Privacy in Practice: The Feasibility of Differential Privacy for Telemetry Analysis

Trey Scheid
tscheid@ucsd.edu

Tyler Kurpanek
tkurpane@ucsd.edu

Bradley Nathanson
bnathanson@ucsd.edu

Christopher Lum
cslum@ucsd.edu

Yu-Xiang Wang
yuxiangw@ucsd.edu

Abstract

This research investigates the implementation implications of differential privacy mechanisms for telemetry data analysis, with a focus on real-world applications. We show, through examples, how knowledge of fundamental privacy-preserving techniques, including randomized response and the Laplace mechanism, is enough to protect sensitive information while maintaining analytical utility. We privatize data tasks from 4 applied research works using Intel telemetry data which encompasses multiple statistical tasks, from user-level rate analysis to logistic regression classification. The study utilizes various (ϵ, δ) budgets (using AutoDP) for precise privacy loss measurement and to quantify the inherent tradeoff between privacy and utility. By demonstrating the feasibility of differential privacy in production environments, we provide a roadmap for organizations seeking to enhance their privacy practices.

Website: https://trey-scheid.github.io/privacy-in-practice/
Code: https://github.com/Trey-Scheid/privacy-in-practice

Table of Contents

1 Introduction

The implementation of differential privacy in production environments presents significant challenges in balancing privacy guarantees with analytical utility. This research addresses these challenges by developing practical privacy-preserving mechanisms for existing telemetry analysis tasks while maintaining the usefulness of their systems. We integrate various differential privacy mechanisms including: Gaussian Composition, the ExponentialLaplace mechanism, and DP-GD to protect sensitive information in telemetry data. Our chosen data tasks encompasses multiple statistical tasks, from user-level rate analysis to logistic regression classification correlation. By evaluating the trade-offs between privacy guarantees and analytical accuracy in production settings, we provide evidence and direction for organizations looking to enhance their privacy practices.

1.1 Motivation

Despite the growing importance of privacy-preserving data analysis Pew Research Center (2019), many practitioners perceive differential privacy implementation as complex and challenging, others note how results only come with dissatisfactory ϵ-level guarantees Bonawitz et al. (2022). Besides the sub-optimality of some DP methods, this perception stems from human difficulties: the mathematical complexity of privacy definitions, the need to carefully calibrate privacy parameters, and concerns about reduced utility (managing the trade-off) Ponomareva et al. (2023). A survey by Smith et al. found that only 23% of data scientists felt confident implementing differential privacy mechanisms in their workflows **?**. Researchers are working to democratize, demystify, and improve usability around Differential Privacy Ponomareva et al. (2023). Recent developments have significantly lowered these barriers to entry. Tools like Google's Privacy on Beam [1], Microsoft's SmartNoise [2], and various open-source libraries like AutoDP[3] provide accessible frameworks for implementing differential privacy. These tools abstract away much of the underlying complexity while maintaining rigorous privacy guarantees. In addition, educational resources and practical tutorials have emerged to guide practitioners through implementation challenges [4].

This research builds upon these recent developments by providing a practical demonstration of differential privacy mechanisms in telemetry data analysis. By implementing privacy-preserving techniques for existing tasks, we aim to show that differential privacy can be seamlessly integrated into production systems without significant utility loss. Our work focuses on two key objectives: privatizing existing telemetry analysis tasks and evaluating the privacy-utility tradeoffs in production settings.

[1] https://codelabs.developers.google.com/codelabs/privacy-on-beam
[2] https://smartnoise.org
[3] https://pypi.org/project/autodp/
[4] https://desfontain.es/blog/friendly-intro-to-differential-privacy.html

1.2 Differential Privacy

Differential privacy by Dwork and Roth (2014) is a framework for data privacy that gives a mathematical guarantee that the information for each individual (record or user) in the dataset is protected[5]. The core idea is to introduce random noise into algorithms so that the data of any individual does not significantly affect the overall result and therefore is not recoverable or identifiable.

Mathematically, a mechanism M is considered (ϵ, δ) differentially private if for all datasets D and D' which differ by at most 1 element when $\mathbb{P}[M(D) \in S] \leq e^{\epsilon}\mathbb{P}[M(D') \in S)] + \delta$ where ϵ and δ are privacy parameters and S is a query solution set. Smaller ϵ and δ imply stronger privacy guarantees. Differential privacy is a *property* of algorithms, not datasets; it is a method that ensures private results to a high degree of probability (whether that is a trained model or a noisy dataset).

This definition applies to data anonymization, but does not cover methods for transparency, use, access or security. By pursuing this property for common data tasks we aim to create a solution which once implemented achieves similar results but removes the need for data access and security. Some settings the predictions themselves are important, but sensitive, meaning a mechanism without anonymity is unusable!

1.3 Telemetry Data

The Intel Data Collection and Analysis (DCA) team derives insights from over 39 million systems! This includes any of their hardware installed in personal, corporate and IoT devices (collected only with consent). Through their partnership with the University of California San Diego's Halicioğlu Data Science Institute, at the foundation of Intel Lab's Telemetry Center of Excellence, they permit study of possibly sensitive device information to develop solutions that benefit the whole ecosystem [6]. Faculty and Intel researchers have published many white papers using the database since the CoE inception in 2020. We selected 4 of interest and found their core data science methods Table 1.

