ClearAudit: A Differentially Private Data Curator

Alexander Sosnkowski (yxv7fp) and Gabriel Gladstone (zwy7ce)
dept. of Computer Engineering
Charlottesville, VA

I. Abstract

We have created ClearAudit, an open source web-based framework that enables businesses to more readily work with differentially private mechanisms for data releases and machine learning tasks. Our tool empowers data publishers to better understand the risks associated with publishing raw data, statistics, or ML models trained off of their datasets before releasing it to the public by clearly visualizing the tradeoff between privacy and utility. The web app is assumed to be run within the user's secure, private network and is capable of handling a number of data statistics and ML tasks from regression to classification.

A. Introduction

Protecting data publisher's privacy is essential, both for the proprietary nature of the data and the sensitive nature of real individuals used to create it. Improper data obfuscation techniques such as data swapping have led to attacks such as the infamous 2010 US Census [3] which exposed sensitive information such as the race, age, sex, and household relationships of US Citizens.

A carefully designed auditing tool such as ClearAudit could have helped detect this problem and prevented attacks. Auditing datasets is a critical step in ensuring the protection of sensitive data and in mitigating the effects of attacks, however, many machine learning / data specialists have limited exposure to differentially private mechanisms / tools. Moreover, for a given dataset it is not always clear what form of DP mechanism will provide the best utility / privacy tradeoff. Therefore, developing a trusted, universal, and easy to use central platform built off-of open-source, community trusted frameworks is an important contribution that aids in the continued prevalence and adoption of differential privacy mechanisms.

II. Related Work

A. OpenDP

OpenDP is an open-source tool for statistical analyses [15] initially developed out of Harvard. We used OpenDP for a variety of reasons. For one, it has a large community that audit its open source implementations of differentially private mechanisms and have developed ample resources for reference. In fact, for an open source project, OpenDP has an incredibly robust vetting process which requires formal verification of a method's privacy proof via domain expert reviewers [4]. While some limitations still exist with OpenDP's implementations regarding hardness against side channel and floating point attacks [11], we are confident that it's open source nature and well managed moderation ensure the most core functionalities are stable and secure. OpenDP's API also supports Python which will integrate seamlessly with our Python based web app backend, DJango.

B. SmartNoise

Built in collaboration with OpenDP (and now integrated into OpenDP's general orginization), SmartNoise is a separate framework that implements a number of DP complaint data synthesizers for tabular data which can be used for numerous downstream data analysis and machine learning tasks. In particular, we leverage SmartNoise-synth, which provides implementations of seven robust data synthesizers that are capable of being fitted to a particular dataset and then sampled from to generate an arbitrary amount of statistically similar data [14].

C. Diffprivlib

To further augment ClearAudit with machine learning centric functionality, we look towards IBM's Differential Privacy Library which implements a number of useful differentially private ML models [9].

1) OpenDP Wizard: As a library demonstration, OpenDP Wizard is a simple graphical interface for observing the effect of various epsilon values on utility so as to aid new analysts with selecting the correct epsilon parameter [5]. While this tool is similar in concept and aim to our work, we believe our tool covers a broader spectrum of use cases (ML, full data release, ect) and available tools (pulled from libraries other than OpenDP, such as Diffprivlib). We also hope that our web app can be more readily deployed in institutions on servers with extra compute available to speed up more arduous computations (such as synthesizer model fitting) in a centralized, efficient manner rahter than locally on each host system (as would be the case with OpenDP Wizard).

D. DP Auditorium

Our idea is, in part, inspired by DP Auditorium [10] and similar tools that seek to automate and streamline privacy analysis. As differential privacy increasingly becomes an expected standard in data releases and model training by both public and private sector entities [1], it is desirable to have services which seek to automate the inclusion of differentially private

mechanisms why also providing data analysts and users meaningful metrics and visualizations that characterize the impact of these mechanisms on preventing attacks on data privacy while retaining utility. In recent years, considerable work has been done in various forms of automation. For example, [16] is capable of automatically adapting non-differentially private code into differentially private code, even providing a rough proof sketch. DP Auditorium is another tool that seeks to automatically test black-box differential privacy mechanisms to look for implementation bugs. Furthermore, for a given dataset it is capable of generating a synthetic dataset so that the two together violate privacy guarantees. These tools serve as important advancements in solving the challenges of differential privacy adoption as noted in [7] which identifies the lack of testing tools and frameworks for evaluation as a key constraint, however, these tools remain unapproachable or obtuse for many entry level data analysts and non-technical employees. Therefore, we propose to develop a centralized, accessible tool capable of intuitively demonstrating the effectiveness of various differential privacy mechanisms on arbitrary datasets through a web application so to enable non-technical users to better understand and use privacy preserving mechanisms.

