Data Anonymization via Differential Privacy

(Ver 1.0)

Contents

I. Introduction 2

I A. Motivation for using Differential Privacy 2

I B. Features of Differential Privacy 3

I C. Strengths of Differential Privacy 4

I D. Adoption of Differential Privacy 4

I E. Basic Definitions in Differential Privacy 5

II. Dataset 6

II A. Features of the ITMS Dataset 6

II B. Pre-Processing of the Dataset 8

II C. H3 Geospatial Indexing System 9

II D. Released Queries 11

III. Experimental Results 12

III A. Choice of H3index Resolution 12

III B. Choice of Time Slot 16

III C. Selection of Subset of H3indices 17

III D. Noise Addition Mechanism 19

III E. Sensitivity and Epsilon Calculation 20

Quantifying Epsilon for Query 1 and Query 2 21

III F. Cumulative-Epsilon-Per-Day 23

IV. Pseudocode 23

IV A. Pre- Processing 23

IV B. Selection of Subset of H3 indices 24

IV C. Query 1 25

IV D. Query 2 26

V. Conclusion 27

I. Introduction

Differential privacy provides a strong mathematical definition of privacy that protects us against the threats of unknown attacks and cumulative loss. With differential privacy, statements about risk are proved mathematically--rather than supported heuristically or empirically. Differential privacy ensures that the ability of an adversary to inflict harm (or good, for that matter)—of any sort, to any set of people—should be essentially the same, independent of whether any individual opts into, or opts out of, the dataset. Therefore, inferring information specific to an individual from the outcome of a differentially private release is extremely hard, including whether the individual's information was used at all.

Differential privacy describes a promise, made by a data holder, or curator, to a data subject: "You will not be affected, adversely or otherwise, by allowing your data to be used in any study or analysis, no matter what other studies, data sets, or information sources, is available." (Ref: C. Dwork, "*The promise of differential privacy: A tutorial on algorithmic techniques*," in Proc. IEEE 52nd Annual Symposium on Foundations of Computer Science, Oct. 2011).

I A. Motivation for using Differential Privacy

The main motivation for the development and adoption of differential privacy is that the traditional approaches to data privacy were vulnerable to linkage attacks where data released by a data provider could be linked or combined with other existing data sources, as well as hard-to-anticipate future source of auxiliary information, that could be used to re-identify the attributes that were previously assumed to be private. Note that traditional approaches to data privacy include (i) data suppression (omission of partial or complete data records), (ii)

generalisation (replacing a given value with a range of values), (iii) swapping (shuffling certain attributes among a set of users), (iv) aggregation (averaging user attributes before data release).

There are many real-world examples of data releases that were thought to be sufficiently protective of privacy but were later shown to carry significant privacy risks. One such example is the reconstruction and re-identification of the 2010 US Census data. In 2018, researchers revealed that the underlying confidential data from the 2010 US Decennial Census can be largely reconstructed. They were able to reconstruct with perfect accuracy the gender, age, race, ethnicity, and fine-grained geographic location (to the block-level) reported by Census respondents for 46 percent of the US population. (Ref: Garfinkel, Simson L., John M. Abowd, and Christian Martindale. 2019. "*Understanding Database Reconstruction Attacks on Public Data*." Communications of the ACM 62 (3): 46–53).

I B. Features of Differential Privacy

Differentially private mechanisms are employed in settings in which an analyst seeks to learn about a population, and not particular individuals (for example, when computing statistical estimates such as counts, averages and histograms).

To achieve differential privacy, carefully crafted random statistical noise must be added to query output. *Higher noise leads to better privacy protection but lowers accuracy*. Note that as the number of samples in a dataset grows, the loss in accuracy due to noise addition in a differentially private release can become much smaller.

Every release of data leaks some information about the individual records used as input regardless of the protection method used. Therefore, the privacy loss accumulates over multiple computations and must be tracked. Differential privacy provides formal methods to help manage this cumulative loss, referred to as the *privacy-loss budget*.

I C. Strengths of Differential Privacy

- Differential privacy is robust against arbitrary external/auxiliary information.
- Differential privacy provides provable bounds with respect to the cumulative risk from multiple data releases and is the only existing privacy approach to do so. These privacy bounds are commonly known as *composition theorems*.
- Differential Privacy is robust to post-processing – any manipulation or transformation of a differentially private release cannot decrease privacy.
- Differential privacy provides transparent tools and does not rely on security-by-obscurity. This enables public scrutiny of the technique where calibrated noise is added to trade some accuracy with privacy.

