta_{f_{Levy}}$.

From the above proposition, we obtain that the amount of Laplace noise added for privacy is much smaller for Levy as compared to Baseline and Array-Averaging, for large enough λ. We believe the relative better performance of Levy is due the additional errors in private estimation of the sample quantiles in Quantile.

B. User Scaling

Next, consider the setting of **user scaling**. The following lemma holds, analogous to Lemma V.1.

Lemma V.2. *Under user scaling, we have that*

$$\widehat{m}_{UB} = m_{UB}, \ \widehat{K} = \lambda \cdot K, \ \text{and} \ \widehat{\overline{K}} = \lambda \cdot \overline{K},$$

for m_{UB} chosen as in (7).

Next, we denote the standard deviation of noise added in each algorithm described in Section III, under user scaling, we use the same notation as for the sensitivities under user scaling, but replacing Δ with σ. The following proposition holds, assuming that the estimation error in ε-DependentQuantile for the sample quantiles due to privacy is zero.

Proposition V.2. *For any fixed ε, for large enough λ, we have that with high probability,*

$$\sigma^{(u)}_{f_{arr,\,best}} < \min \left\{ \sigma^{(u)}_{f_{Levy}}, \sigma^{(u)}_{f_{Quantile}} \right\},$$

under the ε-DependentQuantile subroutine for Quantile*, if the exact sample quantiles are employed.*

The proof of Proposition V.2 is in Appendix F. By arguments similar to those above, we observe that since the FixedQuantile subroutine eliminates a much larger number of data samples in the computation of $[a', b']$, with high probability, we will have that $b' - a'$ is amall, and hence the amount of noise added for FixedQuantile is lower than that for Array-Averaging, resulting in better performance.

[1] An interesting direction for future study is the ϵ-differentially private generation of synthetic data that preserves some statistics of the base dataset (see [10] and references therein).

VI. The OPT-Array-Averaging Algorithm

In this section, we describe the novel OPT-Array-Averaging algorithm, which carefully chooses m_{UB} in the Array-Averaging algorithm, in order to jointly optimize the *worst-case* errors due to clipping and privacy. In the process, we obtain an upper bound on the "best" total estimation error for a pseudo-user creation-based algorithm on *any* dataset. We emphasize, though, that this choice of m_{UB} is optimal in a worst-case setting, or in other words, *minimax* optimal, and not necessarily in a typical (or *average-case*) setting. We mention that a general minimax theory for balancing errors due to estimation and privacy was left open in [11] (see also [12]).

In what follows, we assume that the arrays A_i, $i \in [\overline{K}]$ are fully filled and use $\overline{K} = \frac{\sum_{\ell=1}^{L} \min\{m_\ell, m_{\mathrm{UB}}\}}{m_{\mathrm{UB}}}$ in all subsequent equations. For ease of reading, we set $m = m_{\mathrm{UB}}$.

For a given dataset $\mathcal{D}'$, we set $E_1(\mathcal{D}', m) := |f_{\mathrm{arr}}(\mathcal{D}') - f(\mathcal{D}')|$ to denote the error due to clipping the number of samples per user, and

$$E_2(m) := \sqrt{2}\tilde{\Delta}_{f_{\mathrm{arr}}}/\epsilon = \frac{\sqrt{2} \cdot Um}{\epsilon \cdot \sum_{\ell=1}^{L} \min\{m_\ell, m\}} \tag{2}$$

to denote the standard deviation of Laplace noise added by Array-Averaging to guarantee ϵ-DP. Let the overall error due to clipping and the noise added due to privacy be $E(\mathcal{D}', m) := E_1(\mathcal{D}', m) + E_2(m)$. In what follows, our intention is to minimize the maximum (or worst-case) error E over all datasets, i.e., we seek to solve for

$$m_{\mathrm{UB}}^{(O,1)} = \arg \min_{m_\star \le m \le m^\star} \max_{\mathcal{D}'} E(\mathcal{D}', m). \tag{3}$$

The next lemma, which is proved in Appendix G, exactly characterizes the "worst-case" error $E_1(\mathcal{D}', m)$, over all datasets $\mathcal{D}'$. For $\ell \in [L]$, let $\Gamma_\ell := \min\{m_\ell, m\}$.

Lemma VI.1. *We have that*

$$\max_{\mathcal{D}'} E_1(\mathcal{D}') = U \cdot \left(1 - \frac{\sum_\ell \Gamma_\ell}{\sum_\ell m_\ell}\right).$$

Given Lemma VI.1 as above, our objective reduces to minimizing, over $m_\star \le m \le m^\star$, the function

$$E(m) := U \cdot \left(1 - \frac{\sum_\ell \Gamma_\ell}{\sum_\ell m_\ell}\right) + \frac{\sqrt{2} \cdot Um}{\epsilon \cdot \sum_{\ell=1}^{L} \Gamma_\ell}. \tag{4}$$

We show in Appendix H that the function $E_2(m)$ in (2) is unfortunately non-convex in m, for most $\{m_\ell\}$ values of interest. Hence, analytically solving the optimization problem in (**??**) is difficult, and one has to resort to numerical methods.

