Usability Test of Diffprivlib for Differentially Private Cookies

1st Hemant Rana
Faculty of Mathematics and Computer Science
University of Göttingen
Göttingen, Germany
hemant.rana@stud.uni-goettingen.de

***Abstract*—Users today are more informed and cautious about privacy practices but still are expected to make errors as privacy is secondary to them. Reading long texts about cookies and privacy services is a monotonous task which are often overlooked by users in order to use a service. Ignorance to such privacy aspects can result into disclosure of personal data to websites where user was unwilling to do so or websites can actively track user activity for targeted advertisements.**

In order to address the aforementioned problem, this paper explores a plausible use case of differential privacy for cookies, test its usability among users, discusses potential problems that can arise while testing and selecting a suitable usability mechanism for evaluating the use case. The differentially private cookies were modelled using a library called DiffPrivLib provided by IBM. In the end we compare various aspects associated with the library related to implementation and documentation observed while conducting the study with various users.

***Index Terms*—Differential Privacy, Usability Test, Cookies, DiffPrivLib, Qualitative Coding,**

I. Introduction

Differential Privacy can be described as a mechanism for public data such that it is possible to analyse the whole dataset without inferring something particular about a specific user. Several cases of privacy leaks have been witnessed in the past which can be attributed to the advent of differential privacy in the domain of anonymity and privacy. When Netflix shared their public dataset consisting of movie reviews by 500,000 of their subscribers, the user data was anonymized by assigning each user an ID but researchers [1] at UT Austin were able to correlate the reviews with real user identities by using the reviews in a publicly available IMDB dataset as the background knowledge after which they were able to infer sensitive information about users such as political preferences.

Such privacy leak can be avoided using differential privacy. Noise is added to data while preserving it's holistic integrity and correctness.

Definition 1: *A random function κ is ε-Differential Private if for two neighboring datasets D and D' which differ in atmost one record with possible outcomes S:$\forall$ S $\subseteq$ range(κ)* satisfies the following

$$Pr[\kappa(D) \in S] \leq e^{\varepsilon} \times Pr[\kappa(D') \in S] + \delta \tag{1}$$

In Equation.1 δ can be defined as the probability of privacy leak of an entity in a database. When δ = 0, the system can be referred to as purely ε-differential private [2]. A higher value of ε leads to less privacy and vice-versa. Query on a dataset can be defined as a function that maps data to vector(s) of real numbers based on some property. The influence of alteration in data on these queries can be termed as sensitivity of the differential privacy mechanism.

Definition 2: *For $q : D \rightarrow R^k$, the sensitivity of q is*

$$\Delta q = \max_{D_1, D_2} ||q(D_1) - q(D_2)||_1 \tag{2}$$

For implementation phase diffprivlib has been used, which is an open source library provided by IBM for differential privacy. diffprivlib is written in python3 and inherits underlying models and datastructures from numpy and scikit-learn [3] which makes it a preferred choice for users with knowledge of python. It is easy to install using the command in an integrated terminal or command prompt.

```
pip install diffprivlib
```

The library then can be imported in a python file using the following statement

```
import diffprivlib
```

The diffprivlib provides the following modules [3] for the sake of implementation

- `mechanisms` : a collection of differential privacy mechanisms like Gaussian, Laplace etc.
- `models` : differential private machine learning modules based on scikit-learn.
- `tools` : a collection of differential private mathematical operations like mean, standard deviation, rounding float point number, plotting histogram etc.

Apart from aforementioned functionalities the library also provides sensitivity calculation and privacy budget accounting [4] which makes it one of the most versatile library for implementing differential privacy.

Usability has been previously described multiple times in different connotations, [5] describes that usability can be quantified by decomposing effectiveness, efficiency, satisfaction and components into measurable attributes for human-computer interaction and software systems, [6] proposed heuristic evaluation which is an informal method of usability analysis where a number of users are presented with an interface design and asked to evaluate it.