I D. Adoption of Differential Privacy

Organisations that have adopted differential privacy in practice include:

- US Census Bureau: used DP for its 2020 Census report
- Google: uses DP for sharing historical traffic statistics
- Apple: uses DP for its intelligent personal assistant technology
- Microsoft: uses DP for telemetry in Windows
- LinkedIn: uses DP for answering advertiser queries
- Facebook: uses DP for releasing user datasets to academics and agencies

However, there are still certain challenges that must be overcome:

- There is no industry consensus on the acceptable value of the privacy loss budget.
- Privacy loss accumulates with each data release, so continuous release of information remains an issue.

Researchers are glad that most big-tech companies now acknowledge the need to provide privacy using DP, and hope that companies will implement common guidelines in future.

I E. Basic Definitions in Differential Privacy

A ***query*** is a function to be applied to a database.

A ***privacy mechanism*** is an algorithm that takes a database and a set of queries as input and produces an output string that produces relatively accurate answers to the queries.

Neighbouring Datasets are two datasets that differ in just one *element* $x_i \rightarrow x_i'$. With a temporal dataset, this *element* can correspond to a single user record at a given time instant (in case of event-level privacy), or all the records of a single user for a single day (in case of privacy per user per day), or all the records of a single user over all the days (in case of user-level privacy).

Differential Privacy a property of a protocol/mechanism A which you run on some dataset X producing some output A(X). Mechanism A is ε differentially private if for any two neighbouring datasets X, X', for all outcomes v, we have

$$e^{-\varepsilon} \leq \Pr(A(X)=v) / \Pr(A(X')=v) \leq e^{\varepsilon}$$

The parameter ε corresponds to the **privacy loss** due to release of the privacy mechanism output. Smaller values of epsilon (ε) imply lower privacy loss while higher values of epsilon imply enhanced privacy loss.

Composition Theorem: If a mechanism A_1 operating on dataset X provides ε_1 differential privacy and mechanism A_2 operating on the same dataset provides ε_2 differential privacy,

then the combined mechanism A defined as $A(X) = (A_1(X), A_2(X))$ provides $(\varepsilon_1 + \varepsilon_2)$ differential privacy.

II. Dataset

II A. Features of the ITMS Dataset

We consider the intelligent transport management system (ITMS) dataset for our case study involving differentially private release of statistics. This dataset consists of records generated from transit buses operating in a city. The parameters reported by the buses include licence plate number, last stop ID visited, instantaneous speed, delay incurred with respect to a reference timetable, coordinates (latitude and longitude) etc.

Our goal is to release useful statistics about this dataset in such a way that the privacy of individual bus records is not compromised. We adopt the notion of differential privacy for this task, which is currently a widely accepted notion of privacy underlying several well-known privacy-preserving data releases and frameworks (e.g., Microsoft's telemetry data collection, Google's CoViD mobility reports, US Census Bureau reports etc.). At a high level, differential privacy provides plausible deniability to users whose data is being used to compute the statistics in the sense that the presence or absence of any user in the dataset does not change the statistic by much.

The following are the columns in the dataset and what it signifies.

Column Name	Column Description
trip_direction	It has the values UP and DN, to signify Up and Down Movement across the map
last_stop_id	This signifies the last Stop ID of the bus
speed	The instantaneous speed of the bus
trip_delay	This can be positive and negative in seconds and shows how much the vehicle deviates from the planned one.
last_stop_arrival_time	Time at which the bus was at last_stop_id
actual_trip_start_time	Date and Time Trip should have started
observationDateTime	Date and Time at which the trip started
trip_id	Unique ID for the ongoing trip
id	Identifies the city
Vehicle_label	The label/route number of the bus
license_plate	The license_plate of the bus

Column Name	Column Description
location.coordinates	Instantaneous Location of the bus in the format [Longitude,Latitude]

Table: List of columns in the dataset

II B. Pre-Processing of the Dataset

We obtained an ITMS dataset with 19672924 records, spanning from November 15th, 2021, to December 25th, 2021. Towards pre-processing the data, we use the H3 as a geospatial indexing system (developed by Uber) that partitions the world into hexagonal cells. A latitude and longitude pair can be transformed to a 64-bit H3 index, identifying a hexagonal cell.

We take the following steps to pre-process the data.

1. Extract Latitude and Longitude values from the location.coordinates attribute
2. Choose the H3 resolution to be 7 and assign H3 index value to each record (based on Latitude and Longitude values in the record)
3. Convert the observation Datetime string attribute into a datetime python object (to help in processing the record based on the value of the datetime object).
4. Divide a day into 24 time slots of one hour each. Create a column called Timeslot which will contain the Time Slot at which the data is recorded. For example, a time of 05:30:03 would be in the Time Slot 5 as it occurs between 5 am to 6 am. If the Time Slot is 14, it signifies that the actual observation time is in between 2pm and 3pm.