Moreover, since

$$\begin{aligned} \min_{m_\star \le m \le m^\star} \max_{\mathcal{D}'} E(\mathcal{D}', m) &\ge \max_{\mathcal{D}'} \min_{m_\star \le m \le m^\star} E(\mathcal{D}', m) \\ &\ge \min_{m_\star \le m \le m^\star} E(\overline{\mathcal{D}}, m), \end{aligned}$$

for *any* dataset $\overline{D}$, we obtain an upper bound on the total estimation error of the "best" pseudo-user based algorithm for $\overline{D}$.

In what follows, we present a simpler optimization problem, which replaces $E_2(m)$ by the "worst-case inverse gain" $\tilde{\Delta}_{f_{\mathrm{arr}}}/\Delta_f$ (see (1)), which we call $E_2'(m)$, where

$$E_2'(m) := \max\left\{\frac{m}{m^\star}, \frac{\overline{m}}{m^\star}\right\},$$

where $\overline{m} := \frac{\sum_\ell m_\ell}{L}$. Recall that the worst-case inverse gain is a proxy for the error incurred due to the privacy requirement in Array-Averaging, in comparison with the error incurred by Baseline (which in turn is independent of m).

In our new optimization problem, we minimize the sum of the worst-case (over datasets) error E_1/U and the worst-case inverse gain, over admissible values m. More precisely, our optimization problem is as follows

$$\begin{aligned} &\text{minimize } \overline{E}(m) := 1 - \frac{\sum_\ell \Gamma_\ell}{\sum_\ell m_\ell} + \max\left\{\frac{m}{m^\star}, \frac{\overline{m}}{m^\star}\right\} \\ &\text{subject to } m_\star \le m \le m^\star. \end{aligned} \tag{5}$$

It can easily be argued that $\overline{E}(m)$ is convex in m and hence (4) is a convex optimization problem with linear constraints. We first state a simple lemma, which characterizes the stationary points of a subderivative of $\overline{E}$. Let $q := \frac{\sum_\ell m_\ell}{m^\star}$ and recall that $m_1 \ge \ldots \ge m_L$. The proof of the lemma is in Appendix I.

Lemma VI.2. *We have that* $\left.\frac{\mathrm{d}\overline{E}}{\mathrm{d}m}\right|_{m=\tilde{m}} = 0$*, iff the following conditions hold:*

1) *q is an integer,*
2) *$m_q \ge \overline{m}$, and*
3) *$\tilde{m} = m_q$.*

There is hence a simple procedure, based on the necessity of the KKT conditions (see [13, Sec. 5.5.3]), for obtaining the minimizer in (4), which we call $m_{\mathrm{UB}}^{(O,2)}$.

(i) If q and m_q obey the conditions in Lemma VI.2, set $m_{\mathrm{UB}}^{(O,2)} = m_q$.

(ii) Else, we have that either $m_{\mathrm{UB}}^{(O,2)} = m_\star$ or $= m^\star$ (i.e., $m_{\mathrm{UB}}^{(O,2)}$ is a boundary point). Pick $m_{\mathrm{UB}}^{(O,2)} = \arg \min_{m \in \{m_\star, m^\star\}} \overline{E}(m)$.

Remark. We have that $\overline{E}(m^\star) = 1$ and $\overline{E}(m_\star) = U \cdot \left(1 - \frac{\sum_\ell \Gamma_\ell}{\sum_\ell m_\ell}\right) + \frac{\overline{m}}{m^\star}$. Hence, without additional information on the distribution of $\{m_\ell\}_{\ell \ge 1}$, it is not possible to comment on which of $\overline{E}(m_\star)$ or $\overline{E}(m^\star)$ is smaller, in Step (ii) above.

VII. Conclusion

In this paper, we provided theoretical justification for the algorithms we proposed in [6] for the private release of sample means of real-world spatio-temporal IoT datasets, with particular focus on traffic datasets with speed data samples. We then devised a novel algorithm based on the creation of pseudo-users, which jointly optimizes the worst-case errors due to clipping and additive noise for user-level privacy. As a by-product, we obtained upper bounds on the lowest achievable error of any pseudo-user creation-based algorithm.

REFERENCES

[1] A. Narayanan and V. Shmatikov, "Robust de-anonymization of large sparse datasets," in *2008 IEEE Symposium on Security and Privacy (sp 2008)*, 2008, pp. 111–125.