Usability test can also be a cumbersome task for the use case developer due to following attributes [7]

1) `Sample Size`: Often it is misleading to consider that larger the sample size, better the results but [8] has stated that 85% of problems can be indentified with a user group of 5 people.
2) `Test Length`: Length of the test not only increases the cost and testing scope but also leads to concentration loss of the users.
3) `Environment`:Although constrained environments like laboratories can help conducting the study with desired conditions but setting them up can incur extra costs.
4) `Ignorance towards test results`: Usability tests can produce a vast amount of data, analyzing which data to consider or discard can be a significant task in itself.

Aspects like sample size and test length are quantifiable and hence can be analysed without much problem whereas tasks like analysing environment and filtering out test results are not quantifiable and thus requires method like coding in order to be analysed efficiently. Coding [9] refers to labelling a long organized text with an abbreviated summary of that text. Coding can have two approaches for analysing unquantifiable data

- `Inductive Coding`: Inferring labels from the user reviews.
- `Deductive Coding`: Labels from previous studies is used.

The study discussed in this report uses inductive approach due to following reasons:

1) Smaller sample size and sub tasks cannot make the study complex in inductive coding.
2) Since users have a similar background, hence their reviews tend to use similar vocabularies, thus generating similar labels.
3) Easier to deduce labels for inductive coding than deductive coding.

After coding phase an appropriate framework for analysing the results is required so that labels can be derived into meaningful insights.

DESMET [10] framework developed for evaluating software engineering practices defines evaluations as the following

1) Evaluating a tool's measurable effect.
2) How well the tool is suited for a specific purpose.

The aforementioned definitions can be applied to the usability test conducted in this report, where observations like time defines the quantifiable or measurable effect of the library whereas attributes like the user's reviews can be considered as the tool's appropriateness for the use case, but the framework is still not well defined as it fails to account for user's knowledge and aspects like their understanding of the environment. The aforementioned problems has been addressed in [11] which defined the five usability factors for evaluating a software engineering methodology but can equally be applied to the usability study discussed in this report. The following usability factors have been defined in accordance of the discussed usability test.

1) Understandability : how well a user is able to understand the task based on the description provided by the conductor.
2) Learnability : effort required by the user to gain knowledge to implement the task
3) Applicability : convenience of establishing the environment for the task.
4) Effectiveness/Usefulness : how useful the library is from the perspective of the use case.
5) User Satisfaction : how much the user is satisfied by using the library for the task.

II. Usability Study

A. Background

This Usability study has been conducted as a part of subject *Lab: Usable Security and Privacy* which is a module for Master in Applied computer Science. The subject is offered by Institute of Computer Science and Campus-Institute Data Science, University of Göttingen. The instructor for the course is Prof. Dr.-Ing. Delphine Reinhardt and this study is supervised by Patrick Kühtreiber.

B. Use Case

Fig. 1. represents the workflow of the use case. Several web services require creation and storage of cookies for better user experience. These cookies can store a wide array of information including location, search history, session tokens etc. While accessing websites that are enabled with cookie tracking, they can access the cookies and share the confidential data with third party sites, users may or may not be concerned where their data might be shared or the reputation of the third party, hence this can lead to privacy leaks to malicious users or companies.

Hence differential privacy can be used in this scenario, where data is stored with some added noise. Whenever access to cookies is required by a website it can access the cookie data to improve the user experience while still maintaining user privacy. For eg. An online food delivery service stores a user's precise location in longitude and latitude, adding a slight noise to both values can still provide enough information about their location to the service to display nearby restaurants without inferring the exact location.

C. Study Design

Colab, a free to use jupyter notebook service by Google has been used for conducting the study. python3 has been used for implementing the use case with libraries diffprivlib and geocoder. The sample cookie being used for this study consists of the following

- `Location Data` : stores client's current location using Latitude and Longitude.