5. Select the records that belong to dates spanning from 15th November 2021 to 15th December 2021 to obtain one month data. Further, within this subset, select those records that fall in the time slots ranging from 9am to 9pm.
6. Create a column named HAT (to denote Hexagon and Time slot) which denotes a combination of the H3index and the Time Slot. For example, a record having H3 index 8742d9d69ffffff and Time Slot number 20 will have a HAT id as 20 8742d9d69ffffff.

After pre-processing, 13296107 records are obtained based on the Time Slots (ranging from 9am to 9pm) and the Dates (ranging from 15th November 2021 to 15th December 2021).

II C. H3 Geospatial Indexing System

H3 is a geospatial indexing system that partitions the world into hexagonal cells. H3 was developed to address the challenges of Uber's data science needs. The H3 geospatial indexing system is a discrete global grid system (Ref: Kevin Sahr, Denis White and A. Jon Kimerling (2003) "*Geodesic Discrete Global Grid Systems*", Cartography and Geographic Information Science, 30:2, 121-134) consisting of a multi-precision hexagonal tiling of the sphere with hierarchical indexes.

The level of the hierarchy is called resolution and can take a value from 0 till 15, where 0 is the base level with the largest and coarsest cells. A latitude and longitude pair can be transformed to a 64-bit H3 index, identifying a grid cell. The H3 index is used primarily for bucketing locations and other geospatial manipulations.

The table given below denotes the H3 resolution and the Area covered by each hexagon. The last column denotes the number of unique indexes in the *world* if split into the given number resolution.

H3 Resolution	**Average Hexagon Area (km^2)**	**Average Hexagon Edge Length (km)**	**Number of unique indexes (worldwide)**
0	4,250,546.847700	1,107.712591000	122
1	607,220.9782429	418.676005500	842
2	86,745.8540347	158.244655800	5,882
3	12,392.2648621	59.810857940	41,162
4	1,770.3235517	22.606379400	288,122
5	252.9033645	8.544408276	2,016,842
6	36.1290521	3.229482772	14,117,882
7	5.1612932	1.220629759	98,825,162
8	0.7373276	0.461354684	691,776,122
9	0.1053325	0.174375668	4,842,432,842
10	0.0150475	0.065907807	33,897,029,882
11	0.0021496	0.024910561	237,279,209,162

H3 Resolution	Average Hexagon Area (km^2)	Average Hexagon Edge Length (km)	Number of unique indexes (worldwide)
12	0.0003071	0.009415526	1,660,954,464,122
13	0.0000439	0.003559893	11,626,681,248,842
14	0.0000063	0.001348575	81,386,768,741,882
15	0.0000009	0.000509713	569,707,381,193,16

Table: Area covered by a hexagon with different H3 resolutions

II D. Released Queries

We define a HAT (Hexagon-And-Timeslot) to denote a combination of H3 index and Time Slot. The H3 index is classified using the H3 library based on the resolution defined. The Time Slot denotes a range of time interval, e.g. 1pm -2pm timeslot will contain all the records present in the one-hour time span. We release the following two queries per HAT:

1. The *first query (Query 1)* provides the average bus speed in a given HAT (where the averaging is over the monthly data). The input would be H3 index, Time Slot and the output would be the noisy value of average bus speed in that HAT.
2. The *second query (Query 2)* calculates the average number of buses per day which exceed a user specified speed limit in a HAT (where the averaging is over the monthly data). The input would be H3index, Time Slot and Speed Limit and the output would be the noisy average value of number of buses crossing the speed limit.

Differential privacy provides a formal mathematical relation between the amount of noise added to each query output and the amount of privacy loss due to release of the noisy outputs. In the next section, we discuss the trade-off between the accuracy of the released outputs and privacy loss.

III. Experimental Results

III A. Choice of H3index Resolution

An H3 index represents a hexagon in the H3 grid system at a particular resolution. In the following, we analyse and compare H3 resolutions 6,7 and 8. In our experiments, we choose the H3 resolution to be 7 and provide reasons for this selection.

H3 resolution = 8

With **resolution 8 we obtain 473 unique H3 indices** (where the dataset has ITMS records for time slots 9am to 9pm). The number of **HATs** (H3index and Time Slots) with available bus records are **4835.** Note that with 473 H3 indices, the total number of HATs (corresponding to 12 time slots between 9am to 9pm) are 473 * 12 = 5676. However, the bus records are available only for 4835 HATs (out of 5676 HATs) because the other HATs do not have any bus records in the ITMS dataset.