III. Methodology

Our site was made to aid machine learning researchers in exploring the use of differential privacy on their datasets as shown in Fig 1. We guide users through a 5 step process to explore and configure differential privacy, test for vulnerabilities, estimate model accuracy, and release differentially private general statistics.

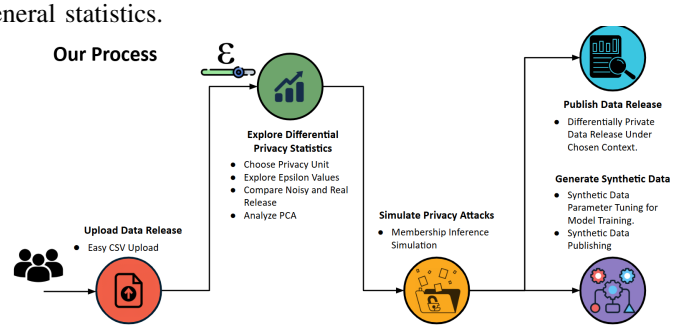

Fig. 1: ClearAudit Process

Our contribution to this project is the integration of a variety of DP tools onto one platform, an easy to use tool for exploration, and a means of one-shot experimentation for model testing. The tool is designed to accept many data sources, but we will be using the Wine dataset [2] for examples in this report. The Wine dataset consists of 13 chemical properties of wine measured from three different cultivators, and thus represents a three class, classification task.

A. Upload Data

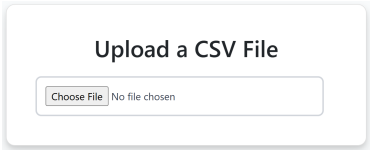

Fig. 2: Upload Screen

A user can upload the dataset they plan to analyze as a comma separated values (CSV) file via the upload component as shown in Fig 2. This is a simple upload functionality for 1 csv file, however in future works it can be extended to handle bulk data or remote data sources. While platforms like [6] allow users to update multiple files, with varying data types, and labels, for our proof of concept implementation we chose this single file approach as to provide a more streamlined pipeline that appeals to a wider audience.

B. Exploring Statistics

After uploading data, users can explore general statistics of the data, a principal component analysis, and a data preview.

The first graph shown is simply the number of records. The user specifies the label column and epsilon value and we run a simulation (n=5000) to simulate a differentially private count Fig 3. We infer the actual count from the label column and apply Laplacian noise at a scale determined from epsilon. We use the same method on the target column for sum Fig **??**.

$$scale = \frac{unique\,records\,in\,label\,column}{\epsilon}$$

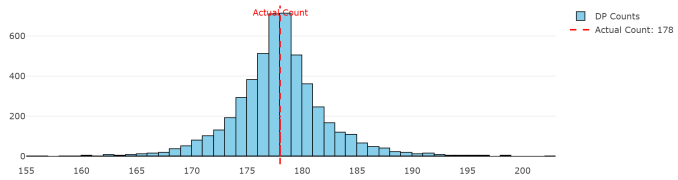

Fig. 3: Count Simulation (n=5000)

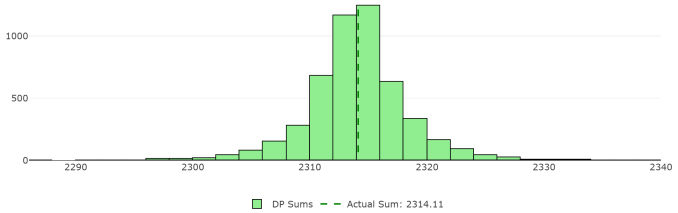

Fig. 4: Sum Simulation (n=5000)

Later in the pipeline, users will have an option to publish the count (number of records), and sum differentially private statistics, so it is important to explore the possible values now.