Differential privacy methods and guarantees are attractive for many domains. Telemetry is the remote data transfer of automated system measurements. As people use technology everyday their machines track usage diagnostics which are used by hardware and software manufacturers to reduce bugs and increase efficiency. System usage information is recorded at regular intervals and usually results in massive quantities of measurements. The identifiability of the specific machine or user of an event is a concern regardless of PIID tags. Dinur Nissim Reconstruction and linkage attacks can be used to recover or reconstruct the original information: the source Dinur and Nissim (2003). This is a breach of privacy for a user which depending on the sensitivity of the information can be concerning. For example, personal laptops may send diagnostics to Intel given that the user opts in to the program [Intel telemetry].

[5] Gadotti et al. (2024) has an in depth explanation of interpretations and attacks.

[6] https://community.intel.com/t5/Blogs/Tech-Innovation/Data-Center/Intel-Labs-Investment-in-Telemetry-Center-of-Excellence-Produces/post/1460669

Table 1: Data Tasks

Data Task	Code	Paper	Citation
Conditional Probabilities	COND_PROB	Exploration of CPU Error Dependencies and Prediction	Kwasnick (Unpublished)
KMeans Clustering	KMEANS	PC Health Impact White Paper	Ryan et al. (Unpublished)
Lasso Regression	LASSO	Power Consumption Patterns in Intel's Telemetry Data: China Burns 2x Energy that of the US	Cheon (Unpublished)
Logistic Regression Significance Tests	LR_PVAL	Product Health Insights Using Telemetry	Su et al. (2024)

We use a secure research database shared be Intel Corporation with consent of its users to generate real results.

1.3.1 Errors

In our paper, we will analyze two different types of errors. The Machine Check Architecture, or MCA, will detect an error and label it as either corrected or uncorrected. A corrected error means the system can observe and correct a detected error. Correction mechanisms include single error correction, double error correction, and more. An uncorrected error is one that was detected but not corrected or there was a computation delay long enough that the MCA treated it as an interrupted computation. Kwasnick (Unpublished)

1.3.2 Hardware Power

Another concept is power, this is the rate of energy consumption by the device. Our analysis is on CPU's produced by Intel and AMD. Kwasnick (Unpublished)

1.4 Applications

Telemetry is one narrow domain which privacy is a concern, many other types of data require sensitive handling and sharing practices. For example the US Census [7].

We will get into the specifics of each task, however note that each one can be applied to datasets in any domain. Probabilities of political party affiliation, Lasso/kmeans for gene identification, or correlation for stock prices.

[7] https://www.census.gov/about/policies/privacy/privacy-policy.html

1.5 Related Work

We are building on a previous analysis on differentially private mechanisms for Logistic Regression Scheid et al. (2199). The paper investigates how privacy affects different mini-batch stochastic gradient descent algorithms for logistic regression classification. It is shown that privacy affects the batch size for optimal performance.

2 Methods

2.1 Utility

Each method used in our study has different measures of success, but they can all be translated into a proportion representing utility compared to the best model, whether that is the non-private baseline or an improved version. In the context of differential privacy, utility refers to the degree to which a privatized model or analysis retains its analytical accuracy and usefulness compared to its non-private counterpart.

2.2 Data Structure

The first sub-task was data ingestion and processing. One key feature of telemetry data is the volume; the research database was already a processed version of raw signals from devices, aggregated and merged for usefulness and practicality. This left us with a schema containing 10's of tables and cryptic metrics, highlighting the importance of documentation by our fellow researchers to help us replicate their work.

The tables once loaded to disc could be processed with SQL queries and filtered and processed with more basic operations such as aggregating by group and finding statistics. These would make up our featurized datasets ready for Python analysis.

2.3 LR_PVAL: Private Correlation (via Logistic Regression Coefficient)

This paperSu et al. (2024) seeks to identify whether a certain variable is disproportionately present for a certain outcome. More specifically, it takes a close look at two variables, max temperature on a day and whether a corrected error was present on that day. They would take one of those two variables and train a logistic regression model with maximum likelihood estimation to predict whether an uncorrected error was present. From the model, they use the coefficient of the variable and make a hypothesis test whether that variable is equal to zero.

For our implementation, we focused only on whether there were corrected errors on a day, and not the variable max temperature on a day. We add privacy to the model by using DP-SGD when training the logistic regression model, where the hypothesis test is then private

by means of post-processing. We replicated the non-private model to get a baseline for utility and compared it to our private model using set intersection over union. Our data came from Intel's telemetry database, which contains information about the hardware and software on millions of devices and we took a subset of one year of the data from Februrary 2020 to 2021.

2.3.1 Nonprivate Logistic Regression

The paper used a univariate logistic regression model, one for each of the top 30 uncorrected error codes. The target variable is whether or not an uncorrected error was present on a day, and the feature a boolean of whether there was a corrected error on that day, making uncorrected error corrected error.

The logistic regression model is defined as

$$\log \frac{P(y=1|x)}{1-P(y=1|x)} = \beta_0 + \beta_1 x \tag{1}$$

and they took β_1 as the coefficient of interest. They used a Wald test to test the null hypothesis that $\beta_1 = 0$ against the alternative that $\beta_1 \neq 0$. The Wald test statistic is defined as

$$W = \frac{\hat{\beta}_1}{SE(\hat{\beta}_1)} \tag{2}$$

where $\hat{\beta}_1$ is the coefficient estimate and $SE(\hat{\beta}_1)$ is the standard error of the coefficient estimate. This statistic can be used to test whether or not the feature is significant in predicting the target variable (whether the presence of a corrected error is associated with the presence of an uncorrected error). For our implementation, we used the statsmodels library to fit the model and calculate the coefficient and standard error with an alpha of 0.05.