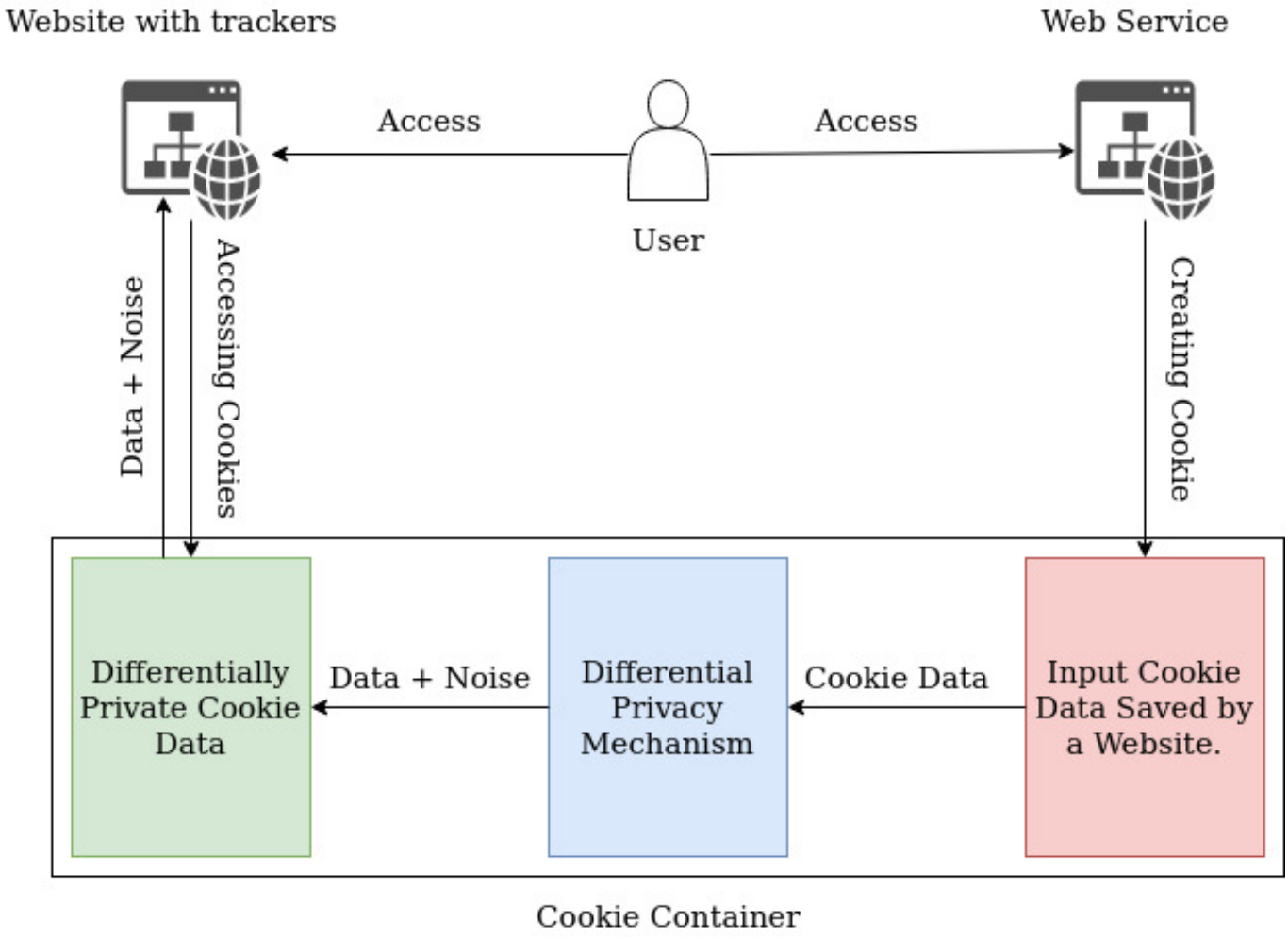

Fig. 1. Creating and Retrieving Differential Private Cookie Data

- `Video Stream Service` : stores user's search history on a video streaming service, length of viewing in seconds and the time of stream.
- `E-commerce Data` : stores user's search query and products viewed for it.

The cookie is made differentially private by following steps

1) Adding noise to longitude and latitude with a higher value of epsilon as even smaller change in location data can lead to significant change in location coordinates. Using a suitable mechanism from `diffprivlib.mechanisms` for randomising the values.
2) Randomising and aggregating the watch time of Video Stream Service data for each search query and calculating mean available in `diffprivlib.tools` module.
3) Randomising number of products viewed for each search query in E-commerce data followed by rounding it to nearest integer provided by `diffprivlib.mechanisms.transforms` module.