Users can also explore a principal component analysis (PCA) visualization of their data in both a non-private and private context. We reduce the dimensionality to 2 axis and plot the resulting dimensionality reduced data. Optionally, the user can select one column of data to act as the colored label of data points (in a classification task, this would be the target class label). We use the Scikit-learn implementation of PCA for our non-private context and IBM's differential privacy library with a user given ϵ value for our private context. The reason we chose to use Diffprivlib over OpenDP is because OpenDP's PCA implementation at the time of development was not functional due to a bug in the classes constructor (this has since been patched with the release of OpenDP 0.13).

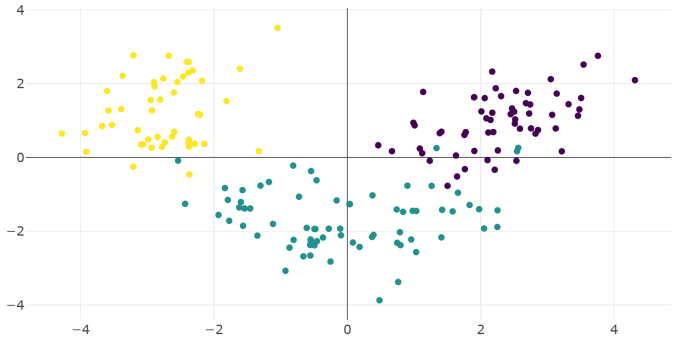

Fig. 5: Non-Private PCA

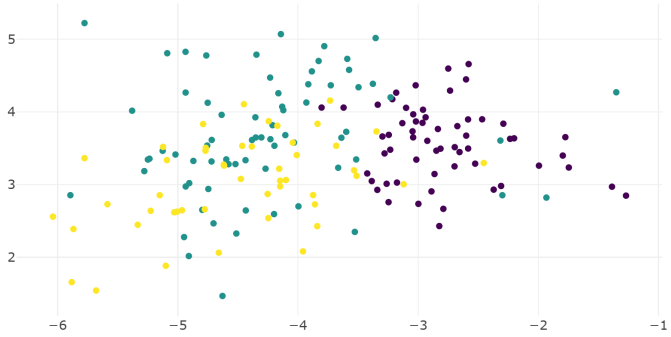

Fig. 6: Private PCA $\epsilon = 1$

Exploring the Wine dataset, we can clearly see 3 distinct clusters of data points that correspond with the 3 types of Wine in the non-private PCA Fig. 5, however, these distinct clusters are less prevalent / separable in the Private PCA, depending on the epsilon value, Fig. 6. Increasing the privacy parameter will create plots that interpolate between these two extremes. Machine learning researchers who want to train a model for classification using differentially private mechanisms can use this visualization to gain insights in how drastically changing the privacy parameter alters the data's utility. Generally with traditional ML techniques such as SVM's and Logisitc Regression, the model's ability to fit / find a meaniingful decision boundary is dictated by the separability of the data (with linear separability being ideal). Through this plot, users can tune a privacy parameter value that captures enough of the data's underline structural information neccessary for classification, while still proving important privacy guarantees.

C. Simulating Privacy Attacks

We explored a couple of differential privacy attack simulation options such as OpenDP's membership and differencing attacks. We ultimately did not include the functionality in the website. One reason was the difficulty involved in implementing the functionality in a generalizable manner. Many of the example Jupyter Notebooks provided by OpenDP have been archived for no longer being compatible with the calling conventions / syntax of newer versions of OpenDP. Furthermore, each demonstration assumes a significant amount of dataset / threat model specific knowledge making automation a non-trivial task left for future work.