[2] L. Sweeney, "Weaving technology and policy together to maintain confidentiality," *Journal of Law, Medicine & Ethics*, vol. 25, no. 2–3, p. 98–110, 1997.

[3] C. Whong. (2014) Foiling nyc's taxi trip data. [Online]. Available: https://chriswhong.com/open-data/foil_nyc_taxi/

[4] V. Pandurangan. (2014) On taxis and rainbow tables: Lessons for researchers and governments from nyc's improperly anonymized taxi logs. [Online]. Available: https://blogs.lse.ac.uk/impactofsocialsciences/2014/07/16/nyc-improperly-anonymized-taxi-logs-pandurangan/

[5] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," *Theory of Cryptography*, p. 265–284, 2006.

[6] P. Gupta, V. A. Rameshwar, A. Tandon, and N. Chakraborty, "Mean estimation with user-level privacy for spatio-temporal iot datasets," *Submitted to the IEEE International Conference on Signal Processing and Communications (SPCOM)*, 2024.

[7] D. A. N. Levy, Z. Sun, K. Amin, S. Kale, A. Kulesza, M. Mohri, and A. T. Suresh, "Learning with user-level privacy," in *Advances in Neural Information Processing Systems*, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., 2021. [Online]. Available: https://openreview.net/forum?id=G1jmxFOtY_

[8] A. J. George, L. Ramesh, A. Vikram Singh, and H. Tyagi, "Continual Mean Estimation Under User-Level Privacy," *arXiv e-prints*, p. arXiv:2212.09980, Dec. 2022.

[9] V. Arvind Rameshwar, A. Tandon, P. Gupta, N. Chakraborty, and A. Sharma, "Mean Estimation with User-Level Privacy for Spatio-Temporal IoT Datasets," *arXiv e-prints*, p. arXiv:2401.15906, Jan. 2024.

[10] M. Boedihardjo, T. Strohmer, and R. Vershynin, "Privacy of synthetic data: A statistical framework," *IEEE Transactions on Information Theory*, vol. 69, no. 1, pp. 520–527, 2023.

[11] L. Wasserman and S. Zhou, "A statistical framework for differential privacy," *Journal of the American Statistical Association*, vol. 105, no. 489, pp. 375–389, 2010. [Online]. Available: http://www.jstor.org/stable/29747034

[12] J. C. Duchi, M. J. Wainwright, and M. I. Jordan, "Minimax optimal procedures for locally private estimation," *Journal of the American Statistical Association*, vol. 113, pp. 182 – 201, 2016. [Online]. Available: https://api.semanticscholar.org/CorpusID:15762329

[13] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge: Cambridge University Press, 2004.

[14] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Foundations and Trends® in Theoretical Computer Science*, vol. 9, no. 3–4, pp. 211–407, 2014. [Online]. Available: http://dx.doi.org/10.1561/0400000042

[15] R. Vershynin, *High-Dimensional Probability: An Introduction with Applications in Data Science*, ser. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2018.

[16] E. Pauwels, "Statistics and optimization in high dimensions," Lecture notes. [Online]. Available: https://www.math.univ-toulouse.fr/~epauwels/M2RI/session1.pdf

[17] A. D. Smith, "Privacy-preserving statistical estimation with optimal convergence rates," in *STOC '11: Proceedings of the forty-third annual ACM symposium on Theory of computing*, 2011. [Online]. Available: https://dl.acm.org/doi/10.1145/1993636.1993743

[18] K. Amin, A. Kulesza, A. Munoz, and S. Vassilvtiskii, "Bounding user contributions: A bias-variance trade-off in differential privacy," in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 09–15 Jun 2019, pp. 263–271. [Online]. Available: https://proceedings.mlr.press/v97/amin19a.html

Appendix A
The BestFit Strategy for Creation of Pseudo-Users

Let m_{UB} be given. We then initialize L empty arrays $A_1, \ldots, A_L$, each of length m_{UB}.

Now, we process each user in turn, beginning with user 1 and populate his/her samples in the arrays, with a maximum of m_{UB} samples from any user being populated in the arrays. For every user $\ell \in [L]$, the BestFit grouping strategy finds the least-indexed array that can accommodate m_ℓ samples and is filled the most. The m_ℓ samples from the user ℓ are then placed in array A contiguously and the process is iterated. The algorithm then returns the non-empty arrays. Let the number of arrays created by populating the users' samples contiguously, starting from the first unoccupied location be

$$K := \left\lfloor \frac{\sum_{\ell=1}^{L} \min\{m_\ell, m_{\text{UB}}\}}{m_{\text{UB}}} \right\rfloor \tag{6}$$

Let the number of non-empty arrays using BestFit be $\overline{K} \geq K$. The key advantage of the BestFit strategy is that a user's samples can lie in only one array A_i, $1 \leq i \leq \overline{K}$.