The user study has been designed as follows

1) Introduction
   a) Brief Introduction to the topic
      - Differential Privacy is a mechanism for public data such that it is possible to analyse the whole dataset without inferring something particular about a specific user
      - The following parameters are used to describe differential privacy
        - epsilon is the privacy budget of our mechanism and governs how much noise will be added to the data.
        - sensitivity refers to the amount of noise to be added to data to ensure it is differentially private.
   b) Library and stack
      - Diffprivlib: the library that is used for implementing all aspects related to differential privacy.
      - Geocoder has been used for fetching client's location in the form of longitude and latitude.

c) Overview of the task

Provided a cookie, the task is to make it differentially private by following the instructions provided by the conductor.

2) Usability Study

a) Questionnaire

- Questions asked before implementing the task
  - Knowledge of python programming language, do they have professional experience or have personally used it for a task?
  - Their educational background and professional experience, if any?
  - Are they familiar with differential privacy and to what extent have knowledge of the topic?
- Questions asked after finishing the task.
  - Which was the most difficult sub task from the given tasks?
  - Which was the easiest of the tasks performed?
  - Usability rating for the library on a scale of 5(1 being the worst and 5 being the best.)?
  - Would they recommend the library to colleagues or seniors who are actively working in similar fields for their project or would they use the library for any project they may be involved in?

b) Observation

- Time taken for following tasks
  - Time taken for reading the documentation throught the implementation phase.
  - Installing diffprivlib in the project
  - Importing libraries in the file
  - Initiating parameters(mechanism with suitable values)

For the aforementioned metrics users are expected to perform the task autonomously whereas for implementing code specific tasks users may seek evaluator's assistance for programming and debugging as it requires language specific knowledge which is not our primary motive for user study. Every step has been individually assessed due to following factors

- Initially users are unfamiliar with the structure of the documentation and hence may commit more mistakes and take longer for instantiating the first mechanism.
- Instantiating any mechanism after first one will be easy for users as they are already familiar with it.
- Searching for other functions may take less time as users have gained insights about the documentation but implementing them can consume different amount of time based on their knowledge of language and previous experience with reading other documentations.
- User's motivation to understand the documentation by themselves and implementing the required functions is also a crucial factor to be taken into account.

D. Study Implementation

Initially users were given a brief introduction to differential privacy and libraries being used, following which they were provided with the following sub tasks for implementing the use case:

1) Install required libraries using pip or terminal/command-prompt
2) Import required libraries
3) Instantiate `Gaussian mechanism` (m_1) with some provided values of ϵ, δ and sensitivity
4) Use m_1 mechanism to randomise latitude and longitude values
5) Instantiate `GaussianAnalytic mechanism`(m_2) with provided values of ϵ, δ and sensitivity.
6) Randomise viewing time for each search using m_2 and calculate mean for the data.
7) Instantiate `Laplace mechanism`(m_3) with provided values of ϵ and sensitivity
8) Randomise the number of products viewed for a search query in `E-commerce Data` with m_3
9) Round off the randomised value to the nearest integer using m_3 and using the required function from diffprivlib.

The time taken and reviews by users for the aforementioned tasks were noted. In which task and for how long user was stuck while implementing it was also observed. The reviews are then coded into labels and their sentiments have been deduced based on the reviews. The sentiments are then analysed to draw insights about the drawbacks of the library.

III. Analysis

The study was conducted as an interview with active interaction between user and conductor. A total of 3 users participated in the study having technological backgrounds with the age group of 22-28 years having 1-2 years of industrial experience and have pursued bachelor's degree in the field of computer science having atleast taken programming, database and object oriented programming classes. The users were completely unfamiliar with the concept of differential privacy .The analysis is based on the following aspects

- Observations : Time taken by users for implementing the sub tasks mentioned in Section II (D).
- Questionnaire: consists of assessing user's skill by interviewing them about the task and prerequisites as stated in Section II(C).

The User reviews and time taken for each tasks can be seen in Table I. Under the review section for each user the comments of that specific user has been recorded whereas inside circular brackets () additional reference to their comments has been mentioned to make it easy to interpret. The Table II consists of sentiment analysis of the users for deduced labels based on the reviews.