2.3.2 Private Logistic Regression

In order to add privacy to the model, we used DP-Gradient Descent to train the model. DP-GD is a method of training a model with differential privacy by adding noise to the gradient updates using Gaussian noise. Each iteration of gradient descent adds a little bit more noise, consuming a proportion of the privacy budget, meaning that higher epsilons require more iterations. We considered using DP-Stochastic Gradient Descent, but the processed datasets were small enough to warrant using DP-GD. We implemented our own DP-GD algorithm using AutoDP[8] to account for our privacy budget. After privately training the model, post-processing enabled us to calculate the wald test the same way as in the non-private model. In order to compare the alpha values, we completed permutation testing on each model

[8] https://pypi.org/project/autodp/

in order to calculate the empirical p-value which was used to identify models that were statistically significant. One of the thirty models had exceptionally more values than the others, so to save compute we removed it from the analysis.

2.3.3 Utility

We defined our utility metric as the set intersection over union of the top 29 uncorrected error codes. This metric has a range of 0 to 1, where 0 is no intersection and 1 is full intersection, meaning that the higher the utility, the more similar the two results are.

2.4 LASSO Private Regression (via DP-Frank-Wolfe)

2.4.1 Utility

2.5 KMEANS Private Clustering (DP-Lloyd's)

K-Means clustering (Lloyd's Algorithm) is applied to group devices based on similarities in their usage patterns. The method leverages Z-scores for standardizing the usage data and calculates L1 distances between weekly usage patterns to identify trends over time. Lloyd's Algorithm clusters devices by assigning them to centroids based on their usage patterns, recalculating the centroids as the mean of assigned points after each iteration.

Differentially Private Lloyd's Algorithm (DP-Lloyd's)[9] modifies the standard K-Means clustering by adding Laplacian noise during the iterative centroid update step to ensure privacy. It introduces noise to both the sum of coordinates and the count of points within clusters, with the amount of noise controlled by the number of iterations and the sensitivity of the data. This is applied after clipping the data, to for ensure that the impact of any single data point is limited.

$$\mathrm{x'}_j = \mathrm{clip}\left(\mathrm{x}_j, -\tau, \tau\right)$$

After clipping the data, the sum of each cluster is computed with added Laplace noise. One point changes the sum, at most, the range of the dataset, which is $2 \times \tau$ (the sensitivty). The number of data points is at least 1 if the cluster exists, and so one data point changes the count by 1. The sensitivity is therefore 1. These sensitivities are then divided by the per-iteration ϵ.

$$\mathrm{new\_centroid}_j = \frac{\sum_{i=1}^{n_j} \mathrm{x'}_{j,i} + \mathrm{Lap}(\tau \times 2, \epsilon_{\mathrm{iter}})}{\max\left(1, n_j + \mathrm{Lap}(1, \epsilon_{\mathrm{iter}})\right)}$$

[9] https://arxiv.org/pdf/1504.05998

2.5.1 Utility

To calculate the utility for the K-Means task, we defined the loss for each epsilon as the sum of squared distances between each data point and its closest centroid. We then divided it by the number of data point so it is agnostic to the dataset size. We then normalized the loss so that each loss value fell between 0 and 1 and defined the utility as 1-loss to inverse it.

2.6 COND_PROB: Private Conditional Probability Release (Laplace Mechanism)

In order to replicate the paper, were are going to apply differential privacy methods to the processes outlined in the paper. In the paper, the authors first separate each GUID (user) into the number of corrected errors observed during a set time period. They then created a histogram where the x-axis was the number of corrected errors observed and the y-axis was the percentage of GUID's that observed an uncorrected error.Kwasnick (Unpublished) This is the process that we are aiming to privatize.

We are releasing a percentage for each bin in the histogram, and in order to guarantee privacy, we must add noise to to both the numerator and denominator where the numerator is the number of GUID's that contained an uncorrected error (number of 1's) and the denominator is the number of GUID's total. However we can not release the number of GUID's in each corrected error count as that would violate differential privacy so instead we are just going to add noise to number of GUID's that contained an uncorrected error as well as the number of GUID's that did not contain an uncorrected an error (number of 1's + number of 0's) so that we have a private denominator.

$$P(Uncorrected) = \frac{\text{GUID's containing an Uncorrected Error}}{\text{Total GUID's}}$$

We are going to apply the Laplace mechanism to to release this percentage privately. The Laplace mechanism adds noise drawn from the Laplace distribution to the output of a function. B is the scale parameter and ∇f is the sensitivity of the function f. Dwork and Roth (2014)

$$noise \sim Lap(b)$$
$$Lap(x|b) = \frac{1}{2b} e^{-\frac{|x|}{b}}$$
$$b = \frac{\Delta f}{\epsilon}$$

The sensitivity of the function is defined as the maximum possible absolute change in the output of the function due to the change in a single user's data. Since we are dealing with a percentage (and we are considering the worst case), this change can be at most

1 user, which corresponds to a sensitivity of 1 (in terms of the scale of the count, not the percentage itself). This is because, in the worst case, the change is a single user being added or removed, and the total number of GUIDs is assumed to be large enough that the effect of one user's change on the output is not too significant.and ∇f is the privacy parameter.

$$\Delta f = \max_{D,D'} ||f(x) - f(x')|| = 1$$

D and D' are neighboring datasets

2.6.1 Utility

To calculate the utility for the Conditional Probability task, we defined the loss for each epsilon as the Mean Absolute error between the Noisy and Original Histograms. We then normalized the loss so that each loss value fell between 0 and 1 and defined the utility as 1/loss.