Appendix B
Performance of Array-Averaging Using the Sample Median

Given a collection of real numbers $(x_1, \ldots, x_n)$, we define its median $\text{med}(x_1, \ldots, x_n)$ as any value x such that $|\{i \in [n] : x_i \geq x\}| = \lceil n/2 \rceil$. Now, we state and prove a simple lemma that shows that choosing $m_{\text{UB}} = \text{med}(m_1, \ldots, m_L)$ (which ensures lower error due to clipping than the choice $m_{\text{UB}} = m_\star$), results in a fairly large "gain".

Lemma B.1. *The choice $m_{UB} = med(m_1, \ldots, m_L)$ results in $\frac{\Delta_f}{\tilde{\Delta}_{farr}} \geq \frac{\mathsf{OPT}}{2}$.*

Proof. For this choice of m_{UB}, setting $\alpha = \frac{m^\star}{m_{\text{UB}}}$ and $B := \{\ell \in [L] : m_\ell < m_{\text{UB}}\}$, we have that

$$\begin{aligned}\frac{\Delta_f}{\tilde{\Delta}_{f_{\text{arr}}}} &= \frac{\alpha}{\sum_\ell m_\ell} \cdot \sum_\ell \min\{m_\ell, m_{\text{UB}}\} \\ &= \frac{\alpha}{\sum_\ell m_\ell} \cdot \left[\sum_{\ell \in B} m_\ell + \frac{m^\star \lceil L/2 \rceil}{\alpha} \right] \geq \frac{\mathsf{OPT}}{2}.\end{aligned}$$

□

Appendix C
Detailed Description of Levy

Let $A_1, \ldots, A_{\overline{K}}$ be the arrays obtained using the BestFit grouping strategy with a certain value of m_{UB} (to be specified). We now clip the range of speed values. As mentioned earlier, the Levy algorithm first privately estimates an interval $[a, b]$ where the speed values lie with high probability, with privacy loss set to $\epsilon/2$.

We then define the function $f_{\text{Levy}}(\mathcal{D}) := \frac{1}{K} \cdot \sum_{i=1}^{\overline{K}} \Pi_{[a,b]}(\overline{A_i})$, where $\Pi_{[a,b]}(x) = \min\{b, \max\{a, x\}\}$, for any $x \in \mathbb{R}$. Note that now the sensitivity is $\Delta_{f_{\text{Levy}}} = \frac{b-a}{\overline{K}}$. Levy then computes $M_{\text{Levy}}(\mathcal{D}) = f_{\text{Levy}}(\mathcal{D}) + \text{Lap}(2\Delta_{f_{\text{Levy}}}/\epsilon)$. Note that in the above expression, the privacy loss is assumed to be $\epsilon/2$. Overall, the privacy loss for both private interval estimation and for private mean estimation is ϵ, following [14, Corollary 3.15], and hence Levy is ϵ-DP.

Observe that when $b - a < U/2$, the standard deviation of the noise added in this case is less than that added using Array-Averaging with BestFit grouping. We now explain the heuristics we employ to select the parameters a, b in the Levy algorithm and m_{UB}.

1) Private Interval Estimation: The subroutine we use in this algorithm to privately compute the "high-probability" interval $[a, b]$ is borrowed from Algorithm 6 in [7], which crucially relies on the following concentration property of the data values provided to the algorithm:

Definition C.1. A random sequence X^n supported on $[0, M]$ is (τ, γ)-concentrated (τ is called the "concentration radius") if there exists $x_0 \in [0, M]$ such that with probability at least $1 - \gamma$, $\max_{i \in [n]} |X_i - x_0| \leq \tau$.

In what follows, we set $\gamma = 0.2$. Although the samples in each array are drawn in a potentially non-i.i.d. fashion from an unknown distribution, we simply rely on heuristics that assume that the data samples are i.i.d. and that each array is fully filled, with $\overline{K} = K := \left\lfloor \frac{\sum_{\ell=1}^{L} \min\{m_\ell, m_{\text{UB}}\}}{m_{\text{UB}}} \right\rfloor$.

By an application of Hoeffding's inequality (see, e.g., [15, Theorem 2.2.6]), we have that each array mean $\overline{A}_i$, $i \in [K]$, is $\frac{U^2}{4m_{\text{UB}}}$-subGaussian (see, e.g., [16, Theorem 2.1.1]). Hence, if the speed samples were i.i.d., we obtain (see, e.g., [16, Theorem 2.2.1]) that the sequence $(\overline{A}_1, \ldots, \overline{A}_K)$ is in fact (τ, γ)-concentrated about the expected value, where

$$\tau = U \cdot \sqrt{\frac{\log(2K/\gamma)}{2m_{\text{UB}}}}. \tag{7}$$

We use this value of τ to compute the "high-probability" interval $[a, b]$, with privacy loss $\epsilon/2$. More precisely, we divide the interval $(0, U]$ into U/τ disjoint bins, each of width τ, and use these bins to compute an estimate $\hat{\mu}$ of the median of the data samples as in Algorithm 6 of [7]. We then set $a = \hat{\mu} - \frac{3\tau}{2}$ and $b = \hat{\mu} + \frac{3\tau}{2}$.