Now, smaller number of buses in a HAT lead to relatively higher privacy loss based on the statistics released for that HAT. For differential private release, we want to select only those HATs for which the average number of unique buses is at least 10.

For the given dataset, it is observed that 55% of the HATs have only 5 or less unique buses. The average number of unique buses per HAT is 8.21711322. The following histogram shows

number of unique buses per HAT per day with H3 resolution = 8 (where the X-axis denotes the number of buses).

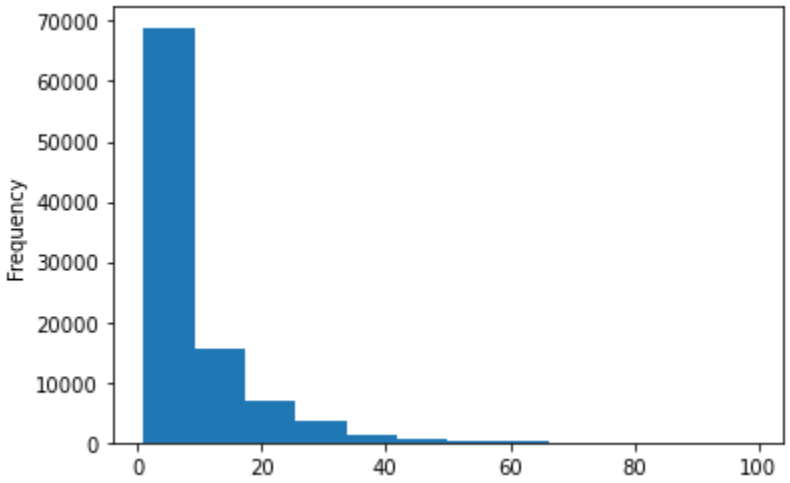

Fig: Histogram for the number of unique buses per HAT per day with H3 resolution = 8

As most of the HATs have relatively small number of buses, we are interested in increasing the hexagon size by reducing the H3 resolution to 7. Note that a larger hexagon size leads to a greater number of buses per hexagon, and thus helps to lower the privacy loss due to release of statistics per HAT.

H3 resolution = 7

With H3 resolution = 7, the number of H3 indices is 117, and the number of HATs across the dataset is 1213. Here, the average number of unique buses per HAT increases to 13.30. The following histogram shows number of unique buses per HAT per day with resolution = 7

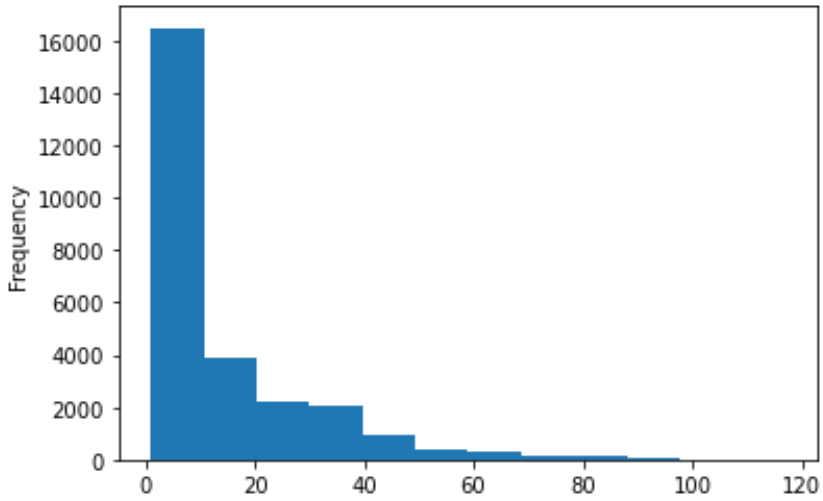

Fig: Histogram for the number of unique buses per HAT per day with H3 resolution = 7

Resolution = 6

The number of H3 indices is only 31 and total possible HATs are 317.The average number of unique buses per HAT increases to 26.62. The following histogram shows number of unique buses per HAT per day with resolution = 6.