3 Results

3.1 All Tasks: Privacy vs Utility

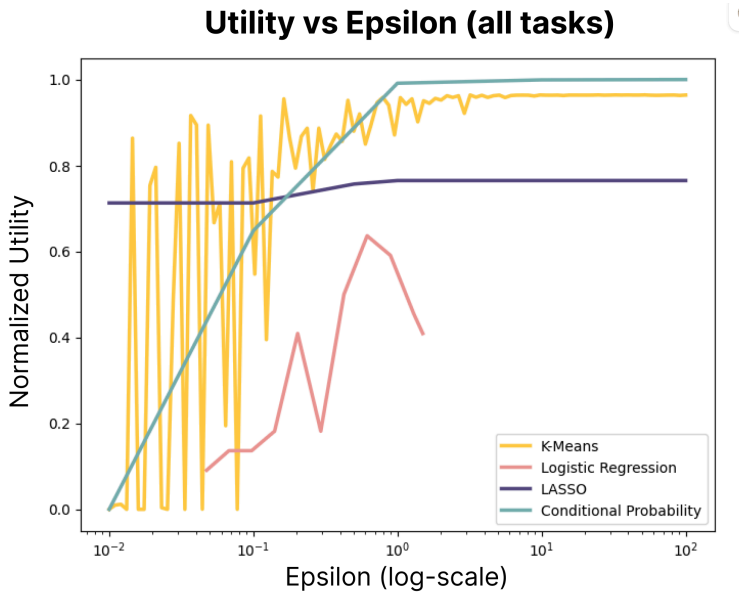

Figure 1: Privacy vs Normalized Utility for all tasks. Non-private models have a utility of 1.

Figure 2: Your plot caption here.

3.2 COND_PROB

3.3 LR_PVAL

Our main method of comparing against the baseline was by calculating the set intersection over union for the private models and the non-private models. For epsilon = 1, the model severely under-shot the number of significant models. This means that it failed to find the correct relationship between corrected and uncorrected errors in a third (0.31) of the models.

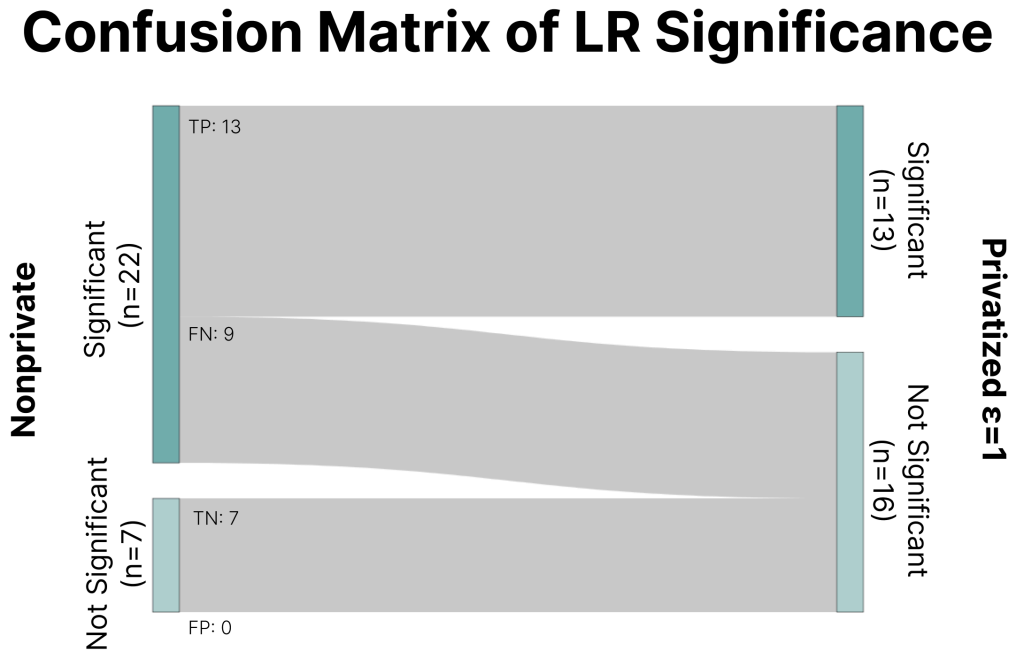

Figure 3: Confusion matrix for epsilon = 1 for Wald Test. Right side has the private model classifications and the left has the non-private baselines. There were no false positives