2) Choosing m_{UB}: The subroutine we use to choose the length of the arrays $m_{\text{UB}} = m_{\text{UB}}^{(L)}$ is tailored to the sensitivity of the function f_{Levy}. For a fixed $m = m_{\text{UB}}$, note that $\Delta_{f_{\text{Levy}}}(m) = \frac{3\tau}{K} = \frac{3U}{K}\sqrt{\frac{\log(2K/\gamma)}{2m}}$. To reduce the sensitivity, our heuristic aims to maximize the $K\sqrt{m}$ term in $\Delta_{f_{\text{Levy}}}(m)$, and sets

$$m_{\text{UB}}^{(L)} = \text{argmax}_{m_\star \leq m \leq m^\star} \frac{\sum_{\ell=1}^{L} \min\{m_\ell, m\}}{\sqrt{m}}, \tag{8}$$

and numerically solves this optimization problem.

Appendix D
Detailed Description of Quantile

The Quantile algorithm first sets m_{UB} to be that value $m_{\text{UB}}^{(L)}$ obtained by solving the optimization problem in (7). Let $\overline{A}_1, \ldots, \overline{A}_{\overline{K}}$ be the arrays (or pseudo-users) obtained using the BestFit strategy.

Next, QUANTILE estimates a "high-probability" interval $[a', b']$ (different from the one used in LEVY) with privacy loss $\epsilon/2$. We then define $f_{\text{Quantile}}(\mathcal{D}) := \frac{1}{K} \cdot \sum_{i=1}^{K} \Pi_{[a',b']}(\overline{A}_i)$, with the sensitivity $\Delta_{f_{\text{Quantile}}} = \frac{b'-a'}{K}$, The QUANTILE algorithm then computes $M_{\text{Quantile}}(\mathcal{D}) = f_{\text{Quantile}}(\mathcal{D}) + \text{Lap}(2\Delta_{f_{\text{Quantile}}}/\epsilon)$, where the privacy loss for mean estimation is set to $\epsilon/2$. We describe two subroutines that we use to privately estimate $[a', b']$.

1) FIXEDQUANTILE*:* This subroutine privately estimates the $\left(\frac{1}{10}, \frac{9}{10}\right)$-interquantile interval of the array means $\overline{A}_1, \ldots, \overline{A}_{\overline{K}}$, using Algorithm 2 in [17]. Note that this algorithm computes estimates of a' and b' separately, and we set the privacy loss of each of these computations to be $\epsilon/4$, so that the overall privacy loss for quantile estimation is $\epsilon/2$.

2) ε-DEPENDENTQUANTILE*:* This subroutine privately estimates (with privacy loss $\epsilon/2$) the interval $[a', b']$, which is now chosen to minimize the sum of the absolute estimation errors due to privacy and due to clipping the speed values, following the work in [18]. In particular, an argument similar to that in Section 3 in [18] advocates that in order to minimize this sum of absolute estimation errors, we need to set b to be the $\lceil \frac{2}{\epsilon} \rceil^{\text{th}}$-largest value among $(\overline{A}_1, \ldots, \overline{A}_{\overline{K}})$, and by symmetry, we need to set a to be the $\lfloor \frac{2}{\epsilon} \rfloor^{\text{th}}$-smallest value among $(\overline{A}_1, \ldots, \overline{A}_{\overline{K}})$. We then privately estimate the $\left(\frac{1}{\overline{K}} \cdot \lceil \frac{2}{\epsilon} \rceil, 1 - \frac{1}{\overline{K}} \cdot \lceil \frac{2}{\epsilon} \rceil\right)$-interquantile interval using Algorithm 2 in [17]. Again, we set the privacy loss of computing a' and b' to be individually $\epsilon/4$, so that the overall privacy loss for private quantile estimation is $\epsilon/2$.

APPENDIX E
NUMERICAL RESULTS

For our ITMS dataset C, we have $m^* = 417$, $\sum_\ell m_\ell = 17166$, and $U = 65$ km/hr. Further, $\mu = \mu(C) = 20.66769$, $\sigma^2(C) = 115.135$, and $\text{med}(m_1, \ldots, m_L) = 46$. Also, for $m_{\text{UB}} = m_{\text{UB}}^{(L)}$ (see (7)), the number of arrays $\overline{K}$ is 164. Figure 1, for the non-i.i.d. ITMS dataset, shows that clipping results in much better performance than the naïve BASELINE approach. However, there is very little difference between the performance of the other algorithms, possibly owing to the relatively small size of the ITMS dataset.