A. User Background

User1 has a minimal understanding of python having worked on some assignments during college with no knowl-

TABLE I
USER ASSESSMENT AND REVIEWS FOR EACH TASK

S.No.	Task	User 1		User 2		User 3	
		Time	***Review***	***Time***	***Review***	***Time***	***Review***
1	Install required libraries using pip or terminal/command-prompt	30sec	Easy to install	43sec	Easy to install	10sec	Easy to install
2	Import libraries	10sec	Easy to import	13sec	Easy to import	48sec	Easy to import
3	Instantiate `Gaussian mechanism` (m_1) with some provided values of ϵ, δ and sensitivity	2min 49sec	Documentation was a bit confusing due to * as first parameter for instantiation of the object for mechanism	6min 53sec	* as the first parameter in the initialisation seemed confusing and parameters like epsilon were required to be mentioned explicitly which is not stated in the documentation.	7min 24sec	Nomenclature/class-hierarchy was a bit confusing in the code base(differentiating between classes and subclasses)
4	Use m_1 mechanism to randomise latitude and longitude values	2min 44sec	Implementing the task took longer due to unfamiliarity with python	2min 55sec	Documentation was good for this part.	6min 43sec	Implementation took longer due to unfamiliarity with python, also initializing parameters was confusing to code
5	Instantiate `GaussianAnalytic mechanism`(m_2) with provided values of ϵ, δ and sensitivity.	30sec	Easy as have been done earlier(referring to task 3)	1min 6sec	Easy after having coded it previously(referring to task 3)	34sec	Easy
6	Randomise viewing time for each search using m_2 and calculate mean for the data.	1min	Also done earlier (referring to task 4)	7min 56sec	Familiarity with documentation required	5min 50sec	Implementation becomes easier after knowing about documentation.
7	Instantiate `Laplace mechanism`(m_3) with provided values of ϵ and sensitivity	51sec	Similar to earlier task(referring to task 3)	1min 19sec	Easy having done previously(referring to task 3)	33sec	Easier to implement having done earlier(referring to task 3).
8	Randomise the number of products viewed for a search query in `E-commerce Data` with m_3	10sec	Similar to earlier task(referring to task 4)	23sec	Easy to implement	58sec	Easy to implement
9	Round off the randomised value to the nearest integer using m_3 and a function from DiffPrivLib.	6min 26sec	Knowledge of modules required	4min 17sec	Took extensive search of documentation to find the appropriate function	6min 12sec	Documentation was not clear, assistance needed.

edge of differential privacy and having 1.5 years of experience as systems engineer.

User2 has also previously worked with python on a project in college and has no knowledge of differential privacy and has 1.5 years of industrial experience in computer science related profile.

User3 self learnt python a few years back and has no knowledge of differential privacy having 2.5 years of industrial experience with a computer science background.

B. What users have to say

For user 1 easiest of task not relevant to documentation was installing and importing the library whereas for task relevant to library it was initiating mechanisms m_2 and m_3. They had the most difficulty with implementation of rounding off integers and rate the library 4 out of 5 as it has all the required functions available but documentation could be more elaborate with code sample and examples, also module `diffprivlib.mechanisms.transforms` was a bit confusing as it requires a fair understanding of object oriented programming in python. They recommend the library if no other library with better documentation is available.

For user 2 also easiest of task not relevant to library was installing and importing it and in relevant one was initialization of mechanisms m_2 and m_3. Most difficult task was randomising with mechanism m_1. They rate the library 3.5 out of 5 and intends to recommend only in case the documentation improves.

For user 3 also the aforementioned observations are applicable with easiest tasks being installing/importing library and initialization of mechanisms m_2, m_3 for task not relevant and task relevant to library respectively. Most difficult of the task was rounding off integers. They give a usability rating of 4 out of 5 because of availability of a plethora of mechanisms

and functions but not descriptive enough. Since they didn't go through the whole documentation, hence cannot comment on whether they can recommend the library or not.

C. Assessment

The results are based on both observations and questionnaire discussed before.

