Privacy-Aware Data Analysis: Recent Developments for Statistics and Machine Learning

Yuliia Lut

Submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy
under the Executive Committee
of the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2022

Abstract

Privacy-Aware Data Analysis: Recent Developments for Statistics and Machine Learning

Yuliia Lut

Due to technological development, personal data has become more available to collect, store and analyze. Companies can collect detailed browsing behavior data, health-related data from smartphones and smartwatches, voice and movement recordings from smart home devices. Analysis of such data can bring numerous advantages to society and further development of science and technology. However, given an often sensitive nature of the collected data, people have become increasingly concerned about the data they share and how they interact with new technology. These concerns have motivated companies and public institutions to provide services and products with privacy guarantees. Therefore, many institutions and research communities have adopted the notion of differential privacy to address privacy concerns which has emerged as a powerful technique for enabling data analysis while preventing information leakage about individuals. In simple words, differential privacy allows us to use and analyze sensitive data while maintaining privacy guarantees for every individual data point. As a result, numerous algorithmic private tools have been developed for various applications. However, multiple open questions and research areas remain to be explored around differential privacy in machine learning, statistics, and data analysis, which the existing literature has not covered.

In Chapter 1, we provide a brief discussion of the problems and the main contributions that are

presented in this thesis. Additionally, we briefly recap the notion of differential privacy with some useful results and algorithms.

In Chapter 2, we study the problem of differentially private change-point detection for unknown distributions. The change-point detection problem seeks to identify distributional changes in streams of data. Non-private tools for change-point detection have been widely applied in several settings. However, in certain applications, such as identifying disease outbreaks based on hospital records or IoT devices detecting home activity, the collected data is highly sensitive, which motivates the study of privacy-preserving tools. Much of the prior work on change-point detection—including the only private algorithms for this problem—requires complete knowledge of the pre-change and post-change distributions. However, this assumption is not realistic for many practical applications of interest. In this chapter, we present differentially private algorithms for solving the change-point problem when the data distributions are unknown to the analyst. Additionally, we study the case when data may be sampled from distributions that change smoothly over time rather than fixed pre-change and post-change distributions. Furthermore, our algorithms can be applied to detect changes in linear trends of such data streams. Finally, we also provide a computational study to empirically validate the performance of our algorithms.

In Chapter 3, we study the problem of learning from imbalanced datasets, in which the classes are not equally represented, through the lens of differential privacy. A widely used method to address imbalanced data is resampling from the minority class instances. However, when confidential or sensitive attributes are present, data replication can lead to privacy leakage, disproportionally affecting the minority class. This challenge motivates the study of privacy-preserving pre-processing techniques for imbalanced learning. In this work, we present a differentially private synthetic minority oversampling technique (DP-SMOTE) which is based on a widely used non-private oversampling method known as SMOTE. Our algorithm generates differentially private synthetic data from the minority class. We demonstrate the impact of our pre-processing technique on the performance and privacy leakage of various classification methods in a detailed computational study.

In Chapter 4, we focus on the analysis of sensitive data that is generated from online internet activity. Accurately analyzing and modeling online browsing behavior play a key role in understanding users and technology interactions. Towards this goal, in this chapter, we present an up-to-date measurement study of online browsing behavior. We study both self-reported and observational browsing data and analyze what underlying features can be learned from statistical analysis of this potentially sensitive data. For this, we empirically address the following questions: (1) Do structural patterns of browsing differ across demographic groups and types of web use?, (2) Do people have correct perceptions of their behavior online?, and (3) Do people change their browsing behavior if they are aware of being observed? In response to these questions, we found little difference across most demographic groups and website categories, suggesting that these features cannot be implied solely from clickstream data. We find that users significantly overestimate the time they spend online but have relatively accurate perceptions of how they spend their time online. We find no significant changes in behavior throughout the study, which may indicate that observation had no effect on behavior or that users were consciously aware of being observed throughout the study.

Table of Contents

Acknowledgments xi

Chapter 1: Introduction and Background 1

1.1 Background on Differential Privacy 4

1.2 Change-Point Detection 8

1.2.1 Summary of our contributions 9

1.3 Imbalanced Learning 10

1.3.1 Summary of our contributions 11

1.4 Measurement and Analysis of Digital Behavior 12

1.4.1 Summary of our contributions 13

Chapter 2: Differentially Private Nonparametric Change-Point Detection 15

2.1 Introduction 15

2.1.1 Our contributions 16

2.1.2 Related Work 19

2.2 Preliminaries 21

2.2.1 Change-point background 21

2.2.2 Concentration inequalities 22

2.3 Offline private nonparametric change-point detection 23
2.3.1 Finite sample accuracy guarantee for the non-private nonparametric estimator 23
2.3.2 Private offline algorithm 29
2.4 Online change point detection 33
2.5 Application: Drift Change Detection 40
2.6 Empirical Results 42
2.6.1 Results of Offline Algorithm with Real Data 42
2.6.2 Offline Results with Synthetic Data 43
2.6.3 Online Results with Synthetic Data 47
2.7 Conclusions 49

Chapter 3: Private Tools for Imbalanced Learning 50
3.1 Introduction 50
3.1.1 Overview of Our Contributions 53
3.1.2 Related work 54
3.2 Preliminaries 55
3.3 DP-SMOTE Algorithm 57
3.4 Empirical Cost of Privacy in SMOTE 61
3.4.1 Methodology 62
3.4.2 DP-SMOTE and SMOTE perform similar to baseline under ROC metrics . 64
3.4.3 DP-SMOTE and SMOTE outperform baseline on minority-focused accuracy metrics 66
3.5 Empirical Comparison of Methods for Private Imbalanced Classification 69

3.5.1 Alternative methods for private imbalanced classification 69
3.5.2 Increased sensitivity from SMOTE as pre-processing 70
3.5.3 Experimental Results 75
3.6 Conclusions 76

Chapter 4: Measurement and Analysis of Digital Behavior 78
4.1 Introduction 78
4.1.1 Our Contributions 79
4.2 Related Work 80
4.3 Methods 82
4.3.1 Study Procedures 82
4.3.2 Data Collection 83
4.3.3 Data Analysis 87
4.3.4 Limitations 90
4.4 RQ1: Does browsing behavior differ across user demographic groups and type of web use? 91
4.4.1 Differences across demographic groups 92
4.4.2 Differences in behavior across types of web use 93
4.5 RQ2: Do people have correct perceptions of their behavior online? 95
4.5.1 Perceptions of time spent browsing 95
4.5.2 Perceptions of browsing activity by website category 97
4.6 RQ3: Do people change browsing behavior if they are aware of being observed? . . 99
4.6.1 Changes in level of activity 99

4.6.2 Changes in type of web use . . . 101
4.7 Discussion and Conclusions . . . 101

References . . . 104

Appendix A: Additional Figures for Chapter 3 . . . 117
A.1 Experimental results on other datasets . . . 117
A.1.1 Datasets . . . 117
A.1.2 Additional figures . . . 117

Appendix B: Additional Results for Chapter 4 . . . 125
B.1 Additional figures and tables . . . 125
B.2 Alternative Methodologies for RQ2 . . . 127
B.3 Screenshots of Study Materials . . . 130

List of Figures

2.1 Real data and accuracy results. 43

2.2 Empirical accuracy $\beta = \Pr[|\tilde{k} - k^*| > \alpha]$ of PNCPD from Monte Carlo simulations using Gaussian data, where pre-change data are drawn from $\mathcal{N}(0, 1)$ and post-change data are drawn from $\mathcal{N}(\mu_1, 1)$. Each simulation involves 10^3 runs of PNCPD with varying ϵ on data generated by 200 i.i.d. samples from appropriate distributions with $\mu_1 = 1$ or 5, and change point $k^* = 50, 100$, or 150. 44

2.3 Value for statistics $V(k)$ with (orange) and without (black) Laplace noise with privacy parameter $\epsilon = 5$ for varying settings for the size change and location of a change point. . . 46

2.4 Empirical accuracy $\beta = \Pr[|\tilde{t} - t^*| > \alpha]$ of PNCPD for drift detection. The data are generated from the drift change model with parameters $\eta = 1$, $\xi_0 = 0$, $\xi_1 = 5$, and e_t drawn from $\mathcal{N}(0, 1)$. These data are then modified as described in Section 2.5 so that the PNCPD algorithm can be applied. 47

2.5 Probability of inaccurate estimation and false alarm (left) and probability of inaccurate report conditioned on raising an alarm correctly (right) for Monte Carlo simulations. Data drawn from $\mathcal{N}(5, 1)$ pre-change and $\mathcal{N}(0, 1)$ post-change, with true change-point $k^* = 5000$. Each simulation involves 10^3 runs of ONLINEPNCPD with $\gamma = 0.1$, window size $n = 500$, threshold $T = 0.8$, and varying ϵ. 49

3.1 ROC Curves for multiple classifiers on diabetes datasets with varying preprocessing techniques. 64

3.2 ROC Curves for (a) Logistic Regression, (b) Random Forest classifiers varying preprocessing techniques: None, SMOTE, and DP-SMOTE on diabetes dataset. . . 65

3.3 Performance metrics (a) accuracy and (b) ROC-AUC for diabetes dataset for no pre-processing, SMOTE and DP-SMOTE under varying values of β with Logistic Regression classifier and $\alpha = 1, \epsilon = 1$ 66

3.4 Performance metrics (a) minority class accuracy, (b) G-mean, (c) F_1 for diabetes dataset for no pre-processing, SMOTE and DP-SMOTE under varying values of β with Logistic Regression classifier and $\alpha = 1, \epsilon = 1$. 68

3.5 ROC Curves SMOTE and DP-SMOTE with privacy budget of the resulting pipeline (a) $\epsilon = 1$, (b) $\epsilon = 5$, (c) $\epsilon = 10$. 76

4.1 Distribution of number of browsing actions (blue) and browsing time in hours (orange) on different website categories averaged over all participants in our study. 92

4.2 Empirical distributions of participants' *click*, *type*, and *urlChange* browsing actions within each website category. 94

4.3 (a) Scatter plot illustrating actual daily average browsing time vs. perceived (self-reported) number of hours spent browsing per day. Each point corresponds to one participant. (b) Distribution of error values δ_i in the participant population. 96

4.4 Confusion matrix showing accuracy of participants perceptions regarding the website categories they most frequently browse. Each participant i self-reported their k_i top website categories, and these were compared with their top k_i categories of observed browsing based on time spent browsing. Blue shaded cells indicate correct perceptions (true positives or true negatives), and orange shaded cells indicate incorrect perceptions (false positives or false negatives). Total correct and incorrect perceptions are also calculated for each category. 99

4.5 Average number of actions and time spent browsing, per participant per active browsing day of the study. 100

4.6 Proportion of (a) browsing actions and (b) time spent browsing on each website category on each day of the study. 101

A.1 ROC Curves for multiple classifiers on phoneme datasets with varying preprocessing techniques. 117

A.2 ROC Curves for multiple classifiers on abalone datasets with varying preprocessing techniques. 118

A.3 ROC Curves for (a) Logistic Regression, (b) Random Forest classifiers varying preprocessing techniques: None, SMOTE, and DP-SMOTE on diabetes dataset. . . 118

A.4 ROC Curves for (a) Logistic Regression, (b) Random Forest classifiers varying preprocessing techniques: None, SMOTE, and DP-SMOTE on phoneme dataset. . 118

A.5 ROC Curves for (a) Logistic Regression, (b) Random Forest classifiers varying preprocessing techniques: None, SMOTE, and DP-SMOTE on abalone dataset. . . 119

A.6 Performance metrics for diabetes dataset for no pre-processing, SMOTE and DP-SMOTE under varying values of balance parameter β with random forest classifier and $\alpha = 1, \epsilon = 1$. 120

A.7 Performance metrics for phoneme dataset for no pre-processing, SMOTE and DP-SMOTE under varying values of balance parameter β with logistic regression classifier and $\alpha = 1, \epsilon = 1$. 121

A.8 Performance metrics for phoneme dataset for no pre-processing, SMOTE and DP-SMOTE under varying values of balance parameter β with random forest classifier and $\alpha = 1, \epsilon = 1$. 122

A.9 Performance metrics for abalone dataset for no pre-processing, SMOTE and DP-SMOTE under varying values of balance parameter β with logistic regression classifier and $\alpha = 1, \epsilon = 1$. 123

A.10 Performance metrics for abalone dataset for no pre-processing, SMOTE and DP-SMOTE under varying values of balance parameter β with random forest classifier and $\alpha = 1, \epsilon = 1$. 124

B.1 Distribution of browsing actions performed on the 100 most browsed websites in the study (as measured by number of browsing actions), color-coded by category. . 125

B.4 (a) Scatter plot illustrating observed daily average browsing time vs. adjusted perceived (self-reported) number of hours spent browsing per day. Each point corresponds to one participant. Adjusted and original (non-adjusted, as in Section 4.5.1) points are shown. Red line $x = y$ corresponds to no error in perceptions. (b) Distribution of error values δ_i in the participant population based on the adjusted perceived values. The average error δ_i is -1.41 hours (SD=3.35), with 77.4% of participants over-estimating their time spent online. 130

B.6 Screenshots of the browsing extension in the Chrome browser as seen by the participants during the study. 131

B.2 Differences between the actual number of hours spent browsing per day and self-reported number of browsing hours per day, for each participant in the study. On non-active browsing days, the time spent browsing was set to zero. The x-axis enumerates the day of the experiment. The red horizontal line is a mean of these differences over the days of experiment. 132

B.3 (a) Scatter plot illustrating observed daily average browsing time vs. perceived (self-reported) number of hours spent browsing per day. Each point corresponds to one participant. Orange dots correspond to analysis with 5 minutes of inactivity as a cutoff, and the blue dots correspond to analysis with 30 minutes as a cutoff time as in Section 4.5.1. Red line $x = y$ corresponds to no error in perceptions. (b) Distribution of error values δ_i in the participant population using 5 minutes of inactivity as a cutoff. 133

B.5 Recruitment flyer 133

List of Tables

3.1 ROC AUC for diabetes dataset for various pre-processing methods: (None, SMOTE($k = 5, \alpha = 1$), DP-SMOTE($\ell = 2, \epsilon = 1, \alpha = 1$)) over different classification methods: logistic regression, random forest. 65

3.2 Pre-processing methods (vanilla, SMOTE($k = 5, \alpha = 1$), DP-SMOTE($\ell = 2, \epsilon = 1, \alpha = 1$)) for diabetes dataset across various metrics: G-Mean, F_1, ROC-AUC. The metrics are averaged for different classification methods: logistic regression and random forest. 68

3.3 Mean ROC-AUC values with standard error for SMOTE and DP-SMOTE with DP logistic regression. 75

3.4 Pre-processing methods (SMOTE($k = 5, \alpha = 1$), DP-SMOTE($\ell = 2, \epsilon = 5, \alpha = 1$)) for diabetes dataset across various metrics. The metrics are computed for DP logistic regression classifier. 76

4.1 Action types collected through the extension 85

4.2 Website categories, Symantec WebPulse Site Review [134] 86

4.3 Participant demographics 87

4.4 Test for equality of means of daily average number of browsing actions 93

4.5 Test for equality of means of daily average browsing time. 93

4.6 p-values for pairwise t-test for equality of means for perceptions δ_is across demographic groups. 97

B.1 p-values for Pearson's χ^2-test for homogeneity based on distribution of browsing actions within websites. Tests were performed pairwise for all website categories. . 126

B.2 Alexa Top Websites [135] categories offered in the pre-study survey, along with number of participants who named each category as among their most frequently browsed and number of participants for whom each category was among their observed top categories of browsing during the study. 127

B.3 p-values for pairwise t-test for equality of means of perception errors δ_is across demographic groups using 5 minutes of inactivity as a cutoff. 129

B.4 p-values for pairwise t-test for equality of means of perception errors δ_is across demographic groups using adjusted self-reports of browsing activity. 129

Acknowledgements

First and foremost, I would like to thank my mom Natella, who, despite the distance, has always been my main support and encouragement. Thank you for always being on my side and for making any place feel like home. I also want to thank my family: my grandma Tatiana, my late grandpa Tariel, my aunt Tanya, my cousin Olya and her husband Sasha, thanks for always being supportive and cheering me up during tough times. I love you all so much. I also want to thank my late dad Igor. I wish you could see the person I've become.

I want to thank my advisor Rachel Cummings for this fantastic opportunity and great five years of work. Thank you for introducing me to the research, always offering your advice and guidance, and your support, motivation, and patience. Thank you to my faculty collaborators, Sara Krehbiel, Elissa M. Redmiles, and Marco Avella-Medina. I am really grateful for your guidance and for sharing your experience with me. Finally, I want to thank the members of my defense and proposal committee: Kaizheng Wang, Shipra Agrawal, Pascal van Hentenryck and Yoa Xie. It was an honor to have you evaluate my work and to receive your valuable comments.

I want to mention my professors from Ukraine, who played a vital role in my academic and life path and helped me to grow as a person. Thank you to Vadym Kirman and Oleh Polyakov for being my teachers and mentors from high school through the university. I owe thanks to Valerii Turchyn, who convinced me to choose the Mechanics and Mathematics department and to major in Statistics.

I want to say a special thank you to Alfredo Torrico. Thank you for being an amazing, loving partner and best friend; for always believing in me and being my biggest fan; for helping me and being there every step of the way. I am grateful for your support and advice that helped me to grow as a researcher. We shared all ups and downs of grad school, academia, and the job market, supporting each other during the most difficult and the happiest times. And I'm excited about what comes next.

A huge thank you to my friends, Jana Boerger and Adrian Rivera Cardoso, who greatly supported me during the last years of my Ph.D. and throughout the job market. Thank you for sharing with me the happiest and the hardest moments.

Finally, I want to say thank you to all my friends. To my Ukrainian friends, Katerina Melnik, Yana Tachanska, and Inna Kurbanova. Thank you to Igor Molybog, who was massive support at the beginning of my Ph.D. I've been lucky to meet amazing people during these five years. To friends who I met during my Ph.D.: Arina Nikitina, Will Zhang, Seyma Gurkan, Idil Arsik, Mohamed El Tonbari, Matias Siebert, Berni Rios, Ramon Auad, Maira Moya Reyes, Alejandro Carderera, Kala Garapati, Sebastian Perez Salazar, Eyes Kareeratana, Wanrong Zhang, and Juba Ziani. Thanks to all of you and anyone I am missing. I am so glad we could share so many great moments together.

Chapter 1: Introduction and Background

The growing concern of big technology companies having too much power over users' data became apparent after the incident with Cambridge Analytica and Meta [1], formerly known as Facebook. This incident was followed by data leakage events in 2018 and 2021 [2, 3]. In public institutions, the interest in data privacy has also grown in the last few years. In particular, a critical turning point was the discovery that with the aid of commercial data, malicious attackers can re-identify precise information of almost 179 million people, corresponding to 58% of the population included in the 2010 Census[1]. Because of the possibility of malicious events, the US Census Bureau, along with academic experts, implemented a series of tools to address privacy concerns. As a result, the 2020 Decennial Census data was released using an algorithmic tool that operates under interpretable formal privacy guarantees known as differential privacy [4].

Differential privacy emerged as a dominant notion to define and quantify privacy in technical terms. It was first introduced in the seminal work of [5] and since then the literature and applications has grown exponentially, including data analysis [6, 7, 8, 9], recommender systems [10, 10, 11], deep learning [12, 13], genomics [14, 15], etc. Differential privacy guarantees that no individual data point can be learned from the output of a computation. This can be achieved by bounding the worst-case probability that a single data entry changes the output of the computation. Differential privacy allows the analyst to use sensitive data in research while addressing people's concerns about privacy. Multiple companies, such as Apple [16, 17, 18], Google [19, 20, 21], Microsoft [22, 23, 24], and Meta [25, 26, 27] have adopted the framework of differential privacy for data analysis, algorithm design and data release. Due to its popularity in research and industry

[1] Alabama v. Department of Commerce (2021)

communities, the study of differential privacy has expanded to multiple fields, such as machine learning, statistics, theoretical computer science, optimization, etc. However, with the increasing need for private tools in applications, there is still an extensive area to be explored around differential privacy, which is not covered by the existing literature. In the following, we briefly describe three settings where novel privacy-preserving tools are needed.

First, we study a commonly encountered in practice problem of identifying the point in which the underlying distribution of data has changed. This setting arises in multiple applications, such as medical condition monitoring (heart rate, sugar level) and video segmentation or speech recognition for smart home devices. For example, for a glucose monitor, the goal is to detect any abrupt change in sugar level and notify the user about this change as soon as possible. In several cases, the data associated with these applications are sensitive; therefore, there is a need to develop privacy-preserving tools that can adequately address privacy concerns while recognizing this change in distributions. One of the main challenges in this setting is that the data distribution is usually unknown. Most existing research has concentrated on parametric models that require full or partial knowledge of the distribution. On the other hand, privacy-preserving tools for nonparametric models, where distributions are unknown, have been studied much less extensively. In Chapter 2 of this thesis, we study the problem of privacy-preserving change-point detection in the nonparametric setting.

Second, as mentioned above, differential privacy has been the predominant technical notion of privacy, and numerous machine learning algorithms have been studied and designed around this definition. However, the commonly encountered problem of imbalanced learning has been overlooked in the differentially private learning literature. Roughly speaking, the problem of imbalanced learning appears when the classification task is performed on binary labeled data, and one of the classes is significantly smaller than the other. This challenge arises in multiple scenarios, such as fraud detection, disease detection, and weather prediction. In non-private settings, standard pre-processing techniques have been widely studied to adequately balance the classes

and, consequently, obtain improved learning outcomes. However, in several applications, such as the ones we just mentioned above, the smaller class contains sensitive information. Much of the research related to privacy-preserving learning for imbalanced data has focused on the task of private classification rather than the pre-processing techniques. However, a combination of non-private oversampling and the private classifier can lead to a higher privacy loss in the resulting model. In Chapter 3 of this thesis, we take a different perspective, and we address the problem of imbalanced learning by studying novel privacy-preserving pre-processing techniques.

Third, Internet activity data from consumers has been crucial for companies and other institutions, in particular, to increase their revenue and impact. For instance, according to the US Digital Ads Market report in 2022, made by Insider Intelligence [28], US digital ad spending will grow by nearly 50% in the next four years, and, by 2025, the digital ad market will top $300 billion. This process largely depends on the collection, sharing, and selling of user data, normally done without the user's approval. Because of this, there has been a growing concern among users about how companies handle information about them. In this context, it is necessary to take a step back and understand users' online behavior, which will consequently allow us to design appropriate tools to prevent the leakage of sensitive information. Towards this, in Chapter 4 of this thesis, we study potentially sensitive browsing data and analyze what underlying features can be learned from statistical analysis. We suggest that our findings can be used towards creating private obfuscation tools that prevent potential misuse of browsing data.

In the remainder of this chapter, in Section 1.1, we briefly recap the notion of differential privacy together with some useful results and algorithms. In Section 1.2, we briefly discuss the change-point detection problem and the need for privacy-preserving tools in the non-parametric setting. In Section 1.3, we discuss the challenge of imbalance learning in private settings and related previous approaches. Finally, in Section 1.4, we provide a brief discussion on online browsing behavior and how new privacy tools are needed to address the challenge of handling sensitive data. In each of the sections above, we also provide a summary of the main contributions that are

presented in this thesis.

1.1 Background on Differential Privacy

Differential privacy emerged as a powerful technical definition in machine learning and theoretical computer science to address privacy concerns using formal algorithmic tools. Informally, differential privacy bounds the effect of any individual's data in computation and ensures that very little can be inferred about an individual from the output of the differentially private analysis. In other words, differential privacy bounds the maximum amount that a single data entry can affect analysis performed on the database. Two databases $D, D' \in \mathcal{D}^n$ are *neighboring* if they differ in one entry. Let $\mathcal{R}$ be a generic output space. Formally, for a given $\epsilon \geq 0$, differential privacy is defined as follows:

Definition 1.1 (Differential Privacy [29])**.** *An algorithm $\mathcal{M} : \mathcal{D}^n \to \mathcal{R}$ is* ϵ-differentially private *if for every pair of neighboring databases $D, D' \in \mathcal{D}^n$, and for every subset of possible outputs $\mathcal{S} \subseteq \mathcal{R}$,*

$$\Pr[\mathcal{M}(D) \in \mathcal{S}] \leq e^{\epsilon} \cdot \Pr[\mathcal{M}(D') \in \mathcal{S}].$$

In general, differential privacy is achieved algorithmically by adding noise that scales with the *sensitivity* of a computation, which is the maximum change in the function's value that can be caused by changing a single entry in the database. In the following, we formally define the sensitivity of a function f that maps a dataset D to the space of real numbers $\mathbb{R}$:

Definition 1.2 (Sensitivity)**.** *Given a function or query $f : \mathcal{D}^n \to \mathbb{R}$, we define its* sensitivity *as:*

$$\Delta f = \max_{\textit{neighbors } D, D'} |f(D) - f(D')|.$$

One of the most well-known techniques for achieving differential privacy is adding Laplace noise. The *Laplace distribution* with scale b is the distribution with probability density function:

$\text{Lap}(x|b) = \frac{1}{2b}\exp\left(-\frac{|x|}{b}\right)$. We will write $\text{Lap}(b)$ to denote the Laplace distribution with scale b, or (with a slight abuse of notation) to denote a random variable sampled from $\text{Lap}(b)$. The choice of Laplace noise is not arbitrary, the distribution has two important qualities: it is symmetric, and the tails decay exponentially, which ensures that there is a small probability that we add a large amount of noise. We now formally define the Laplace Mechanism [29]:

Definition 1.3. *(Laplace Mechanism [29]). Given $\epsilon \geq 0$ and any function $f : \mathcal{D}^n \to \mathbb{R}$, the* Laplace Mechanism *is defined as:*

$$\mathcal{M}_{Lap}(D, f, \epsilon) = f(D) + \textit{Lap}(\Delta f/\epsilon).$$

For algorithms with non-numeric outputs, a common technique to achieve privacy is the Exponential Mechanism introduced in [30]. The output of the Exponential Mechanism is sampled from a range R with a probability that depends exponentially on a given score function q. This ensures that the algorithm is exponentially more likely to output outcomes with higher scores, which preserves high accuracy while adding privacy.

Definition 1.4 (Exponential mechanism [30])**.** *Let $q : \mathcal{D}^n \times R \to \mathbb{R}$ be a utility function defined over a space of datasets $\mathcal{D}^n$ and a domain of discrete outputs R. Given $\epsilon \geq 0$, the* Exponential Mechanism *is defined as a mechanism $\mathcal{M}_{Exp}$ that returns the output $r \in R$ with probability proportional to $e^{\frac{\epsilon q(D,r)}{2\Delta q}}$, where* sensitivity Δq *is defined as*

$$\Delta q = \max_{r \in R} \max_{D, D' \in neighbors} |q(D, r) - q(D', r)|.$$

Now we discuss some useful properties of differential privacy that play an essential role in designing privacy-preserving algorithms. One valuable property of differential privacy is that it *composes*, meaning that the privacy parameter degrades gracefully as additional computations are performed on the same database. This property allows us to design a privacy-preserving algorithm

by combining multiple differentially private algorithms and using them as building blocks.

Theorem 1.1 (Basic Composition [29]). *Consider $\mathcal{D}_1^n, \mathcal{D}_2^n$ two dataset spaces and $\mathcal{R}_1, \mathcal{R}_2$ two generic output spaces. Let $\mathcal{M}_1 : \mathcal{D}_1^n \to \mathcal{R}_1$ be an algorithm that is ϵ_1-differentially private, and let $\mathcal{M}_2 : \mathcal{D}_2^n \to \mathcal{R}_2$ be an algorithm that is ϵ_2-differentially private. Then their composition $\mathcal{M}_{1,2}(D) = (\mathcal{M}_1(D), \mathcal{M}_2(D))$ is $(\epsilon_1 + \epsilon_2)$-differentially private.*

A second important property of differential privacy is robustness to *post-processing*. This means that any further analysis of the differentially private output cannot decrease the privacy guarantees. Formally, the post-processing property is defined as follows:

Theorem 1.2 (Post-processing [29]). *Let $\mathcal{M} : \mathcal{D}^n \to \mathcal{R}$ be an algorithm that is ϵ-differentially private. Let $f : \mathcal{R} \to \mathcal{R}'$ be an arbitrary deterministic mapping. Then $f \circ \mathcal{M} : \mathcal{D}^n \to R'$ is ϵ-differentially private.*

To finish this Section, we briefly review some practical differentially private algorithms that we use later in Chapters 2 and 3.