Figures 2a and 2b show the performance of our algorithms on the large artificial dataset $\mathcal{D}$ generated via **sample scaling** and **user scaling**, respectively. We mention that in Figure 2b, we use $m_{\text{UB}} = m_{\text{UB}}^{(L)}$ (see (7)) for ARRAY-AVERAGING as well, to maintain uniformity.

APPENDIX F
PROOF OF PROPOSITION V.2

Recall that for $X \sim \text{Lap}(b)$, the standard deviation of X is $\sqrt{2}b$. We now prove Proposition V.2.

Proof. First consider ARRAY-AVERAGING under user scaling. We recall first that $\Delta_{f_{\text{arr, best}}}^{(u)} = \frac{U}{\widehat{\overline{K}}} = \frac{U}{\lambda \cdot \overline{K}}$, where the last equality follows from Lemma V.2. Hence, we have that $\sigma_{f_{\text{arr, best}}}^{(u)} = \frac{\sqrt{2}U}{\lambda \overline{K} \varepsilon}$.

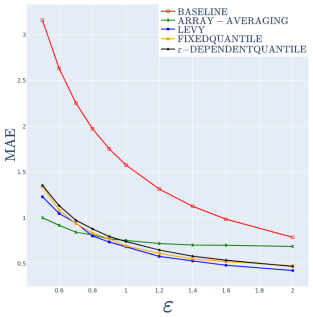
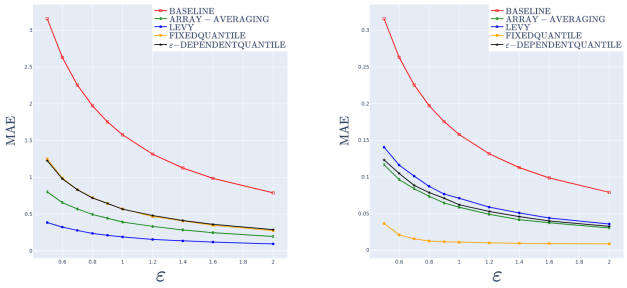


Fig. 1: Plots comparing the performance of algorithms on real-world ITMS data



(a) (b)

Fig. 2: (a) Plots comparing the performance of algorithms under **sample scaling** (b) Plots comparing the performance of algorithms under **user scaling**

Now, consider LEVY under user scaling. From the expression for $\Delta_{f_{\text{Levy}}}$, we have that

$$\sigma_{f_{\text{Levy}}}^{(u)} = \min\left\{\frac{6\sqrt{2} \cdot \tau}{\widehat{\overline{K}} \cdot \varepsilon}, \frac{2\sqrt{2} \cdot U}{\widehat{\overline{K}} \cdot \varepsilon}\right\} = \frac{2\sqrt{2} \cdot U}{\lambda \overline{K} \varepsilon},$$

for $\lambda \geq \frac{\gamma}{2\overline{K}} \cdot e^{2m_{\text{UB}}/9}$ (see (6)). Here, in the last equality above, we once again invoke Lemma V.2 to argue that

$$\widehat{\tau} = U \cdot \sqrt{\frac{\log(2\lambda \overline{K}/\gamma)}{2m_{\text{UB}}}}.$$

Hence, for $\lambda \geq \frac{\gamma}{2\overline{K}} \cdot e^{m_{\text{UB}}/18}$, we have that $\sigma_{f_{\text{arr, best}}}^{(u)} < \sigma_{f_{\text{Levy}}}^{(u)}$.

Now consider the ε-DEPENDENTQUANTILE under user scaling. Fix some $\delta > 0$ such that $\delta < U/4$. We claim that for large enough λ, the true $\left(\widehat{\overline{K}}^{-1} \cdot \lceil \frac{2}{\varepsilon} \rceil, 1 - \widehat{\overline{K}}^{-1} \cdot \lceil \frac{2}{\varepsilon} \rceil\right)$-interquantile interval of the samples contains $[\delta, U - \delta]$, with high probability. To this end, let $t := \lceil \frac{2}{\varepsilon} \rceil$ and let B denote the event that $\left(\widehat{\overline{K}}^{-1} \cdot t, 1 - \widehat{\overline{K}}^{-1} \cdot t\right)$-interquantile interval is contained in $[\delta, U - \delta]$. We shall now show that $\Pr[B]$ decays to zero as λ increases to infinity. Observe that

$$\Pr[B] \leq \Pr\left[\exists\ \widehat{\overline{K}} - 2t \text{ array means all lying in } [\delta, U - \delta]\right]. \quad (9)$$