3.4 KMEANS

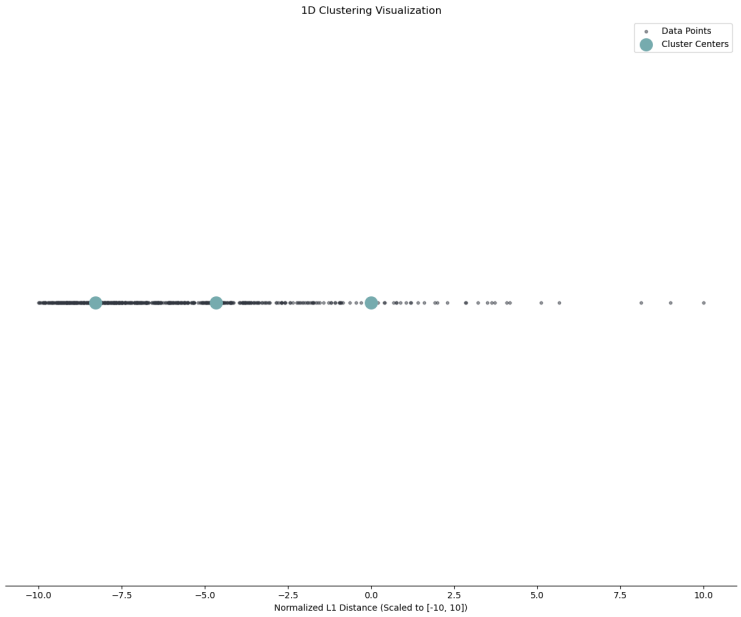

Figure 4: Centroid locations overlaid over data points $\epsilon = 1$

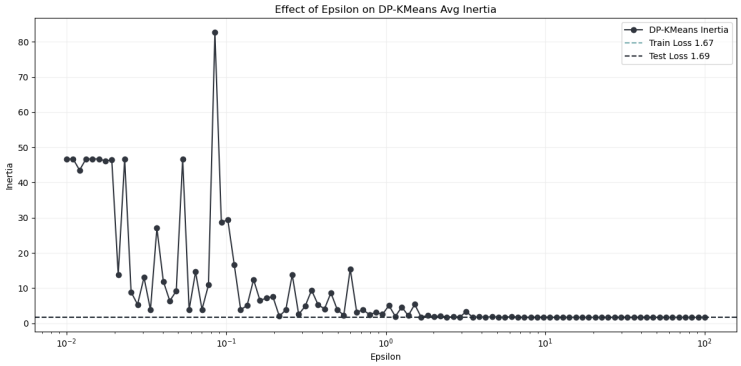

Figure 5: Epsilon vs. Inertia compared to baseline non-private model

3.5 LASSO

4 Discussion

4.1 Interpretation

Our work provides a road map for integrating differential privacy into production environments, ensuring that sensitive telemetry data can be analyzed without compromising user privacy. This is particularly relevant for industries handling large volumes of user data.

4.1.1 COND PROB

Similar to established differential privacy theory and literature /citeDworkroth, we observed a tradeoff between privacy and utility. However, our focus on conditional probability tasks allowed us to explore this tradeoff in a more specific context, providing insights into how differential privacy impacts analytical accuracy in this domain.

Our results show a significant increase in utility as ε increases. For instance, at $\varepsilon = 0.1$, the utility was approximately 0.65, which rose to nearly 0.99 at $\varepsilon = 1$ and reached almost 1 at $\varepsilon = 100$. This demonstrates that less stringent privacy guarantees (higher ε values) lead to higher utility.

For each epsilon, we found that as the number of corrected errors increased, the utility decreased. This is because the distribution of number of corrected errors is very heavily skewed left. The error amount with the largest amount of unique GUIDs is 0. This trend continues; as the number of corrected errors increases, the number of GUIDs with that exact amount of corrected errors decreases. This means the utility of the function is bounded by the maximum number of errors included in the histogram. We bounded our largest bin to be at 30 and found that an epsilon of 1 gave a sufficient utility. If we wanted to use a stricter epsilon, we would need to reduce the maximum corrected error to maintain the same utility.

4.1.2 LR$_{pVAL}$

Due to high compute costs, we were only able to get up to epsilon 1.5 for each of the 29 models. As we expected, the utility would increase from very low epsilon but around $\varepsilon = 0.5$, it is unclear whether or not the utility would continue to increase. At $\varepsilon = 1$, the model does somewhat poorly as seen in 3. The model has an intersect over union of around 0.60 and notably identifies a majority of the models as not significant, a result contrary to the nonprivate model. Overall, this analysis task did not seem to be replicable privately, at least under the strict privacy constraints.

It is important to note that this analysis would not be nearly as computationally hungry if we weren't trying to compare the models to a non-private model. In the practical setting, we wouldn't need to do permutation testing to find our p-values empirically. Similarly to the non-private setting, our alpha would be a hyperparameter that we would be able to select ourselves. This means that we may be able to use much higher values of epsilon in practice

4.1.3 KMEANS

The clustering algorithm is unsupervised so the best way to quantify performance is using inertia. It is the squared differences between data points and their nearest centroid. The smaller this value is, the better performing the cluster locations are, and therefore the model.

As you can see from figure 4, the cluster locations are located where the most data points are. Given that the distribution of the data is right skewed, it makes sense that the centroids, or cluster locations, are also right skewed to follow the data.

From the figure 5, we can see that as epsilon increases, the inertia for the privatized model slowly converges to the non-private train and test loss. This is expected, and given its convergence close to an epsilon of 1, the model performs quite well in terms of both privacy and accuracy.

4.1.4 LASSO

4.1.5 Meta

For each of our tasks, the privacy-utility tradeoff is not identical. Each task has different sensitivities to added noise. Some tasks can tolerate higher amounts of noise without significantly affecting the utility of the results, while others rely on precision and can degrade very fast with even slight noise. Given by our combined plot, each task has signifcantly different curves as epsilon increases. One potential factor that also is not considered is that the added noise may reduce overfitting and could help the model generalize better.

4.2 Reflecting On The Process

A part of assessing the feasibility of applying differential privacy is necessarily going to be a discussion surrounding the process of applying DP itself. We have gotten together to discuss what went well, what went poorly, and what limitations we had to concede.

4.2.1 What was helpful

In the process of applying differential privacy, we found three main things the most helpful: mathematical foundations of DP result in a consistent comparison across methods, differential privacy algorithms are intuitive at a high level, and that many papers exist on different DP algorithms ready to be implemented.

