Novel Techniques in Private 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

Is the abstract supposed to be in light gray as well? Seems difficult to read. This research investigates the practical implementation of differential privacy mechanisms for telemetry data analysis, with a focus on real-world applications. We propose a comprehensive framework that employs various privacy-preserving techniques, including randomized response and the Laplace mechanism, to protect sensitive information while maintaining analytical utility. Our methodology encompasses multiple statistical tasks, from user-level rate analysis to logistic regression classification. The study utilizes AutoDP for precise privacy loss measurement and documents the inherent tradeoffs between privacy guarantees and analytical accuracy in production environments. By demonstrating the feasibility of differential privacy in telemetry analysis, we provide a roadmap for organizations seeking to enhance their privacy practices.

Website: `https://endurable-gatsby-6d6.notion.site/DP-Telemetry-14556404e74780818747cbe76de2e04a?pvs=4`[Notion until actual site is created]
Code: `https://github.com/Trey-Scheid/Novel-Techniques-in-Private-Data-Analysis`

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 identify comprehensive frameworks that integrate various differential privacy mechanisms, including Guassian Composition and the Laplace mechanism, to protect sensitive information in telemetry data. Our methodology encompasses multiple statistical tasks, from user-level rate analysis to logistic regression classification, and utilizes AutoDP for precise privacy loss measurement. By evaluating the tradeoffs between privacy guarantees and analytical accuracy in production settings, we provide a roadmap for organizations looking to enhance their privacy practices.

1.1 Motivation

Despite the growing importance of privacy-preserving data analysis, many practitioners perceive differential privacy implementation as complex and challenging **?**. This perception stems from several factors: the mathematical complexity of privacy definitions, the need to carefully calibrate privacy parameters, and concerns about reduced utility **?**. A survey by Smith et al. found that only 23% of data scientists felt confident implementing differential privacy mechanisms in their workflows **?**. However, 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. Additionally, educational resources and practical tutorials have emerged to guide practitioners through implementation challenges **?**. 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] linkneeded

[2] linkneeded

[3] linkneeded

1.2 Background and Literature Review

1.3 Differential Privacy

Differential privacy is a framework for data privacy that gives a mathematical guarantee that the sensitivity of each individual in the dataset is preserved. The core idea is to introduce random noise to the output of algorithms so that any single individual's data does not significantly affect the overall result. Mathematically, a mechanism is considered (,) 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 loss parameters. Smaller and imply stronger privacy guarantees.

Differential privacy is applied to algorithms, not datasets. One common and foundational algorithm is logistic regression. Many privatized implementations of logistic regression exist, leaving data scientists with a host of convoluted choices and complex language about parameters they may not fully understand. We hope to show some examples that will help practitioners implement this model on their own datasets.

1.4 Intel Telemetry Data

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 **?** and linkage attacks can be used to recover or reconstruct the original information: the source. 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].

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

1.4.1 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. **?**

2 Methods

2.1 Data Preprocessing

Add some language here about steps that were universal between all tasks if any

For each of the follow sections we will describe the task, the algorithm used, and the implementation details.

2.1.1 Logistic Regression (DP-SGD)

This paper[4] 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.

2.2 Correlation (via Logistic Regression Coefficient)

This paper [5] 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.

2.2.1 LASSO Regression (DP-FW)

will add lots of detail about lasso, then talk about adapting franke-wolfe to be differentially private.

2.2.2 K-Means (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

[4] needs citation not a footnote

[5] needs citation

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)[6] 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.

2.2.3 Z-score (Additive Noise)

As Z-score is computed before performing K-means clustering,

$$Z = \frac{X-\mu}{\sigma}$$

One can privatize this clustering task by simply adding Laplacian noise to the Z-scores, though the privacy gurantee and performance between the two methods, DP-Lloyd's and Additive Noise are likely to be different.

$$Z_{\text{private}} = \frac{X-\mu}{\sigma} + \text{Lap}\left(\frac{\Delta f}{\epsilon}\right)$$

$$\begin{aligned} \Delta f &\text{ is the global sensitivity of the Z-score computation,} \\ \epsilon &\text{ is the privacy parameter.} \end{aligned}$$

2.3 Probabilty (Additive Noise)

The probability of one or more uncorrected errors occurring on a system given the number of corrected errors it experienced during a specific time period is given as

$$P(Uncorrected|Corrected) = \frac{P(Corrected \cap Uncorrected)}{P(Corrected)}$$

To privatize this mechanism, one can apply two methods, adding noise to the numerator and denominator, as well as clipping. The noise is sampled from a Laplace distribution with scale parameter $\frac{\nabla f}{\epsilon}$, where ∇f is the sensitivity and ∇f is the privacy paramater. The sensitivity in this case for both the numerator and denominator will be equal to 1 because the max change in probability is equal to 1.

In addition to the noise that is added we will regularize the data by adding a constant, λ. This constant will also be added to both the numerator and denominator so that the denominator does not get too close to 0. Clipping the dataset involves capping the values of data points to a predefined range, preventing extreme values from disproportionately influencing analysis or model training. We will make sure that each GUID is limited in the amount of datapoints that it can contribute.

[6] linkneeded

3 Results

Should we do a results section for each task separately again?

3.1 Combined Results

We have discussed with Yu-Xiang a plot we can create which combines all the tasks into 1.

4 Discussion

4.1 Interpretation

4.2 Limitations

5 Conclusion

5.1 Summary

5.2 Impact

5.3 Future Direction

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. implemented the algorithms in ... B.N. analyzed the experimental results ... T.K. analyzed the experimental results ... 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 the laplace mechanism and clipping on both datasets
-
  Todo: Figure out how to privatize heat map

Bradley Nathanson

- Replicated K-means clustering using Z-scores from the Clustering Devices Together Using To Detect Change Patterns Paper
- Implemented non-private K-means clustering
  Todo: Implementation of private K-means clustering using either private Lloyd's algorithm or privatizing Z-scores before clustering

Christopher Lum

- Replicated Logistic Regression analysis using DP-SGD
- Introduced methods for engineering data
  Todo: Implement private Logistic Regression and compare to non-private model

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 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 such as: __, __, __, and __. Especially those which engaged in discussion with us about the field (Smith, Ulman, Guatam et al.). We are grateful for their contributions.

References

Appendices

A.1 Project Proposal

[12pt,letterpaper]article style/dsc180reportstyle

B Proposal

B.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.

B.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 analyzing 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.

B.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.

B.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.

B.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.