Now, let B_i denote the event that $\widehat{\widehat{A}}_i$, for $i \in [\widehat{\widehat{K}}]$ lies in $[\delta, U-\delta]$, and let $i^\star$ denote that array index i that maximizes $\Pr[B_i]$. Note that

$$\Pr[B_{i^\star}] \leq 1 - \max\left\{\Pr[\widehat{\widehat{A}}_{i^\star} = 0] + \Pr[\widehat{\widehat{A}}_{i^\star} = U]\right\}.$$

Further, we have that

$$\begin{aligned}\Pr[\widehat{\widehat{A}}_{i^\star} = 0] &= \left(\Pr[\widehat{S}_1^{(1)} = 0]\right)^{w(\widehat{A}_{i^\star})} \\ &= (1 - Q(\mu/\sigma))^{w(\widehat{A}_{i^\star})}.\end{aligned}$$

where we use the fact that the samples $\widehat{S}_j^{(\ell)}$, $\ell \in [\widehat{L}]$, $j \in [\widehat{m}_\ell]$ are drawn i.i.d. according to the "projected" Gaussian distribution described earlier. Since $w(\widehat{A}_{i^\star}) > 0$, we get that $\Pr[\widehat{\widehat{A}}_{i^\star} = 0] > 0$ and hence that $\Pr[B_{i^\star}] < 1$.

Employing a union bound argument to upper bound the probability in (9), we get that

$$\begin{aligned}\Pr[B] &\leq \binom{\widehat{\widehat{K}}}{\widehat{\widehat{K}} - 2t} \cdot \prod_{i=1}^{\widehat{\widehat{K}}} \Pr[B_i] \\ &\leq \binom{\widehat{\widehat{K}}}{\widehat{\widehat{K}} - 2t} \cdot (\Pr[B_{i^\star}])^{\widehat{\widehat{K}}} = \binom{\widehat{\widehat{K}}}{2t} \cdot (\Pr[B_{i^\star}])^{\widehat{\widehat{K}}}\end{aligned}$$

where the second inequality holds by the definition of $i^\star$. Since $\binom{\widehat{\widehat{K}}}{2t}$ grows polynomially in λ, for a fixed t (see Lemma V.2), we have that for any fixed $\delta > 0$, and for a fixed, small $\beta \in (0,1)$, there exists λ_0, such that for all $\lambda \geq \lambda_0$, we have that $\Pr[B] < \beta$.

Therefore, for $\lambda \geq \lambda_0$, we have that with probability at least $1 - \beta$,

$$\sigma^{(u)}_{f_{\text{Quantile}}} \geq \frac{2(U-2\delta)}{\widehat{\widehat{K}} \cdot \varepsilon} > \frac{U}{\widehat{\widehat{K}} \cdot \varepsilon} = \sigma^{(u)}_{f_{\text{arr, best}}},$$

where the second inequality holds since $\delta < U/4$.

Hence, by picking $\lambda \geq \max\left\{\lambda_0, \frac{\gamma}{2\overline{\widehat{K}}} \cdot e^{2m_{\text{UB}}/9}\right\}$, we get $\sigma^{(u)}_{f_{\text{arr, best}}} < \min\left\{\sigma^{(u)}_{f_{\text{Levy}}}, \sigma^{(u)}_{f_{\text{Quantile}}}\right\}$, with probability at least $1 - \beta$. □

Appendix G
Proof of Lemma V.1

Proof. First, recall that $m_1 \geq m_2 \geq \ldots m_L$; let $m_{\ell^\star}$ denote the smallest index ℓ such that $m_\ell < m$. We use the notation $\Gamma_\ell := \min\{m_\ell, m\}$. Then,

$$\begin{aligned}&E_1(\mathcal{D}') \\ &= \left|\frac{1}{\overline{K}m}\sum_{\ell<\ell^\star}\sum_{j=1}^{m} S_j^{(\ell)} + \frac{1}{\overline{K}m}\sum_{\ell\geq\ell^\star}\sum_{j=1}^{m} S_j^{(\ell)}\right. \\ &\qquad \left. - \left(\frac{1}{\sum_\ell m_\ell}\sum_{\ell<\ell^\star}\sum_{j=1}^{m_\ell} S_j^{(\ell)} + \frac{1}{\sum_\ell m_\ell}\sum_{\ell\geq\ell^\star}\sum_{j=1}^{m_\ell} S_j^{(\ell)}\right)\right| \\ &= \left|\left(\frac{1}{\overline{K}m} - \frac{1}{\sum_\ell m_\ell}\right)\cdot\sum_{\ell=1}^{L}\sum_{j=1}^{\Gamma_\ell} S_j^{(\ell)} - \frac{1}{\sum_\ell m_\ell}\sum_{\ell<\ell^\star}\sum_{j>m} S_j^{(\ell)}\right|.\end{aligned}$$