First, we would like to highlight the REPORTMAX algorithm originally introduced in [31]. We formally state this method in Algorithm 1. Simply put, this method takes a collection of queries as input, computes a noisy answer to each query, and returns the index of the query with the largest noisy value.

Algorithm 1 Report Noisy Max: REPORTMAX$(X, \Delta, \{f_1, \ldots, f_m\}, \epsilon)$

Input: database X, set of queries $\{f_1, \ldots, f_m\}$ each with sensitivity Δ, privacy parameter ϵ

for $i = 1, \ldots, m$ **do**

  Compute $f_i(X)$

  Sample $Z_i \sim \text{Lap}(\frac{\Delta}{\epsilon})$

end for

Output $i^* = \underset{i \in [m]}{\text{argmax}} \, (f_i(X) + Z_i)$

Theorem 1.3 ([31]). REPORTMAX *is ϵ-differentially private.*

The second method that we would like to formalize is the ABOVETHRESHOLD algorithm which was first introduced in [32] and later refined into its current form in [33]. We formally define this method in Algorithm 2. This algorithm takes as an input a potentially unbounded stream of queries, compares the answer of each query to a fixed noisy threshold, and halts when it finds a noisy answer that exceeds the noisy threshold.

Algorithm 2 Above Noisy Threshold: ABOVETHRESHOLD$(X, \Delta, \{f_1, f_2, \ldots\}, T, \epsilon)$

Input: database X, stream of queries $\{f_1, f_2, \ldots\}$ each with sensitivity Δ, threshold T, privacy parameter ϵ

Let $\hat{T} = T + \text{Lap}(\frac{2\Delta}{\epsilon})$

for each query i **do**

  Let $Z_i \sim \text{Lap}(\frac{4\Delta}{\epsilon})$

  if $f_i(X) + Z_i > \hat{T}$ **then**

    Output $a_i = \top$

    Halt

  else

    Output $a_i = \perp$

  end if

end for

Theorem 1.4 ([32]). ABOVETHRESHOLD *is ϵ-differentially private.*

Privacy has become an important issue that must be addressed. Multiple technology companies and other institutions, such as Google, Apple, and the U.S. Census Bureau, have increasingly used differential privacy as a primary technical gold standard for designing their algorithmic tools and products. This increasing interest from the private and public sectors shows the usefulness and impact that differential privacy has had over the last years. Research around differential privacy has

grown exponentially in recent years and has expanded over different fields, including optimization, computer science, and machine learning. For other fundamental results and a more comprehensive discussion about differential privacy, we refer the interested reader to the exciting book by Cynthia Dwork and Aaron Roth [31].

1.2 Change-Point Detection

In statistics, we often observe a problem when a stream of random points changes distribution abruptly or smoothly over time. For instance, climate time series often exhibit long-term trends from changes in climate conditions; however, they also show that abrupt changes in distribution can occur from undocumented changes in recording instruments or station settings [34, 35]. To adjust for these artificial shifts, analysts need to detect the precise time when they occurred, which is an example of a change-point detection problem. Formally, the *change-point detection problem* seeks to identify distributional changes in data streams. This problem assumes that data points are initially sampled from a pre-change distribution, and then at an unknown time that we call a *change point*, data points are sampled from a post-change distribution. The task is to quickly and accurately identify the change point. The change-point detection problem has been extensively studied in theoretical statistics [36, 37, 38, 39, 40] as well as practical applications including climatology [35], econometrics [41], and DNA analysis [42].

Much of the previous work on change-point detection has focused on the setting where the pre- and post-change distributions are perfectly known to the analyst. This setting is undesirable and unrealistic in practice as it assumes perfect distributional knowledge. In practice, an analyst may only have access to the current (pre-change) distribution and may wish to detect a change to any distribution that is sufficiently far from the current distribution without making specific parametric assumptions on the future (post-change) distribution. On the other hand, in many applications, change-point detection algorithms are applied to sensitive data and may require formal privacy guarantees. For example, the Center for Disease Control (CDC) may wish to analyze hospital

records to detect disease outbreaks, or the Census Bureau may wish to analyze income records to detect changes in employment rates. As our privacy notion, we use differential privacy, as previously defined in Section 1.1.

In Chapter 2, we study the change-point detection problem under the *nonparametric* setting, where these distributions are unknown to the analyst. Specifically, we address the challenge of nonparametric change-point detection in both offline and online settings. In the offline case, the entire database is given up front, and the analyst seeks to estimate the change-point with a small additive error. In the online case, the analyst observes the stream of data points sequentially over time.

1.2.1 Summary of our contributions

We provide differentially private algorithms for accurate nonparametric change-point detection in offline and online settings. To design our main algorithmic tools, we first focus on the non-private setting. In this case, we improve the best previously-known [43] finite sample accuracy guarantee of this estimation procedure from sub-linear to constant in the sample size additive error. With this improved accuracy bound, we design an algorithm to make this estimation procedure differentially private in offline and online settings. We show that for the offline setting, adding privacy to the estimation procedure does not create any dependence on the sample size in the accuracy guarantee. Our offline algorithm uses the REPORTMAX framework which is formally described in Algorithm 1 in Section 1.1. In the online case, we show that our differentially private procedure achieves a logarithmic additive error (in the sample size). The main challenge in this setting is that we receive one data point at a time, and if the true change occurs later in the data stream, there is a high chance of false positive error. To prevent this, our algorithm uses the ABOVETHRESHOLD framework described in Algorithm 2. The ABOVETHRESHOLD algorithm chooses a fixed size interval of the data stream that contains the true change point with high probability. When this interval is found, our online algorithm runs the offline algorithm on it. We also

show how our results can be applied to settings where data are not sampled i.i.d., but are instead sampled from distributions changing smoothly over time.

Finally, we present an empirical study, which validates our theoretical guarantees, and provides evidence that our algorithms perform well on both synthetic and real data. For instance, our empirical findings show that our algorithm for the offline setting achieves a much higher detection accuracy when the true change point is closer to one of the edges of the data stream.

1.3 Imbalanced Learning

The problem of imbalanced learning arises when one class of the data is significantly smaller than the other class. This often happens when samples from one class rarely appear, e.g., fraudulent bank transactions or natural disasters such as earthquakes. In particular, for classification tasks that aim to predict a class label for a given instance of data, standard methods such as Random Forest and Logistic Regression fail to achieve high classification accuracy when data is imbalanced [44]. Moreover, in many applications, such as the detection of rare diseases, the data is not only imbalanced but also sensitive. Despite the broad implementations of differentially private tools in machine learning (we refer the interested reader to the survey [45] and the references therein), privacy-preserving imbalanced classification has been much less studied in the literature [46, 47].

Imbalanced learning has been widely studied in non-private settings [48, 49, 50]. In practice, we usually aim to balance classes, which can be achieved by randomly removing samples from the majority class, under-sampling, or by increasing the number of samples in the minority class, called over-sampling, which can be done by random replication or more sophisticated algorithmic tools. A common oversampling method is the Synthetic Minority Over-sampling Technique (SMOTE) [44]. Simply put, the algorithm generates synthetic minority examples by sampling them from the lines that connect minority samples with their k nearest neighbors.

In private settings, private imbalanced classification is usually achieved by a pipeline of a non-private pre-processing method and the corresponding privacy-preserving classification method.

However, over-sampling treatment increases the privacy loss of the private classification [51]. The sensitivity of the classifier increases when we add a large amount of synthetically generated data that heavily depends on the minority class samples. Therefore, we need to add more noise to the privacy-preserving classification method to keep the chosen privacy budget.

In Chapter 3, we take a different approach to privacy-preserving imbalanced learning. We specifically focus on the pre-processing task and design a private version of SMOTE, which we call DP-SMOTE, which ensures privacy guarantees while not losing the classification accuracy. In an empirical study, we focus on two main directions to validate the performance of our method. Firstly, we assess the performance of DP-SMOTE in a pipeline with a non-private classifier by comparing our method to the original SMOTE. While the combination of DP-SMOTE and a non-private classifier is not fully differentially private, it shows that adding privacy does not hurt the accuracy of the classifier. Moreover, in some cases, it improves the accuracy of the minority class. Secondly, we show that the treatment of data with DP-SMOTE before differentially private classification improves the accuracy of the classifier when compared to the original SMOTE.

1.3.1 Summary of our contributions

We develop a new differentially private tool for tackling the problem of imbalanced learning. Our algorithm is based on a classic framework for imbalanced data known as SMOTE [44]. Roughly speaking, our technique works as follows: first, it creates a private noisy histogram of the minority samples, then the algorithm uses it as a differentially private proxy of the minority class and applies SMOTE directly to it. This method ensures that we achieve good accuracy while using a minimal amount of privacy budget. Our main contributions are twofold:

1. We propose a novel framework for the generation of differentially private synthetic data that can be used to balance data with classes of different sizes. We show that for a non-private classifier, our approach improves the accuracy of the minority class while not considerably affecting the overall accuracy.

2. We create an alternative method for DP imbalanced classification that consists of DP-SMOTE and a privacy-preserving classifier. Our algorithm ensures a lower sensitivity of the resulting pipeline and, as a result, achieves better accuracy than a combination of non-private SMOTE with the corresponding private classifier.

1.4 Measurement and Analysis of Digital Behavior

In the last decade, the use of internet activity data has been crucial for companies and other institutions. However, users have raised multiple concerns related to the use of their data. For instance, a recent survey has shown that at least 86% of the respondents show some degree of concern about the way tech companies handle information about them [52]. In particular, Internet Service Providers (ISPs) collect, share, and sell sensitive user data without the user's approval. Other institutions, such as insurance companies or banks, may use this data to infer users' sensitive characteristics. While users cannot hide their browsing data from ISPs, researchers have proposed several solutions to address this concern [53, 54]. In particular, one possible solution is to obfuscate browsing data with noise by issuing randomized search queries or randomly clicking on ads. However, these methods do not consider the structure of the data and attributes that require protection. Therefore, there is a need to understand what kind of noise can obfuscate or hide users' sensitive attributes and browsing habits. Towards this goal, we must take a step back and learn what underlying attributes can be implied from the collected browsing data and, therefore, have to be protected.

A vast majority of prior work measuring users' browsing behavior was conducted using proprietary, industry data to which the majority of academic researchers do not have access (see, e.g., [55, 56, 57]). As an alternative, researchers often rely on users' self-reports of their online behavior [58, 56] or observe and measure users' behavior directly in an experimental study [56]. However, both methods are not without limitations. The accuracy of self-report data can suffer due to response bias [59, 60, 61]. Additionally, due to limitations of most experimental studies, participants are

usually aware of being observed and, as a result, may behave differently (e.g., [62, 63, 64, 65, 66]). Therefore, such observed browsing data may not represent real browsing behavior.

Intending to contribute towards creating private tools that prevent potential misuse of browsing data, in Chapter 4, we aim to understand users' online behavior by doing a field study. To approach the understanding of browsing data, we apply and compare both self-report and observational methods to provide an up-to-date understanding of both users' browsing behavior and compare two academically-feasible methods for measuring this behavior. To the best of our knowledge, the most recent measurement study of online browsing behavior was conducted in 2013 [67]. To do so, we designed and conducted a user experiment ($n = 31$) in which we surveyed participants about their browsing behavior and continuously observed participants' browsing behavior for 14 days. Using these data, we address the following research questions:

(RQ1) Does browsing behavior differ across user groups (i.e., demographics) and types of web use?

(RQ2) Do people have accurate perceptions of their behavior online? Does perception accuracy differ by user group or type of web use?

(RQ3) Do people change their browsing behavior if they are aware of being observed?

1.4.1 Summary of our contributions

Based on the results of our study, we observe that people spend much more time online relative to prior work conducted in 2010 [55]. We found little difference across demographic groups, suggesting that demographic features cannot be implied solely from clickstream data. Similarly, we find few significant differences in within-website browsing behavior across different categories of websites.

Also, we found that, in general, participants do not have an accurate perception of their browsing habits, and therefore, self-reported data is not an accurate substitute for the original data. Specifically, we find that people substantially overestimate their time spent online and that this

inaccurate perception does not differ among the demographic groups. However, we find that people have roughly accurate perceptions of their top-browsed website categories. Finally, we do not find changes in either level of browsing activity or in the distribution of browsing across website categories over time during the study, which could indicate that people do not change their behavior when aware of being observed.

Chapter 2: Differentially Private Nonparametric Change-Point Detection

2.1 Introduction

As we discussed in Section 1.2, the change-point problem has been widely studied in theoretical statistics [36, 37, 38, 39, 40] and finds numerous applications including climatology [35], econometrics [41], and DNA analysis [42]. The goal in the *change-point detection problem* is to identify distributional changes in streams of data. Formally, given a dataset $X = \{x_1, \ldots, x_n\}$, where n can be finite or infinite, such that data points $\{x_1, \ldots, x_{k^*}\}$ are sampled from a pre-change distribution P_0, and data points $\{x_{k^*+1}, \ldots, x_n\}$ are sampled from a post-change distribution P_1, the goal is to quickly and accurately identify the unknown time k^*, called a *change point time*, where the distribution changes from P_0 to P_1.

Despite the broad literature on change-point detection in statistics, in privacy literature, there is much less work on differentially private methodologies for change-point detection problem. To the best of our knowledge, the only two prior works on differentially private change-point detection [68, 69] considered the parametric setting, in which the analyst has access to P_0 and P_1 in advance. In this structured setting, the analyst could use algorithms tailored to details of these distributions, such as computing the maximum log-likehood estimator (MLE) of the change-point time.

In this chapter, we focus on the *nonparametric* setting, where these distributions are unknown to the analyst. This setting is closer to practice, as it removes the unrealistic assumption of perfect distributional knowledge. The nonparametric setting requires different test statistics, as common approaches like computing the MLE do not work without full knowledge of P_0 and P_1.

Unfortunately, most nonparametric estimation procedures are not amenable to differential privacy. Indeed, all prior work on private change-point detection has been in the parametric setting,

where P_0 and P_1 are known [68, 69]. A standard approach in the nonparametric setting is to first estimate a parametric model, and then perform parametric change-point detection using the estimated model. Common nonparametric estimation techniques include kernel methods and spline methods [70, 71] or nonparametric regression [72]. These methods are difficult to make private in part because of the complexity of finite sample error bounds combined with the effect of injecting additional noise for privacy. In contrast, simple rank-based statistics–which order samples by their value–have an easy sensitivity analysis.

In this chapter, we estimate nonparametric change-points using the Mann-Whitney test [73, 74], which is a rank-based test statistic, which we formally present in Section 2.2.1. This test picks an index k and measures the fraction of points before k that are greater than points after k. For the change-point problem, this statistic should be largest around the true change-point k^*, and smaller elsewhere (under mild non-degeneracy conditions on the pre- and post-change distributions). Also note that this statistic simply computes pairwise comparisons of the observed data, and it does not require any additional knowledge of P_0 or P_1 beyond the assumption that a data point from P_0 is larger than a data point from P_1 with probability $> 1/2$. The test statistic has sensitivity $O(1/n)$ for a database of size n, and is known to have lower sensitivity than most other test statistics for the same task [74].

2.1.1 Our contributions

In this chapter, we provide differentially private algorithms for accurate nonparametric change-point detection in both the offline and online settings. We also show how our results can be applied to settings where data are not sampled i.i.d., but are instead sampled from distributions changing smoothly over time.

In the offline case, the entire database is given up front, and the analyst seeks to estimate the change-point with small additive error. We use the Mann-Whitney rank-sum statistic and its extension to the change-point setting due to [43]. At every possible change-point time k, the

test measures the fraction of points before k that are greater than points after k, using statistic $V(k) = \frac{\sum_{j=k+1}^{n}\sum_{i=1}^{k} I(x_i > x_j)}{k(n-k)}$. The test then outputs the index $\hat{k}$ that maximizes this statistic. For the non-private settings, we improve the best previously-known finite sample accuracy guarantees of this estimation procedure. The previous non-private accuracy guarantee has $O(n^{2/3})$ additive error [43], whereas our Theorem 2.2 in Section 2.3.1 achieves $O(1)$ additive error.

Next, we present in Algorithm 3 the first, to the best of our knowledge, differentially non-parametric private change-point detection algorithm for finite dataset, which we call PNCPD. Our algorithm uses the REPORTMAX framework of [31] that takes in a collection of queries, computes a noisy answer to each query, and returns the index of the query with the largest noisy value. We formally present this method in Algorithm 3 in Section 2.3.2. We instantiate this framework with our test statistics $V(k)$ as queries, to privately select the argmax of the statistics. One challenge is ensuring that the test statistics $V(k)$ have low enough sensitivity that the additional noise required for privacy does not harm the estimation error by too much. We show that our PNCPD algorithm is differentially private (Theorem 2.3) and has $O(\frac{1}{\epsilon^{1.01}})$ additive accuracy (Theorem 2.4), meaning that adding privacy does not create any dependence on n in the accuracy guarantee.

In the online case, the analyst starts with an initial database of size n, and receives a stream of additional data points, arriving online. The analyst's goal here is to accurately estimate the change-point quickly after it occurs. This is a more challenging setting because the analyst will have very little post-change data if they want to detect changes quickly. In this setting, we design our method ONLINEPNCPD formally stated in Algorithm 4 in Section 2.4. This algorithm uses the ABOVETHRESHOLD framework of [32, 33]. The ABOVETHRESHOLD algorithm takes in a potentially unbounded stream of queries, compares the answer of each query to a fixed noisy threshold, and halts when it finds a noisy answer that exceeds the noisy threshold. We formally state ABOVETHRESHOLD in Algorithm 2 in Chapter 1. Our algorithm computes the test statistic $V(k)$ for the middle index k of each sliding window of the last n data points. Once the algorithm finds a window with a high enough test statistic, it waits for enough additional data points to meet

the requirements of our *offline* algorithm PNCPD for accuracy, and then calls PNCPD on the n most recent data points to estimate the change-point time. One technical challenge in the online setting is finding a threshold that is high enough to prevent false positives before a change occurs, and low enough that a true change will trigger a call to the offline algorithm. We show that our ONLINEPNCPD algorithm is differentially private (Theorem 2.5) and has $O(\log n)$ additive error (Theorem 2.6).

In Section 2.5, we apply our results to privately solve the problem of *drift change detection*, where points are not sampled i.i.d. pre- and post-change, but instead are sampled from smoothly changing distributions whose means are shifting linearly with respect to time, and the linear drift parameter changes at an unknown change-time k^*. We show how to reduce an instance of the drift change detection problem with non-i.i.d. samples to an instance of the change-point detection problem to which our algorithms can be applied. We show in Corollary 2.1 that our algorithms also provide differential privacy and accurate estimation for the drift change detection problem. We also suggest extensions of this reduction technique so that our algorithms may also be applied for non-linear drift change detection for other smoothly changing distributions that exhibit sufficient structure.

In Section 2.6, we report experimental results that empirically validate our theoretical results. We start by applying our PNCPD algorithm to a real-world dataset of stock price time-series data that appear by visual inspection to contain a change-point, and find that our algorithm does find the correct change-point with minimal loss in accuracy, even for small ϵ values. We then apply our PNCPD algorithm to simulated datasets sampled from Gaussian distributions, varying the parameters corresponding to the size of the distributional change, the location of the change-point in the dataset, and ϵ. We also perform simulations for our application to drift change detection by simulating data points drawn from the drift change model, performing the reduction described in Section 2.5, and applying our PNCPD algorithm to the resulting dataset. Lastly we apply our ONLINEPNCPD algorithm to streaming simulated datasets drawn from Gaussian distributions,

again varying the parameters that correspond to the size of the distributional change, the location of the change-point in the dataset, and ϵ. In all cases, the empirical accuracy corresponds qualitatively to what our theoretical results predict. Our empirical findings show that PNCPD achieves a much higher detection accuracy when the true change-point is closer to one of the edges of the data stream.

2.1.2 Related Work

Change-point detection is a canonical problem in statistics that has been studied for nearly a century; selected results include [36, 37, 38, 75, 76, 77, 39, 78, 79, 80, 81, 82, 40, 83, 84]. The problem originally arose from industrial quality control, and has since been applied in a wide variety of other contexts including climatology [35], econometrics [41], and DNA analysis [42]. In the parametric setting where pre-change and post-change distributions P_0 and P_1 are perfectly known, the Cumulative Sum (CUSUM) procedure [37] is among the most commonly used algorithms for solving the change-point detection problem. It follows the generalized log-likelihood ratio principle, calculating $\ell(k) = \sum_{i=k}^{n} \log \frac{P_1(x_i)}{P_0(x_i)}$ for each $k \in [n]$, and declaring that a change occurs if and only if $\ell(\hat{k}) \geq T$ for MLE $\hat{k} = \operatorname{argmax}_k \ell(k)$ and appropriate threshold $T > 0$. Nonparametric change-point detection has also been well-studied in the statistics literature [43, 85, 86], and requires different test statistics that do not rely on exact knowledge of the distributions P_0 and P_1.

The only two prior works on differentially private change-point detection [68, 69] both considered the parametric setting and employed differentially private variants of the CUSUM procedure and the change-point MLE underlying it. [68] directly privatized non-private procedures for the offline and online settings. [69] gave private change-point detection as an instantiation of a solution to the more general problem of private hypothesis testing, partitioning time series data into batches of size equal to the sample complexity of the hypothesis testing problem, and then outputs the batch number most consistent with a change-point. Both works assumed that the pre- and post-distributions were fully known in advance.

In our nonparametric setting, we use the Mann-Whitney test [73, 74] instead of the MLE that the CUSUM procedure is built on. The Mann-Whitney test was originally proposed as a rank-based nonparametric two-sample test, to test whether two samples were drawn from the same distribution using the null hypothesis that after randomly selecting one point from each sample, each point is equally likely to be the larger of the two. It was extended to the change-point setting by [43], for testing whether samples from before and after the hypothesized change-point were drawn from the same distribution. Given a database $X = \{x_1, \ldots, x_n\}$, for each possible change-point k, the test statistic $V(k) = \frac{\sum_{j=k+1}^{n} \sum_{i=1}^{k} I(x_i > x_j)}{k(n-k)}$ counts the proportion of index pairs (i, j) with $i \leq k < j$ for which $x_i > x_j$. This is a nonparametric test because it does not require any additional knowledge of the distributions from which data are drawn. Additionally, the Mann-Whitney test is known to be more efficient [87] and have lower sensitivity [74] than most other test statistics for the same task, including the Wald statistic [88] and the Kolmogorov-Smirnov test [89]. Differentially private versions of related test statistics have been used in recent unpublished work in the context of hypothesis testing, but they have not been applied to the change-point problem [90, 91].

Although the current paper largely follows the same structure as [68] for privatizing the change-point procedure, the analysis of the algorithm is vastly different, due to new challenges introduced by the nonparametric setting. Most test statistics for nonparametric estimation have high sensitivity, and therefore require large amounts of noise to be added to satisfy differential privacy. This means that off-the-shelf applications of nonparametric test statistics to the differentially private change-point framework of [68] would result in high error. Indeed, even with our use of the Mann-Whitney test statistic which was chosen for its low sensitivity, an immediate application of the best known finite-sample accuracy bounds [43] yielded additive error $O(n^{2/3})$ in the offline setting for databases of size n. To achieve our much tighter $O(\epsilon^{-1.01})$ error bounds required a new analysis.

2.2 Preliminaries

This section provides the necessary background for interpreting our results for the problem of private nonparametric change-point detection. Section 2.2.1 defines the nonparametric change-point detection problem, Section 1.1 describes the differentially private tools that will be brought to bear, and Section 2.2.2 gives the concentration inequality which will be used in our proofs.

2.2.1 Change-point background

Let $X = \{x_1, \ldots, x_n\}$ be n real-valued data points. The *change-point detection problem* is parametrized by two distributions, P_0 and P_1. The data points in X are hypothesized to initially be sampled i.i.d. from P_0, but at some unknown change time $k^* \in [n]$, an event may occur (e.g., epidemic disease outbreak) and change the underlying distribution to P_1. The goal of a data analyst is to announce that a change has occurred as quickly as possible after k^*. Since the x_i may be sensitive information—such as individuals' medical information or behaviors inside their home—the analyst will wish to announce the change-point time in a privacy-preserving manner.

In the standard non-private offline change-point literature, the analyst wants to test the null hypothesis $H_0 : k^* = n$, where $x_1, \ldots, x_n \sim_{\text{iid}} P_0$, against the composite alternate hypothesis $H_1 : k^* < n$, where $x_1, \ldots, x_{k^*} \sim_{\text{iid}} P_0$ and $x_{k^*+1}, \ldots, x_n \sim_{\text{iid}} P_1$. If P_1 and P_0 are known, the log-likelihood ratio of $k^* = \infty$ against $k^* = k$ will be given by

$$\ell(k, X) = \sum_{i=k+1}^{n} \log \frac{P_1(x_i)}{P_0(x_i)}.$$

The maximum likelihood estimator (MLE) of the change time k^* is given by $\operatorname{argmax}_{k \in [n]} \ell(k, X)$. However, note that to perform this test, the analyst must have complete knowledge of distributions P_0 and P_1 to compute the log-likelihood ratio.

In this chapter, we consider the situation that we do not know both the pre-change distribution

and the post-change distribution. We require no knowledge of the pre- and post- change distributions, and assume only that the probability of an observation from P_0 exceeding an observation from P_1 is different than the probability of an observation from P_1 exceeding an observation from P_0, which is necessary for technical reasons. The Mann-Whitney test [73] is a commonly used nonparametric test of the null hypothesis that it is equally likely that a randomly selected value from one sample will be less than or greater than a randomly selected value from a second sample. [43] proposed a modification of the Mann-Whitney test to solve the change-point estimation problem. For each possible change-point k, a test statistic counting the proportion of index pairs (i, j) with $i \leq k, j > k$ for which $x_i > x_j$ is calculated as follows:

$$V(k, X) = \frac{\sum_{j=k+1}^{n} \sum_{i=1}^{k} I(x_i > x_j)}{k(n-k)} \tag{2.1}$$

For data X drawn according to the change-point model with distributions P_0, P_1, this statistic is largest or smallest in expectation at the true change-point k^* depending on the value $a = \Pr_{x_0 \sim P_0, x_1 \sim P_1}[x_0 > x_1]$. If $a > 1/2$, we estimate the change-point by taking the arg max of the Mann-Whitney statistics; otherwise we take the arg min. When X is clear from context, we will simply write $V(k)$. The estimator $\hat{k}$ is understood to denote the argmax or argmin of $V(k)$ depending on whether $a > 1/2$.

We will measure the additive error of our estimations of the true change-point as follows.

Definition 2.1 ((α, β)-accuracy)**.** *A change-point detection algorithm that produces a change-point estimator $\tilde{k}$ is* (α, β)-accurate *if $\Pr[|\tilde{k} - k^*| > \alpha] \leq \beta$, where the probability is taken over randomness of the data X sampled according to the change-point model with true change-point k^* and (possibly) the randomness of the algorithm.*

2.2.2 Concentration inequalities

Our proofs will also use the following concentration inequality.

Theorem 2.1 (McDiarmid [92]). *Define the discrete derivatives of the function $f(X_1, \ldots, X_n)$ of independent random variables $X_1, \ldots, X_n$ as*

$$D_i f(x) := \sup_z f(x_1, \ldots, x_{i-1}, z, x_{i+1}, \ldots, x_n) - \inf_z f(x_1, \ldots, x_{i-1}, z, x_{i+1}, \ldots, x_n). \quad (2.2)$$

Then for $X_1, \ldots, X_n$ independent, $f(X_1, \ldots, X_n)$ is subgaussian with variance proxy $\frac{1}{4}\sum_{i=1}^n ||D_i f||_\infty^2$. In particular,

$$\Pr[f(X_1, \ldots, X_n) - \mathbb{E}\left[[\right] f(X_1, \ldots, X_n)] \geq t] \leq \exp(-\frac{2t^2}{\sum_{i=1}^n ||D_i f||_\infty^2}).$$

2.3 Offline private nonparametric change-point detection

In this section, we give an offline private algorithm for change-point detection when the pre- and post-change distributions are unknown. In Section 2.3.1, we first offer the finite sample accuracy guarantee for the non-private nonparametric algorithm given by $\hat{k} = \operatorname{argmax} V(k)$ for the test statistic $V(k)$ given in Equation (2.1), which will serve as the baseline for evaluating the utility of our private algorithm. Then in Section 2.3.2 we present our private algorithm, and give privacy and accuracy guarantees.