Comparing across methods was made easy because of the groundedness of DP in its definition and its reliance on epsilon. We could easily compare across tasks and observe that some tasks work well with an epsilon of 1 and some tasks didn't. The structure enabled us to have a strong idea that each of our privacy guarantees were identical.

These mathematical foundations do mean that a lot of research is theory and math oriented, but we found that at a high level, DP algorithms are intuitive and straight forward. DP algorithms rely on three main things: add noise/randomness, bound sensitivity, and privacy accounting. The specific math of how much noise to add or what to clip might be tough, but boiling down an algorithm is often as simple as knowing where the data gets clipped and where the noise gets added.

Many traditional analysis tasks have been privatized and written about. Applying differential privacy oneself often relies on finding a paper detailing the mechanism and molding it for your specific use case. The authors of the papers have been especially nice as well, being available to email and talk to about their methods. One small thing to note, several times have we found minor errors in papers which did make applying the methods occasionally difficult, but overall, the methods already existed and we just needed to implement it.

4.2.2 What was difficult

In the process, we also found two main difficulties that hindered our ability to complete our analysis tasks: epsilon is difficult to interpret and it is hard to quantify utility loss.

Epsilon, as a value in the differential privacy equation, is straightforward with how it compares two probabilities. The issue is what this really looks like in real life. It is hard to get an intuition for what an arbitrarily bad event is and at what probability that would occur. We may know that our epsilons are the same, but what protections does that practically assure us? We know that an epsilon of 10 is bad, but how bad is it really? Sure, e^{10} is a massive value, but what is the probability that something terrible actually happens?

On the other hand, we sought to measure utility loss as compared to a baseline. This had a set of difficulties in its own right. For some analysis tasks, the non-private version may not be the ground truth. For example, a lot of deep learning models generalize better when noise is added during training. Establishing what is exactly maximum utility was a long conversation. Secondly, for a given amount of utility, it's hard to quantify how bad is bad. For example in our paper, the logistic regression models had an IOU of around 0.60. This is an example of a task that does have a more solid baseline, but how solid terrible is it to have 0.60 IOU? At a more abstract level, what if being 1% off is the difference betewen 100 million and 99 million lives saved? It's difficult to have a good intuition of what exactly we're losing.

4.2.3 Limitations

There were a couple of limitations surrounding our ability to forge our analyses: a general lack of knowledge and replication vs. novel analysis.

Six months ago, differential privacy was a new concept to each of us. None of our backgrounds delved greatly into the rigor of mathematical proofs. Telemetry data was new to us and we suffered from lack of domain knowledge. Researchers or analysts who wish to accomplish similar comparisons may benefit greatly from more knowledge in either differential privacy and/or the domain in question itself.

The applicability of our study as a commentary of the feasibility of DP methods must be framed knowing that we replicated papers and did not seek to do novel analyses from scratch. Having the guidance of the original paper meant that there were some steps that we did not attempt or do privately ourselves. We did not try to tune hyperparameters privately, a task that would rely on high amounts of domain knowledge or using some of

the privacy budget in order to find valuable hyperparameters. Further, we already knew what features we wanted, private EDA might take up plenty of privacy budget itself. One could argue that the analyst implementing a DP algorithm is already private and need not consider privacy in their analysis, but then the question arises, whom are we protecting against?

Additionally on replicating papers, some papers were difficult to replicate due to obscurity in their writing or lack of general information. A common pitfall was not knowing exactly which ”temperature” a paper was referring to. One of us had to assess several papers before being able to find one that would be able to be replicated.

4.3 State of DP and how it relates to us

The field of differential privacy is rapidly growing as more organizations and governments recognize the importance of protecting individuals' data in an increasingly data-driven world. It is crucial to be knowledgable about this field as it equips us with the understanding of how to protect sensitive data while enabling meaningful analysis.

5 Conclusion

5.1 Summary

Overall, we found mixed results on how feasible applying differential privacy was. Some tasks were hardly affected and others would result in much different conclusions. There seems to be no universal solution for applying DP in tasks, it is task dependent. Different tasks have different success criteria, different methods have varying levels of ability to be privatized.

The feasibility of applying differential privacy seems to rely heavily on the practitioner's knowledge of both DP methods and their own domain. There is a high barrier to entry with differential privacy. An analyst who is familiar with their domain but completely new to DP would struggle greatly switching their workflow from their non-private methods to their private counterparts. Further, if the guarantees of DP aren't adequately understood, there would be a lack of desire in putting in the effort to lose utility and gain privacy. A path forward to private analyses across the board would not be able to be done bottom up, smart people would need to hold the hand of the typical analyst.

5.2 Impact

We recreated baseline models and algorithms used in previous research papers with their associated private models in a practical setting, providing valuable insights into how these privacy-preserving techniques perform in real-world applications. As we are not PhD-level

researchers, with more academic rigor it could lead to more promising findings and a deeper understanding of the privacy-utility balance in applied machine learning. Nevertheless, our work demonstrates that with just a few months of practice and an understanding of differential privacy, it is possible to implement privacy-preserving methods that showcase the best epsilon that balances privacy and utility. As DP becomes even more accessible, it will make implementation faster, improving both performance and computation.

5.3 Future Direction

Future research could explore alternative differential privacy methods for our tasks, such as applying Lasso regression using the Functional Mechanism to improve utility while maintaining privacy. Additionally, investigating different privacy accounting regimes, such as Rényi differential privacy or zero-Concentrated DP, could provide a more flexible trade-off between privacy and accuracy, optimizing the overall performance of the model.