Now, since each of the two terms in the maximization above is non-negative, with $S_j^{(\ell)} \in (0, U]$, for all $\ell \in [L]$ and $j \in [m_\ell]$, we get that

$$\begin{aligned}&\max_{\mathcal{D}'} E_1(\mathcal{D}') \\ &= \max\left\{\left(\frac{U}{\overline{K}m} - \frac{U}{\sum_\ell m_\ell}\right)\sum_{\ell=1}^{L}\Gamma_\ell, \frac{U}{\sum_\ell m_\ell}\sum_{\ell<\ell^\star}(m_\ell - m)^+\right\}.\end{aligned} \tag{10}$$

In (9), the first expression on the right is attained when $S_j^{(\ell)} = U$, for all $\ell \in [L]$ and $j \in [\Gamma_\ell]$, and $S_j^{(\ell)} = 0$, otherwise. Analogously, the second expression on the right is attained when $S_j^{(\ell)} = U$, for all $\ell < \ell^\star$ and $j > m$, and $S_j^{(\ell)} = 0$, otherwise. We use the notation $(c)^+$ to denote $\max\{0, c\}$, for $c \in \mathbb{R}$. Next, observe that $\sum_\ell (m_\ell - m)^+ = \sum_\ell m_\ell - \sum_\ell \Gamma_\ell$. Plugging this into (9), we obtain that

$$\begin{aligned}\max_{\mathcal{D}}{}' E_1(\mathcal{D}') &= U \cdot \max\left\{\left(\frac{1}{\overline{K}m} - \frac{1}{\sum_\ell m_\ell}\right)\sum_{\ell=1}^{L}\Gamma_\ell, 1 - \frac{\sum_\ell \Gamma_\ell}{\sum_\ell m_\ell}\right\} \\ &= U \cdot \left(1 - \frac{\sum_\ell \Gamma_\ell}{\sum_\ell m_\ell}\right),\end{aligned}$$

where in the last equality we use the fact that $\overline{K}m = \sum_\ell \Gamma_\ell$. □

Appendix H
On the Non-Convexity of E_2

Recall from (2) that

$$E_2(m) = \sqrt{2}\tilde{\Delta}_{f_{\text{arr}}}/\epsilon = \frac{\sqrt{2}\cdot Um}{\epsilon \cdot \sum_{\ell=1}^{L}\min\{m_\ell, m\}}.$$

In this section, we shall show that for most distributions of the number of speed samples $\{m_\ell\}_{\ell=1}^{L}$, the function E_2, or equivalently, the function

$$\eta(m) := \frac{m}{\sum_{\ell=1}^{L}\min\{m_\ell, m\}},$$

is non-convex in m.

Indeed, consider the setting where there exists some $k \in [L]$ such that $m_k - m_{k+1} > 2$ (recall that $m_1 \geq \ldots \geq m_L$). In what follows, we show that the function $\eta(m)$ is in fact *concave*, for $m \in (m_{k+1}, m_k)$, when the argument m is treated as a real number. In particular, this then implies that $\eta(m)$ is non-convex when m takes an integer value in the range (m_{k+1}, m_k).

Lemma H.1. *We have that $\eta(m)$ is concave in m, for $m \in (m_{k+1}, m_k)$.*

Proof. Observe that for $m \in (m_{k+1}, m_k)$, we have that

$$\eta(m) = \frac{m}{c + km},$$

where $c := \sum_{\ell=k+1}^{L} m_\ell$. For this range of m values, hence, $\frac{\mathrm{d}^2\eta}{\mathrm{d}m^2} = -2ck \cdot (c + km)^{-3}$. Since $c, k > 0$, we obtain that $\frac{\mathrm{d}^2\eta}{\mathrm{d}m^2} < 0$, implying the concavity of η for the given range of m values. □

Furthermore, observe from Lemma VI.1 that $\max_{\mathcal{D}'} E_1(\mathcal{D}')$ is *convex* in m, for all integer values of m. Combining this with the lemma above, we have that $E(m)$ as in (**??**) is non-convex in m in general, and hence minimizing $E(m)$ over m analytically is hard.

Appendix I
Proof of Lemma VI.2

Proof. First, we take the subderivative of $\overline{E}$ (see also [18]), to obtain

$$\frac{\mathrm{d}\overline{E}}{\mathrm{d}m} = \frac{-|\{\ell : m_\ell \geq m\}|}{\sum_\ell m_\ell} + \frac{1}{m^\star} \cdot \mathbb{1}\{m \geq \overline{m}\}.$$

Hence, we have that $\left.\frac{\mathrm{d}\overline{E}}{\mathrm{d}m}\right|_{m=\tilde{m}} = 0$ only if $|\{\ell : m_\ell \geq m\}| = \frac{\sum_\ell m_\ell}{m^\star}$; in other words, $\tilde{m}$ is a stationary point if and only if the conditions stated in the lemma hold. □