D. Generate Synthetic Data

After the user has completed some initial exploration, they have the option to generate and download synthetic data. We create synthetic data with OpenDP's SmartNoise [13] and use the Multiplicative Weights Exponential Mechanism (MWEM) synthesizer to fit and sample synthetic data [8]. Future work can explore the impacts / tradeoffs of various synthesizers. Unfortunately, SmartNoise comes with a few drawbacks. For one, even on the relatively small Wine dataset, the library could take minutes to fit a model for synthetic data generation. It could also impose significant memory constraints (at times crashing our test Jupyter Notebooks) - this problem can be addressed by increasing the Split Factor parameter which adapts the model to operate on more manageable, less memory intensive units of data (this problem is most pronounced in datasets containing floating point values that have to be added to discrete bins). Another issue we encountered was that, without user provided domain specific knowledge (such as bounds on min/max data values which OpenDP can then use to estimate sensitivity), a large portion of the privacy budget is allocated for pre-processing. We were able to significantly reduce fit time by providing the synthesizer with context such as the max and min values of columns so it doesn't have to infer for its transformations automatically. The synthesizer would also fail if the pre-preocessor is not given enough of the privacy budget, so we built in a retry mechanism t hat increments the pre-processor allocation after failing to fit the synthesizer for a given dataset. Our final major issue originated from using multiple different data privacy libraries. In particular, we encountered dependency conflicts with SmartNoise-Synth. As such, we had to run our synhesizers in a separate virtual environment (calling a sub-process from wihin he web app).

Fig. 9 shows the R^2 per epsilon of training a logistic regression model and Fig. 10 shows performance of training on a random forest model. A baseline model is trained on the complete dataset. Next, synthetic dataset are generated for each epsilon value plotted and the same ML model is fitted to each one. Finally, we use a differentially private version of the ML model on the original dataset. This visual allows the user to visualize possible model performance with differing epsilon values and DP techniques which will further inform the user of best epsilon / architecture choices. We should see that the model performance increases as epsilon increases, but this may not be the case, especially with synthetic and noisy data that is randomly sampled. In general, it is not always clear wether for a given dataset it is better to use a DP version of a ML model on the original data, or to train a non-DP ML model on synthetic data. Therefore, our hope is that we can help an ML analyst make these decisions for themselves by plotting these tradeoffs specifically for their data.

We explore a couple model performance options like linear regression, logistic regression, and random forest training to allow users to review the right type of performance for their use cases.

E. Publish Data Release

DP General Statistics for Alcohol Column

Statistic	DP Value	True Value
Number of Responses (Count)	177	178
Sum of Alcohol	2,184.078	2,314.110
Mean of Alcohol	12.339	13.001

Fig. 7: DP Statistics

A user can also publish general statistics about the dataset. We simulated possible values in the Explore Statistics step, but now we generate differentially private values. Using OpenDP, the user can provide configuration that the app uses to generate statistics under one context Fig. 8. First, we infer the privacy unit (contributions) from the users selected label column. Next, we set the maximum partition of the dataset to the total number of records as this is a data release.

```
context = dp.Context.compositor(
    data=df,
    privacy_unit=dp.unit_of(contributions=contributions),   # len Unique Rows in label column
    privacy_loss=dp.loss_of(epsilon=epsilon),
    split_evenly_over=NUM_QUERIES,
    margins=[
        dp.polars.Margin(
            max_partition_length=true_count # the biggest partition
        ),
    ],
)
```

Fig. 8: OpenDP Context Setup

We originally left setting partitions and contributions to the user, but learned results would be very inaccurate if set incorrectly. These parameters can be modified but we noticed drastic differences in results including negative sum values and values on the order of 1000x difference from the true value, so parameters should be set carefully.

IV. Results

We run our pipeline on the Wine dataset [2] to better understand model performance on a common ML dataset when trained under different circumstances.