Future work could focus on privatizing additional data tasks to enhance privacy while maintaining analytical utility. One potential task for future privatization is identifying the owning group for addressing a telemetry-detected issue, which could benefit from group-level differential privacy. This approach would help protect sensitive organizational information while still enabling efficient issue resolution.

We could explore several directions to improve and expand differential privacy applications. One avenue is scaling up computations and applying privatization methods to different domains, enabling broader adoption in diverse fields such as gaming analytics, hardware performance, and behavioral studies. Additionally, investigating tasks with varying sensitivity levels could lead to more nuanced privacy strategies, where higher-sensitivity tasks receive stronger protections while lower-sensitivity tasks maintain higher utility.

Another promising direction is leveraging off-the-shelf differential privacy packages, such as Google's DP library or PySyft, to streamline implementation and improve accessibility. This could facilitate the more widespread adoption and standardization of privacy-preserving methods.

Beyond technical advancements, think-aloud studies and longitudinal research could provide valuable insights into how users interact with differentially private systems in real-world settings. By observing users over time, we can refine privacy mechanisms to better align with practical workflows. Finally, validating utility results through alternative testing methods would help ensure that privacy-preserving models maintain effectiveness across different evaluation metrics, strengthening confidence in their real-world applicability

6 Contributions

6.1 Author Contributions

: T.S. focused on task22 LASSO Regression to highlight the exploratory capabilities of private data while implementing a previously theoretical framework (Franke-Wolfe). C.L. focused on private linear regression to be able to do Wald tests to asses the relationship between input and target variables B.N. analyzed and implemented non-private and private K-Means clustering. T.K. analyzed the experimental results of applying privacy mechanisms to a conditional probability histogram. Y.W. supervised the research and provided guidance on the mathematical foundations. All authors contributed to writing and reviewing the manuscript.

6.2 Task Details

Trey Scheid

- Replication of
- Implementation of non-private franke-wolfe lasso regression
- Ethics considerations webpage
  Todo: Implementation of private franke-wolfe lasso regression

Tyler Kurpanek

- Replication of Exploration of CPU Error Dependencies and Prediction
- Implementation of Laplace Mechanism

Bradley Nathanson

- Replication of k-means clustering with Lloyd's algorithm
- Implementation of privatized k-means clustering

Christopher Lum

- Preprocess and replicate logistic regression paper privately and non-privately
- Develop website

Yu-Xiang Wang

- Concept ideation
- Data Access
- Provided guidance on the mathematical foundations
- Proofing and editing all content

6.3 Acknowledgements

We would like to recognize the support of our instructor, Yu-Xiang Wang, for his guidance and feedback throughout the project. We would also like to thank the teaching staff Umesh

Bellur and Shriniwas Kulkarni for their support and feedback. The tasks database was a foundational part of our work and was created by another student researcher: Qiyu Li. We also attended a workshop called "Workshop on Defining Holistic Private Data Science for Practice" hosted by ENCORE at UC San Diego, which helped greatly with our broad understanding of the state of the field of differential privacy in practice.

We also would like to thank the authors of the papers we referenced in our literature review. Their work was instrumental in our understanding of the topic and the development of our project. Our understandings of differential privacy has been built on the work of many researchers in the field. Especially those which engaged in discussion with us about the field (Smith, Ulman, Guatam et al.). We are grateful for their contributions.

References

Bonawitz, Kallista, Peter Kairouz, Brendan Mcmahan, and Daniel Ramage. 2022. "Federated learning and privacy." *Commun. ACM* 65 (4), p. 90–97. [Link]

Cheon, Seung Hyun. Unpublished. "Power Consumption Patterns in Intel's Telemetry Data: China Burns 2x Energy that of the US."

Dinur, Irit, and Kobbi Nissim. 2003. "Revealing information while preserving privacy." In *Proceedings of the Twenty-Second ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*. New York, NY, USA Association for Computing Machinery. [Link]

Dwork, Cynthia, and Aaron Roth. 2014. "The Algorithmic Foundations of Differential Privacy." *Foundations and Trends® in Theoretical Computer Science* 9 (3–4): 211–407. [Link]

Gadotti, Andrea, Luc Rocher, Florimond Houssiau, Ana-Maria Creţu, and Yves-Alexandre de Montjoye. 2024. "Anonymization: The imperfect science of using data while preserving privacy." *Science Advances* 10 (29), p. eadn7053. [Link]

Kwasnick, Robert. Unpublished. "Exploration of CPU Error Dependencies and Prediction."

Pew Research Center. 2019. "Americans and Privacy: Concerned, Confused and Feeling Lack of Control Over Their Personal Information." report, Pew Research Center. [Link]

Ponomareva, Natalia, Hussein Hazimeh, Alex Kurakin, Zheng Xu, Carson Denison, H. Brendan McMahan, Sergei Vassilvitskii, Steve Chien, and Abhradeep Guha Thakurta. 2023. "How to DP-fy ML: A Practical Guide to Machine Learning with Differential Privacy." *Journal of Artificial Intelligence Research* 77, p. 1113–1201. [Link]

Ryan, Jacob M., Shuangquan Feng, Marie McCusker, Benjamin L Smarr, Rayan Saab, and Virginia de Sa. Unpublished. "PC Health Impact White Paper."

Scheid, Trey, Tyler Kurpanek, Christopher Lum, Bradley Nathanson, and Yu-Xiang Wang., "Exploring Tradeoffs in Differential Privacy: An Empirical Study of Logistic Regression on Telemetry Data."