2.3.1 Finite sample accuracy guarantee for the non-private nonparametric estimator

In this section, we provide error bounds for the non-private nonparametric change-point estimator when the data are drawn from two unknown distributions P_0, P_1 with true change-point $k^* \in \{\lceil \gamma n \rceil, \ldots, \lfloor (1-\gamma)n \rfloor\}$, for some known $\gamma < 1/2$. This γ bounds away from the change-point occurring too early or too late in the sample, and is necessary to ensure sufficient number of samples from both the pre-change and post-change distributions. Without loss of generality, we assume that $a := \Pr_{x_0 \sim P_0, x_1 \sim P_1}[x_0 > x_1] > 1/2$.

For the non-private task, we use the following estimation procedure of [43], which calculates

the estimated change-point $\hat{k}$ as the argmax of $V(k)$ over all k in the range permitted by γ:

$$\hat{k} = \text{argmax}_{k \in \{\lceil \gamma n \rceil, \ldots, \lfloor (1-\gamma)n \rfloor\}} V(k),$$

for test statistic $V(k)$ defined in Equation (2.1). We show in Theorem 2.2 that the additive error of this procedure is constant with respect to the sample size n.

Our result is much tighter that the previously known finite-sample accuracy result in [43], which gave an estimation error bound of $O(n^{2/3})$. This sublinear result comes from a connection between the accuracy and the maximal deviation of $V(k)$ from the expected value over $[\gamma n, (1-\gamma)n]$. To bound the maximal deviation, [43] first analyzed the variance approximation of $V(k)$ to bound the deviation for a single point k. Then they utilized a Lipschitz property to partition $[\gamma n, (1-\gamma)n]$ to small intervals, and took a union bound over these intervals to yield a high probability guarantee. In contrast, we better leverage the connection between $V(k)$ and $V(k^*)$ for improved accuracy and a simplified proof. At a high level, we show that the expectation of $V(k)$ is single-peaked around k^*, and $V(k) - V(k^*)$ is subgaussian. We carefully analyze the discrete derivative as a function of $|k^* - k|$, γ, and n to use a concentration bound yielding our constant error result.

Theorem 2.2. *For data $X = \{x_1, \ldots, x_n\}$ drawn according to the change-point model with any distributions P_0, P_1 with $a = \Pr_{x \sim P_0, y \sim P_1}[x > y] > 1/2$, constraint $\gamma \in (0, 1/2)$, and change-point $k^* \in \{\lceil \gamma n \rceil, \ldots, \lfloor (1-\gamma)n \rfloor\}$, we have that the estimator*

$$\hat{k} = \text{argmax}_{k \in \{\lceil \gamma n \rceil, \ldots, \lfloor (1-\gamma)n \rfloor\}} \frac{\sum_{j=k+1}^{n} \sum_{i=1}^{k} I(x_i > x_j)}{k(n-k)}$$

is (α, β)-accurate for any $\beta > 0$ and

$$\alpha = C \cdot \left(\frac{1}{\gamma^4 (a - 1/2)^2} \right)^c \cdot \log \frac{1}{\beta}$$

for any constant $c > 1$ and some constant $C > 0$ depending on c.

If $a < 1/2$ we achieve the same error bound using $\hat{k} = \operatorname{argmin} \frac{\sum_{j=k+1}^{n} \sum_{i=1}^{k} I(x_i > x_j)}{k(n-k)}$.

Proof. We will show that for $\hat{k} = \operatorname{argmax} V(k)$ and α as in the theorem statement,

$$\Pr[\left|\hat{k} - k^*\right| > \alpha] \leq \sum_{k:|k-k^*|>\alpha} \Pr[V(k) > V(k^*)] \leq \beta.$$

To do this, we fix any $k \in \{\lceil \gamma n \rceil, \ldots, \lfloor (1-\gamma)n \rfloor\}$ and show that $f(X) = V(k) - V(k^*)$ is subgaussian. In particular, for k at least α away from k^*, the expectation of $V(k^*) - V(k)$ is sufficiently large and its discrete derivative is sufficiently small that the probability of $V(k) > V(k^*)$ can be tightly bounded as a function of α by application of Theorem 2.1.

First we give a lower bound the difference in expectation of $V(k^*)$ and $V(k)$. Observe that

$$\begin{aligned}
\mathbb{E}\left[\right] V(k)] &= \frac{\sum_{i \leq k, j > k} \Pr[x_i > x_j]}{k(n-k)} \\
&= \begin{cases} \frac{\frac{1}{2}(k^*-k)+a(n-k^*)}{n-k} & k \leq k^* \\ \frac{ak^*+\frac{1}{2}(k-k^*)}{k} & k > k^* \end{cases},
\end{aligned}$$

achieving its maximum at $\mathbb{E}\left[\right] V(k^*)] = a$. Therefore, we can bound

$$\begin{aligned}
\mathbb{E}\left[\right] V(k^*) - V(k)] &= \begin{cases} (a - \frac{1}{2})\frac{k^*-k}{n-k} & k \leq k^* \\ (a - \frac{1}{2})\frac{k-k^*}{k} & k > k^* \end{cases} \\
&\geq (a - \frac{1}{2})\frac{|k^* - k|}{n}. \qquad (2.3)
\end{aligned}$$

In the following bounds on the discrete derivative of $f(X) = V(k) - V(k^*)$, we will make use

of the fact that f can be written as:

$$\begin{aligned}
f(X) =& \frac{\sum_{j=k+1}^{n}\sum_{i=1}^{k} I(x_i > x_j)}{k(n-k)} - \frac{\sum_{j=k^*+1}^{n}\sum_{i=1}^{k^*} I(x_i > x_j)}{k^*(n-k^*)} \\
=& \left(\frac{1}{k(n-k)} - \frac{1}{k^*(n-k^*)}\right)\left(\sum_{\substack{i=1,\ldots,k,\\ j=k+1,\ldots,n}} I(x_i > x_j)\right) \\
&+ \frac{1}{k^*(n-k^*)}\left(\sum_{\substack{i=1,\ldots,k,\\ j=k+1,\ldots,n}} I(x_i > x_j) - \sum_{\substack{i=1,\ldots,k^*,\\ j=k^*+1,\ldots,n}} I(x_i > x_j)\right)
\end{aligned}$$

We bound the discrete derivative $D_i f$ separately for $i \leq \min\{k, k^*\}$, $i \in (\min\{k, k^*\}, \max\{k, k^*\}]$, and $i > \max\{k, k^*\}$. When x_i changes arbitrarily for $i \leq \min\{k, k^*\}$, we note that $\sum_{j=k+1}^{n} I(x_i > x_j)$ can change by at most $\pm(n-k)$ and $\sum_{j=k+1}^{k^*} I(x_i > x_j)$ can change by at most $\pm(k^*-k)$. These counts are normalized in f, and the normalization ensures this former count contributes at most $\frac{|k^*-k|}{k^*(n-k^*)} + \frac{|k^*-k|}{kk^*}$ to the discrete derivative. We bound the discrete derivative for $i \leq \min\{k, k^*\}$ as follows:

$$\begin{aligned}
D_i f &\leq \left|\frac{1}{k(n-k)} - \frac{1}{k^*(n-k^*)}\right|(n-k) + \frac{|k^*-k|}{k^*(n-k^*)} \\
&= \left|\frac{1}{k} - \frac{n-k}{k^*(n-k^*)}\right| + \frac{|k^*-k|}{k^*(n-k^*)} \\
&= \left|-\frac{|k-k^*|}{k^*k} + \frac{|k-k^*|}{k^*(n-k^*)}\right| + \frac{|k-k^*|}{k^*(n-k^*)} \\
&\leq \frac{|k-k^*|}{\gamma^2 n^2} + \frac{2\,|k-k^*|}{\gamma(1-\gamma)n^2} \\
&\leq \frac{3\,|k-k^*|}{\gamma^2 n^2}
\end{aligned}$$

We bound the discrete derivative for $i > \max\{k, k^*\}$ similarly, noting that an arbitrary change

in x_i changes $\sum_{i'=1}^{k} I(x_{i'} > x_i)$ by at most $\pm k$ and $\sum_{i'=k^*+1}^{k} I(x_{i'} > x_i)$ by at most $\pm(k - k^*)$:

$$\begin{aligned}
D_i f &\leq \left| \frac{1}{k(n-k)} - \frac{1}{k^*(n-k^*)} \right| \cdot k + \frac{|k^* - k|}{k^*(n-k^*)} \\
&= \left| \frac{1}{n-k} - \frac{k}{k^*(n-k^*)} \right| + \frac{|k^* - k|}{k^*(n-k^*)} \\
&= \left| -\frac{|k-k^*|}{(n-k^*)(n-k)} + \frac{|k-k^*|}{k^*(n-k^*)} \right| + \frac{|k-k^*|}{k^*(n-k^*)} \\
&\leq \frac{|k-k^*|}{\gamma^2 n^2} + \frac{2\,|k-k^*|}{\gamma(1-\gamma)n^2} \\
&\leq \frac{3\,|k-k^*|}{\gamma^2 n^2}
\end{aligned}$$

Finally, we bound the discrete derivative for $\min\{k, k^*\} < i \leq \max\{k, k^*\}$. To do this, we note that the first summation in f changes by k if $k < k^*$ or $n - k$ if $k > k^*$, and the difference of summations in the second term changes by at most $n - (k + k^*)$ in either case. Then we achieve our bound as follows:

$$\begin{aligned}
D_i f &\leq \left| \frac{1}{k(n-k)} - \frac{1}{k^*(n-k^*)} \right| \cdot \max\{k, n-k\} + \frac{n - (k^* + k)}{k^*(n-k^*)} \\
&\leq \frac{|k-k^*|}{\gamma^2 n^2} + \frac{n}{\gamma(1-\gamma)n^2} \\
&\leq \frac{2}{\gamma^2 n}
\end{aligned}$$

Then since $D_i f$ is finite for each i, we have that f is subgaussian with variance proxy as follows:

$$\begin{aligned}
\frac{1}{4}\sum_{i=1}^{n} (D_i f)^2 &\leq \frac{n - |k^* - k|}{4} \cdot \frac{9\,|k-k^*|^2}{\gamma^4 n^4} + \frac{|k^* - k|}{4} \left(\frac{|k-k^*|}{\gamma^2 n^2} + \frac{1}{\gamma(1-\gamma)n} \right)^2 \\
&\leq \frac{9\,|k-k^*|^2}{4\gamma^4 n^3} + \frac{|k^* - k|}{\gamma^4 n^2} \\
&\leq \frac{13\,|k^* - k|}{4\gamma^4 n^2}
\end{aligned}$$

We can now bound the probability of outputting any particular $k = \lceil \gamma n \rceil, \ldots, \lfloor (1-\gamma)n \rfloor$ as a function of $|k - k^*|$ by applying Theorem 2.1, recalling our bound on $\mathbb{E}\left[[]\, V(k^*) - V(k)\right]$ from Equation (2.3).

$$\begin{aligned}
\Pr[V(k) > V(k^*)] &= \Pr\left[V(k) - V(k^*) - \mathbb{E}\left[[]\, V(k) - V(k^*)\right] > \mathbb{E}\left[[]\, V(k^*) - V(k)\right]\right] \\
&\leq \Pr\left[V(k) - V(k^*) - \mathbb{E}\left[[]\, V(k) - V(k^*)\right] > (a - \frac{1}{2})\frac{|k - k^*|}{n}\right] \\
&\leq \exp(-\frac{2\gamma^4}{13}(a - \frac{1}{2})^2 |k - k^*|).
\end{aligned}$$

We complete the proof by bounding the probability of any incorrect $\hat{k}$ such that $|\hat{k} - k^*| > \alpha$ by β.

$$\begin{aligned}
\Pr[\left|\hat{k} - k^*\right| > \alpha] &\leq 2 \sum_{|k-k^*|=\alpha}^{n} \exp(-\frac{2\gamma^4}{13}(a - \frac{1}{2})^2 \, |k - k^*|) \\
&\leq \frac{2\exp(-\frac{2\gamma^4}{13}(a - \frac{1}{2})^2 \alpha)}{1 - \exp(-\frac{2\gamma^4}{13}(a - \frac{1}{2})^2)} \\
&\leq \beta
\end{aligned}$$

Rearranging shows that our accuracy result will hold for

$$\alpha \geq \frac{13}{2\gamma^4(a - 1/2)^2} \left(\log \frac{2}{\beta} + \log \frac{1}{1 - \exp(-\frac{2\gamma^4}{13}(a - \frac{1}{2})^2)} \right)$$

We achieve our final bound by simplifying the above expression as follows. We observe that $\gamma < 1/2, a < 1$ implies $x = 2\gamma^4(a - 1/2)^2/13 \leq 1/416$, and for small x we have $\log(1/(1 - \exp(-x))) \leq 2\log(1/x)$. For any $c > 0$, we have $\log(1/x) \leq C(1/x)^c$ for any $1/x \geq 416$ and $C \geq (\log 416)/(416^c)$, which can be applied to get our final bound.

□

2.3.2 Private offline algorithm

We now give a differentially private version of the nonparametric estimation procedure of [43], in Algorithm 3. Our algorithm uses REPORTMAX as a private subroutine, instantiated with queries $V(k)$ to privately compute $\operatorname{argmax} V(k)$. We show that our algorithm is differentially private (Theorem 2.3) and produces an estimator with additive accuracy that is constant with respect to the sample size n (Theorem 2.4).

The crux of the privacy proof involves analyzing the sensitivity of the Mann-Whitney statistic to ensure that sufficient noise is added for the REPORTMAX algorithm to maintain its privacy guarantees. The low sensitivity of this test statistic plays a critical role in requiring only small amounts of noise to preserve privacy. The accuracy proof extends Theorem 2.2 for the non-private estimator to incorporate the additional error due to the Laplace noise added for privacy. Since the event $V(k) > V(k^*)$ is less probable for k that are further away from k^*, our analysis permits larger values of Laplace noise Z_k for k far from k^*, allowing privacy "for free" with respect to accuracy, for small constants ϵ.

Algorithm 3 Private Nonparametric Change-Point Detector: PNCPD(X, ϵ, γ)

Input: Database $X = \{x_1, \ldots, x_n\}$, privacy parameter ϵ, contraint parameter γ.

for $k = \{\lceil \gamma n \rceil, \ldots, \lfloor (1-\gamma)n \rfloor\}$ **do**

Compute $V(k) = \frac{1}{k(n-k)} \sum_{j=k+1}^{n} \sum_{i=1}^{k} I(x_i > x_j)$

Sample $Z_k \sim \text{Lap}(\frac{2}{\epsilon \gamma n})$

end for

Output $\tilde{k} = \underset{k \in \{\lceil \gamma n \rceil, \ldots, \lfloor (1-\gamma)n \rfloor\}}{\operatorname{argmax}} \{V(k) + Z_k\}$

Theorem 2.3. *For arbitrary data $X = \{x_1, \ldots, x_n\}$, privacy parameter $\epsilon > 0$, and constraint $\gamma \in (0, 1/2)$,* PNCPD*(X, ϵ, γ) is ϵ-differentially private.*

Proof. Privacy follows by instantiation of REPORTMAX (Algorithm 1) with queries $V(k)$ for $k \in$

$\{\lceil \gamma n \rceil, \ldots, \lfloor (1-\gamma)n \rfloor\}$, which have sensitivity $\Delta V = 1/(\gamma n)$, with the observation that noise parameter $2\Delta V/\epsilon$ suffices for non-monotonic statistics. We include a proof for completeness.

Fix any two neighboring databases X, X' that differ on index t. For any $k \in \{\lceil \gamma n \rceil, \ldots, \lfloor (1-\gamma)n \rfloor\}$, denote the respective rank statistics as $V(k)$ and $V'(k)$. By the definition of $V(k)$, we have

$$|V(k) - V'(k)| = \begin{cases} \frac{1}{k(n-k)} \left| \sum_{j=k+1}^{n} \mathbb{I}(x_t > x_j) - \mathbb{I}(x'_t > x_j) \right| \leq \frac{1}{k} & \text{if } t \leq k \\ \frac{1}{k(n-k)} \left| \sum_{i=1}^{k} \mathbb{I}(x_i > x_t) - \mathbb{I}(x_i > x'_t) \right| \leq \frac{1}{n-k} & \text{if } t > k, \end{cases}$$

and it follows that $\Delta V = 1/(\gamma n)$.

Next, for a given $1 \leq t \leq n$, fix Z_{-t}, a draw from $[\text{Lap}(2/\gamma\epsilon n)]^{n-1}$ used for all the noisy rank statistics values except the lth one. We will bound from above and below the ratio of the probabilities that the algorithm outputs $\tilde{k} = t$ on inputs X and X'. Define the minimum noisy value in order for t to be selected with X:

$$Z_t^* = \min\{Z_t : V(t) + Z_t > V(k) + Z_k \quad \forall k \neq t\}$$

For all $k \neq t$ we have

$$V'(t) + \Delta V + Z_t^* \geq V(t) + Z_t^* > V(k) + Z_k \geq V'(k) - \Delta V + Z_k.$$

Hence, $Z'_t \geq Z_t^* + 2\Delta V$ ensures that the algorithm outputs t on input X', and the theorem follows from the following inequalities for any fixed Z_{-t}, with probabilities over the choice of $Z_t \sim \text{Lap}\,(2/(\gamma\epsilon n))$.

$$\Pr[\tilde{k} = t \mid X', Z_{-t}] \geq \Pr[Z'_i \geq Z_i^* + 2\Delta V \mid Z_{-t}] \geq e^{-\epsilon} \Pr[Z_t \geq Z_i^* \mid Z_{-t}] = e^{-\epsilon} \Pr[\tilde{k} = t \mid X, Z_{-i}]$$

□

Next, we provide an accuracy guarantee for our private algorithm PNCPD when the data are drawn according to the change-point model. The first term in the error bound of Theorem 2.4 comes from the randomness of the n data points, and the second term is the additional cost that comes from the randomness of the sampled Laplace noises, which quantifies the cost of privacy. To ensure that the cost of privacy is as small as possible, we use *k-specific thresholds* t_k in the proof to optimize the trade-off between how much to tolerate the Laplace noise added for privacy versus the randomness of the data. As $|k - k^*|$ increases, $V(k)$ is less likely to be close to $V(k^*)$, so we can allow more Laplace noise rather than set a universal tolerance for all k.

Theorem 2.4. *For data $X = \{x_1, \ldots, x_n\}$ drawn according to the change-point model with any distributions P_0, P_1 with $a = \Pr_{x\sim P_0, y\sim P_1}[x > y] > 1/2$, constraint $\gamma \in (0, 1/2)$, change-point $k^* \in \{\lceil \gamma n \rceil, \ldots, \lfloor (1-\gamma)n \rfloor\}$, and privacy parameter $\epsilon > 0$, we have that* PNCPD*(X, ϵ, γ) is (α, β)-accurate for any $\beta > 0$ and*

$$\alpha = \max\{C_1 \cdot \left(\frac{1}{\gamma^4(a-1/2)^2}\right)^c \cdot \log\frac{1}{\beta}, C_2 \cdot \left(\frac{1}{\epsilon\gamma(a-1/2)}\right)^c \cdot \log\frac{1}{\beta}\},$$

for any constant $c > 1$ and some constants $C_1, C_2 > 0$ depending on c.

As with our analysis of the non-private estimator, we can take the argmin and get the same error bounds (with $a - 1/2$ replaced by $|a - 1/2|$) if $\Pr_{x\sim P_0, y\sim P_1}[x > y] < 1/2$.

Proof. We will show that for $\tilde{k} = \operatorname{argmax}\{V(k) + Z_k\}$ and α as in the theorem statement,

$$\Pr[\left|\tilde{k} - k^*\right| > \alpha] \leq \sum_{k:|k-k^*|>\alpha} \Pr[V(k) + Z_k > V(k^*) + Z_{k^*}] \leq \beta$$

by showing that $V(k) - V(k^*)$ is subgaussian as in Theorem 2.2, and we will additionally show that the Laplace noise does not introduce too much additional error. For the algorithm to output

an incorrect $\tilde{k}$, it must either be the case that the statistic $V(k)$ is nearly as large as $V(k^*)$ because of the randomness of the data points, or that Z_k is much larger than Z_{k^*}. For each value of k, we choose a threshold t_k increasing in $|k-k^*|$ specifying how much to tolerate bad Laplace noise versus bad data, and we bound the probability that the algorithm outputs k as follows:

$$\Pr[V(k)+Z_k > V(k^*)+Z_{k^*}] \leq \Pr[V(k^*)-V(k) < t_k] + \Pr[Z_k - Z_{k^*} > t_k] \tag{2.4}$$

Setting $t_k = (a-1/2)|k-k^*|/(2n)$, we can bound the first term as in Theorem 2.2 using Theorem 2.1 as follows:

$$\begin{aligned}\Pr[V(k)-V(k^*) > -t_k] &= \Pr\left[V(k)-V(k^*) - \mathbb{E}\left[\right] V(k)-V(k^*)] > \left(a-\frac{1}{2}\right)\frac{|k-k^*|}{2n}\right] \\ &\leq \exp\left(-\frac{\gamma^4\left(a-\frac{1}{2}\right)^2|k-k^*|}{26}\right).\end{aligned}$$

We bound the second term of (2.4) by analyzing the Laplace noise directly.

$$\begin{aligned}\Pr[Z_k - Z_{k^*} > t_k] &\leq \Pr\left[2\left|\text{Lap}(2/(\epsilon\gamma n))\right| > \left(a-\frac{1}{2}\right)\frac{|k-k^*|}{2n}\right] \\ &\leq \exp\left(-\frac{\left(a-\frac{1}{2}\right)\epsilon\gamma|k-k^*|}{8}\right)\end{aligned}$$

We complete the proof by bounding the probability of any incorrect $\tilde{k}$ such that $\left|\tilde{k}-k^*\right| > \alpha$ by β.

$$\begin{aligned}\Pr\left[\left|\tilde{k}-k^*\right| > \alpha\right] &\leq 2\sum_{k:|k-k^*|=\alpha}^{n} \exp\left(-\frac{\gamma^4\left(a-\frac{1}{2}\right)^2|k-k^*|}{26}\right) + \exp\left(-\frac{\left(a-\frac{1}{2}\right)\epsilon\gamma|k-k^*|}{8}\right) \\ &\leq \frac{2\exp\left(-\frac{\gamma^4}{26}\left(a-\frac{1}{2}\right)^2\alpha\right)}{1-\exp\left(-\frac{\gamma^4}{26}\left(a-\frac{1}{2}\right)^2\right)} + \frac{2\exp\left(-\frac{\epsilon\gamma}{8}\left(a-\frac{1}{2}\right)\alpha\right)}{1-\exp\left(-\frac{\epsilon\gamma}{8}\left(a-\frac{1}{2}\right)\right)} \\ &\leq \beta\end{aligned}$$

We bound each term above by $\beta/2$. Rearranging shows that our accuracy result will hold for

$$\alpha \geq \max\left\{\frac{26}{\gamma^4\left(a-1/2\right)^2}\left(\log\frac{4}{\beta}+\log\frac{1}{1-\exp\left(-\frac{\gamma^4}{26}\left(a-\frac{1}{2}\right)^2\right)}\right),\right.$$
$$\left.\frac{8}{\epsilon\gamma\left(a-1/2\right)}\left(\log\frac{4}{\beta}+\log\frac{1}{1-\exp\left(-\frac{\epsilon\gamma}{8}\left(a-\frac{1}{2}\right)\right)}\right)\right\}.$$

We achieve our final bound by simplifying the above expression as follows. For the first term, we observe that $\gamma < 1/2, a < 1$ implies $x = \gamma^4(a-1/2)^2/26 \leq 1/1664$, and for small x we have $\log(1/(1-\exp(-x))) \leq 2\log(1/x)$. For any $c > 0$, we have $\log(1/x) \leq C(1/x)^c$ for any $1/x \geq 1664$ and $C \geq (\log 1664)/(1664^c)$, which can be applied to get our final bound. For the second term, we observe that $x = \epsilon\gamma(a-\frac{1}{2})/8 \leq \epsilon/32$. When ϵ is small and the corresponding $x \leq 4/5$, we have $\log(1/(1-\exp(-x))) \leq 2\log(1/x)$, and for any $c > 0$, we have $\log(1/x) \leq C(1/x)^c$ for any $1/x \geq 5/4$ and $C \geq (\log 4/5)/((4/5)^c)$. When ϵ is large and the corresponding $x > 4/5$, we have $\log(1/(1-\exp(-x))) \leq \log 2$, which can be incorporated into the constant in our final bound.

□

2.4 Online change point detection

In this section, we show how to extend our results for change-point detection with unknown distributions to the *online setting*, where the database X is not given in advance, but instead data points arrive one-by-one. We assume the analyst starts with a database of size n, and receives one new data point per unit time.

Our algorithm uses the Above Noisy Threshold algorithm of [32, 33] (ABOVETHRESHOLD, Algorithm 2) instantiated with queries of the Mann-Whitney test statistic $V(k)$ in the center of a sliding window of the most recent n points. With each new data point $k > n$, the algorithm computes $V(k)$ for database $X = \left\{x_{k-n/2+1}, \ldots x_{k+n/2}\right\}$, and compares this statistic against a

noisy threshold for significance. When this statistic is sufficiently high, the online algorithm calls the offline algorithm PNCPD on this window to estimate k^*. For simplicity in indexing and to avoid confusion with the notation of the previous section, we define $U(k) = V(k)$ when $V(k)$ is taken over database X for each $k > n/2$. Since the algorithm only checks for a change-point in the middle of the window, we assume that $k^* \geq n/2$ to ensure that the change-point does not occur too early to be detected.

We note that the offline subroutine PNCPD assumes that a change point occurs sometime after the first γn and before the last γn of the n data points on which it is called. We will show that for an appropriate choice of T, ONLINEPNCPD exceeds $\hat{T}$ for some k such that $k^* \in [k, k + n/2]$. Therefore, by waiting for an additional γn data points, we ensure that the assumptions of PNCPD are met as long as $\gamma < 1/4$.

Algorithm 4 Online Private Nonparametric Change-Point Detector: ONLINEPNCPD($X, n, \epsilon, \gamma, T$)

Input: Data stream X, starting size n , privacy parameter ϵ, constraint parameter γ, threshold T.

Let $\hat{T} = T + \text{Lap}\left(\frac{8}{\epsilon n}\right)$

for each new data point $x_{k+n/2}$, $k > n/2$ **do**

Compute $U(k) = \frac{4}{n^2} \sum_{j=k+1}^{k+n/2} \sum_{i=k-n/2+1}^{k} I(x_i > x_j)$

Sample $Z_k \sim \text{Lap}(\frac{16}{\epsilon n})$

if $U(k) + Z_k > \hat{T}$ **then**

Wait for γn new data points to arrive

Output PNCPD $\left(\left\{x_{k-n/2+1+\gamma n}, \ldots x_{k+n/2+\gamma n}\right\}, \epsilon/2, \gamma\right)$

Halt

end if

end for

Privacy follows immediately from the privacy guarantees of ABOVETHRESHOLD and PNCPD.

Theorem 2.5. *For arbitrary data stream X with starting size n, privacy parameter $\epsilon > 0$, and constraint $\gamma \in (0, 1/2)$,* ONLINEPNCPD(X, n, ϵ, γ) *is ϵ-differentially private.*

Proof. By Theorem 1.4, ABOVETHRESHOLD is ϵ-differentially private, and by Theorem 2.3, the statistics $V(k)$ and $U(k)$ have sensitivity $2/n$. Also by Theorem 2.3, PNCPD is ϵ-differentially private. Thus the algorithm ONLINEPNCPD is simply ABOVETHRESHOLD instantiated with privacy parameter $\epsilon/2$, composed with PNCPD also instantiated with privacy parameter $\epsilon/2$. By Basic Composition (Theorem 1.1), ONLINEPNCPD(X, n, ϵ, γ) is ϵ-differentially private. □

To give accuracy bounds on the performance of ONLINEPNCPD, we need to bound several sources of error. First we need to set the threshold T such that the algorithm will not raise a false alarm before the change-point occurs (i.e., control the false positive rate) and that the algorithm will not fail to raise an alarm on a window containing the true change-point (i.e., control the false negative rate). This must be done taking into account the additional error from the private ABOVETHRESHOLD subroutine. Finally, we can use the accuracy guarantees of PNCPD to show that conditioned on calling a window that contains the true change-point, we are likely to output an estimator $\hat{k}$ that is close to the true change-point k^*.