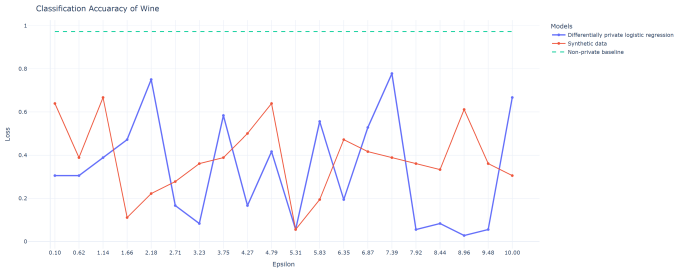

Fig. 9: R^2 of Logistic Regression Models

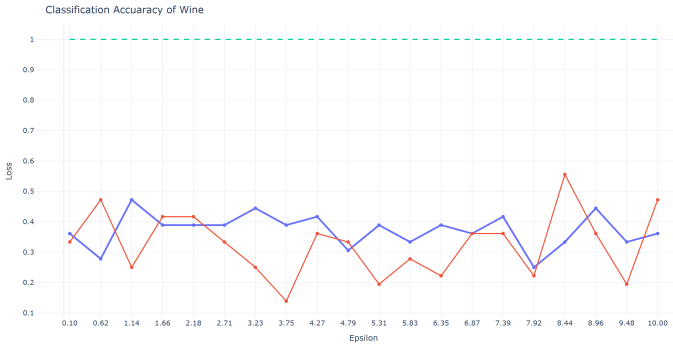

Fig. 10: R^2 of Random Forest Models

V. Discussion

ClearAudit is made as a general solution to explore datasets for user in machine learning applications while incorporating differential privacy. This means that the implementation of the site can be applied to a wide range of datasets. We did however test our site on the Wine dataset. Fig. 10 and Fig. 9 show the model performance results of training on a non-private dataset, a non-private dataset with a private training mechanism, and a synthetic dataset. As we can see, for certain epsilon values it is ideal to use Synthetic data over DP ML models, while at others the opposite is true. Thus, our tool can help an ML analyst correctly find the correct technique to use for their desired level of privacy and performance. Increasing epsilon makes the dataset less private (and improves utility), so we expect the model performance to approach the baseline performance (in green). However, the two plots were quite noisy and non-deterministic. On subsequent runs, sometimes the trend would be slightly upwards and sometimes the trend would be negligible as epsilon increases. One attributing factor is that the size of the training data is small, so noise will produce a lot of variance in the result especially for the randomly sampled synthetic datasets. This is of course not ideal as the visuals we show the user should be robust. This can be alleviated by re-running the results for each data point (say 10 or 50 times) and averaging the result (at the expense of web app latency since the backend needs to do significantly more processing).

A. Evaluation

Our goal for the auditing platform was to accurately and clearly communicate security risks to a data publisher to help them best decide on the privacy mechanisms they need for their dataset and threat model. While we can improve on the accuracy of the risks, we think our platform effectively communicates the use case for differential privacy in a visual manner so that even ML analysts with minimal exposure to DP can understand the privacy / utility tradeoff.

B. Limitations and Challenges

During the development of our Web App, we faced a number of challenges integrating features across datasets. As many of these libraries are in active development, they can at times either lack features or have bugs in their implementation.

We found one such example when testing the aforementioned differentially private PCA functionality implemented in OpenDP which would not run (thus, we opted to use the Diffprivlib implementation). Submitting the issue to the Github [12], we quickly received a response and in the subsequent release, the bug was patched.

The pipeline also only supports cleaned datasets with numerical data types and no missing values. To reach a wider audience and more realistic range of datasets, we would need to incorporate data cleaning and transformation options.

Generating model accuracies can also take time. As described in the methodology, we dramatically reduced synthetic data creation time by providing more context to the synthesizer, however, this should likely be better communicated to the current user.

A big problem working with differential privacy and synthetic data is that the results were non-deterministic and too noisy during testing. We could mitigate the noise be rerunning simulations at the same epsilon and taking the average or generating synthetic data with very high epsilon values if it is a regression task, however, this can sacrifice the real time nature of the web app. More work needs to be done to develop a satisfying compromise between speed and resolution.

C. Ethical Considerations

The website, in its current state, is made to run locally on a user's private network. We did not focus on the security and data management aspects of the website. Thus, ClearAudit should be used as an initial exploration tool of differential privacy before model training in an already private environment. Significant work would need to be done to adapt our tool into a trusted, multi-user online platform.

D. Future Work

Developing web apps is an endless cycle which must account for changes in the web space, incorporate new features as libraries grow, and maintain compatibility for older devices / systems. Significant improvements could be made in the providing the user greater freedom to configure the web app's parameters. As this was a demo, the website was streamlined to core functionality, but with how expansive OpenDp and Diffprivlib have become in recent years, there is boundless new features that could provide users with incredibly valuable tools.