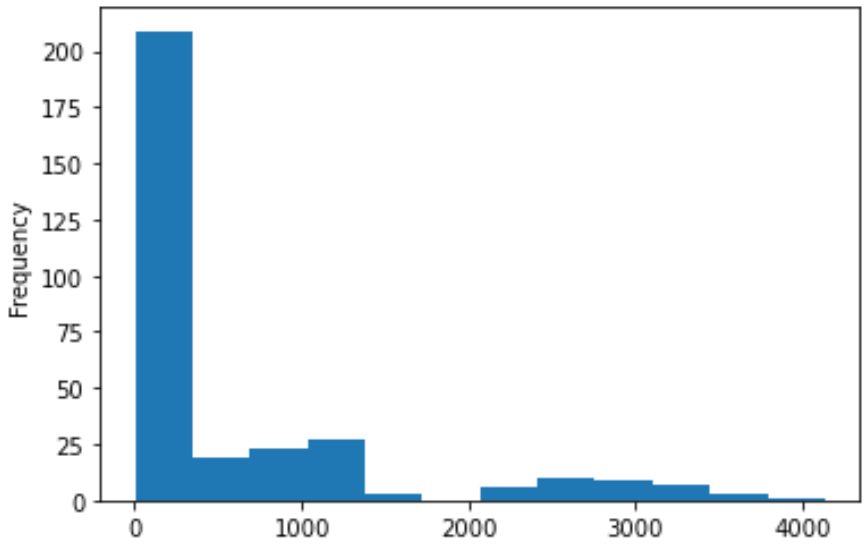

Fig: Histogram for the number of unique buses per HAT per day with H3 resolution = 6

The following is a table summarising the results of the comparative analysis:

Description	Resolution 8	Resolution 7	Resolution 6
Average Size of Each Hexagon	0.7373276 km square	5.1612932 km square	36.1290521 km square
Total Number of H3 indices in Dataset	473	117	33
Total Number of HAT in Dataset	4835	1213	317
Average number of unique buses per HAT per day	8.2	13.3	26.62
Percentage of H3 index with 10 or more unique buses per day per hat for each time slot	11.6%	26%	30%

Table: Comparison of different H3 resolutions

Remarks

- Increase in hexagon area (by lowering the H3 resolution) helps us lower the privacy loss (when releasing statistics per HAT using differential privacy).

- However, if the hexagon area size is too large, the fine-grained information about specific hot-spot areas (with a slow-moving traffic) may get lost.
- The reason why we do not choose resolution 6 is because area specificity is not maintained. Each H3 index corresponds to an area of roughly 36 Km Square. This large area results in a very coarse output, where the fine-grained traffic information about specific hot-spot area may be lost.
- The reason why we do not choose resolution 8 is because only 11.6% of the H3 indices meet the base condition that there should be at least 10 unique buses in each of the 12 timeslots from 9am to 9pm (when the time-slot size is chosen to be 1-hour). Hence, with resolution = 8, the released statistics will correspond to only 11.6% of the total city area. In contrast, with H3 resolution 7, the fraction of H3 indices with at least 10 buses per HAT per day is more than 26%.

Hence it is preferable to choose H3 resolution 7 over resolution 6 or 8.

III B. Choice of Time Slot

The ITMS dataset contains records from 5am to 9pm. In our work we choose a 12-hour window between 9am and 9pm. The reason why we chose the 9am to 9pm slot is because most of the buses travel during these hours. Note that small number of buses operating in a time slot imply higher privacy loss based on statistics released per HAT. Hence, to lower the privacy loss, we only select the bus records from 9am to 9pm (with relatively high number of buses).

Statistics

Total Number of Records (5am to 9pm) = 16977444

Total Number of Records Selected (9am to 9pm) = 13296107

Total Number of Records Rejected (5am to 9am) = 3681337

Statistics for the Number of Unique Buses Per HAT per Day when each time slot is for 1 hour, the selected time range is 9am to 9pm, and H3 resolution is 7:

mean	8.217113
std	9.537184
min	1.000000
25%	2.000000
50%	4.000000
75%	11.000000
max	99.000000

Remarks

- For convenience, we choose the time interval 9am - 9pm. Corresponding to this time interval we have 12 time slots, where each time slot corresponds to 1 hour.
- The python code can be adapted to select different time intervals and time slot size.

III C. Selection of Subset of H3indices

Out of the 117 possible H3 index we select a subset of H3indices with relatively large number of unique buses in each of the 12 time slots.

Condition for Selection of H3indices

The selected H3 index should have an average of more than 10 Unique Buses per day for all Time Slots from 9 am to 9 pm.

Process to Choose H3index

1. Identify all the H3 indices possible
2. Count the number of unique buses per day, per time slot and per H3 index
3. Add up the number of unique buses per time slot and per H3 index date wise
4. Set a limit on the required number of buses over the entire month (31*10 = 310), and select only those H3 indices with more than 310 buses (over the entire month) for all the 12 time slots.