Theorem 2.6. *For data stream X with starting size n drawn according to the change-point model with any distributions P_0, P_1 with $a = \Pr_{x\sim P_0, y\sim P_1}[x > y] > 1/2$, constraint $\gamma \in (0, 1/4)$, change-point $k^* \geq n/2$, privacy parameter $\epsilon > 0$, and threshold $T \in [T_L, T_U]$ such that*

$$T_L = \frac{1}{2} + \sqrt{\frac{2}{n}\log\left(\frac{8(k^* - n/2)}{\beta}\right)} + \frac{32\log((k^* - n/2)/\beta)}{n\epsilon}$$
$$T_U = a - \sqrt{\frac{2}{n}\log\left(\frac{8}{\beta}\right)} - \frac{32\log(8(k^* - n/2)/\beta)}{n\epsilon},$$

we have that ONLINEPNCPD$(X, n, \epsilon, \gamma, T)$ *is (α, β)-accurate for any $\beta > 0$ and*

$$\alpha = \max\left\{C_1 \cdot \left(\frac{1}{\gamma^4(a - 1/2)^2}\right)^c \cdot \log\frac{n}{\beta}, C_2 \cdot \left(\frac{1}{\epsilon\gamma(a - 1/2)}\right)^c \cdot \log\frac{n}{\beta}\right\},$$

for any constant $c > 1$ and some constants $C_1, C_2 > 0$ which depend only on c.

Proof. First, we find an interval $[T_L, T_U]$ for the threshold T that ensures that the algorithm neither calls PNCPD before the true change-point has occurred nor fails to call PNCPD on the window containing k^* somewhere in the middle $(1 - 2\gamma)n$ data points. For now we will ignore the error from ABOVETHRESHOLD, and use T'_L, T'_U to denote the desired thresholds ignoring this additional source of noise. For ease of notation and reindexing, we define $U(k) = V(k)$ when $V(k)$ is computed over database $X = \left\{x_{k-n/2+1}, \dots x_{k+n/2}\right\}$ for the Mann-Whitney test statistic $V(\cdot)$ as defined in Equation (2.1).

Thus we aim to find a range $[T'_L, T'_U]$ such that

$$\Pr[U(k) > T'_L | X_{k-n/2+1}, \dots X_{k+n/2} \sim P_0] \leq \frac{\beta}{8(k^* - n/2)}, \tag{2.5}$$

$$\Pr[U(k) < T'_U | X_{k-n/2+1}, \dots X_k \sim P_0,\ X_{k+1}, \dots X_{k+n/2} \sim P_1] \leq \frac{\beta}{8}. \tag{2.6}$$

Condition (2.5) means that after taking a union bound over all the windows that do not contain k^*, the probability that ABOVETHRESHOLD raises the alarm on the window that does not contain the true change point k^* does not exceed $\beta/8$. Condition (2.6) means that on the window containing the true change-point k^* in the center of the window, ABOVETHRESHOLD will fail to raise the alarm with probability at most $\beta/8$.

It will be helpful to have high probability bounds that the test statistics $U(k)$ are close to their means. Using McDiarmid's Inequality (Theorem 2.1) we can obtain that for any $k > n$

$$\Pr[U(k) - \mathbb{E}\left[\right] U(k)] > t] \leq \exp(-t^2 n/2), \tag{2.7}$$

$$\Pr[U(k) - \mathbb{E}\left[\right] U(k)] < -t] \leq \exp(-t^2 n/2) \tag{2.8}$$

Using these bounds, we will first find T'_L. Note that Condition (2.5) on T'_L considers the setting where all points in the current window are drawn from P_0. Under this condition, $\mathbb{E}\left[\right] U(k)] = 1/2$.

Then by plugging in $t = T'_L - 1/2$ into Inequality (2.7), we get the following expression:

$$\Pr\left[U(k) \geq T'_L | X_{k-n/2+1}, \ldots X_{k+n/2} \sim P_0\right] \leq \exp\left(-\frac{n}{2}\left(T'_L - \frac{1}{2}\right)^2\right)$$

Setting the right hand side of this to less than or equal to $\frac{\beta}{8(k^*-n/2)}$ and solving for T'_L gives the following lower bound, which satisfies Condition (2.5):

$$T'_L = \frac{1}{2} + \sqrt{\frac{2}{n}\log\left(\frac{8(k^* - n/2)}{\beta}\right)}.$$

Next we find the upper bound T_U. Note that Condition (2.6) on T'_U considers the setting where the first $n/2$ points in the window are drawn from P_0 and the remaining $n/2$ points are drawn from P_1. Under this condition, $\mathbb{E}\left[[]\, U(k)\right] = a$. Then plugging $t = a - T'_U$ in Inequality (2.8) and using Condition (2.6), we get the following bound:

$$\Pr[U(k) \leq T'_U | X_, \ldots X_{n/2} \sim P_0,\ X_{n/2+1}, \ldots X_n \sim P_1] \leq \exp\left(-(a - T'_U)^2 n/2\right) \leq \frac{\beta}{8}.$$

Solving this for T'_U gives the following Inequality which satisfies Condition (2.6):

$$T'_U \leq a - \sqrt{\frac{2}{n}\log\left(\frac{8}{\beta}\right)}.$$

We now return to account for the error from ABOVETHRESHOLD. To ensure that this error does not cause a window to be called before the true change-point and also does not skip the window with the true change-point, we require the following conditions to both hold with probability $\frac{\beta}{4}$

$$\text{For } T \geq T_L, \;\; U_k < T - \alpha' \text{ when } k < k^*$$

$$\text{For } T \leq T_U, \;\; U_{k^*} > T + \alpha'$$

Thus we obtain that the new interval for T is $[T_L, T_U]$, where $T_L = T'_L + \alpha'$, and $T_U = T'_U - \alpha'$. If both those conditions hold then for $\alpha' = \frac{32\log(8(k^*-n/2)/\beta)}{n\epsilon}$, ABOVETHRESHOLD will identify the window which contains the true change point with probability $(1-\beta/4)$ by the following theorem shown in [32].

Theorem 2.7 ([32])**.** *For any sequence of m queries $f_1, \ldots, f_m$ with sensitivity Δ such that $|\{i < m : f_i(X) \geq T - \alpha\}| = 0$,* ABOVETHRESHOLD *outputs with probability at least $1-\beta$ a stream of $a_1, \ldots, a_m \in \{\top, \perp\}$ such that $a_i = \perp$ for every $i \in [m]$ with $f(i) < T - \alpha$ and $a_i = \top$ for every $i \in [m]$ with $f(i) > T + \alpha$ as long as*

$$\alpha \geq \frac{8\Delta \log(2m/\beta)}{\epsilon}.$$

Taking a union bound over the failure probabilities of Conditions (2.5) and (2.6), and the statement above, we can see that ONLINEPNCPD will call PNCPD on the right window except with small probability $\beta/2$.

Finally, we can use the accuracy guarantees of PNCPD to show that conditioned on raising an alarm in the correct window, we are likely to output an estimate $\hat{k}$ that is close to the true change-point k^*. Slightly more careful accounting is needed here, because conditioning on raising an alarm and calling PNCPD, the data points in the chosen window are no longer distributed according to the change-point model. Let $W(k)$ denote the event that ONLINEPNCPD calls PNCPD$(\left\{x_{k-n/2+1+\gamma n}, \ldots x_{k+n/2+\gamma n}\right\}, \epsilon/2, \gamma)$ on the window centered at k. Then

$$\begin{aligned}
\Pr\left[\left|\tilde{k} - k^*\right| > \alpha\right] &= \sum_{k > n/2} \Pr\left[W(k) \cap \{|\tilde{k} - k^*| > \alpha\}\right] \\
&\leq \sum_{k \notin (k^*-n/2, k^*]} \Pr\left[W(k)\right] + \sum_{k \in (k^*-n/2, k^*]} \Pr\left[W(k) \cap \left\{\left|\tilde{k} - k^*\right| > \alpha\right\}\right] \\
&\leq \frac{\beta}{2} + \frac{n}{2} \Pr\left[\text{PNCPD fails}\right] < \beta
\end{aligned}$$

To achieve the inequality above, the probability of PNCPD fails to report the change point within the α-window around k^* has to be bounded by β/n. Thus by Theorem 2.4 we set the error to be,

$$\alpha = \max\left\{C_1 \cdot \left(\frac{1}{\gamma^4\left(a-1/2\right)^2}\right)^c \cdot \log\frac{n}{\beta}, C_2 \cdot \left(\frac{1}{\epsilon\gamma\left(a-1/2\right)}\right)^c \cdot \log\frac{n}{\beta}\right\},$$

for any constant $c > 1$ and some constant $C_1, C_2 > 0$ depending on c. □

We have proved the theorem, but we should also show that the window $[T_L, T_U]$ is non-empty, and there exists a good range in which to choose the threshold T. The condition that $T_L < T_U$ is equivalent to,

$$a - \frac{1}{2} > \sqrt{\frac{2}{n}\log\left(\frac{8(k^* - n/2)}{\beta}\right)} + \sqrt{\frac{2}{n}\log\left(\frac{8}{\beta}\right)} + \frac{64\log(8(k^* - n/2)/\beta)}{n\epsilon}. \tag{2.9}$$

We can simplify Inequality (2.9) as,

$$\begin{aligned}\sqrt{\frac{2}{n}\log\left(\frac{8(k^* - n/2)}{\beta}\right)} + \sqrt{\frac{2}{n}\log\left(\frac{8}{\beta}\right)} + \frac{64\log(8(k^* - n/2)/\beta)}{n\epsilon} \\ < \sqrt{\frac{2}{n}\log\left(\frac{8k^*}{\beta}\right)} + \sqrt{\frac{2}{n}\log\left(\frac{8}{\beta}\right)} + \frac{64\log(8k^*/\beta)}{n\epsilon} < a - \frac{1}{2}.\end{aligned}$$

Finally, solving the right hand side for n, we find the following bound on n that satisfies Inequality (2.9).

$$n > \frac{1}{(a-1/2)^2}\left(\sqrt{2\log\left(\frac{8k^*}{\beta}\right)} + \sqrt{2\log\left(\frac{8}{\beta}\right)} + \frac{64}{\epsilon}\log\left(\frac{8k^*}{\beta}\right)\right)^2.$$

For any starting database size that is at least this large (only $n = \Omega((\frac{\log(k^*/\beta)}{\epsilon(a-1/2)})^2)$), the acceptable region $[T_L, T_U]$ for a threshold T will be non-empty. Moreover, the $\log k^*$ dependence of T_L and T_U means that only a rough estimate of the true change-point is necessary in practice to choose an acceptable threshold T.

2.5 Application: Drift Change Detection

In this section, we extend our consideration of the change-point problem to the setting where data are not sampled i.i.d. from fixed pre- and post-change distributions, but instead are sampled from distributions that are changing smoothly over time. In particular, we consider distributions with *drift*, where the parameter of the distribution changes linearly with time, and the rate of linear drift changes at the change-point. Since the samples are not i.i.d., we consider differences between successive pairs of samples in order to apply the algorithms from the previous sections.

The *drift change detection problem* is parametrized by error terms e_t independently sampled from a mean-zero distribution $\mathcal{S}$, two drift terms ξ_0 and ξ_1, a drift change-point $t^* \in [n]$, and a mean η associated with t^*. Independent random variables $X = \{x_1, \ldots, x_n\}$ are said to be drawn from the drift change detection model if we can write

$$x_t = \mu_t + e_t,$$

for μ_t piecewise linear as follows:

$$\mu_t = \begin{cases} \eta - (t^* - t)\xi_0 & t \leq t^* \\ \eta + (t - t^*)\xi_1 & t > t^* \end{cases}.$$

Our goal is to detect the drift change-point t^* with the smallest possible error.

In order to apply our algorithms which require i.i.d. samples, we will transform the sample X by considering differences of consecutive pairs of x_i. These differences are i.i.d. with mean ξ_0 before t^*, and i.i.d. with mean ξ_1 after t^*, and we can now apply PNCPD to this instance of change-point detection. For ease of presentation, we will assume n is even and t^* is odd.

Formally, define a new sample $Y = \{y_1, \ldots, y_{n/2}\}$ with sample points $y_t = x_{2t} - x_{2t-1}$, for

$t = 1, \dots n/2$. Then we have

$$y_t = \begin{cases} \xi_0 + e_{2t} - e_{2t-1}, & \text{for } t = 1, \dots, \frac{t^*-1}{2}, \\ \xi_1 + e_{2t} - e_{2t-1}, & \text{for } t = \frac{t^*+2}{2}, \dots, \frac{N}{2}. \end{cases}$$

Note that random variables $(e_{2t} - e_{2t-1})$ are independent and identically distributed. Thus the y_t are independent, and they are sampled from a fixed distribution before the change point, and from another distribution after the change-point. We can then apply the PNCPD algorithm and privately estimate the drift change-point $\hat{t}$ as twice the output of PNCPD($\{y_1, \dots, y_{n/2}\}, \epsilon, \gamma$). This estimation procedure will inherit the privacy and accuracy results of Theorems 2.3 and 2.4.[1]

As a concrete example, consider points sampled from a Gaussian distribution with mean $\mu_t = \xi_0 t + \eta_0$ and standard deviation σ for $t \leq t^*$, and from a Gaussian distribution with mean $\mu_t = \xi_1 t + \eta_1$ and standard deviation σ for $t > t^*$. Then $y_t = x_{2t} - x_{2t-1}$ will be Gaussian with variance $2\sigma^2$ and mean ξ_0 before the change-point and ξ_1 after it. If any of the parameters ξ_0, ξ_1, or σ are unknown, this would require nonparametric change-point estimation.

Corollary 2.1. *For data $X = \{x_1, \dots, x_n\}$ drawn according to the drift change model with drift terms $\xi_0 > \xi_1$, constraint $\gamma \in (0, 1/2)$, drift change time $t^* \in (\lceil \frac{\gamma}{2} n \rceil \dots \lceil (1 - \frac{\gamma}{2}) n \rceil)$, and privacy parameter $\epsilon > 0$, there exists an ϵ-differentially private nonparametric change point estimator that is (α, β)-accurate for any $\beta > 0$ and*

$$\alpha = \max \left\{ C_1 \cdot \left(\frac{1}{\gamma^4 (a - 1/2)^2} \right)^c \cdot \log \frac{1}{\beta}, C_2 \cdot \left(\frac{1}{\epsilon \gamma (a - 1/2)} \right)^c \cdot \log \frac{1}{\beta} \right\},$$

for any constant $c > 1$ and some constant $C_1, C_2 > 0$ depending on c.

[1] This procedure finds a change-point in the sample Y, which corresponds to a pair (x_{2t-1}, x_{2t}) such that one of them is the estimated change point. Under our assumption that t^* is odd, we should output $\hat{t} = 2t - 1$. If t^* is even, then the estimated change-point may be off by one, and $y_{t^*/2}$ is distributed differently than other data points. However, since the PNCPD algorithm is differentially private, its performance is guaranteed be in insensitive to a single outlier in the database, so this fact will not affect the result of the algorithm by too much.

We note that this approach is not restricted solely to offline linear drift detection. The same reduction in the online setting would allow us to use ONLINEPNCPD to detect drift changes online. Additionally, a similar approach could be used to for other types of smoothly changing data, as long as the smooth changes exhibited enough structure to allow for reduction to the i.i.d. setting. For example, if data were sampled of the form $x_t = f(\mu_t + e_t)$ for any one-to-one function $f : \mathbb{R} \to \mathbb{R}$, we could define $y_t = f^{-1}(x_{2t}) - f^{-1}(x_{2t-1})$, and these y_ts would again be i.i.d.. This includes random variables of the form $\exp(\mu_t + e_t)$, $\log(\mu_t + e_t)$, and arbitrary polynomials $(\mu_t + e_t)^k$ (where even-degree polynomials must be restricted to, e.g., only have positive range).

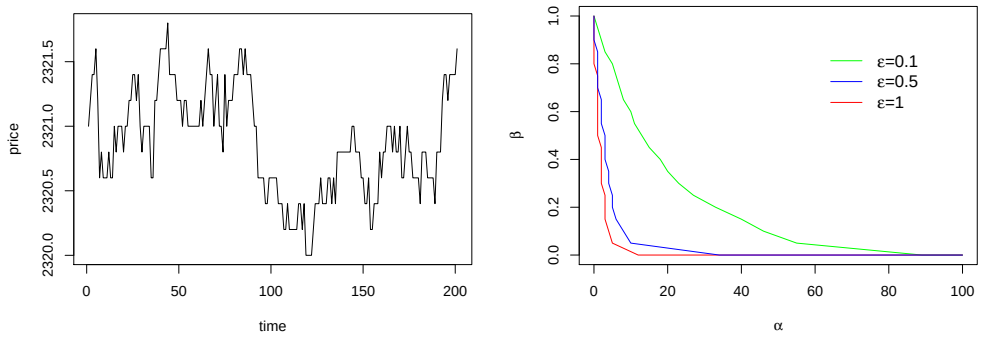
2.6 Empirical Results

We now report the results of an experiment on real data followed by Monte Carlo experiments designed to validate the theoretical results of previous sections. We only consider our accuracy guarantees because the nature of differential privacy provides a strong worst-case guarantee for all hypothetical databases, and therefore is impractical and redundant to test empirically. Our simulations consider both offline and online settings for detecting a change in the mean of Gaussian distribution.

2.6.1 Results of Offline Algorithm with Real Data

First, we illustrate the effectiveness of our offline algorithm on real data by applying it to a window of stock price data including a sudden drop in price, and we use it to determine approximately when this change-point occurred. We use a dataset from [93], which contains stock price data over time, with prices collected every second over a span of 5 hours on October 9, 2012. We identified by visual inspection a window of $n = 200$ seconds (indexed 6900 to 7100 in the dataset, reindexed 0 to 200 here) that appears to include a discrete change in distribution from higher mean price to lower mean price. We then calculated the argmax of the Mann-Whitney statistic $V(k)$ to identify the most likely change-point as time $\hat{k} = 92$, assuming the pre-change data were drawn i.i.d. from

one distribution and the post-change data were drawn i.i.d. from a distribution with lower mean. We used this estimate as the ground truth ($k^* = \hat{k} = 92$) in error analysis of our private offline algorithm. We ran our PNCPD algorithm with $\gamma = 0.1$ on the selected dataset 10^3 times for each privacy value $\epsilon = 0.1, 0.5, 1$. Figure 2.1(a) plots the data in our selected window, and Figure 2.1(b) plots the empirical accuracy $\beta = \Pr[|\tilde{k} - k^*| > \alpha]$ as a function of α for our PNCPD simulations.



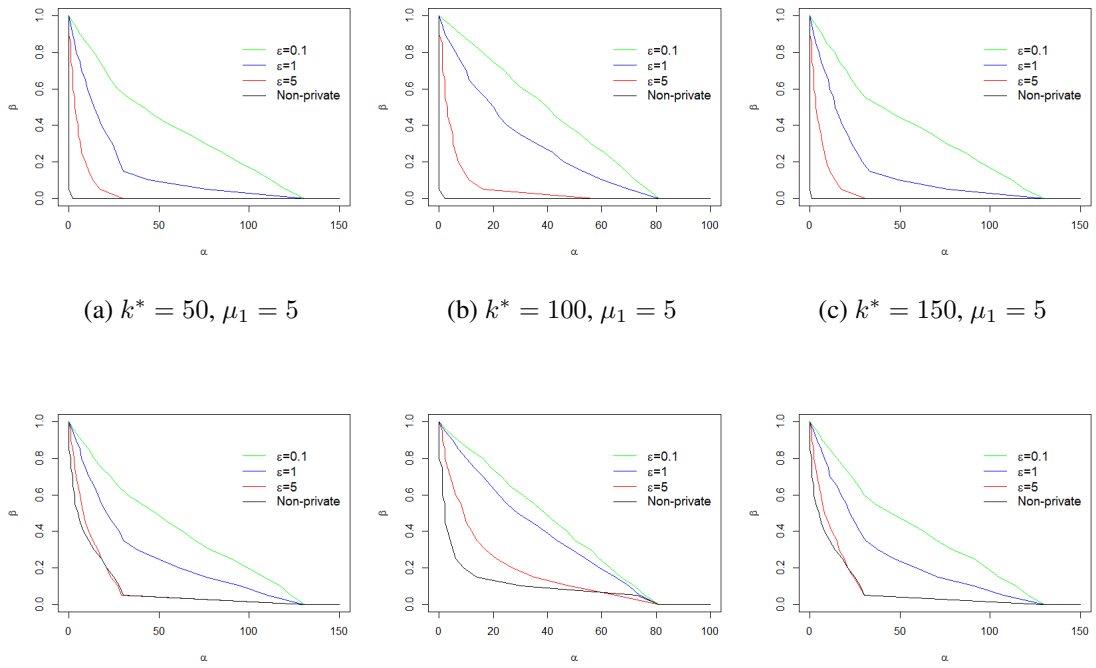
(a) Data trajectory

(b) Accuracy of PNCPD on data from (a)

Figure 2.1: Real data and accuracy results.

2.6.2 Offline Results with Synthetic Data

We now provide simulations of our algorithms using many synthetic datasets drawn exactly according to the change-point model. In the following simulations for PNCPD, we use an initial distribution of $\mathcal{N}(0, 1)$ and post-change distributions of the form $\mathcal{N}(\mu_1, 1)$, considering both a small change $\mu_1 = 1$ and a large change $\mu_1 = 5$. We use $n = 200$ observations where the true change occurs at time points $k^* = 50, 100, 150$. This process is repeated 10^3 times for each value of k^* and μ_1. We consider the performance of our algorithm for $\gamma = 0.1$ and $\epsilon = 0.1, 1, 5, \infty$, where $\epsilon = \infty$ corresponds to the non-private problem, which serves as our baseline. The results are summarized in Figure 2.2, which plots the empirical probabilities $\beta = \Pr[|\tilde{k} - k^*| > \alpha]$ as a function of α.



(a) $k^* = 50$, $\mu_1 = 5$ (b) $k^* = 100$, $\mu_1 = 5$ (c) $k^* = 150$, $\mu_1 = 5$

(d) $k^* = 50$, $\mu_1 = 1$ (e) $k^* = 100$, $\mu_1 = 1$ (f) $k^* = 150$, $\mu_1 = 1$

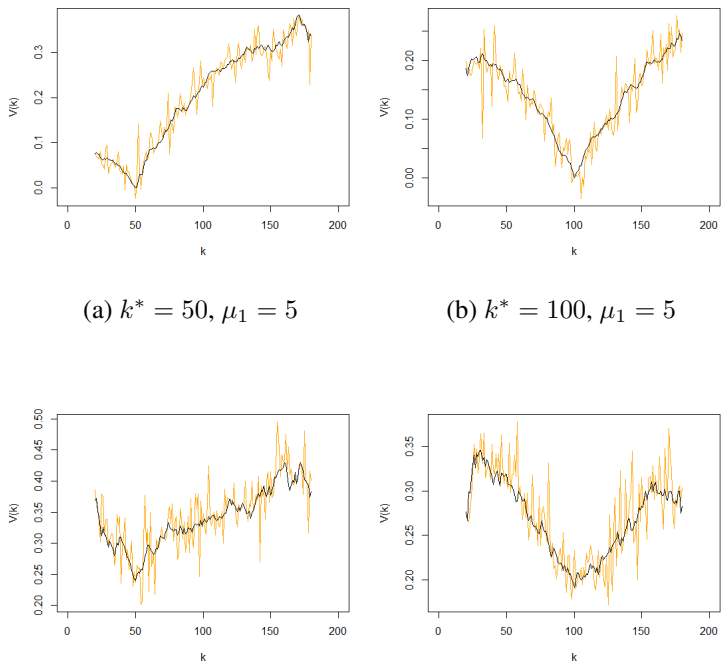
Figure 2.2: Empirical accuracy $\beta = \Pr[|\tilde{k} - k^*| > \alpha]$ of PNCPD from Monte Carlo simulations using Gaussian data, where pre-change data are drawn from $\mathcal{N}(0, 1)$ and post-change data are drawn from $\mathcal{N}(\mu_1, 1)$. Each simulation involves 10^3 runs of PNCPD with varying ϵ on data generated by 200 i.i.d. samples from appropriate distributions with $\mu_1 = 1$ or 5, and change point $k^* = 50, 100$, or 150.

As expected, the algorithm finds the change-point accurately, with better performance when the distributional change is larger or the ϵ value is larger. Performance is slightly diminished when the change-point is at the center of the window, corresponding to $k^* = 100$ in our experiments. This is due to the scaling factor $\frac{1}{k(n-k)}$ in the expression of $V(k)$ as seen in Equation (2.1), which places relatively higher weight on k that are close to the beginning and end of the window. This scaling factor could be removed and our algorithm would still be differentially private and our accuracy result would (qualitatively) continue to hold for change-points near the center of the window. However, if an analyst already has reason to believe that the change-point occurs in the middle of her selected window, she is unlikely to need a change-point detection algorithm.

We also note that our simulations use slightly larger ϵ values and distributional changes than previous work on *parametric* private change-point detection, where the pre- and post-change distributions are given explicitly as input to the algorithm [68].[2] This is to be expected since the nonparametric problem is information theoretically harder to solve, because the test statistic cannot be tailored to the pre- and post-change distributions.

To illustrate these accuracy guarantees, Figure 2.3 show the values of the true test statistic $V(k)$ and the noisy test statistic $V(k) + Z_k$ for the same distributions. We still use $n = 200$ observations and $k^* = 50, 100$ and $\mu_1 = 1, 5$, and run the process only once for each pair of parameter values. We note that for the chosen distributions, $a < 1/2$ so our test statistic $V(k)$ should be minimized at k^*, and we use the variant of PNCPD that outputs $\tilde{k} = \operatorname{argmin}\{V(k) + Z_k\}$ rather than the argmin as described in Algorithm 3. The smoother black line in the figures corresponds to the true test statistic $V(k)$ and the more jagged orange line corresponds to the noisy test statistic $V(k) + Z_k$ for $\epsilon = 5$. Figure 2.3 shows that in all cases, the true statistic is minimized at the true change k^*. This is even more prominent when the distributional change is larger ($\mu_1 = 5$), so more noise can be tolerated. Under smaller distributional changes ($\mu_1 = 1$) the minimization of $V(k)$ around k^* is less dramatic, and there is more opportunity for the noise terms Z_k to introduce estimation error when minimizing the noisy statistic. This also illustrates the structure of the proof of Theorem 2.4, and in particular Equation (2.4), where we separate out the failure probability of the algorithm into two terms: the probability of bad data and the probability of bad draws from the Laplace distribution.

[2]The simulations of [68] used $\epsilon = 0.1, 0.5, 1$ and $\mu_1 = 0.5, 1$, whereas we use $\epsilon = 0.1, 1, 5$ and $\mu_1 = 1, 5$.



(a) $k^* = 50$, $\mu_1 = 5$ (b) $k^* = 100$, $\mu_1 = 5$

(c) $k^* = 50$, $\mu_1 = 1$ (d) $k^* = 100$, $\mu_1 = 1$

Figure 2.3: Value for statistics $V(k)$ with (orange) and without (black) Laplace noise with privacy parameter $\epsilon = 5$ for varying settings for the size change and location of a change point.

Finally, we provide simulations of our PNCPD algorithm for our application of drift change detection, as described in Section 2.5. Recall that our drift change detection model involved data points $X = \{x_1, \ldots, x_n\}$ defined as $x_t = \mu_t + e_t$ where

$$\mu_t = \begin{cases} \eta - (t^* - t)\xi_0 & t \leq t^* \\ \eta + (t - t^*)\xi_1 & t > t^* \end{cases},$$

for drift change-point t^*, and e_t are mean-zero noise terms. In our simulation we use parameters $\eta = 1$, $\xi_0 = 0$, $\xi_1 = 5$, and $e_t \sim_{i.i.d.} \mathcal{N}(0, 1)$. We use $n = 200$ observations where the true drift change occurs at time $t^* = 100$, and repeat the process 10^3 times. We modify the observations X to create a new sample $Y = \{y_1, \ldots, y_{n/2}\}$ as described in Section 2.5, and apply our PNCPD al-

gorithm to this new sample. Figure 2.4 plots the empirical accuracy $\beta = \Pr[|\tilde{t} - k^*| > \alpha]$ as a function of α for $\gamma = 0.1$ and $\epsilon = 0.1, 1, 5, \infty$, where $\epsilon = \infty$ is our non-private baseline.