Su, Fei, Robert Kwasnick, John Holm, William Penner, Hermann Gartler, Josh Boelter, Yufei Zhou, Bijan Arbab, and Michael Rothberg. 2024. "Product Health Insights Using

Telemetry." *IEEE Design Test* 41 (4): 56–64. [Link]

Appendices

A.1 Project Proposal

A.1.1 Problem Statement

Telemetry data is important to privatize as it encodes personally identifiable information which could be used to discover sensitive information. This data is collected from various IT devices, from satellites to personal computers. For our project, the telemetry data includes hardware and software performance metrics, monitoring, and errors.

We will privatize 22 analysis tasks for the Intel telemetry dataset, ensuring a reasonable privacy budget (). We will implement mechanisms that balance data utility and privacy, ensuring sensitive information is protected, and allocate a reasonable privacy budget (), a parameter that governs the trade-off between accuracy and privacy.

One example of a task is to predict CPU failure. This would require a privatized logistic regression model that predicts the probability of a failure from 0-1. The model would analyze data such as CPU temperature, usage patterns, error logs, or other performance indicators. If non-privatized, this model could expose this data, as a malicious individual could do a reconstruction attack, a method to reconstruct the training data by repeatedly querying the model with various synthetic inputs. The attacker could query this model with different sets of CPU-related inputs, and, over time, the attacker could gain information such as the CPU temperature threshold for an error to occur, or whether certain system configurations have a distinct failure pattern.

A.1.2 Methods

Our methodology for privatizing the 22 telemetry analysis tasks will employ multiple privacy mechanisms, such as the exponential mechanism and the Laplace mechanism, with AutoDP serving as our core privacy accounting tool. For each analysis task, we will first evaluate the sensitivity of the computation and determine the optimal privacy mechanism to maintain utility while satisfying privacy requirements. The implementation process requires careful privacy budget allocation across multiple components of each analysis to ensure the total privacy loss remains within acceptable bounds.

The evaluation of each privatized implementation will involve a comprehensive comparison with non-private baselines to document the privacy-utility tradeoff. This includes analyz-

ing performance metrics before and after applying privacy mechanisms, measuring accuracy degradation at various privacy budget levels, and considering computational efficiency challenges specific to telemetry data analysis. AutoDP will help quantify the privacy guarantees and guide the noise calibration process throughout implementation.

Each privatized task will be thoroughly documented with implementation details, privacy guarantees, and performance metrics. This documentation will include privacy budget allocation strategies, noise mechanism selection rationale, and practical guidelines for future implementations. The goal is to create a comprehensive resource demonstrating how different privacy mechanisms can be effectively applied to various telemetry analysis scenarios while maintaining practical utility and ensuring strong privacy protections.

A.1.3 Deliverable

The privatized analysis tasks will be stored and shared in a public repository, (without release of source data from Intel). This is our primary contribution, to offer tools in a privatized manner. In collaboration with the accessible programs, we will publish a website that will serve to educate our peers on differential privacy. The variety of analysis tasks done in the telemetry domain can be generalized and applied to many types of data; therefore, descriptions of privacy algorithms, their motivations, and limitations can teach practitioners new methods for their own tasks.

The Intel data as mentioned is not public (due to the customer privacy and proprietary nature). Therefore our data processing, tasks, and report will include only some metrics of performance and data quality (size, distribution, features, etc). For the information we can share, we will compare the performance of the task with that of the non-private baseline. This gives analysts a sense of the utility-privacy tradeoff in each application.

A.1.4 Impact

By implementing differential privacy across telemetry we will create a significant impact by maintaining data confidentiality. This project will establish novel approaches to common tasks enabling hardware manufacturers to analyze system performance data while preserving strong privacy guarantees. This advances the field by demonstrating how to maintain data utility while protecting sensitive information in real-world applications.

The research contribution includes documenting privacy-utility trade-offs and establishing guidelines for privacy budget allocation across multiple analysis tasks. Our work will demonstrate practical privacy considerations in telemetry analysis while protecting users' participation in datasets. The methodologies developed can be adapted by other researchers working with sensitive telemetry data.

A.1.5 Success Criteria

The success of this project is dependent on a few factors. The first two are team collaboration and schedule adherence. There are many tasks that can be privatized and there may be unique challenges for each (hence the value in sharing these!). With one-quarter complete with group work on our privatized logistic regression paper, our group is confident in our communication, task management, and problem-solving abilities. Paired with our mentor Yu-Xiang Wang, an expert in the field of differential privacy, and a seasoned professor, we are equipped to find innovative and theoretically founded methods for privatizing data tasks.

The other requirements for this project rely on data access and task availability. The Intel data is proprietary, and we have signed agreements to use the data for research, however strict access and usage terms have not been given to us yet. Previous students have worked with the contact/program at Intel successfully and we are reassured by them that we will have a usable telemetry dataset by the start of the quarter. Similarly, there is a set of non-privatized tasks completed on this dataset by previous data scientists, their work is the foundation which we will build off of to show utility is possible even with privacy. These projects were successful implementations on the specific dataset we will have access to, this pairing therefore will continue to bear fruit as we privatize the tasks and compare baselines.

Lastly, although we have not reviewed the dataset and tasks yet (no access), the intel program is sharing genuine telemetry information from devices with given consent as part of their program. Additionally, this HDSI-Intel partnership has been cooperating since 2020 and HDSI has used hundreds of terabytes of information.