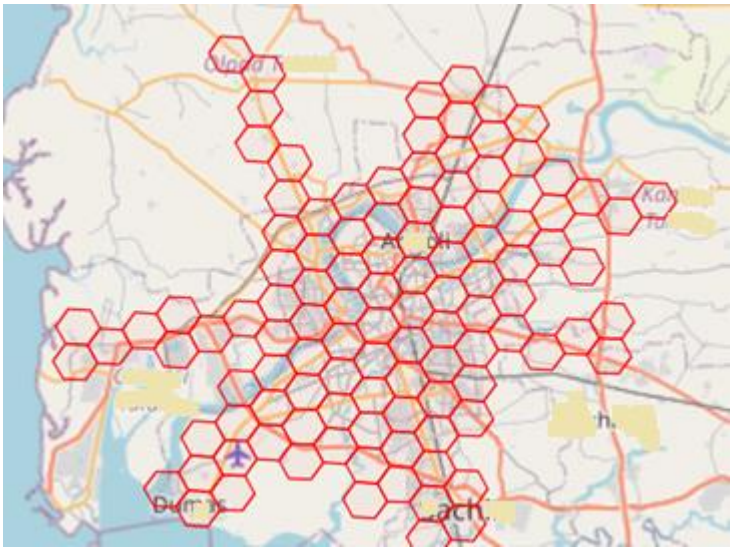

Fig: All possible hexagons covered by bus records in the dataset with H3 resolution = 7

Fig: Selected hexagons (in green) with more than 10 buses per day in all time slots

There were 30 H3 indices selected based on the selection condition: the average number of unique buses travelling in every hourly time slot from 9am to 9pm per day is more than 10. Since we selected records from 15 Nov to 15 Dec (data for 31 days), the H3 index selection condition can be equivalently stated as follows: the number of unique buses travelling in every hourly time slot from 9am to 9pm (added over all the 31 days) is more than 310. From the above figures, we notice that the selected H3 indices are in the middle of the city. This is not

unexpected as the central area of a city is expected to have relatively higher number of operating buses compared to the out-skirts of the city.

Remark: Our experimental results may be extended to a multi-resolution framework where two (or more) H3 resolutions are jointly considered. The higher H3 resolution (with smaller hexagon size) may be used to answer queries for hot-spot areas with relatively heavy bus traffic, while the lower H3 resolution (with larger hexagon size) covers areas with sparse traffic.

III D. Noise Addition Mechanism

In our work, we choose the Laplacian mechanism for achieving differential privacy. The Laplace distribution is a symmetric version of the exponential distribution. The Laplacian mechanism will simply compute the desired function f and perturb each coordinate of the output with noise drawn from the Laplace distribution. The scale of the noise will be calibrated to the [(sensitivity of f (query))/ε].

We generate Laplacian noise samples with zero mean and variance equal to $2B^2$. These noisy samples are added to each query output. We can guarantee ε differential privacy (Epsilon-per-HAT-per-Day) by selecting Laplace parameter B as B = Sensitivity/ε (Ref: C. Dwork, F. McSherry, K. Nissim, A. Smith, "*Calibrating noise to sensitivity in private data analysis*", Conference on Theory of Cryptography, 2006). Given the sensitivity value of a specific query, a user can choose the Laplacian B parameter value and then compute the value of ε (or vice versa).

Laplacian noise with parameter B (and zero mean) ensures that the absolute value of noise is less than 3B with more than 95% probability. For our implementation of the average speed

query per HAT, we recommend that value of B may be chosen in the interval [0.5, 2]. Note that lower value of B implies higher value of ε (and hence higher privacy loss). On the other hand, lower value of B implies lower standard deviation for the noisy samples, and hence relatively higher accuracy of the released outputs.

Remark: Researchers have shown the existence of privacy attacks (when Laplacian noise is added to query output for providing differential privacy) that exploit the irregularities in double-precision floating point arithmetic implementations and use the least significant bits of the released output for breaching privacy. However, such attacks can be mitigated by truncating the query output to a coarser precision level (Ref: Ilya Mironov, "*On significance of the least significant bits for differential privacy*," Proc. 2012 ACM Conf. Computer and Communications Security, pp. 650—661, Oct. 2012).

III E. Sensitivity and Epsilon Calculation

Sensitivity is defined as the maximum absolute variation in the query output if a certain number of records of a single user are replaced or omitted. Recall the privacy loss due to release of query output is captured by epsilon (ε) value. We will be calculating two types of epsilons: (i) Epsilon-per-HAT-per-Day, and (ii) Epsilon-per-Day.

Epsilon-per-HAT-per-Day – Here, the sensitivity is calculated by considering a neighbouring database that allows changes in records of any given user for a single HAT in a given day.

Epsilon-per-Day – In this case, the sensitivity is calculated by considering a neighbouring database that allows changes in records of any given user on a given day for all possible HATs.