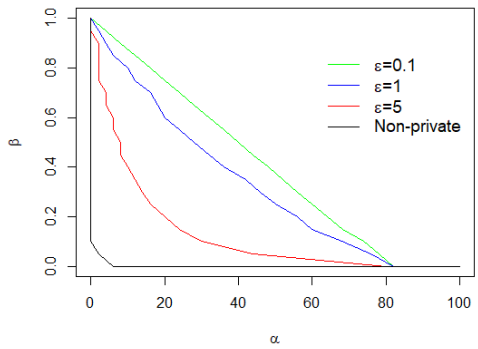


Figure 2.4: Empirical accuracy $\beta = \Pr[|\tilde{t} - t^*| > \alpha]$ of PNCPD for drift detection. The data are generated from the drift change model with parameters $\eta = 1$, $\xi_0 = 0$, $\xi_1 = 5$, and e_t drawn from $\mathcal{N}(0, 1)$. These data are then modified as described in Section 2.5 so that the PNCPD algorithm can be applied.

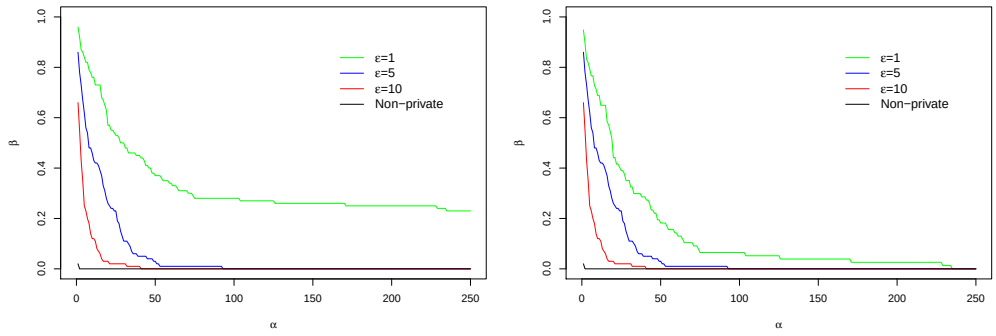
2.6.3 Online Results with Synthetic Data

We also perform simulations for our online private change-point detection algorithm ONLINEPNCPD when the data points arrive sequentially. We use an initial distribution of $\mathcal{N}(5, 1)$ and post-change distribution of $\mathcal{N}(0, 1)$, where the true change occurs at time $k^* = 5000$. To help ensure that the range of the appropriate threshold T in ONLINEPNCPD is non-empty, we choose a larger window size $n = 500$, and larger privacy parameter $\epsilon = 1, 5, 10, \infty$.

We choose the appropriate threshold T by setting a constraint that an algorithm must have positive and negative false alarm rates both at most 0.1, which can be ensured by setting $\beta = 0.4$. (Recall from the proof of Theorem 2.6 that our false alarm rates are each $\beta/4$.) Since we know k^* and a, we can compute the theoretical upper and lower bounds on the threshold exactly for the distributions used in our simulations using the expressions given in the statement of Theorem 2.6. The resulting lower bounds are $T_L = 1.28, 0.80, 0.74, 0.69$ and the upper bounds are $T_U = 0.16, 0.74, 0.81, 0.89$ for $\epsilon = 1, 5, 10, \infty$, respectively. Although the theoretical range of T is empty

for $\epsilon = 1, 5$, our empirical results show that $T = 0.8$ is sufficient to control both false alarm rates, as the theoretical bounds are overly conservative. We choose $T = 0.8$ for all $\epsilon = 1, 5, 10, \infty$. In practice when a and k^* are unknown, the analyst should set a to be the smallest interesting magnitude of distributional change, and k^* to be the analyst's estimate of the time of the change, and similarly compute T_L and T_U using these estimates. Larger values of k^* correspond to more conservative estimates and result in smaller windows for the threshold. We also note the analyst can also choose the lower and upper bounds of T via numerical methods as in [68].

We run our ONLINEPNCPD algorithm 10^3 times with $\gamma = 0.1$ and privacy parameters $\epsilon = 1, 5, 10, \infty$. Figure 2.5 summarizes these simulation results. As in the proof of Theorem 2.6, we can separate the error into two possible sources within the algorithm: halting on an incorrect window, and producing an incorrect estimate of the change-point, even conditioned on halting on the correct window. Figure 2.5(a) shows the error from both of these sources, and Figure 2.5(b) shows the error from only the latter source. These figures show that our algorithm works well with privacy parameters $\epsilon = 5, 10, \infty$. For $\epsilon = 1$, we can control the overall error rate to be less than 0.4 as desired, but not much lower. Figure 2.5(b) shows that this error mainly comes from the failure to halt on the window that contains the true change-point, because the error decreases dramatically after conditioning on the algorithm halting on a correct window that contains the true change-point.



(a) Error probability from inaccurate estimate and false alarm

(b) Error probability from inaccurate estimate only

Figure 2.5: Probability of inaccurate estimation and false alarm (left) and probability of inaccurate report conditioned on raising an alarm correctly (right) for Monte Carlo simulations. Data drawn from $\mathcal{N}(5,1)$ pre-change and $\mathcal{N}(0,1)$ post-change, with true change-point $k^* = 5000$. Each simulation involves 10^3 runs of ONLINEPNCPD with $\gamma = 0.1$, window size $n = 500$, threshold $T = 0.8$, and varying ϵ.

2.7 Conclusions

In this chapter, we present differentially private algorithms for accurate nonparametric change-point detection in offline and online settings. Additionally, we provide an improved analysis of the accuracy of the non-private change-point detection algorithm that leads to a tighter bound on the additive error of the computation. Our accuracy analysis shows that the addition of privacy to the change-point detection algorithm does not hurt the overall performance. Specifically, we show that the additive error of non-private and private offline algorithms does not depend on the sample size. In the online case, this error is logarithmic in the initial sample size.

While our results are well-suited for real-life problems, as they do not require any distributional knowledge, one could extend the results by considering the possibility of multiple change points. Additionally, to make this more suitable for real data, this framework can be considered in high-dimensional space.

Chapter 3: Private Tools for Imbalanced Learning

3.1 Introduction

The problem of imbalanced learning typically refers to classification tasks that involve data with classification classes or categories that are not equally represented. This often happens when some instances appear very rarely in real-world data, such as fraudulent transactions in banks or natural disasters like earthquakes. The challenge that arises while analyzing imbalanced data is that standard machine learning algorithms assume that classes of the data are approximately balanced or have equal misclassification costs. Consequently, such standard algorithms fail to achieve equally good accuracy across both classes and to represent the minority class in the resulting model.

When we do not account for imbalances of the data in a machine learning model, the cost of the false labeling for different classes can vary substantially. Consider the problem of detecting spam tweets. In a study conducted in 2009 [94], it was shown that the proportion of spam posts on Twitter is approximately 3%. If we base our model choice only on predictive accuracy, a valid choice is to have a classifier that labels every tweet as not spam, which achieves 97% accuracy. However, this solution would not satisfactorily solve the problem of spam detection, and would still lead to user dissatisfaction with the number of unnecessary tweets they see in their feed. However, in the case of cancer recognition [95, 96] or fraud detection in bank transactions [97, 98] such mistakes can lead to more significant consequences. Therefore, we cannot assess the performance of machine learning algorithms using only predictive accuracy. To better handle imbalanced data, in this work, along with accuracy, we consider other performance metrics such as the Receiver Operating Characteristic (ROC) curve, the Area Under the ROC Curve (AUC), G-mean, F_1 score, and specifically minority class accuracy. We provide more details about the advantages of these

metrics for the setting of imbalanced learning in Section 3.4.1.

In the machine learning community, the problem of imbalanced learning has been widely studied in non-private settings [48, 49, 50]. This issue has been tackled with two main approaches. The first approach is to modify the machine learning model itself, by assigning appropriate different weights to the classes. The second one is to balance the original dataset, which can be achieved by randomly removing samples from the majority class, known as *under-sampling*, or by increasing the number of samples from the minority class, known as *over-sampling*, which can be done by random replication of points in the sample or more sophisticated algorithmic tools. A well-known over-sampling method is the Synthetic Minority Over-sampling Technique (SMOTE) [44]. The main idea of the algorithm is to generate new minority samples that are similar to the input samples, rather than simply replicating them. More formally, SMOTE iterates over the minority class instances and for every instance finds its k nearest neighbors. Then it generates synthetic minority examples by sampling them from the line segments that connect the selected minority sample with its nearest neighbors. The number of new samples generated depends on the amount of synthetic minority data we want to introduce to the dataset. This minority oversampling algorithm has a simple implementation and is intuitive to interpret. It has been empirically shown that SMOTE helps machine learning models to recognize minority samples and improves the performance overall [44].

Another advantage of SMOTE is that it does not rely on a specific structure of the data, and therefore can be widely used for almost any dataset. However, it also comes with an accuracy tradeoff because SMOTE does not account for any underlying structure of the data, for instance, it does not take into consideration that neighboring samples can be from the majority class which can increase the overlap between minority and majority classes.

In cases when data are sensitive and imbalanced, for instance, detection of rare diseases, we need machine learning tools that preserve privacy while maintaining high classification accuracy. As we have mentioned in previous chapters, differential privacy has emerged as a prevailing privacy

notion in machine learning. Various private implementations of the traditional machine learning algorithms are widely studied in the literature [45], specifically for standard classification methods such as Random Forest [46] and Logistic Regression [47]. However, privacy-preserving methods for imbalanced learning have been much less studied in the literature. In private settings, private imbalanced classification has been usually achieved by applying a pipeline of a non-private pre-processing method and the corresponding privacy-preserving classification method. However, it has been shown that pre-processing techniques, such as oversampling, increase the level of privacy loss in private classification tasks [51]. Moreover, it has been shown that these methods can amplify the loss of accuracy of a minority class [99, 100, 101, 102] as well as magnify bias and unfairness [103, 99, 104, 105]. One of the reasons why accuracy is lost is because the sensitivity of the classifier increases when we add a large amount of synthetically generated data (as in over-sampling) that heavily depends on the minority class samples. There is a need for novel privacy-preserving techniques for imbalanced learning that improve the model performance with respect to the minority class without overuse of the privacy budget.

In this chapter, we take a different approach to privacy-preserving imbalanced learning. We provide a private variant of SMOTE, which we call DP-SMOTE. In an empirical study, we focus on two main directions to validate the performance of our method. First, we assess the performance of DP-SMOTE in a pipeline with a non-private classifier by comparing our method to the non-private SMOTE. While this method is not fully differentially private, as we still include the original data when training the non-private classifier, we show that adding privacy does not hurt the accuracy of the classifier and additionally, it improves the accuracy of the minority class. Second, we show that the treatment of data with DP-SMOTE before differentially private classification, improves the accuracy of the classifier, compared to the non-private SMOTE.

3.1.1 Overview of Our Contributions

In this chapter, we develop, to the best of our knowledge, the first minority class oversampling technique that generates differentially private synthetic data instances. Our algorithm is based on a widely used non-private tool for synthetic minority oversampling algorithm known as SMOTE [44]. Roughly speaking, our technique works in the following way. In the first step, it creates a private synopsis of the minority samples, which is essentially a noisy histogram. This private synopsis is used as a differentially private proxy of the minority class. In the second step, the algorithm applies our variation of SMOTE directly to the private synopsis and generates synthetic minority instances. Note that in our algorithm, privacy is added only at the first step, therefore, this method ensures that we achieve a good accuracy while using a minimal amount of differentially privacy budget.

Our main contributions are twofold. Firstly, in Section 3.3 we propose a novel framework for the generation of differentially private synthetic data that can be used as an oversampling technique for minority instances when data is imbalanced. In Section 3.4, we assess the performance of the original SMOTE and our DP-SMOTE relative to a baseline. We empirically show that when used in a pipeline with a non-private classifier SMOTE and DP-SMOTE perform similarly to the baseline when the data is not significantly imbalanced. However, in the setting of highly imbalanced data, both SMOTE and DP-SMOTE improve the accuracy of the minority class while not considerably affecting the overall accuracy. As the performance of SMOTE and DP-SMOTE are similar, we conclude that the addition of privacy does not affect significantly the performance of the algorithm. Secondly, we create an alternative method for DP imbalanced classification that consists of DP-SMOTE and a privacy-preserving classifier.

3.1.2 Related work

The problem of imbalanced data often arises in machine learning when the size of one data class is considerably smaller than the other data class. We refer the reader to comprehensive reviews for imbalanced learning [48, 49, 50]. One of the techniques for tackling imbalanced learning is over-sampling, which is a variant of synthetic data generation. This approach aims to overcome the imbalance in the original data sets by artificially generating data samples. A common oversampling method is the Synthetic Minority Over-sampling Technique (SMOTE) [44]. It generates synthetic minority examples by sampling them from the lines that connect minority samples with their k nearest neighbors. In empirical studies [44, 106], it is shown that SMOTE achieves a higher AUC when combined with under-sampling techniques. A detailed empirical comparison of different SMOTE-based techniques is presented in [107].

Differential privacy [31] is a widely accepted notion of theoretical data privacy for machine learning models. Privacy is achieved by adding noise calibrated to the sensitivity of the query to the output of the algorithm. Various private implementations of the traditional differentially private algorithms are widely studied in the literature [45]. Specifically, in this work we use private Random Forest [46] and Logistic Regression [47] implementation.

Synthetic data generation is a widely used tool for approaching imbalanced learning. In this work, we create our private synthetic data generation technique based on a noisy histogram method [108, 109]. This histogram would effectively be synthetic minority data, where the quality of representation would depend on the granularity parameter α. In general, there exist better – and more complex – methods for private synthetic data generation [110, 111, 112, 113, 114, 115]. The simplicity of this phase is two-fold. First, more nuanced and realistic-looking data will be generated in the second phase through the application of SMOTE. Second, for the purposes of minority oversampling and imbalanced classification, the resulting synthetic minority data need not fully capture the richness of the true minority data. Instead, they need only enable accurate

downstream classification. It is for this second reason that the synthetic minority data generated by DP-SMOTE are serving a different objective than typically considered in the private synthetic data generation literature.

The challenge of imbalanced data in machine learning becomes much harder when we want to apply privacy. The accuracy for minority class is low even for non-private classification. There is a line of work that shows that differentially private algorithms can disproportionally affect minority groups by amplifying the loss of accuracy of a minority class [99, 100, 101, 102] as well as magnify bias and unfairness [103, 99, 104, 105, 116].

In the machine learning pipeline, the choice of pre-processing model and the decision to add pre-processing are heavily based on the knowledge about data. This creates a possibility for additional privacy leakage. In [51] the authors emphasize a lack of private methodologies for pre-processing techniques. In their empirical assessment of various pre-processing techniques, it is shown that the application of resampling methods on imbalanced data leads to an increase in privacy leakage. Specifically, they show that the higher ratio of oversampling corresponds to a higher level of privacy loss.

In our differentially private version of SMOTE, we use the uniform grid (UG) synopsis methodology described in [108]. This method converts the original data to a noisy histogram by dividing the data domain into M same-size cells. The variations of the UG method are also discussed in [109, 117].

3.2 Preliminaries

In the following, we provide the necessary background on imbalanced learning and the oversampling method SMOTE. Let $D = (X, y)$ to denote a dataset, where X is a d-dimensional set of instances from a known range $[-r, r]^d$ and y is a set of labels from $\{0, 1\}$ associated with X. Here we assume that dataset D is imbalanced and we assign the positive class (with label "1") to be the *minority* class and the negative class (with label "0") to be the *majority* class. Let X_1 and

$n_1 = |X_1|$ be the subset and number, respectively, of instances that correspond to label "1" or minority class instances. Similarly, let X_0 and $n_0 = |X_0|$ be the subset and number of the subset of instances that corresponds to label "0" or majority class instances, respectively. In the context of imbalanced learning, we assume that $n_0 > n_1$. Finally, for an oversampling technique that aims to balance both classes, we denote the desired ratio between the number of oversampled minority instances and the number of majority instances by $\alpha \in [n_1/n_0, 1]$, where $\alpha = 1$ corresponds to a completely balanced data after oversampling. The parameter α is also known as the sampling strategy.

SMOTE. As we mentioned earlier, SMOTE [44], formally stated in Algorithm 5, is a widely used oversampling technique in machine learning. The algorithm consists of the following steps: (1) on every iteration, it considers an instance from a minority class, (2) finds the k nearest neighbors of each instance, (3) for each minority class instance, it randomly selects $N = (\alpha \cdot n_0 - n_1)/n_1$ samples from the k nearest neighbors, (4) it randomly generates new minority samples from the lines that connect the data point with the chosen nearest neighbors.

Algorithm 5 SMOTE(X_1, α, k) [44]

Input: minority class instances $X_1 = \{x_1, \ldots x_{n_1}\}$, dataset dimension d, sampling strategy α, number of nearest neighbors k.

Output: $(\alpha \cdot n_0 - n_1)$ synthetic minority class samples.

$N = (\alpha \cdot n_0 - n_1)/n_1$

for $i = 1, \ldots, n_1$ **do**

Compute k nearest neighbors for x_i: $\bar{X}_i = (x_i^1, \ldots, x_i^k)$

while $N \neq 0$ **do**

Choose x_i' uniformly at random from $\bar{X}_i$

for $j = 1, \ldots d$ **do**

Sample u uniformly from $[0, 1]$

$z = x_{i,j} + u \cdot (x'_{i,j} - x_{i,j})$

return z

end for

$N = N - 1$

end while

end for

3.3 DP-SMOTE Algorithm

In this section, we present our differentially private synthetic minority oversampling technique, DP-SMOTE, which we formally present in Algorithm 6. The algorithm consists of two phases: first, it creates a noisy histogram of the discretized minority class data, and then it performs SMOTE using the noisy histogram data as input.

One might wonder why this two-phase approach is used, rather than directly applying differentially private mechanisms to SMOTE (Algorithm 5), and then using a private variant of SMOTE

to generate new points one by one. If this approach were taken, the privacy guarantees would degrade as new points are generated, with a resulting privacy guarantee of either $(\Omega(N\epsilon), 0)$-DP or $(\tilde{\Omega}(\sqrt{N}\epsilon), \delta)$-DP when N new points are generated. For most practical use cases, this may yield an unacceptable privacy parameter. Instead, our approach creates a private data structure – namely, the noisy histogram – which can then be used to generate an arbitrary number of new data points. In this way, the algorithm's privacy budget and accuracy guarantees do not depend on the number of synthetic minority points generated.

Before stating our main algorithm, we introduce some notation. First, to construct the noisy histogram, we need to partition the space $[-r, r]^d$ into M equal-width cells. Formally, for dataset D from $[-r, r]^d$, we define a *uniform grid* with granularity parameter ν as $\mathcal{G} = I^d$, where $I = [-r, -r + 2r\nu) \cup [-r + 2r\nu, -r + 4r\nu) \cup \cdots \cup (r - 2r\nu, r]$. For each cell in the grid, we define its *histogram query* as follows:

Definition 3.1 (Histogram query)**.** *Given dataset X partitioned by uniform grid $\mathcal{G}$ into M equal-width cells with cell centers $\{y_i\}_{i=1}^M$, the histogram query $\mathcal{H}_{\mathcal{G},q_i}(X)$ returns the number of instances from X in the cell with center q_i from uniform grid $\mathcal{G}$.*

We now provide a notion of connectivity between the centers of each cell.

Definition 3.2 (ℓ-connectivity)**.** *Given uniform grid $\mathcal{G}$ into M equal-width cells with cell centers $\{q_i\}_{i=1}^M$, we call cell center q_i and q_j* ℓ-connected*, if by moving along grid $\mathcal{G}$, minimal the distance between q_i and q_j in no more than ℓ edges.*

Algorithm 6 DP-SMOTE(X_1, α, ℓ, ν ϵ)

Input: minority class instances $X_1 = \{x_1, \dots x_{n_1}\}$, dimension of the dataset d, data range $[-r, r]^d$, granularity parameter ν, sampling strategy α, connectivity parameter ℓ, privacy parameter ϵ

Output: $(\alpha \cdot n_0 - n_1)$ differentially private synthetic minority class data points

Phase 1: Privately generate noisy histogram

Let $\mathcal{G}$ be a uniform grid with granularity parameter ν on X_1

Let $Q = \{q_1, \dots, q_M\}$ be centers of the cells of $\mathcal{G}$ where $M = \left(\frac{1}{\nu}\right)^d$

for $i = 1, \dots M$ **do**

Sample $Z \sim Lap(1/\epsilon)$

Let $c_i = \mathcal{H}_{\mathcal{G},q_i}(X_1) + Z$

end for

Phase 2: Apply SMOTE to noisy histogram

$N = \alpha \cdot n_0 - n_1$

while $N \neq 0$ **do**

Choose a center $q_i \in Q$ with probability $\propto c_i$

Let Q^ℓ be the set of all ℓ-connected cell centers from $\mathcal{G}$ to q_i

Randomly choose $q_k \in Q^\ell$ with probability $\propto c_k$

for $j = 1, \dots d$ **do**

Sample u uniformly from $[0, 1]$

$z = q_{i,j} + u \cdot (q_{k,j} - q_{i,j})$

return z

end for

$N = N - 1$

end while

The first phase of Algorithm 6 partitions each dimension of the dataset into equal-width cells based on the granularity parameter ν. Then, each point in the dataset D is assigned to the cell in which it appears, and the noisy counts of each cell are computed by adding Laplace noise to the true count in each cell. Since this is a *histogram query*, the differential privacy guarantees do not degrade with the number of cells used, and it suffices to add Laplace noise with parameter $1/\epsilon$ to each count. The granularity parameter ν should be chosen to balance the accuracy of the histogram with the computational time of the algorithm, which both increase with finer granularity.

In the second phase of DP-SMOTE, this noisy histogram (with noisy cell counts projected as multiple points at the corresponding cell center) is used as input for the non-private SMOTE. We emphasize that this application of SMOTE uses *only* the noisy histogram data, and not the original dataset D, so this phase maintains the privacy guarantee of Phase 1 via post-processing.

Privacy of DP-SMOTE (Theorem 3.1) follows immediately from privacy of the Laplace Mechanism [5], stated in Definition 1.3 in Chapter 1 applied to a histogram query in Phase 1, and by post-processing in Phase 2; recall the post-processing result stated in Chapter 1.

Theorem 3.1. *DP-SMOTE is ϵ-differentially private.*

To evaluate the accuracy of DP-SMOTE, we use empirical evaluations in the remainder of this chapter. We empirically evaluate the performance of the pre-processing techniques by comparing ROC curves and Area Under ROC Curves (ROC AUC), similarly to [44], where non-private SMOTE is presented. Additionally, we use other evaluation metrics that are discussed in more detail in Section 3.4. Since both SMOTE and DP-SMOTE output a synthetic dataset of minority points, there is not a natural notion of "accuracy" of the output. Rather, accuracy is measured instead as the accuracy of downstream classification using these synthetic data. In Section 3.4, we compare the performance of DP-SMOTE against non-private pre-processing methods including SMOTE. In Section 3.5, we compare the performance of DP-SMOTE against other methods for private imbalanced classification.

Remark 1. *The two-phased approach of DP-SMOTE can also be extended to other pre-processing methods for imbalanced learning. For example, it could be extended to create a differentially private version of ADASYN [118]. Roughly speaking, the idea of ADASYN is similar to SMOTE, except for two main differences: (1) on the step of finding k nearest neighbors for each minority instance x_i, ADASYN considers both classes as candidates for the neighbors; (2) the number of synthetic data instances that need to be generated for each minority instance x_i is determined by the ratio of majority instances among k nearest neighbors of x_i. In this setting, we need a private synopsis or noisy histogram for both classes. For this, in a similar fashion as in DP-SMOTE, we can construct a shared uniform grid for both classes and compute noisy counts of minority and majority instances in each cell by splitting the privacy budget.*

3.4 Empirical Cost of Privacy in SMOTE

In this section, we assess the performance of DP-SMOTE as a pre-processing step for imbalanced classification. In this process, we first apply DP-SMOTE to increase the size of the minority class, and then binary classification is performed on the resulting augmented dataset. To the best of our knowledge, SMOTE was not studied in the private setting before in the literature, therefore, we compare classification accuracy using DP-SMOTE and non-private SMOTE as pre-processing steps. We also compare our method with a no pre-processing benchmark that consists of a classification method applied to the original (imbalanced) dataset as a baseline. The goal of these experiments is to evaluate the impact on the classification's performance when adding privacy during the oversampling step.

Broadly speaking, in Section 3.4.2, we find little empirical difference in performance under DP-SMOTE and non-private SMOTE, suggesting that there is minimal to no loss in performance from adding privacy to this minority oversampling technique. Perhaps surprisingly, we also find that there is little difference in performance between non-private SMOTE and no pre-processing when using population-level metrics such as ROC or classification accuracy. However, in Section

3.4.3, when considering metrics that prioritize the minority class, such as the false positive rate or classification accuracy of the minority class, we do observe considerable improvements under DP-SMOTE and SMOTE. This suggests that both oversampling techniques improve the performance of the minority class without substantially compromising the accuracy of the majority class. Moreover, our numerical simulations demonstrate the practical relevance of SMOTE and DP-SMOTE and show that our algorithms outperform the baseline in the setting of high imbalance in data in the literature.

3.4.1 Methodology

Dataset. For our experimental results in the main body, we use the Pima Indians diabetes dataset [119] which has been extensively used in the literature on imbalanced learning [107, 120, 121]. This dataset was collected to predict diabetes in females of Pima Indian heritage who are at least 21 years old. The dataset has eight real-valued features (*Pregnancies*, *Glucose*, *BloodPressure*, *SkinThickness*, *Insulin*, *BMI*, *DiabetesPedigreeFunction*, *Age*) and 768 observations, where 500 belonging to the majority class (i.e., patients without diabetes) and 268 in the minority class (i.e., patients with diabetes). In Appendix A.1, we also replicate our findings on two additional real-world datasets, both commonly used in the imbalanced learning literature, to show that our results are not specific to the Pima Indians diabetes dataset. For ease of presentation, these results are deferred to Appendix A.1.

To use these datasets as input to DP-SMOTE, they must first be re-scaled so that each feature contains entries in $[-r, r]$ (we use $r = 1$). To achieve this, we apply standard transformations such as normalization to training samples and normalization with the same parameters to test samples. To mitigate the effect of randomness we use multiple folds of cross-validation; we provide specific details for each classification task below. These steps can be performed privately, e.g., [122, 123, 124], however, we perform them without differential privacy in our experiments to better focus only on the impact of adding privacy in oversampling methods, and the empirical difference between

DP-SMOTE and SMOTE.

Classification methods and performance metrics. For the classification step in the experiments, we used non-private[1] logistic regression, decision tree, and random forest classifiers with 5-fold cross-validation using all features. We use various metrics to measure the accuracy of the classification outcomes relative to the true labels of the data. As we mentioned earlier, in the imbalanced learning setting, predictive accuracy alone cannot assess the performance of machine learning algorithms. Therefore we use Receiver Operating Characteristics (ROC) graphs together with the Area Under the ROC Curve (AUC) that provide a more comprehensive evaluation [125, 126]. Additionally, we use the following metrics, defined below in terms of True Positives (TP), False Positives (FP), False Negatives (FN), and True Negatives (TN):

Precision:

$$\text{Precision} = \frac{TP}{TP + FP}$$

Minority class accuracy or Recall:

$$\text{Recall} = \frac{TP}{TP + FN}$$

G-Mean:

$$\text{G-Mean} = \sqrt{\frac{TP}{TP + FN} \cdot \frac{TN}{TN + FP}}$$

F_1:

$$\text{F}_1 = \frac{TP}{TP + \frac{1}{2}(FP + FN)}$$

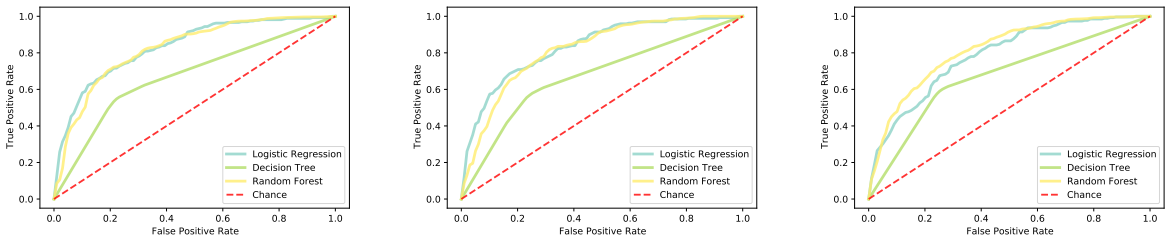
Parameters. Except where specified otherwise, the default parameter settings in the experiments were as follows: The connectivity parameter was set to $\ell = 2$ for DP-SMOTE and the nearest neighbors parameter was set to $k = 5$ for SMOTE. The sampling strategy was $\alpha = 1$, which would create parity between the majority and oversampled minority classes. The granular-

[1]Private classification methods are compared in Section 3.5.

ity parameter was set to $\nu = 0.25$, which is the smallest value (i.e., coarsest granularity) that yields satisfactory accuracy results. This parameter balances accuracy with computational efficiency, as the runtime of DP-SMOTE grows as $O(\nu^{-d})$ for a dataset of dimension d. To assess the impact of privacy, we varied the privacy parameter ϵ in $\{0.1, 1, 10\}$. We also varied each of these parameters individually in our findings to show the impact of each choice.