VI. Conclusion

We have proposed and developed ClearAudit, a multi-library framework for visualizing and understanding the privacy utility tradeoff of differentially private mechanisms in the machine learning space. Using industry standard tools, we developed a streamlined, easy to use tool that incorporates a wide variety of open-source, well audited DP mechanisms. While numerous limitations and areas of for further development exist, we hope ClearAudit can lay the foundation for more user friendly, robust tools in the machine learning space that help address the divide between the usage of DP compliant algorithms in academia and industry.

References

[1] Amina A. Abdu et al. *Algorithmic Transparency and Participation through the Handoff Lens: Lessons Learned from the U.S. Census Bureau's Adoption of Differential Privacy*. 2024. arXiv: 2405.19187 [cs.CY]. URL: https://arxiv.org/abs/2405.19187.

[2] Stefan Aeberhard and M. Forina. *Wine*. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5PC7J. 1992.

[3] US Census Bureau. *The Census Bureau's Simulated Reconstruction-Abetted Re-identification Attack on the 2010 Census*. Census.gov. Section: Government. URL: https://www.census.gov/data/academy/webinars/2021/disclosure-avoidance-series/simulated-reconstruction-abetted-re-identification-attack-on-the-2010-census.html (visited on 03/23/2025).

[4] *Contribution Process &x2014; OpenDP — docs.opendp.org*. https://docs.opendp.org/en/stable/contributing/contribution-process.html. [Accessed 30-04-2025].

[5] *DP Wizard: An Easy Way to Get Started with Differential Privacy and OpenDP — opendp.org*. https://opendp.org/blog/dp-wizard-easy-way-get-started-differential-privacy-and-opendp. [Accessed 30-04-2025].

[6] *Edge Impulse - The Leading Edge AI Platform*. URL: https://edgeimpulse.com/ (visited on 04/28/2025).

[7] Simson L. Garfinkel, John M. Abowd, and Sarah Powazek. "Issues Encountered Deploying Differential Privacy". In: *Proceedings of the 2018 Workshop on Privacy in the Electronic Society*. CCS '18. ACM, Jan. 2018. DOI: 10.1145/3267323.3268949. URL: http://dx.doi.org/10.1145/3267323.3268949.

[8] Moritz Hardt, Katrina Ligett, and Frank McSherry. "A Simple and Practical Algorithm for Differentially Private Data Release". In: ().

[9] Naoise Holohan et al. "Diffprivlib: the IBM differential privacy library". In: *ArXiv e-prints* 1907.02444 [cs.CR] (July 2019).

[10] William Kong et al. *DP-Auditorium: a Large Scale Library for Auditing Differential Privacy*. Dec. 18, 2023. DOI: 10.48550/arXiv.2307.05608. arXiv: 2307.05608[cs]. URL: http://arxiv.org/abs/2307.05608 (visited on 03/23/2025).

[11] *Limitations &x2014; OpenDP — docs.opendp.org*. https://docs.opendp.org/en/stable/api/user-guide/limitations.html. [Accessed 30-04-2025].

[12] OpenDP Shoeboxam. *Opendp/Issues Running Example PCA code provided in the documentation*. URL: https://github.com/opendp/opendp/issues/2363.

[13] *SmartNoise Synthesizers — OpenDP SmartNoise Synthesizers*. URL: https://docs.smartnoise.org/synth/index.html (visited on 04/28/2025).

[14] *smartnoise-sdk/synth at main · opendp/smartnoise-sdk — github.com*. https://github.com/opendp/smartnoise-sdk/tree/main/synth. [Accessed 30-04-2025].

[15] *The OpenDP White Paper — Harvard University Privacy Tools Project*. URL: https://privacytools.seas.harvard.edu/publications/opendp-white-paper (visited on 03/23/2025).

[16] Yuxin Wang et al. "DPGen: Automated Program Synthesis for Differential Privacy". In: *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*. CCS '21. ACM, Nov. 2021, pp. 393–411. DOI: 10.1145/3460120.3484781. URL: http://dx.doi.org/10.1145/3460120.3484781.