Users took around an aggregate of 5-7 minutes for reading the documentation. From observation there were both advantages and disadvantages to a short documentation provided by diffprivlib. Users were able to look up for functions and classes either manually or using the search bar provided with documentation without much difficulty whereas they were sometimes confused with the implementation due to lack of code samples in the documentation. In case of implementing mechanisms, users struggled initially but after implementing first one, they were able to instantiate any mechanism with quite ease whereas for finding functions for rounding and mean, users were not able to find them efficiently due to lack of description of modules and struggled while implementing due to lack of code samples. As users were getting familiar with the documentation they were able to work their way to implement tasks more efficiently.

Based on the reviews the following labels can be considered for qualitative coding of the library

- `Installing`
- `Importing`
- `Documentation`
- `Code-Structure`
- `Python-Proficiency`
- `Parameter-Initialization`

For conducting sentiment-analysis for each of the aforementioned labels ratings are assigned based on the reviews in the following manner:

- 0 : User failed to implement the task, and requires assistance with implementation.
- 1 : User struggled to implement the task and maybe required hint(s) but was able to complete the task.
- 2 : User was able to complete the task with minor or no struggle on their own.

From Table II it can be concluded that users had a negative sentiment associated with documentation for most of the tasks implemented for the first time, parameter-initialization is another drawback of the library as users were confused while implementing it, also it cannot be overlooked that even though users were somewhat familiar with python, they struggled with some tasks due to incomprehension of the python code described in the documentation.

Following are the reviews for usability parameters stated in [11]

1) Understandability : Users were able to understand the underlying principle of differential privacy on introduction if not in depth.
2) Learnability : Users struggled somehow due to lack of knowledge of python programming language but with minor assistance they were able to carry out the tasks successfully.
3) Applicability : Initially users were provided with a jupyter notebook on google colab but for the installing phase they were able to complete the task without difficulty.
4) Effectiveness/Usefulness : Users had a positive feedback for the availability of functions required for the use case.
5) User Satisfaction : Users were not so satisfied with the library due to shortcomings of the documentation.

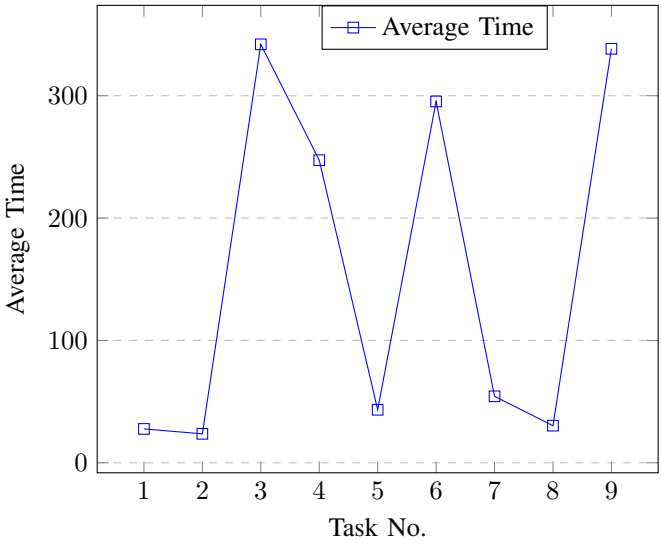

Fig. 2. Average time taken by users for each task

IV. Conclusion

Library overall has mixed reviews with positive aspect of providing multiple functions that are relevant to the task and differential privacy whereas has a negative aspect of documentation which at certain points was misleading as in the case of parameters in the initialization of mechanisms, with * as the first parameter, as it was not clearly mentioned what it denotes and whether users have to explicitly mention the parameter names(eg. epsilon) or can parameters be automatically indexed by the program. The mixed sentiment for approval of the documentation can also be accurately inferred from the inductive coding ratings given in Table II, according to which accumulative rating for the documentation is 29/36. Also based on Fig. 2. and Table II, it can be observed that after implementing a task users were able to implement another task of similar nature with quite ease, for eg. instantiating mechanisms m_2, m_3 and randomising values using it. Apart from documentation content user should have a good understanding of python to implement the required functionalities efficiently. A short synopsis of the documentation for each module might have helped users to gain insight into documentation beforehand the implementation step. From the usability parameters in Section III(C), it can be inferred that diffprivlib as a library is easy to setup and install and