3.4.2 DP-SMOTE and SMOTE perform similar to baseline under ROC metrics

We begin by showing the performance of DP-SMOTE with $\epsilon = 1$, SMOTE, and no pre-processing using all three classification methods of logistic regression, decision trees, and random forests. Figure 3.1 compares ROC curves of each classification method for each pre-processing method, where the lines represent the mean ROC. First, we observe that logistic regression and random forests have similar performance, and that they both significantly outperform decision trees across all three pre-processing methods. For this reason, we will focus only on logistic regression and random forests as classification methods for the remainder of our experiments. Second, we observe that all three pre-processing methods have comparable performance.

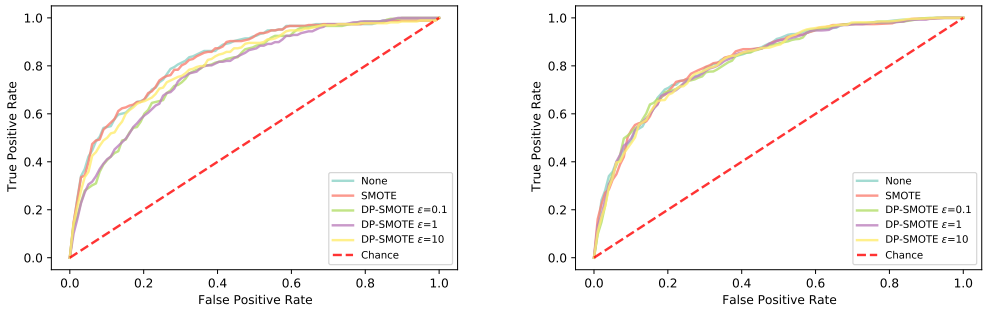


(a) No pre-processing (b) SMOTE pre-processing (c) DP-SMOTE pre-processing

Figure 3.1: ROC Curves for multiple classifiers on diabetes datasets with varying preprocessing techniques.

Figure 3.2 shows ROC curves for all three pre-processing methods, separated by classification method. DP-SMOTE is now run with varying ϵ at values of $\{0.1, 1, 10\}$. We observe more clearly

that there is no significant difference in performance between DP-SMOTE with $\epsilon = \{0.1, 1, 10\}$, SMOTE, and no pre-processing, for either classification method.



(a) Logistic Regression

(b) Random Forest

Figure 3.2: ROC Curves for (a) Logistic Regression, (b) Random Forest classifiers varying preprocessing techniques: None, SMOTE, and DP-SMOTE on diabetes dataset.

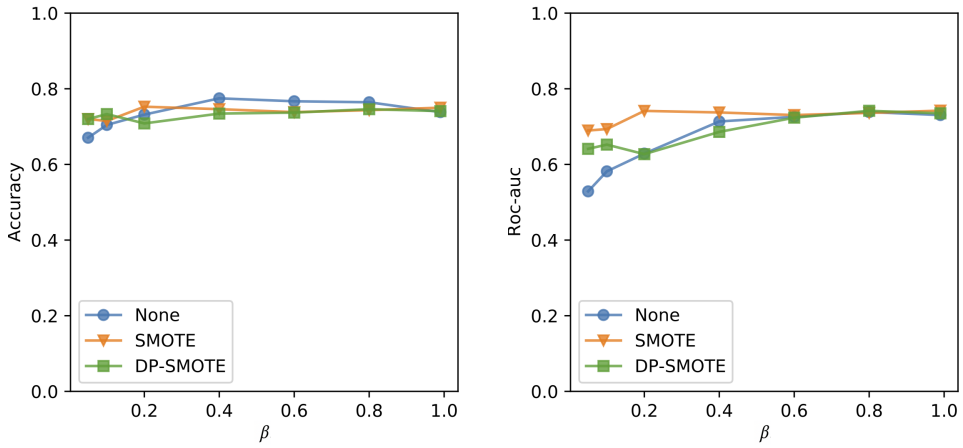
For explicit numerical comparison, Table 3.1 shows the ROC-AUC of each combined pre-processing method and classification method with standard errors. We observe numerically that there is no significant difference across any of these methods at $\epsilon = 1$.

Pre-processing	Logistic Regression	Random Forest
None	0.71 ± 0.03	0.75 ± 0.02
SMOTE($k = 5, T = 1$)	0.71 ± 0.04	0.73 ± 0.03
DP-SMOTE($k = 5, \epsilon = 1, T = 1$)	0.71 ± 0.03	0.73 ± 0.02

Table 3.1: ROC AUC for diabetes dataset for various pre-processing methods: (None, SMOTE($k = 5, \alpha = 1$), DP-SMOTE($\ell = 2, \epsilon = 1, \alpha = 1$)) over different classification methods: logistic regression, random forest.

To explore the impact of the level of imbalance in the data on the classification task, we randomly under-sample the diabetes dataset such that we can control the ratio between minority and majority classes that we denote by $\beta = n_1/n_0$. For this case, sampling strategy parameter α, which controls the quantity of synthetic minority data points created by SMOTE and DP-SMOTE is set to 1 which means full balance in the over-sampled data. Figure 3.3 shows the impact of varying ratio

β between 0.05 and 1 for baseline, SMOTE, and DP-SMOTE under Logistic Regression classifier. To highlight the impact of β, ϵ is held fixed at 1 for DP-SMOTE. We observe that the accuracy does not differ substantially between baseline, SMOTE, and DP-SMOTE for different levels of imbalance in the dataset. However, we observe a difference in ROC-AUC values between the pre-processing methods at very high levels of imbalance, $\beta < 0.4$, which suggests that this metric might be the most valuable with highly imbalanced data. These findings confirm that when analyzing a learning procedure for imbalanced data, we need to consider various performance metrics to account for both classes.



(a) Accuracy (b) ROC-AUC

Figure 3.3: Performance metrics (a) accuracy and (b) ROC-AUC for diabetes dataset for no pre-processing, SMOTE and DP-SMOTE under varying values of β with Logistic Regression classifier and $\alpha = 1, \epsilon = 1$

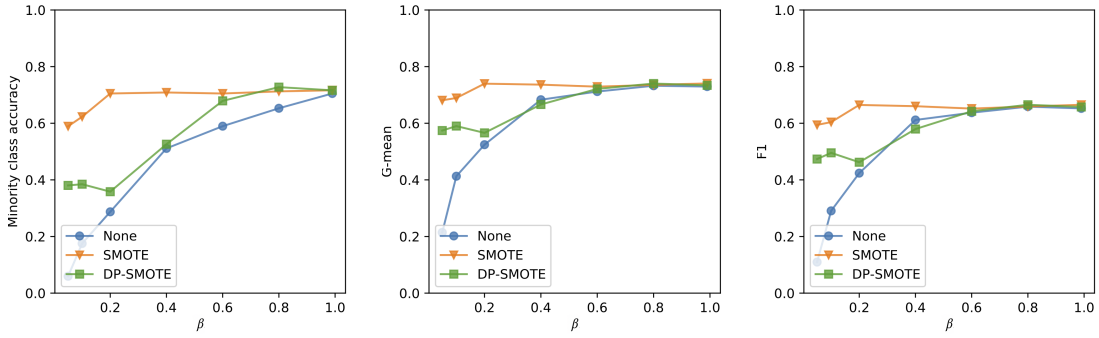
3.4.3 DP-SMOTE and SMOTE outperform baseline on minority-focused accuracy metrics

The results of Section 3.4.2 have been focused on the ROC-AUC metric, which captures global performance across the full population. This is a potential explanation for observing no difference in performance even between non-private SMOTE and the baseline without any oversampling techniques applied: since the minority class is smaller in size than the majority class, global accuracy

can still be high while providing poor performance on the minority class. Indeed, we observe here that when using metrics that require accurate classification of the minority class, there is a statistically significant difference between the two SMOTE-based methods and the baseline.

Figure 3.4 shows the performance of each pre-processing method across a variety of metrics: minority class accuracy or Recall, G-mean, F_1 score. These metrics are more sensitive to the minority class and therefore, we can see how SMOTE and DP-SMOTE outperform the baseline when the imbalance in the dataset is high (which corresponds to smaller values of β). For every metric, we compute an average score over 5 cross-validation folds for each value of β.

We observe that for performance metrics that focus on global accuracy, such as accuracy and ROC-AUC, performance is comparable or even slightly improved under the three pre-processing methods. However, for metrics such as minority class accuracy or G-mean, which depend heavily on the classification performance for the minority class, DP-SMOTE, and SMOTE are both observed to outperform the baseline of no pre-processing. This suggests that the benefits of using minority oversampling techniques are primarily realized as improved accuracy for the minority class. This can be particularly beneficial in applications where accurately identifying members of the minority class is of elevated importance, such as identifying fraudulent financial transactions or diagnosis of a rare disease.



(a) Minority class accuracy (b) G-Mean (c) F_1

Figure 3.4: Performance metrics (a) minority class accuracy, (b) G-mean, (c) F_1 for diabetes dataset for no pre-processing, SMOTE and DP-SMOTE under varying values of β with Logistic Regression classifier and $\alpha = 1, \epsilon = 1$.

Table 3.2 presents the results of Figures 3.3 and 3.4 numerically with an average score and standard deviation over all 5 cross-validation folds. Again, we observe that SMOTE and DP-SMOTE improve the minority class accuracy compared to the baseline. However, we do not see such a drastic improvement as of Figure 3.4 because here we use the original data with $\beta = 0.54$. We can conclude that SMOTE and DP-SMOTE offer a drastic improvement for minority class accuracy, G-Mean, and F_1 when the dataset is heavily imbalanced.

Pre-processing	Classifier	ROC-AUC	Accuracy	Minority class accuracy	G-Mean	F_1
None	LR	0.72±0.02	0.77±0.02	0.56±0.05	0.70±0.03	0.63±0.03
	RF	0.73±0.04	0.76±0.03	0.61±0.08	0.72±0.05	0.64±0.05
SMOTE	LR	0.73±0.03	0.74±0.02	0.71±0.07	0.73±0.03	0.66±0.04
	RF	0.75±0.04	0.76±0.04	0.69±0.07	0.74±0.05	0.67±0.06
DP-SMOTE	LR	0.71±0.03	0.73±0.02	0.63±0.07	0.70±0.03	0.62±0.04
	RF	0.75±0.04	0.77±0.03	0.65±0.09	0.74±0.05	0.67±0.06

Table 3.2: Pre-processing methods (vanilla, SMOTE($k = 5, \alpha = 1$), DP-SMOTE($\ell = 2, \epsilon = 1, \alpha = 1$)) for diabetes dataset across various metrics: G-Mean, F_1, ROC-AUC. The metrics are averaged for different classification methods: logistic regression and random forest.

3.5 Empirical Comparison of Methods for Private Imbalanced Classification

In this section, we compare DP-SMOTE against other methods for private imbalanced classification. The data analysis pipeline for imbalanced classification includes first applying a pre-processing method such as SMOTE, and then performing classification on the augmented dataset. To privatize this process, one could privatize the pre-processing step – as we do with DP-SMOTE – and then the classification step with privacy is done via post-processing, or one could privatize the classification step directly using existing methods for differentially private classification [46, 47].

When applying DP-SMOTE as a private pre-processing step, special care must be paid to the treatment of the original imbalanced dataset. If the original data are included when training the classifier, then private classification is still required to ensure end-to-end privacy of the full algorithmic process. In this section, we compare these three different methods for private imbalanced learning: (1) no pre-processing with DP classification, (2) SMOTE with DP classification, (3) DP-SMOTE with DP classification including original data.

3.5.1 Alternative methods for private imbalanced classification

Before presenting the experimental results, we formally describe the private imbalanced classification methods that will be compared here.

- **Baseline + DP classification.** We keep our baseline method of applying differentially private classification tools without any pre-processing to increase the size of the minority class. This is a natural baseline to show the performance of private classification on imbalanced datasets. One would expect poor performance from this method in applications where the minority class is very small relative to the size of the dataset, which is the typical use case for oversampling methods.

- **SMOTE + DP classification.** This method first applies non-private SMOTE with parame-

ters ($k = 5, \alpha = 1$) and then applies differentially private classification to the resulting augmented data. This method must account for the increased sensitivity of applying SMOTE before classification. We use privacy budget $\epsilon = 1$ for the private classifier.

- **DP-SMOTE + DP classification including original data.** This method first applies DP-SMOTE with parameters ($\ell = 2, \alpha = 1$) and then applies a differentially private classification tool to the original dataset augmented by the synthetic data generated by DP-SMOTE. The privacy budget is split equally between DP-SMOTE and the private classifier.

Experiment setup. Similarly to the previous section, the parameter settings in the experiments are as follows: The nearest neighbors parameter was set to $k = 5$ for SMOTE and the connectivity parameter is set to $\ell = 2$ for DP-SMOTE. The sampling strategy was $\alpha = 1$, which would create parity between the majority and oversampled minority classes. The granularity parameter was set to $\delta = 0.25$ for DP-SMOTE.

Unlike the previously presented experiments, in this section, we consider differentially private classification methods. Specifically, we use differentially private logistic regression [47]. We use the implementation in Python package *diffprivlib* [124]. For the SMOTE and the DP-SMOTE pipelines, we compare differentt privacy levels $\epsilon = 1, 5, 10$.

3.5.2 Increased sensitivity from SMOTE as pre-processing

In this section, we emphasize the necessity of managing the increase in sensitivity that arises, when the input data is treated with a pre-processing technique such as over-sampling, before a differentially private algorithm is applied. This detail is often overlooked in private machine learning, even though it affects the correctness of the privacy guarantees, and resampling methods on imbalanced data have been shown to cause an increase in privacy leakage [51]. Specifically, SMOTE generates synthetic minority samples that are highly dependent on the original data, which, as we show in Theorem 3.2, leads to an increase in the sensitivity of analysis on the resulting dataset. In

contrast, we show in Theorem 3.3 that the application of DP-SMOTE leads to a dratically smaller increase in the privacy budget of the resulting model, meaning that DP-SMOTE causes significantly less privacy leakage than SMOTE.

Theorem 3.2. *Let X be a d-dimensional dataset with n entries, where n_1 entrees correspond to minority class and n_0 entrees correspond to majority class. Let M be an arbitrary ϵ-DP algorithm. Then the instantiation of algorithm M on the dataset X with synthetically generated data by SMOTE(X, α, k) is (ϵ', δ)-differentially private, where $\epsilon' = \epsilon(1+\gamma)^{\frac{l(d,k)\cdot(\alpha\cdot n_0 - n_1)}{n_1 \cdot k}}$, $\delta = e^{\frac{l(d)(\alpha\cdot n_0 - n_1)}{n_1}\left(\epsilon - \frac{\gamma^2}{k(2+\gamma)}\right)}$, $l(d,k)$ is the maximum number of times one point from $\mathbb{R}^d$ can appear among k-nearest neighbors of other points from $\mathbb{R}^d$, and $\gamma \geq 0$ controls the trade-off between ϵ' and δ.*

Proof. Let X, X' be two neighboring datasets such that $X' = X \cup \{x\}$. For the rest of the proof, fix SMOTE parameters $\alpha \in (0, 1]$ and $k \in \mathbb{Z}^+$. To compare the outputs of SMOTE(X, α, k) and SMOTE(X', α, k), we fix the randomness that comes from drawing a uniform sample in the inner for-loop of Algorithm 5. In the worst case, when all $l(d,k)$ points are affected by adding a point x, the added point x is a k-nearest neighbor to all points from X, which means x will replace one of the k nearest neighbors for every point from X. Let $\{x_i\}_{i=1}^{l(d,k)}$ be minority points that have x as their k-nearest neighbor. Note, that SMOTE(X', α, k) has a different output from SMOTE(X, α, k) only if on some iteration the algorithm draws x. Then, for every iteration i of the SMOTE(X', α, k), the probability of choosing x as the nearest neighbor for x_i is $1/k$. Now let $Y = |\text{SMOTE}(X, \alpha, k) \oplus \text{SMOTE}(X', \alpha, k)|$, where $\oplus$ denotes a symmetric difference. Then $Y = \sum_{k=1}^{l\cdot N/n_1} \mathbb{I}\{\text{draw the nearest neighbor } x \text{ for } x_k\}$. Note that $Y \sim Binomial\left(\frac{l\cdot N}{n_1}, 1/k\right)$ and $\mathbb{E}[Y] = \frac{l\cdot N}{n_1 \cdot k}$. To simplify notations, we remove dependence of $l(\cdot,\cdot)$ on d and k when it is clear from the context, and denote $l = l(d,k)$.

Note that if the value of Y were known then we could obtain a group privacy guarantee for the output of SMOTE. However, Y is a random variable and therefore we find a high probability bound

for Y. Using the one-sided Chernoff bound, we constrain the probability that Y is significantly greater than its average and later we incorporate this probability in δ:

$$\Pr\left[Y \geq (1+\gamma)\frac{l \cdot N}{n_1 \cdot k}\right] \leq e^{-\frac{\gamma^2}{2+\gamma}\frac{l \cdot N}{n_1 \cdot k}}$$

Let $M(\cdot)$ be an ϵ-differentially private algorithm with the space of outputs $\mathcal{M}$. Denote $N_l = l \cdot N/n_1$. Let $T(X) = X \cup \text{SMOTE}(X, \alpha, k)$. Then for an arbitrary set of M outputs $S \subset \mathcal{M}$

$$\begin{aligned}
\Pr\left[M(T(X)) \in S\right] &= \sum_{j=1}^{N_l} \Pr\left[M(T(X)) \in S | Y = j\right] \cdot \Pr\left[Y = j\right] \\
&\leq \sum_{j=1}^{N_l} e^{\epsilon \cdot j} \Pr\left[M(T(X')) \in S | Y = j\right] \cdot \Pr\left[Y = j\right] \\
&= \sum_{j=1}^{(1+\gamma)N_l/k} e^{\epsilon \cdot j} \Pr\left[M(T(X')) \in S | Y = j\right] \cdot \Pr\left[Y = j\right] \\
&\quad + \sum_{j=(1+\gamma)N_l/k+1}^{N_l} e^{\epsilon \cdot j} \Pr\left[M(T(X')) \in S | Y = j\right] \cdot \Pr\left[Y = j\right] \\
&\leq e^{\epsilon \cdot (1+\gamma)N_l/k} \sum_{j=1}^{(1+\gamma)N_l/k} \Pr\left[M(T(X')) \in S | Y = j\right] \Pr\left[Y = j\right] \\
&\quad + e^{\epsilon \cdot N_l} \Pr\left[Y \geq (1+\gamma)\frac{N_l}{k}\right] \\
&\leq e^{\epsilon \cdot (1+\gamma)N_l/k} \Pr\left[M(T(X')) \in S\right] + e^{\epsilon \cdot N_l - \frac{\gamma^2}{2+\gamma}\frac{N_l}{k}},
\end{aligned}$$

where the first equality is due to the law of total probability; the second inequality is due to the definition of group privacy; in the third and fourth inequalities, we apply earlier derived Chernoff bound to the components with large j and use the law of total probability and differential privacy definition for the components with small j.

Therefore, we show that $M(T(X))$ is $\left(\epsilon(1+\gamma)\frac{l \cdot N}{n_1 \cdot k}, e^{\epsilon \cdot \frac{l \cdot N}{n_1} - \frac{\gamma^2}{2+\gamma}\frac{l \cdot N}{k \cdot n_1}}\right)$-differentially private. □

Notice, that the bound achieved in Theorem 3.2 provides an upper bound on the epsilon and delta parameters of the resulting DP guarantee, but due to the use of the Chernoff inequality is not necessarily tight.

In Lemma 3.1, we give a lower bound for a parameter $l(d, k)$ that describes the maximum number of times one point from $\mathbb{R}^d$ can appear among k-nearest neighbors of other points from $\mathbb{R}^d$.

Lemma 3.1. *Let $l(d, k)$ be the maximum number of times one point from $\mathbb{R}^d$ can appear among k-nearest neighbors of other points from $\mathbb{R}^d$. Then $l(d, k) \geq 2^{0.2075d(1+o(1))}$.*

Proof. Let X be a d-dimensional dataset with n entries, where n_1 entrees correspond to minority class and n_0 entrees correspond to majority class. Consider a minority class instance x_0. Now we construct an instance of a dataset such that x_0 is a 1-nearest neighbor to at least $2^{0.2075d(1+o(1))}$ minority points.

Let K denote a kissing number for $\mathbb{R}^d$ [128], which is defined as the greatest number of non-overlapping unit spheres that can be arranged in the space such that they each touch a common unit sphere. Let $x_1, \ldots, x_K$ be the centers of such non-overlapping spheres corresponding to the minority points that have x_0 as their nearest neighbor, where x_0 is the center of the central unit sphere, with which the non-overlapping K spheres intersect only in one point. Then for all $i = 1, \ldots K$, $||x_0 - x_i||_2 = 2$ and for all $i, j = 1, \ldots K$, $i \neq j$ $||x_i - x_j||_2 \geq 2$.

Note that in this case x_0 is 1-nearest neighbor for instances $x_1, \ldots x_K$ and thus is also a k-nearest neighbor for any $k \geq 1$. This example shows that there exists a d-dimensional dataset such that one point from this dataset can be the nearest neighbor to at least K points. Therefore, $l(d, k) \geq K$. In [127, 128] it was shown that the lower bound for a kissing number K in a d-dimensional space is $2^{0.2075d(1+o(1))}$, which is a desired lower bound for $l(d, k)$. □

Remark 2. *Lemma 3.1 provides a lower bound for $l(d, k)$, which means that the amount of noise added to the private classifier according to this bound is necessary, but not sufficient. Additionally,*

the combination of results from Theorem 3.2 and Lemma 3.1 may not be sufficient to guarantee differential privacy of the composition of SMOTE and DP algorithm. Specifically, the upper bound for the privacy budget from Theorem 3.2 is given as a function of $l(d, k)$ and Lemma 3.1 gives a lower bound on the value of $l(d, k)$, however, it is not currently known whether either of these bounds is tight. In Section 3.5.3, we empirically show that even with less noise added to the DP classifier with SMOTE, the private classification pipeline with DP-SMOTE performs significantly better.

Now we show a privacy guarantee for the pipeline of the private classifier with DP-SMOTE:

Theorem 3.3. *Let X be a d-dimensional dataset with n entries, where n_1 entrees correspond to minority class and n_0 entrees correspond to majority class. Let M be an arbitrary ϵ_1-DP algorithm. Then the instantiation of algorithm M on the dataset X with synthetically generated data by DP-SMOTE$(X, \ell, \alpha, \epsilon_2)$ is $(\epsilon_1 + \epsilon_2)$-differentially private.*

Proof. By Theorem 3.1, synthetically generated data by DP-SMOTE$(X, \ell, \alpha, \epsilon_2)$ is ϵ-differentially private. Formally, the composition of M and DP-SMOTE is computed in two steps. The first step is an instantiation of DP-SMOTE$(X, \ell, \alpha, \epsilon_2)$. The second step is a differentially private algorithm M instantiated on a dataset $X \cup$ DP-SMOTE$(X, \ell, \alpha, \epsilon_2)$ with privacy budget ϵ_1, which is ϵ_1-defferentially private by post-processing (Theorem 1.2). In other words, DP algorithm M applied to the output of DP-SMOTE retains the ϵ_2 privacy guarantee of DP-SMOTE due to the post-processing guarantees of differential privacy; the instantiation of M on dataset X incurs an additional ϵ_1 privacy cost. Basic composition over these two uses of the same database X yields to the $(\epsilon_1 + \epsilon_2)$ privacy guarantee.

□

According to the results in this section, we adjust privacy parameters in the experiments in Section 3.5.3.

3.5.3 Experimental Results

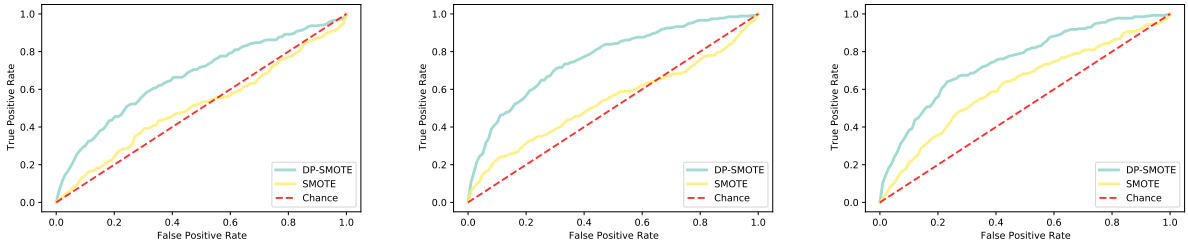
In this experimental study, we compare the performances of the SMOTE and DP-SMOTE pipelines under differentially private classifications. Unlike in the previously considered experiment, these pipelines offer full privacy guarantees to the whole dataset. To accommodate the privacy conditions from Theorem 3.2, according to Remark 2, for a given diabetes dataset, we have to lower the privacy budget of the classifier in the SMOTE pipeline by a scale of 0.024, which corresponds to a kissing number $l(8) = 240$. For the DP-SMOTE pipeline, we split the privacy budget equally between DP-SMOTE and DP classifier, according to Theorem 3.3.

	$\epsilon = 1$	$\epsilon = 5$	$\epsilon = 10$
SMOTE	0.51 ± 0.09	0.55 ± 0.07	0.59 ± 0.04
DP-SMOTE	0.68 ± 0.06	0.71 ± 0.05	0.73 ± 0.05

Table 3.3: Mean ROC-AUC values with standard error for SMOTE and DP-SMOTE with DP logistic regression.

To have a better understanding of how pre-processing techniques compare, in Figure 3.5 we show ROC curves for SMOTE and DP-SMOTE pre-processing methods with varying privacy budget ϵ for the resulting model at values of $\{1, 5, 10\}$. In Table 3.3, we present the mean ROC-AUC metric for both pre-processing methods with different ϵ of the resulting model. Contrasting the results from Section 3.4.2, we can see a clear difference between the pre-processing techniques. We observe that DP-SMOTE outperforms SMOTE for all values of $\epsilon = 1, 5, 10$.

Finally, to assess the performance of SMOTE and DP-SMOTE compared to the baseline with DP classifiers, we consider metrics that prioritize accuracy on the minority class. We compare SMOTE and DP-SMOTE pipelines for the privacy budget value $\epsilon = 10$. In Table 3.4, we present the performance of each pre-processing method across a variety of metrics: ROC-AUC, accuracy, minority class accuracy or Recall, G-mean, F_1 score. For each metric, we compute an average score and standard error over all 5 cross-validation folds.



(a) $\epsilon = 1$ (b) $\epsilon = 5$ (c) $\epsilon = 10$

Figure 3.5: ROC Curves SMOTE and DP-SMOTE with privacy budget of the resulting pipeline (a) $\epsilon = 1$, (b) $\epsilon = 5$, (c) $\epsilon = 10$.

We observe that for DP logistic regression DP-SMOTE outperforms SMOTE for all considered metrics. In particular, our methodology leads to a better performance in terms of the metrics such as minority class accuracy, G-mean, and F_1, which depend heavily on the classification performance of the minority class.

sampler	ROC-AUC	Accuracy	Minority class accuracy	G-Mean	F_1
SMOTE	0.59±0.04	0.5±0.06	0.46±0.13	0.48±0.06	0.39±0.08
DP-SMOTE	0.73±0.05	0.71±0.02	0.63±0.09	0.69±0.03	0.61±0.04

Table 3.4: Pre-processing methods (SMOTE($k = 5, \alpha = 1$), DP-SMOTE($\ell = 2, \epsilon = 5, \alpha = 1$)) for diabetes dataset across various metrics. The metrics are computed for DP logistic regression classifier.