Quantifying Epsilon for Query 1 and Query 2

Query 1: Average bus speed in a given HAT (where the averaging is over the monthly data)

Epsilon-Per-HAT-Per-Day

For the **average speed query**, the following formulas are used for computing **Epsilon-per-HAT-per-Day**:

Sensitivity = (Maximum Speed -Minimum Speed)/N

$$N = N_1 + N_2 \ldots\ldots + N_{31}$$

N_i **= Number of Unique Buses in a HAT on Day i**

EpsilonPerHATperDay = Sensitivity/B (where B is the Laplace noise parameter)

The Epsilon-per-HAT-per-Day value is inversely proportional to the value of N (see above formula). The lesser the value of N, the higher the Epsilon. Maximum speed of a bus in the ITMS Dataset is 65 kmph while the minimum speed is 0. The smallest value of N (over all HATs) is 314, and therefore the maximum value of Epsilon-per-HAT-per-Day is (65/314)/B.

Epsilon-Per-Day

An upper bound on Epsilon-Per-Day can be computed as follows:

1. Compute Epsilon-per-HAT-per-day for each HAT. Create a list of Epsilon-per-HAT-per-day values for all HATs.
2. Denote K to be the maximum number of HATs covered by a bus in a day.

3. An upper bound on Epsilon-Per-Day is computed by summing up the K largest values of Epsilon-per-HAT-per-day.

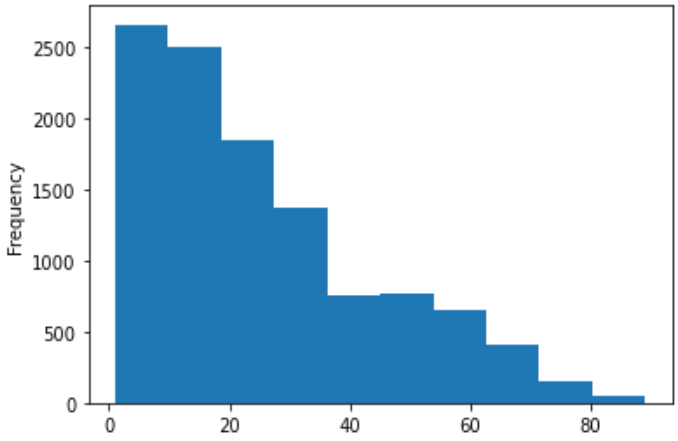

Fig: Histogram of the number of HATs covered by a bus in a day

The maximum number of HATs where a bus participates in a day is K=89. For our dataset, the sum of the worst (largest) 89 values of epsilon-per-HAT-per-day is **12.361/B**.

Query 2: Average Number of buses per day that exceed a speed threshold for a HAT (where the averaging is performed over the monthly data)

Epsilon-Per-HAT-Per-Day

For query 2 the sensitivity corresponding to Epsilon-Per-HAT-Per-Day is computed as follows:

Sensitivity = 1/ (Number of Days) = 1/31

EpsilonPerHATperDay = Sensitivity/B

Note that for Query 2, the sensitivity value is independent of the number of buses operating in a HAT and is equal to 1/31 for all HATs (where 31 is the number of days in a month). Epsilon-Per-HAT-Per-Day = (1/31)/B.

Epsilon-Per-Day

Let K denote the maximum number of HATs in which any bus operates in any given day. We compute Epsilon-Per-Day for Query 2 by multiplying K with the computed Epsilon-Per-HAT-Per-Day. Note that for query 2, the value of Epsilon-Per-HAT-Per-Day is same for all HATs. We know that 89 is the maximum number of HATs in which a bus operates in a day. Therefore, Epsilon-Per-Day is equal to 89 times (1/31)/B.

III F. Cumulative-Epsilon-Per-Day

The Cumulative-Epsilon-Per-Day is the sum of Epsilon-Per-Day values for the two queries.

Epsilon	**Query 1**	**Query 2**	**Cumulative-Epsilon**
Epsilon-Per-Day	12.361	2.871	15.232

Table: Epsilon-Per-Day values corresponding to B=1.

IV. Pseudocode

IV A. Pre- Processing

Pseudocode for steps to follow to convert raw dataset into pre-processed dataset

Input - Dataset

Output - Pre-processed Dataset with H3 index, timeslot, and HAT.

pre_process(raw dataset)

```
Extract Latitude and Longitude values from the location.coordinates

Choose the H3 resolution to be 7 and assign H3 index value to each record
based on latitude and longitude

Convert the observationDatetime string attribute into a datetime object.

Divide a day into 24 time slots of one hour each. Create a column called
TimeSlot which will contain the Time Slot at which the data is recorded.

Select the records that belong to dates spanning from 15th November 2021 to
15th December 2021 to obtain one month data.

Select the records from the above records that fall in the time slots ranging
from 9am to 9pm.

Create a column named HAT (to denote Hexagon and Time slot) which denotes a
combination of the H3 index and the Time Slot.
```