TABLE II
LABELS AND THEIR RATINGS

S.No.	Task	User 1		User 2		User 3	
		Label(s)	***Rating***	***Label(s)***	***Rating***	***Label(s)***	***Rating***
1	Install required libraries using pip or terminal/command-prompt	Installing	2	Installing	2	Installing	2
2	Import libraries	Importing	2	Importing	2	Importing	2
3	Instantiate `Gaussian mechanism` (m_1) with some provided values of ϵ, δ and sensitivity	Documentation	1	Documentation	1	Code-Structure	1
4	Use m_1 mechanism to randomise latitude and longitude values	Python-Proficiency	1	Documentation	2	Python-Proficiency, Parameter-Initialization	1, 1
5	Instantiate `GaussianAnalytic mechanism`(m_2) with provided values of ϵ, δ and sensitivity.	Documentation	2	Documentation	2	Documentation	2
6	Randomise viewing time for each search using m_2 and calculate mean for the data.	Documentation	2	Documentation	1	Documentation	2
7	Instantiate `Laplace mechanism`(m_3) with provided values of ϵ and sensitivity	Documentation	2	Documentation	2	Documentation	2
8	Randomise the number of products viewed for a search query in `E-commerce Data` with m_3	Documentation	2	Documentation	2	Documentation	2
9	Round off the randomised value to the nearest integer using m_3 and a function from DiffPrivLib.	Documentation	1	Documentation	1	Documentation	0

is useful for implementing the use case but requires certain training eg. python language and requires improvements in documentation. Hence it can be stated that the library is suitable for a differential privacy project but with a scope of improvement in documentation.

REFERENCES

[1] A. Narayanan en V. Shmatikov, "How to break anonymity of the netflix prize dataset", arXiv preprint cs/0610105, 2006.

[2] M. Xie, J. Wang, en J. Chen, "A Practical Parameterized Algorithm for the Individual Haplotyping Problem MLF", 04 2008, vol 4978, bll 433–444.

[3] N. Holohan, S. Braghin, P. Mac Aonghusa, en K. Levacher, "Diffprivlib: the IBM differential privacy library", arXiv preprint arXiv:1907. 02444, 2019.

[4] G. M. Garrido, J. Near, A. Muhammad, W. He, R. Matzutt, en F. Matthes, "Do I get the privacy I need? Benchmarking utility in differential privacy libraries", arXiv preprint arXiv:2109. 10789, 2021.

[5] N. Bevan, J. Carter, en S. Harker, "ISO 9241-11 Revised: What Have We Learnt About Usability Since 1998?", in Human-Computer Interaction: Design and Evaluation, 2015, bll 143–151.

[6] J. Nielsen en R. Molich, "Heuristic Evaluation of User Interfaces", in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Seattle, Washington, USA, 1990, bll 249–256.

[7] N. Wolfsheim , Challenges of classical usability test methods and how we solve them, Centigrade, Dec. 20, 2019. Accessed on: Mar. 10, 2022. [Online]. Available: https://www.centigrade.de/en/blog/challenges-of-classical-usability-test-methods-and-how-we-solve/

[8] J. Nielsen en T. K. Landauer, "A Mathematical Model of the Finding of Usability Problems", in Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems, Amsterdam, The Netherlands, 1993, bll 206–213.

[9] M. Linneberg en S. Korsgaard, "Coding qualitative data: a synthesis guiding the novice", Qualitative Research Journal, 05 2019.

[10] B. A. Kitchenham, "Evaluating Software Engineering Methods and Tool Part 1: The Evaluation Context and Evaluation Methods", SIGSOFT Softw. Eng. Notes, vol 21, no 1, bll 11–14, Jan 1996.

[11] Z. Masood, X. Shang, en J. Yousaf, "Usability Evaluation Framework for Software Engineering Methodologies", Lecture Notes on Software Engineering, vol 2, bll 225–232, 01 2014.