3.6 Conclusions

In this chapter, we develop a new differentially private tool for tackling the problem of imbalanced learning based on a classic framework known as SMOTE (Synthetic Minority Class Over-sampling Technique). We propose a novel framework for the generation of differentially private synthetic data that can be used to balance data with classes of different sizes. We show that for a non-private classifier, our approach improves the accuracy of the minority class while not considerably affecting the overall accuracy. We create an alternative method for DP imbalanced

classification that consists of DP-SMOTE and a privacy-preserving classifier. Our algorithm ensures a lower sensitivity of the resulting pipeline and, as a result, achieves better accuracy than a combination of non-private SMOTE with the corresponding private classifier.

As we discussed in this chapter, the original algorithm SMOTE has multiple variations [129, 130] that can achieve better accuracy for a classification model by taking into consideration the internal structure of the data. We hope that the algorithm developed in this thesis can be a base for the differentially private versions of these SMOTE variations. Additionally, in this work, we considered only the case of continuous data, and therefore the next open problem is to extend our result to the case of categorical data.

Chapter 4: Measurement and Analysis of Digital Behavior

4.1 Introduction

The amount of time users spend online, and how they spend that time, has been found to relate to their digital skills [131], to the amount of social capital and other benefits they can derive from online activity [132], and to students' academic performance [133]. Thus, to draw inferences and conclusions about a variety of different digital constructs, researchers seek to measure people's digital behavior.

However, given that some companies may use user's browsing data to make inferences about user's sensitive characteristics, users have raise multiple concerns related to the use of their data. While users cannot hide their browsing data from ISPs, researches have proposed several solutions to address this concern [53, 54]. In particular, one possible solution is to obfuscate browsing data with noise by issuing randomized search-queries or randomly clicking on ads. However, these methods do not consider the structure of the data and attributes that require protection. Therefore, there is a need to understand what kind of noise can obfuscate or hide users' sensitive attributes and browsing habits. Towards this goal, we must take a step back and learn what underlying attributes can be implied from the collected browsing data and, therefore, have to be protected.

While ideally researchers would be able to directly observe users' browsing behavior, due to difficulties obtaining access to such data, researchers often rely of users' self reports of their online behavior [58, 56]. Potential concerns have been raised about the accuracy of such self-report data [59, 60, 61]. An alternative to self-reports that are feasible in an academic research setting, are observational methods such as having participants install a browser plugin that observes and measures their behavior. Such methods are not without limitations, however. A broad literature

in behavioral economics has shown that people behave differently when they are aware that their actions are observed (e.g., [62, 63, 64, 65, 66]). However, this literature has focused primarily on behavior in incentivized economic games, not web behavior.

To understand user's browsing behavior, we designed and conducted a user experiment ($n = 31$) in which we both surveyed participants about their browsing behavior and observed participants' browsing behavior continuously for 14 days. Using this empirical data we examine the relationship between participants' observed and self-reported behavior. Specifically, we address the following research questions:

(RQ1) Does browsing behavior differ across user groups (i.e., demographics) and types of web use?

(RQ2) Do people have accurate perceptions of their behavior online? Does perception accuracy differ by user group or type of web use?

(RQ3) Do people change their browsing behavior if they are aware of being observed?

4.1.1 Our Contributions

For RQ1, we discover that, comparing to prior work conducted in 2010, people spend much more time online: median of 2.9 hours per day in our study versus one hour per day in [55]. We found little difference across demographic groups by race and gender, but did find significant differences by age, with older participants (aged 35-44) browsing less than younger groups (aged 18-24 and 25-34) across multiple metrics of browsing activity. We find few significant differences in within-website browsing behavior across different categories of websites. One notable exception is the Security Concerns category, which had significantly different within-website browsing patterns than all other categories ($p < 10^{-5}$). We suggest ways that this finding can be used to automate detection of security concerns online.

For RQ2, we find that people substantially overestimate their time spent online (80.6% of our participants, by an average overestimate of 4.5 hours). This overestimation effect persists, even

after controlling for various methodological alternatives. We find no significant difference across demographic groups, meaning that all groups overestimate their time spent online. However, we find that people have approximately accurate perceptions of their top-browsed website categories: 50.3% of reported top browsing categories were indeed in the participant's observed top browsing categories.

We are unable to directly test RQ3 because we obtained informed consent for data collection from all participants in our study. Instead, we test whether behavior changed over time during the study, under the hypothesis that participants will have higher awareness of observation early in the study, shortly after providing consent, and lower awareness later in the study. We do not find changes in either level of browsing activity or in distribution of browsing across website categories, over time during the study. This could indicate that people do not change their behavior when aware of being observed, or it could be that a 14-day study is not sufficient time for participants to forget that their browsing data are being collected.

4.2 Related Work

In this section, we review prior work on users' browsing behavior and perceptions of that browsing behavior.

Browsing Behavior Measurements. Prior work finds correlations between users' browsing behavior and their demographic or behavioral type. In [56] the authors use large-scale measurement data to study the differences in how various demographic groups use the internet, and show that use of different website types depends more on users' level of education than on their demographic features. [55] propose taxonomy of page views for popular website categories and study the behavior of Yahoo! users based on search and toolbar log data. Their analysis includes website categorization but does not include demographic data. They show that the distribution of page views across website categories is skewed, with the top five categories (news, portals, games, ver-

ticals, multimedia) accounting for more than half of all Web activity. [57] offer a methodology to predict certain demographic features such as gender and age from a user's observed browsing behavior, which provides 30.4% and 50.3% improvements on gender and age prediction respectively compared to baseline algorithms. [67] show that a user's pattern of web browsing behavior can be uniquely identified by the types of websites they access and the time-of-day they access those websites with at least 75% accuracy.

While these works lay a foundation for measuring user behavior online, the most recent work in this literature [67] is nearly a decade old. A more updated understanding of browsing behavior under modern internet usage is needed. Additionally, these works measure user browsing behavior, but do not elicit user perceptions of their own browsing behavior. In this chapter, we both elicit self-perceptions of browsing and measure browsing behavior, which allows us to evaluate accuracy of users' perceptions.

Accuracy of user perceptions. The accuracy of user perceptions of browsing behavior has been previously studied in limited contexts. [60] study how accurately users estimate their time spent on Facebook. They show that self-reported data can be often unreliable, and that people tend to overestimate the time they spend on Facebook but underestimate the number of times they visit. [133] study dependencies between academic grades and the time students spend on an online educational platform, and find that longer times spent online are associated with higher grades. We extend this work in two key ways. First, we examine the accuracy of users' perceptions of their overall browsing behavior rather than their behavior on one specific website. Second, we test not only the accuracy of users' perceptions about the time they spend online, but also which categories of websites they most frequently browse.

Outside of studies of online browsing behavior, prior work in the security domain has examined the accuracy of people's self reports. [61] follow a methodology similar to our own, comparing observed behavior via software that participants installed and which logged their behavior over

six weeks to self-reports from those same participants about their digital security behavior. They find low correlation between participants' self-reported and actual behavior across a majority of behaviors that were observed. [59] specifically examine software updating behavior, comparing self-reported behavioral intentions in response to software update prompts with behavioral responses to the same prompts observed through proprietary industry data. They find a significant correlation between responses, but find that self-reporting participants reported that they would update significantly faster than observed users did.

4.3 Methods

To answer our research questions we observed the browsing behavior of 31 participants over a period of 14 days in August and September 2019, and additionally assessed participants' self-reported perceptions of their browsing behavior. In this section, we describe our study procedures (Section 4.3.1), data collection from both the Chrome extension and self-reported data (Section 4.3.2), data analysis (Section 4.3.3), and the limitations of our work (Section 4.3.4). All study procedures were approved by the Georgia Institute of Technology's Institutional Review Board (IRB).

4.3.1 Study Procedures

We recruited participants by advertising flyers on bulletin boards and student gathering spaces on the campus of the Georgia Institute of Technology, which is a large public institution for higher education. The flyers advertised an "Internet Browsing Study" stating: "The purpose of this study is to determine how real people interact with the internet so that we can better protect user data." The flyer also included a link to the online screening survey where participants could verify eligibility and sign up for the study. This flyer is shown in Figure B.5 in Appendix B.3.

The brief screening survey was hosted on Qualtrics, and verified that participants met the eligibility criteria for our study: participants needed to be aged 18 or older, native English speakers,

and needed to browse the internet at least 5 hours per week. The age requirement ensured we did not have any minors in our study; the English language requirement ensured that our collected data would focus primarily on English-language websites; and the browsing activity requirement ensured that our study participants would generate sufficient browsing data during the study.

Participants who met the eligibility criteria were invited to come into the lab to complete a consent form for the experimental portion of the study, complete a pre-study survey, and install the Chrome extension that would collect their browsing data. We asked participants to come in-person so that we could provide support in installing the extension, as an effort to mitigate issues of digital inequity. The pre-study survey asked participants to self-report their demographic information and perceptions of their own browsing habits.

Over the next 14 days, the extension collected data on participants' web browsing, including their websites visits, actions within each website, and timestamps of each browsing action. Finally, on the 14th day of the study, participants returned to the lab, where they uninstalled the Chrome extension and completed a brief post-study survey that asked whether participants changed their browsing behavior over the course of the study. For those individuals who were not able to return to the lab in-person on their 14th day, we truncated data collection after 14 days. More details on the pre-study survey, post-study survey, and extension-based data collection are all given in Section 4.3.2 below.

Participants were paid $200 for their full participation in the study. Participants had the option of exiting the study early and receiving payment proportional to the length of their participation. No participants exercised this option.

4.3.2 Data Collection

In this section, we describe the data that were collected in our study. Data from the browsing extension are described first in Section 4.3.2, and then survey data are summarized in Section 4.3.2.

Extension-based Data

In order to record participants' browsing behavior and the metadata related to it, we developed a system that includes a Chrome browser extension for data collection and a server where the collected data are sent and stored. The extension monitors events in a browser using scripts in its background service worker.

We chose a set of user browsing actions to observe through this extension, which included: hitting the back button or forward button (*backButton*), creating a new tab either manually or by opening a link in a new tab (*newTab*), changing tabs (*tabChange*), typing in the address bar (*omniBox*), going to a new URL either by using the address bar or clicking a link in a page (*urlChange*), clicking a button in a webpage that does not change the URL (e.g., 'Like' on social media) (*click*), and typing in a textbox within a webpage (*type*). We chose these because they are common browsing actions that generate or affect internet packets from a user's browsing activity. Note that some of these actions occur within a fixed webpage (*urlChange, click, type*), while others are not necessarily affiliated with a specific website (*backButton, newTab, omniBox, tabChange*). We added an additional *awake* action which the extension would generate every 5 minutes if the browser was open and their computer was connected to the internet. This signal was intended to verify that participants had not uninstalled or disabled the extension during the study. No participants were removed from the analysis from missing *awake* actions. These observable events are summarized in Table 4.1.

When one of these actions occurred in a user's browsing, the Chrome extension recorded the action and relevant metadata including the time the event occurred, the URL (if any) on which the action occurred, and participant performing this action, and forwarded these data to a secure server.

To ensure privacy of the participants, each one was assigned a random ID that was used to associate them with their browsing actions. In this way, we could track the actions of each participant

awake	This action indicates that a user is online. Appears every 5 minutes when browser is open and online. Can occur when a user is not actively browsing.
backButton	Clicking on the back button
click	Click that does not cause URL change.
newTab	Opening new tab
omnibox	Typing in omnibox (address bar / search engine)
tabChange	Alternating between existing tabs
type	Typing a single character
urlChange	Click that causes URL change.

Table 4.1: Action types collected through the extension

without linking these data to his or her identity. The pre- and post-study surveys described next were also associated with participants' random IDs, rather than their names or other identifiers. However, it is still possible that the URLs of visited websites may be disclosive, particularly for long URLs that embed information beyond the domain name. To protect participants, we truncated the URLs in our collected data to contain only the domain name of the website that was visited, and removed the subdirectory information. For example, `www.facebook.com/UserName` became `www.facebook.com`. This did not affect our analysis because we are still able to track URL changes within a fixed domain name with the *urlChange* action.

To enable analysis of patterns of web use, we categorized the websites browsed by participants using the Symantec WebPulse Site Review tool [134]. This tool offers three levels of website categorization: categories, subgroups, groups of categories. For this work, we focus on subgroups of categories because they give the right level of granularity for our analysis: groups are too broad and not informative, while categories are too narrow and do not allow for statistical significance of tests due to the large number of categories. For ease of presentation, we refer to the subgroups simply as "categories", since these are our unit of measure for website categorization. These categories – along with examples of each – are presented in Table 4.2.

The JavaScript code for the extension, along with a description, can be found at `https://github.com/mzywang/browsing-experiment-extension`. This code can be used

Category	Subcategory (examples)
Adult Related	Adult/Mature Content, Gore/Extreme
Liability Concerns	Piracy/Copyright Concerns, Violence/Intolerance
Security Threats	Malicious Outbound Data/Botnets, Phishing
Security Concerns	Compromised Sites, Hacking, Spam
File Transfer	File Storage/Sharing, Peer-to-Peer (P2P)
Society/Government	Charitable/Non-Profit, Government/Legal
Social Interaction	Personal Sites, Social Networking
Multimedia	Audio/Video Clips, Media Sharing
Communication	Email, Internet Telephony, Online Meetings
Health Related	Health, Restaurants/Food, Tobacco
Leisure	Art/Culture, Entertainment, Games
Commerce	Cryptocurrency, Job Search/Careers, Shopping
Technology	Cloud Infrastructure, Computer/Information Security
Information Related	Education, News, Reference, Search Engines/Portals

Table 4.2: Website categories, Symantec WebPulse Site Review [134]

for replication of our study, and may be of independent interest for future research involving browsing data collection.

Self-reported Data

Two surveys were used to assess participants' self-perceptions of their browsing behavior and to collect demographic data. They were conducted immediately before and after the period of browsing data collection.

In the pre-study survey, we asked participants to report their age, gender, ethnicity, and race. We also asked participants to report "How many hours per day, on average, would you say that you spend online?", reported on a slider from 0 to 24 hours; and "What are your most frequented categories of websites to visit? Please select all that apply.", with answer choices: "Social Network (Facebook, Instagram, Reddit, etc)", "Business (Onenote, Dropbox, Linkedin, etc)", "Entertainment (Youtube, Netflix, IMDB, etc)", "News (CNN, ESPN, etc)", "Search (Google, Bing, etc)", "Banking (Paypal, Any personal bank, etc)", "Shopping (Amazon, Walmart, etc)", "Blogging (Tumblr, Wordpress, etc)", "Reference (Wikipedia, Weather, etc)". Additional questions related

to internet use and identity were asked, but are not analyzed in this work.

In the post-study survey, we asked participants whether: "During the course of the study, did you change your browsing behavior to prevent information from being learned about you?", with answer choices "Yes" and "No". (As in the pre-study survey, additional questions related to internet use were asked but are not analyzed in this work.)

Demographic	**Group**	**Number (%)**
Gender	Female	12 (38.7%)
	Male	19 (61.3%)
Age	18-24	14 (45.2%)
	25-34	14 (45.2%)
	35-44	3 (9.7%)
Race[1]	Asian	7 (22.6%)
	Black or African American	10 (32.3%)
	White	9 (29.0%)
	Two or more races	3 (9.7%)
Nationality	USA	16 (51.6%)
	Other	15 (48.4%)

Table 4.3: Participant demographics

Our participants were primarily undergraduate and graduate students at the Georgia Institute of Technology. Demographics features of our participants are presented in Table 4.3.

4.3.3 Data Analysis

RQ1: Differences in behavior. We first examine differences in web use between participants of different genders, races, and ages, using two metrics of browsing activity: time spent browsing and number of browsing actions. To compute the amount of time participants spent browsing, we convert sequences of instantaneous browsing actions into *clickstreams* of consecutive actions performed by one participant, which represent a period of continuous active browsing. Prior work ended a clickstream after periods of inactivity ranging from 30 seconds on Facebook [60] to 30

[1]One participant preferred not to disclose their race, and one participant responded with their ethnicity instead of race.

minutes across all websites. We chose to use 30 minutes of inactivity as a cutoff because we considered the full range of internet browsing.

We use a one-sided t-test with the null hypothesis of no difference between the mean number of daily browsing actions (or mean number of hours spent browsing) on average across days with observed online behavior, for each relevant pair of demographic groups. For brevity, we refer to these two metrics respectively as "daily average number of browsing actions" and "daily average browsing time," as we use these same metrics when measuring browsing activity for the other RQs. We apply bootstrapping to account for smaller sample sizes, and we correct for multiple testing using Bonferonni-Holm correction. For statistical significance reasons, for race we compare only Asian, Black or African American, and white groups, since these have sufficient representation in our sample.

To investigate patterns of web use, we test whether participants had similar distribution of browsing actions within websites of different categories. Only three types of browsing actions could occur within a website: *click*, *type*, and *urlChange*.[2] We measure the empirical distribution of these actions across category, and use a Pearson's χ^2-test for homogeneity to test whether the distribution of actions were the same (pairwise) across categories.

RQ2: Accuracy of perceptions. To address RQ2, we compare participants' observed browsing behavior with their self-reported daily time spent browsing and most browsed website categories. To measure differences in terms of time spent browsing, we introduce a value δ_i for each participant i, which is defined as the difference between their daily average browsing time, and their self-reported daily time spent online. That is, if participant i spent S_i total hours of active browsing across n_i days of the study (i.e., they were active for n_i out of the 14 days), and they self-report spending t_i hours per day online, then δ_i is defined as $\delta_i = \frac{S_i}{n_i} - t_i$. We use a t-test to determine whether the mean of the δ_is among participants in our study is significantly different from

[2]All other actions involved changing websites or actions outside of a website, such as creating a new tab, and thus could not be associated with a particular website category.

0, to determine whether there is a significant difference in observed and perceived browsing time. We additionally examine whether differences between observed and perceived browsing time vary based on demographic group. We use a t-test for mean equality of the δ_is between demographic groups, and apply bootstrap techniques to improve the statistical power of our small sample.

We then examine whether participants' perceptions of their most commonly browsed types of websites are correct. Our pre-study survey asked participants to select any number of website categories that they most frequently browse. For participant i who reported k_i top browsing categories, we compare their reports to their top k_i categories of observed browsing time. We report true/false positive/negative rates for each website category, as well as overall percentage of participants with correct/incorrect perceptions. To determine browsing time in each category, we separate clickstream browsing time by category, by dividing browsing sessions when the participant began browsing a URL of another category. For this portion of the analysis, we use categorization from Alexa Top Websites [135] because our pre-study survey listed these category choices. Unfortunately, this Alexa Top Websites tool was retired after our study was conducted. Thus we use Alexa Top Websites categorization here for consistency with the survey, and we use the Symantec WebPulse Site Review Request for the remainder of analysis in the paper to enable reproducibility.

RQ3: Changes with observation. With RQ3, we aim to test whether people behave differently online when they are consciously aware of being observed. We hypothesize that participants' conscious awareness of being observed may be the most salient early in the study, shortly after they provide informed consent for data collection, and that this awareness may diminish over time. This is consistent with evidence that the behavioral effects of observation are amplified by interpersonal reminders of the observation [66, 65]. If this were the case, we would expect to observe a change in participants' behavior over time during the study. As a proxy measure for this analysis, we test whether the distribution of participants' activity during the first half of the study (Days 1-7) is different from the second half (Days 8-14).

We first test for changes in participants' level of browsing activity, as measured by both the

daily average number of browsing actions and daily average browsing time. For both activity metrics, we test for changes in the mean and the variance in participants' level of browsing activity, using a t-test and Levene test, respectively. We then test for changes in the distribution of website categories browsed under both metrics. We use a Wilcoxon signed-rank test to evaluate if the aggregated distribution of activity across categories from the first and the second halves of the study are different.

4.3.4 Limitations

As with any user study, our findings are subject to multiple practical limitations. First, our sample population was relatively small and consisted primarily of undergraduate and graduate students. We used bootstrapping, which is a robust and commonly-used technique for calculating estimators when sample sizes are small or assumptions about normality of the sampling distribution cannot be made [136]. Our sample is not fully representative of the internet-using population, and our results should be interpreted in this context.

Second, our surveys relied on self-reported user data. While one of the goals of this work was to measure whether users' perceptions of their browsing behavior were accurate, certain self-reported data (e.g., demographics) were unverifiable in the study. Participants could also have misinterpreted the survey questions, or changed their answers due to desirability bias towards a more acceptable behavior [137].

Third, users could turn off the data collecting extension at any point during the study. While this feature was necessary for ethical data collection — participants must have the option to opt out of the study at any time — it also allowed for a potential bias in the collected data, as participants could disable data collection on embarrassing or sensitive websites.

Finally, our metric of active browsing time converts a series of instantaneous events into an aggregate measure of time spent browsing, as is the convention in prior work [60, 133, 67]. This approach does not capture passive browsing activities, such as streaming a movie or reading an

article, although we do explore alternative methodologies for measuring active browsing time in Appendix B.2.

4.4 RQ1: Does browsing behavior differ across user demographic groups and type of web use?

In this section, we measure our participants' browsing behavior in terms of time spent browsing and number of browsing actions. We test whether that behavior differs based on the type of user (i.e., demographics) in Section 4.4.1 and type of web use (i.e., website category) in Section 4.4.2.

Overall, we observe that our participants spent an average of 146 minutes (SD = 100.5) browsing daily during the course of the study. Participants averaged 968 browsing actions per day (SD = 1529). Figure 4.1 illustrates the distribution of time participants spent on each website category and number of browsing actions in each category. We see that "Information Related", "Commerce", and "Technology" websites are the most popular according to both metrics of activity. Some categories, such as "Social Interaction," "Leisure," and "Multimedia", are popular under one metric, but not the other. This suggests, for example, that browsing "Leisure" and "Multimedia" websites does not involve as many click actions as other categories of websites. Recall the descriptions and examples of each website category given in Table 4.2. Figure B.1 in Appendix B.1 shows the distribution of browsing actions and website category for the top 100 most browsed websites by all users during the study.

Compared to prior work conducted 6 and 9 years prior to our study, respectively, we observe that our participants visit similar numbers of pages per session (5-151 per session vs. 14-130 in [67]), but spend more time online (median of 2.9 hours per day vs. a median of an hour per day in [55]).

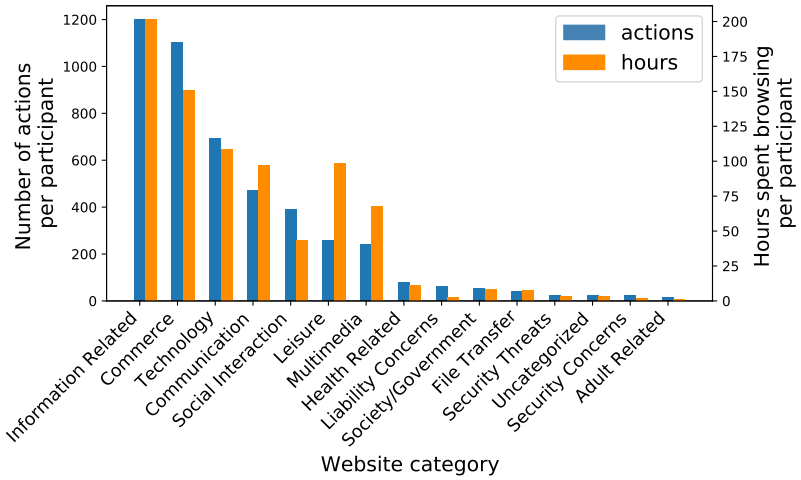


Figure 4.1: Distribution of number of browsing actions (blue) and browsing time in hours (orange) on different website categories averaged over all participants in our study.

4.4.1 Differences across demographic groups

First, we consider the differences in participants' behavior across demographic groups, motivated by prior work showing relationships between browsing behavior and demographic features [56, 57]. We explore this by testing for differences in daily average number of browsing actions and daily average browsing time (per person) across demographic groups.

With daily average number of browsing actions, we find no significant difference between genders ($t = -0.228$, $p = 0.822$), between Black or African American and Asian participants ($t = 0.688$, $p = 0.502$), between white and Asian participants ($t = -0.321$, $p = 0.754$), and between white and Black or African American participants ($t = -0.760$, $p = 0.461$). We also observe no significant difference in daily activity level between those aged 18-24 and those aged 25-34 ($t = 0.322$, $p = 0.999$). However, the daily activity of older participants (aged 35-44) is significantly lower than that of those aged 18-24 years and 25-34 years ($t = 3.301$, $p = 0.007$ and $t = 2.994$, $p = 0.051$, respectively).

With daily average browsing time, we find that there is no significant difference between genders ($t = -0.073$, $p = 0.950$) and among races (pairwise, $t = -0.842$, $p = 0.381$; $t = -0.642$,

$p = 0.501$; $t = 0.114$, $p = 0.918$). We do not observe a significant difference in daily average browsing time between those 18-24 years old and those aged 25-34 ($t = -0.467$, $p = 0.999$). However, participants aged 35-44 on average spend significantly less time online daily than younger participants ($t = 3.297$, $p = 0.007$ in comparison to those aged 18-24, and $t = 3.187$, $p = 0.013$ in comparison to those aged 25-34, respectively). See Table 4.4 and 4.5 below for a full presentation of these tests and their p-values.

Feature	p-value
Gender	
Male vs Female	0.822
Race	
Asian vs Black or African American	0.502
Asian vs White	0.754
Black or African American vs White	0.461
Age	
18-24 vs 25-34	0.999
18-24 vs 35-44	**0.007**
25-34 vs 35-44	**0.051**

Table 4.4: Test for equality of means of daily average number of browsing actions

Feature	p-value
Gender	
Male vs Female	0.950
Race	
Asian vs Black or African American	0.381
Asian vs White	0.501
Black or African American vs White	0.918
Age	
18-24 vs 25-34	0.999
18-24 vs 35-44	**0.007**
25-34 vs 35-44	**0.013**

Table 4.5: Test for equality of means of daily average browsing time.

4.4.2 Differences in behavior across types of web use

Next, we explore how users' behavior within a website changes across different categories of websites. This is motivated in part by existing literature showing that users interact differently with different websites [55, 67]. We aim to understand whether this behavior varies structurally by website category. We measure behavior by the distribution of browsing actions within each website category, rather than total number of browsing actions or time spent browsing because our goal is to measure differences in how users interact with a website, rather than their level of interaction.

Figure 4.2 shows the empirical distribution of click events on different website categories. We observe qualitatively that for most website categories, participant actions were mostly *click*, slightly fewer *urlChange*s, and a small number of *type* actions. A notable exception is Security

Concerns websites, which saw significantly different behavior from all categories ($\chi^2 > 27$, $p < 10^{-5}$ for all categories). Behavior on Multimedia websites was found to be significantly different from behavior on File Transfer ($\chi^2 = 15.349$, $p = 0.036$) and Security Threats ($\chi^2 = 15.969$, $p = 0.027$) websites. The observed differences between all other pairs of websites were not significant. Table B.1 in Appendix B.1 shows the p-values from this test, presented for each pair of website categories.

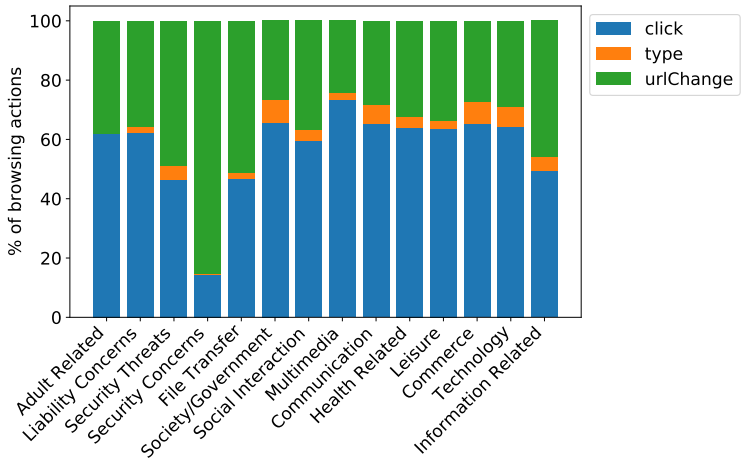


Figure 4.2: Empirical distributions of participants' *click*, *type*, and *urlChange* browsing actions within each website category.

Security Concerns. In our study, 10 out of 31 participants visited a total of 62 Security Concerns websites during the two-week period during which they were observed. There were four Security Concerns subcategories that were visited by these participants: Suspicious (e.g., `www.netlflix.com`), Placeholders (e.g., `www.canvas.com`, `www.richvideos.com`), Potentially Unwanted Software (e.g., `www2.securybrowse.com`), and Hacking (e.g., `www.recoverlostpassword.com`). The majority of the Security Concerns websites visited in our study were in the Suspicious subcategory (48 out of 62).