IV B. Selection of Subset of H3 indices

Choose the H3 indices which pass the following base condition:

Base Condition = Has average more than 10 Unique Buses Per Day for all Time Slots from 9 am to 9 pm

Input - Pre-processed Dataset

Output – Set of Selected H3 indices and the corresponding dataset

selected(dataset)

```
Create a dataframe that depicts the number of unique buses operating in a
given HAT (H3index and TimeSlot) for a given date

Create a dataframe that depicts the sum of the number of unique buses
operating in a given HAT over all 31 days (from 15 Nov to 15 Dec 2021)
```

Select the HATs that satisfy the following condition: the sum of the number of unique buses operating in a given HAT over all 31 days is at least 31*10 = 310

Select the subset of H3 indices for which the above condition is satisfied for all the 12 time slots from 9am to 9pm

IV C. Query 1

Query 1: Average bus speed in a given HAT (where the averaging is over the monthly data)

Input - H3 index, Time Slot, and Laplacian noise parameter B

Output - Average Speed of Bus Passing through the specific HAT plus an additive noise (where the noise is sampled from Laplace distribution with zero mean and variance $2B^2$), Computed value of Epsilon-per-HAT-per-Day for Query 1 based on the choice of HAT and B

query1(H3 index, timeslot, b)

Select records where H3 index and time slot matches the user input

For each bus operating in the chosen HAT on a given day (from 15 Nov to 15 Dec), compute the average speed of the bus on that day in the chosen HAT

Number the days from 1 to 31 corresponding to 15 Nov to 15 Dec. Denote the number of unique buses operating in the chosen HAT on day i as N_i.

Let S_i denote the average bus speed on day i in the chosen HAT (computed as the average value of the average speeds of the Ni buses operating in that HAT)

Compute the overall average bus speed in the chosen HAT (averaged over the monthly data) as $Sp = (S_1 + S_2 + \ldots + S_{31})/31$

Define $N = N_1 + N_2 + \ldots + N_{31}$

Compute Sensitivity as (Maximum Speed - Minimum Speed)/N = (65/N)

Compute Epsilon-per-HAT-per-Day as Sensitivity/B

Add Laplacian noise (with parameter B) to Sp

Return the noisy average speed

IV D. Query 2

Average number of buses per day exceeding a certain speed threshold for a given HAT

Input - H3 index, Time Slot, Choice of Laplace parameter B, speed limit

Output - Average number of buses per day exceeding the speed limit for the selected HAT (H3 index and Time Slot) with an added Laplace noise (with parameter B), Computed value of Epsilon-per-HAT-per-Day for Query 2 based on the choice of noise parameter B

query2 (H3 index, timeslot, B, speed_limit)

Select records where H3 index and time slot matches the user input

Number the days from 1 to 31 corresponding to 15 Nov to 15 Dec. Denote the number of unique buses operating in the chosen HAT on day i as N_i. Compute the maximum speed for each of the N_i buses on day i and append these computed values in the dataframe for each of the 31 days

Select records from the above dataframe (for each of the 31 days) for which the maximum bus speed crosses the threshold speed (speed_limit) given by the user. Let Mi denote the number of buses that exceed the speed limit on day i.

Let $M = (M_1 + M_2 + \ldots + M_{31})/31$ denote the average number of buses that exceed the speed limit.

Add Laplacian noise (with parameter B) to M

Calculate Epsilon-per-HAT-per-Day = 1/(31*B)

Return the noisy value of M and Epsilon-per-HAT-per-Day

V. Conclusion

For the dataset obtained by us, we observed a high variation in the number of records reported by different buses (can be attributed to different sensors fitted in different buses). Secondly, there was also a high variation in the number of records reported by the same bus over different days. Thirdly, there was high variation in the number of buses operating in a HAT for different H3index and Time-Slot values. These variations had to be considered when deciding on system parameters (such as Hexagon size, Time slot size, Selection of subset of HATs with relatively high number of operating buses, choice of noise distribution and its impact on cumulative privacy loss).

We implemented Differential Privacy by adding noise to the released queries and quantified the privacy loss due to the release of all the query outputs by computing the Cumulative-Epsilon-Per-Day value (which considered a neighbouring dataset that allowed for changes in the records of a single user (bus) for all HATs on any given day).

Differential Privacy has been widely adopted by technology giants. However, there is still no industry consensus on acceptable privacy loss. Another issue is that continuous release of statistics remains a challenge due to accumulation of privacy loss over time. However, the future of differential privacy looks optimistic, and researchers hope that companies will soon adopt common guidelines on the acceptable privacy loss.