The fact that user behavior on Security Concerns websites differs from all other types of browsing makes sense given known typical behavior of malicious websites, which aim to redirect users to further malicious pages and/or capture their credentials [138]. The behavior-distribution signals we

observe may serve to augment existing approaches to detecting new or unknown Security Concerns websites [139, 140, 141]. Additionally, such signals may be useful for developing just-in-time in-browser warnings about a potential security concern based on observed browsing behavior on the website.

4.5 RQ2: Do people have correct perceptions of their behavior online?

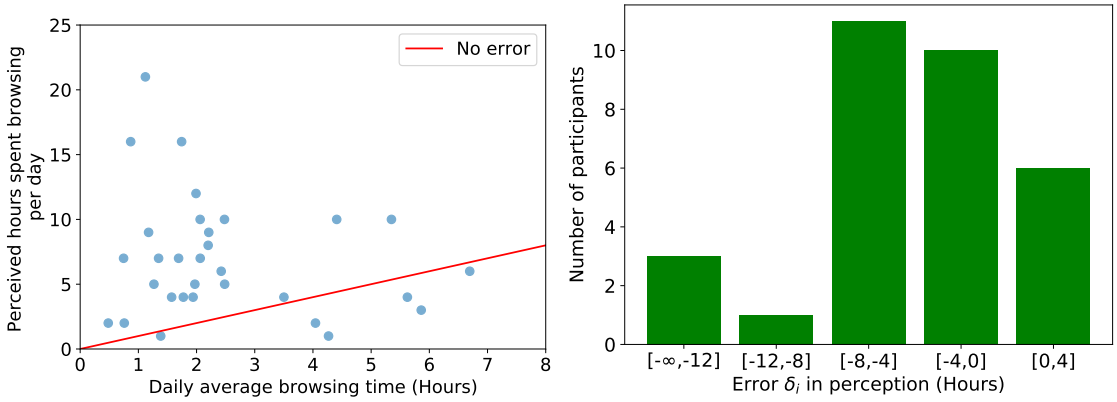
In this section, we test whether participants had accurate perceptions of their online browsing behavior. We first evaluate participants' perceptions of their time spent online and how this varies by demographic group in Section 4.5.1, and then we measure participants' perceptions of the website categories that they most frequently browse in Section 4.5.2.

Overall, we observe that our participants think they spend on average 6.87 hours (SD = 4.6) per day browsing. In response to the question, "What are your most frequented categories of websites to visit?", the most common answers were "Entartainment", "Search", and "Social Network" (respectively from 27, 27, and 23 participants out of 31). The full list of categories with the number of participants who chose each category as their most frequently visited can be found in Table B.2 in Appendix B.1.

4.5.1 Perceptions of time spent browsing

We find that the majority of participants (26 out of 31, 80.6%) significantly over-reported their daily browsing time. Figure 4.3 illustrates the relationship between participants' observed time spent browsing and their perceived (self-reported) time spent browsing. Figure 4.3a shows a scatter plot with one dot corresponding to each participant, where the x-coordinate is their daily average browsing time, and the y-coordinate is the self-reported daily time spent browsing. The red line ($x = y$) corresponds to no error in perceptions, and points further from this line have larger error between perceptions and actual browsing behavior. We observe that most participants substantially overestimated their time spent browsing, as evidenced by the number of points above the red line.

Figure 4.3b aggregates this information to illustrate the error in participants' perceptions of their browsing behavior at the population-level. Recall that δ_i is the difference between the actual (observed) daily average browsing time of participant i, and their self-reported (perceived) daily browsing time. Since a large fraction of participants had a negative value of δ_i, we see that most participants over-reported their time spent browsing. The average error δ_i among our participants is -4.5 hours (SD=5.24). A more detailed visualization of this error at the participant-level is illustrated in Figure B.2 in Appendix B.1.



(a) Observed vs perceived browsing time

(b) Distribution of error in perceptions

Figure 4.3: (a) Scatter plot illustrating actual daily average browsing time vs. perceived (self-reported) number of hours spent browsing per day. Each point corresponds to one participant. (b) Distribution of error values δ_i in the participant population.

Alternative measures of activity. For measurements of active browsing time, we followed the convention of [67] to assume that a browsing session ends after 30 minutes of inactivity (i.e., no browsing actions aside from the *awake* action were recorded by the browser extension for 30 minutes). Other existing literature used cutoffs ranging from 30 seconds [60] to 20 minutes [133] of inactivity. Using shorter cutoff times to indicate inactivity would only reduce the recorded time spent browsing, and thus increase overestimation of browsing activity. In Figure B.3 in Appendix B.2, we consider the impact of using 5 minutes of inactivity as a cutoff. Since this only reduces the recorded time spent browsing, unsurprisingly, we find that overestimation of browsing activity

increases.

In the analysis above, we only counted time spent browsing on a laptop or desktop, as measured by our browsing extension, and did not include mobile browsing activity. Recent 2021 data [142] show that users spend 55.9% of their browsing time on a desktop or laptop. We repeat the analysis above with this adjustment factor, by scaling down each participant's self-report by a factor of 0.559 to account for only measuring desktop/laptop browsing. Even after the adjustment, most participants still overestimate the amount of time they spend online, relative to our observational measurements (80.6% of without adjustment vs. 77.4% with adjustment). A more detailed analysis can be found in Appendix B.2.

Demographic variance. We additionally investigate whether the biases in participants' perceptions differ across demographic groups. For each feature, we test for equality of means for δ_i across groups. We find no significant difference among genders ($t = -0.536$, $p = 0.599$), age groups (pairwise, $t = -0.796$, $p = 0.433$; $t = 0.300$, $p = 0.791$; $t = 0.508$, $p = 0.659$), and races (pairwise, $t = 0.038$, $p = 0.970$; $t = -0.831$, $p = 0.420$; $t = -0.673$, $p = 0.511$). A complete presentation of these results is given in Table 4.6.

Feature	p-value
Gender	
Male vs Female	0.599
Race	
Asian vs Black or African American	0.970
Asian vs White	0.420
Black or African American vs White	0.511
Age	
18-24 vs 25-34	0.433
18-24 vs 35-44	0.791
25-34 vs 35-44	0.659

Table 4.6: p-values for pairwise t-test for equality of means for perceptions δ_is across demographic groups.

4.5.2 Perceptions of browsing activity by website category

Next, we investigate whether participants had correct perceptions about the type of websites they browse most frequently. In the pre-study survey, we asked participants "What are your most

frequented categories of websites to visit?" Each participant could select as many as they desired from the list of: Social Network, Business, Entertainment, News, Search, Banking, Shopping, Blogging, and Reference. These options correspond to categories on Alexa Top Websites by Category [135] (see Section 4.3.3 for details). Each participant i reported their k_i most frequently visited categories; we compared this with their top k_i website categories, as measured by total time spent browsing. Participants on average chose 4.53 categories (SD = 1.34). See Table B.2 in Appendix B.1 for the number of participants who chose each category and the number of participants for whom each category was among their top k_i.

A table on Figure 4.4 presents a confusion matrix that summarizes whether participants' self-reported top browsing categories were among their actual most browsed categories. Figure 4.4 also shows the percentage of participants with correct and incorrect perceptions for each category. In the table on Figure 4.4, orange shaded cells indicate incorrect perceptions. Specifically, the orange shaded column on the left (observed in top k_i categories of browsing, but not self-reported in top k_i) corresponds to participants who underestimated the amount of time they spent on each category, which indicates false positive rate. The orange shaded column on the right (not observed in top k_i categories of browsing, but self-reported to be in top k_i) corresponds to participants who overestimated the time they spent on each category, which indicates false negative rate.

We observe that while participants over-report their time online (see prior section), they are relatively accurate in their perceptions of where they spend their time. Out of 145 total top categories reported in total by our 31 participants, 50.3% truly were top categories of the participant's observed behavior. The categories are sorted in the table on Figure 4.4 by accuracy of participant perceptions. We see that participants had the most accurate perceptions of their browsing on Blogging and News websites, and the least accurate perceptions of Shopping and Business websites. We also see that most of the error in perceptions came from participants overestimating their level of browsing a particular category (i.e., false positive), which happened uniformly across website categories.

Category	Observed Among Top k_i		Observed Not Among Top k_i		Correct perception	Incorrect perception
	Self-Report Top k_i	Self-Report Not Top k_i	Self-Report Top k_i	Self-Report Not Top k_i		
Blogging	0%	3.1%	12.5%	84.4%	84.4%	15.6%
News	0%	3.1%	18.8%	78.1%	78.1%	21.9%
Search	65.6%	9.4%	18.7%	6.3%	71.9%	28.1%
Social Network	53.1%	9.4%	18.7%	18.8%	71.9%	28.1%
Banking	6.3%	0%	31.2%	62.5%	68.8%	31.2%
References	28.1%	15.6%	21.9%	34.4%	62.5%	37.5%
Entertainment	56.3%	9.4%	28.1%	6.2%	62.5%	37.5%
Shopping	12.5%	6.3%	40.6%	40.6%	53.1%	46.9%
Business	6.3%	18.7%	34.4%	40.6%	46.9%	53.1%

Figure 4.4: Confusion matrix showing accuracy of participants perceptions regarding the website categories they most frequently browse. Each participant i self-reported their k_i top website categories, and these were compared with their top k_i categories of observed browsing based on time spent browsing. Blue shaded cells indicate correct perceptions (true positives or true negatives), and orange shaded cells indicate incorrect perceptions (false positives or false negatives). Total correct and incorrect perceptions are also calculated for each category.

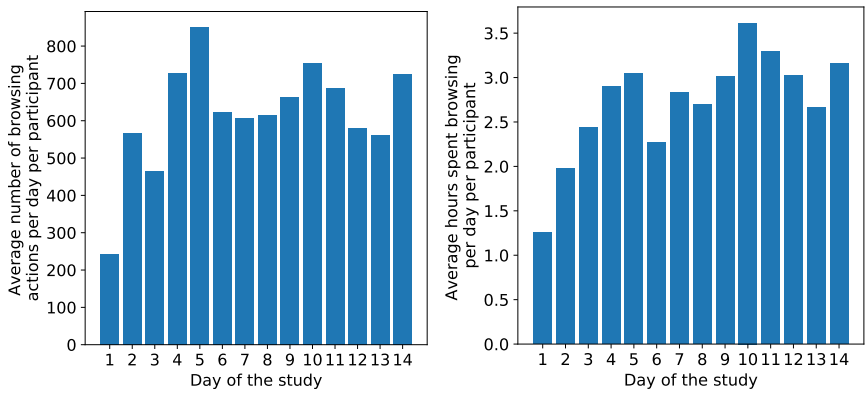
4.6 RQ3: Do people change browsing behavior if they are aware of being observed?

In this section, we aim to test whether people behave differently online when they are consciously aware of being observed. We hypothesize that participants' conscious awareness of being observed may be the most salient early in the study, shortly after they provide informed consent for data collection, and that this awareness may diminish over time. If this were the case, we would expect to observe a change in participants' behavior over time during the study. As a proxy measure for this analysis, we test a hypothesis that the distribution of participants' activity during the first half of the study (Days 1-7) is different from the second half (Days 8-14). We first test for changes in level of browsing activity in Section 4.6.1, and then for differences in website categories browsed in Section 4.6.2.

4.6.1 Changes in level of activity

We investigate whether participants changed their level of browsing activity during the course of the study. We use as activity metrics both daily average number of browsing actions and daily average browsing time. Figure 4.5 shows the daily average number of browsing actions and the

daily average browsing time, both averaged across all participants. We note that participants installed the browsing extension during the first day of the study, so browsing activity is noticeably lower on Day 1 since a full day of browsing was not captured.



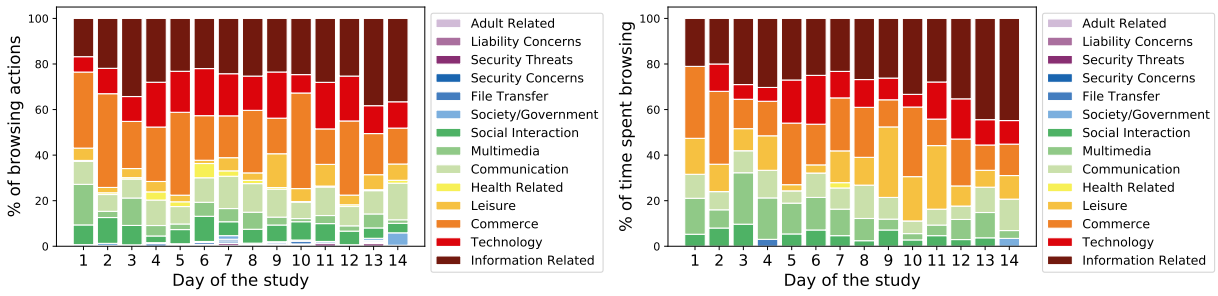
(a) Number of browsing actions (b) Time spent browsing

Figure 4.5: Average number of actions and time spent browsing, per participant per active browsing day of the study.

While we observe variance in average daily activity, we do not find significant differences in the level of browsing activity observed over the duration of the study. Specifically, under the activity metric of daily average number of browsing actions per active browsing day, we find that neither the mean number of actions ($t = -0.915$, $p = 0.348$) nor variance in number of actions ($L = 2.009$, $p = 0.182$) differs significantly between the first and second half of the study. Under the metric of daily average browsing time, similarly, both mean of the number of hours (-1.230, $p = 0.208$) and variance ($L = 2.191$, $p = 0.165$) do not show a significant difference between the first and second half of the study. Participants' reports in our post-study survey support these findings, with only 2 of 31 participants reporting that that they altered their behavior during the study.

4.6.2 Changes in type of web use

We also investigate whether participants' browsing activity across website categories changed over the course of the study. Figure 4.6 shows the proportion of browsing activity across website categories as measured by both the number of browsing actions and hours spent browsing. With number of browsing actions, we do not observe a significant change in distribution of web use across website categories between the first half of the study (Days 1-7) and the second half (Days 8-14) ($W = 37.0, p = 0.357$); similarly, under time spent browsing, we do not observe a significant difference ($W = 38.0, p = 0.390$).



(a) Browsing actions

(b) Time spent browsing

Figure 4.6: Proportion of (a) browsing actions and (b) time spent browsing on each website category on each day of the study.

4.7 Discussion and Conclusions

In this chapter, we provide an up-to-date picture of a young (under 45 years old) sample of internet users' browsing behavior (RQ1). We find that people are viewing similar numbers of pages today as in prior work but are spending significantly more time online (those in our sample spent an average of three hours a day online compared to prior work conducted nearly a decade ago, which observed an average of one hour of daily online activity [67, 55]). Echoing prior work conducted nearly a decade ago [56], we find relatively little demographic variance in browsing activity, although differently from prior work that leveraged demographic inference data [56], we

do observe that the older users (35-44) in our sample browse significantly less than those who are younger.

Our work adds to the body of knowledge on digital browsing behavior in that we examine not only how much time people spend online and how many pages they view, but *what* they do online. Prior work has studied user behavior in terms of webpage access time [67] and number of page revisits [55]. To the best of our knowledge, this is the first work to study user behavior in terms of the proportion of types of actions performed on websites and across different website categories. We find that people spend most of their time on Information Related, Commerce, Technology, Leisure, and Communication websites, with Social Interaction (social media) websites ranking seventh. While people spend different amounts of time on different types of pages, they behave quite similarly in terms of the actions they take (clicks, typing, urlChanges) on these pages. There are a few exceptions: Multimedia websites see less typing and urlChanges, as people are primarily clicking on and watching videos; File Transfer websites see a high number of urlChanges characterizing file uploads; and Security Concern websites, which include suspected phishing URLs and misspellings of popular URLs, and can be characterized by a high number of urlChanges, in line with prior findings that malicious websites aim to redirect users to additional malicious pages and opportunities for credential capture [138]. These findings suggest that one potentially promising direction for augmenting existing approaches [139, 140, 141] to keeping people safe online is to add common website interaction patterns as signals for detecting malicious websites.

Further, our work addresses a critical question for the study of online behavior: we examine the relationship between participants' self-reported online browsing — in terms of time spent online and types of web uses — and their actual behavior as we observe it using our measurement tools (RQ2). We find that participants significantly over-report their daily time spent online, by an average of 4.5 hours per day; this overreporting does not vary with demographics (age, gender, or race). This finding aligns with prior work that examined the accuracy of people's self-reports about their Facebook behavior, specifically, finding that people overestimated their time spent on

the platform [60]. This suggests that findings regarding the relationship between various digital constructs (e.g., social capital, digital skill [131, 132]) and self-reported time spent online should be interpreted with care: people's perceptions of how much time they spend online may over-represent the time they actually spend online.

While participants in our study over-reported their time spent online, they were relatively accurate in their reports about the types of websites where they spent the most time. This suggests, in line with prior work examining the accuracy of people's self reports about the speed with which they update their computers [59], that people may have an accurate *relative* sense of their digital behavior, but inaccurate absolute perceptions (i.e., about the exact amount of time they spend online or the precise strength of their passwords [61]). This suggests that observational methods of measurement may be most appropriate for use when precise absolute measurements are necessary, but that self-report measurements may be an appropriate proxy when only relative measurements are required.

Finally, given prior findings from other fields on possible observational biases that may occur when participants are aware that their behavior is being observed [62], we examine whether participants' observed behavior changed over the course of our experiment to see whether we could detect such observational biases in our measurements of web behavior (RQ3). We find no significant changes in participant behavior over the course of the study. It is possible that we observe no behavior change because 14 days is not a sufficiently long period of time for participants to forget that they are being observed. Alternately, people may have such a pervasive sense of being observed online [143] that even installing a browser plugin that they know observes their behavior may not change their activity. Future work is necessary to further explore the question of observation bias in measurements of digital behavior, perhaps through comparison of proprietary industry measurement data – which a user is not actively aware is being collected – with measurement data from a disclosed browser plugin such as the one we use in this study.

References

[1] N. Confessore, "Cambridge analytica and facebook: The scandal and the fallout so far," *The New York Times*, Apr. 4, 2018.

[2] M. Isaac and S. Frenkel, "Facebook security breach exposes accounts of 50 million users," *The New York Times*, Sep. 28, 2018.

[3] K. O'Flaherty, "Facebook data breach: Here's what to do now," *Forbes*, Apr. 6, 2021.

[4] J. Abowd, R. Ashmead, R. Cumings-Menon, S. Garfinkel, M. Heineck, C. Heiss, R. Johns, D. Kifer, P. Leclerc, A. Machanavajjhala, B. Moran, W. Sexton, M. Spence, and P. Zhuravlev, "The 2020 Census Disclosure Avoidance System TopDown Algorithm," *Harvard Data Science Review*, no. Special Issue 2, 2022, https://hdsr.mitpress.mit.edu/pub/7evz361i.

[5] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of cryptography conference*, Springer, 2006, pp. 265–284.

[6] A. Friedman and A. Schuster, "Data mining with differential privacy," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2010, pp. 493–502.

[7] R. Bassily, A. Smith, and A. Thakurta, "Private empirical risk minimization: Efficient algorithms and tight error bounds," in *2014 IEEE 55th annual symposium on foundations of computer science*, IEEE, 2014, pp. 464–473.

[8] A. Beimel, K. Nissim, and U. Stemmer, "Learning privately with labeled and unlabeled examples," in *Proceedings of the twenty-sixth annual ACM-SIAM symposium on Discrete algorithms*, SIAM, 2014, pp. 461–477.

[9] K. Chaudhuri and C. Monteleoni, "Privacy-preserving logistic regression," *Advances in neural information processing systems*, vol. 21, 2008.

[10] F. McSherry and I. Mironov, "Differentially private recommender systems: Building privacy into the netflix prize contenders," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2009, pp. 627–636.

[11] A. Machanavajjhala, A. Korolova, and A. D. Sarma, "Personalized social recommendations-accurate or private?" *arXiv preprint arXiv:1105.4254*, 2011.

[12] R. Shokri and V. Shmatikov, "Privacy-preserving deep learning," in *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, 2015, pp. 1310–1321.

[13] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, 2016, pp. 308–318.

[14] C.-A. Azencott, "Machine learning and genomics: Precision medicine versus patient privacy," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 376, no. 2128, p. 20 170 350, 2018.

[15] B. Berger and H. Cho, *Emerging technologies towards enhancing privacy in genomic data sharing*, 2019.

[16] V. Feldman, A. McMillan, and K. Talwar, "A simple and nearly optimal analysis of privacy amplification by shuffling," 2021.

[17] K. Talwar, *Differential secrecy for distributed data and applications to robust differentially secure vector summation*, 2022.

[18] V. Feldman and T. Zrnic, *Individual privacy accounting via a renyi filter*, 2021.

[19] A. Beimel, H. Kaplan, Y. Mansour, K. Nissim, T. Saranurak, and U. Stemmer, "Dynamic algorithms against an adaptive adversary: Generic constructions and lower bounds," in *STOC 2022*, 2022.

[20] J. Gillenwater, M. Joseph, A. M. Medina, and M. R. Diaz, "A joint exponential mechanism for differentially private top-k," in *International Conference on Machine Learning (ICML) 2022*, 2022.

[21] E. Tsfadia, E. Cohen, H. Kaplan, Y. Mansour, and U. Stemmer, "Friendlycore: Practical differentially private aggregation," in *ICML 2022*, 2022.

[22] A. Bietti, C.-Y. Wei, M. Dudik, J. Langford, and Z. S. Wu, "Personalization improves privacy-accuracy tradeoffs in federated optimization," *arXiv preprint arXiv:2202.05318*, 2022.

[23] M. Geppert, V. Larsson, J. L. Schönberger, and M. Pollefeys, "Privacy preserving partial localization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 17 337–17 347.

[24] F. Niu, H. Nori, B. Quistorff, R. Caruana, D. Ngwe, and A. Kannan, "Differentially private estimation of heterogeneous causal effects," *arXiv preprint arXiv:2202.11043*, 2022.

[25] G. Kamath, X. Liu, and H. Zhang, "Improved rates for differentially private stochastic convex optimization with heavy-tailed data," in *International Conference on Machine Learning*, PMLR, 2022, pp. 10 633–10 660.

[26] A. Hannun, C. Guo, and L. van der Maaten, "Measuring data leakage in machine-learning models with fisher information," in *Uncertainty in Artificial Intelligence*, PMLR, 2021, pp. 760–770.

[27] M. Malek, I. Mironov, K. Prasad, I. Shilov, and F. Tramèr, "Antipodes of label differential privacy: Pate and alibi," *arXiv preprint arXiv:2106.03408*, 2021.

[28] I. Intelligence, "Digital advertising in 2022: Market trends and predictions," Apr. 20, 2022.

[29] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Proceedings of the 3rd Conference on Theory of Cryptography*, ser. TCC '06, 2006, pp. 265–284.

[30] F. McSherry and K. Talwar, "Mechanism design via differential privacy," in *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, IEEE, 2007, pp. 94–103.

[31] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Foundations and Trends in Theoretical Computer Science*, vol. 9, no. 3–4, pp. 211–407, 2014.

[32] C. Dwork, M. Naor, O. Reingold, G. N. Rothblum, and S. P. Vadhan, "On the complexity of differentially private data release: Efficient algorithms and hardness results," in *Proceedings of the 41st ACM Symposium on Theory of Computing*, ser. STOC '09, 2009, pp. 381–390.

[33] C. Dwork, M. Naor, T. Pitassi, and G. N. Rothblum, "Differential privacy under continual observation," in *42nd ACM Symposium on Theory of Computing*, ser. STOC '10, 2010.

[34] C. Gallagher, R. Lund, and M. Robbins, "Changepoint detection in climate time series with long-term trends," *Journal of Climate*, vol. 26, no. 14, pp. 4994–5006, 2013.

[35] R. Lund and J. Reeves, "Detection of undocumented changepoints: A revision of the two-phase regression model," *Journal of Climate*, vol. 15, no. 17, pp. 2547–2554, 2002.

[36] W. A. Shewhart, *Economic Control of Quality of Manufactured Product*. D. Van Norstrand Company, Inc., 1931.

[37] E. S. Page, "Continuous inspection schemes," *Biometrika*, vol. 41, no. 1/2, pp. 100–115, 1954.

[38] A. N. Shiryaev, "On optimum methods in quickest detection problems," *Theory of Probability & Its Applications*, vol. 8, no. 1, pp. 22–46, 1963.

[39] M. Pollak, "Average run lengths of an optimal method of detecting a change in distribution," *The Annals of Statistics*, vol. 15, no. 2, pp. 749–779, 1987.

[40] Y. Mei, "Is average run length to false alarm always an informative criterion?" *Sequential Analysis*, vol. 27, no. 4, pp. 354–419, 2008.

[41] J. Bai and P. Perron, "Computation and analysis of multiple structural change models," *Journal of Applied Econometrics*, vol. 18, no. 1, pp. 1–22, 2003.

[42] N. Zhang and D. O. Siegmund, "Model selection for high-dimensional, multi-sequence change-point problems," *Statistica Sinica*, vol. 22, no. 4, pp. 1507–1538, 2012.

[43] B. Darkhovskh, "A nonparametric method for the a posteriori detection of the "disorder" time of a sequence of independent random variables," *Theory of Probability & Its Applications*, vol. 21, no. 1, pp. 178–183, 1976.

[44] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[45] M. Gong, Y. Xie, K. Pan, K. Feng, and A. K. Qin, "A survey on differentially private machine learning," *IEEE computational intelligence magazine*, vol. 15, no. 2, pp. 49–64, 2020.

[46] S. Fletcher and M. Z. Islam, "Differentially private random decision forests using smooth sensitivity," *Expert systems with applications*, vol. 78, pp. 16–31, 2017.

[47] K. Chaudhuri, C. Monteleoni, and A. D. Sarwate, "Differentially private empirical risk minimization.," *Journal of Machine Learning Research*, vol. 12, no. 3, 2011.

[48] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on knowledge and data engineering*, vol. 21, no. 9, pp. 1263–1284, 2009.

[49] Y. Sun, A. K. Wong, and M. S. Kamel, "Classification of imbalanced data: A review," *International journal of pattern recognition and artificial intelligence*, vol. 23, no. 04, pp. 687–719, 2009.

[50] N. V. Chawla, N. Japkowicz, and A. Kotcz, "Special issue on learning from imbalanced data sets," *ACM SIGKDD explorations newsletter*, vol. 6, no. 1, pp. 1–6, 2004.

[51] A. Lau and J. Passerat-Palmbach, "Statistical privacy guarantees of machine learning preprocessing techniques," *arXiv preprint arXiv:2109.02496*, 2021.

[52] B. Galarita, "How to protect your student data in college," *Forbes*, Jul. 28, 2022.

[53] H. Nissenbaum and H. Daniel, "Trackmenot: Resisting surveillance in web search," 2009.

[54] D. C. Howe and H. Nissenbaum, "Engineering privacy and protest: A case study of adnauseam.," in *IWPE@ SP*, 2017, pp. 57–64.

[55] R. Kumar and A. Tomkins, "A characterization of online browsing behavior," in *Proceedings of the 19th International Conference on World Wide Web*, ser. WWW '10, 2010, pp. 561–570.

[56] S. Goel, J. Hofman, and M Sirer, "Who does what on the web: A large-scale study of browsing behavior," in *Proceedings of the International AAAI Conference on Web and Social Media*, ser. ICWSM '12, 2012, pp. 130–137.

[57] J. Hu, H.-J. Zeng, H. Li, C. Niu, and Z. Chen, "Demographic prediction based on user's browsing behavior," in *Proceedings of the 16th International Conference on World Wide Web*, ser. WWW '07, 2007, pp. 151–160.

[58] P. A. Williams, J. Jenkins, J. Valacich, and M. D. Byrd, "Measuring actual behaviors in HCI research–A call to action and an example," *Transactions on Human-Computer Interaction*, vol. 9, no. 4, pp. 339–352, 2017.

[59] E. M. Redmiles, Z. Zhu, S. Kross, D. Kuchhal, T. Dumitras, and M. L. Mazurek, "Asking for a friend: Evaluating response biases in security user studies," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '18, 2018, pp. 1238–1255.

[60] S. K. Ernala, M. Burke, A. Leavitt, and N. B. Ellison, "How well do people report time spent on Facebook? An evaluation of established survey questions with recommendations," in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, ser. CHI '20, 2020, pp. 1–14.

[61] R. Wash, E. Rader, and C. Fennell, "Can people self-report security accurately? Agreement between self-report and behavioral measures," in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, ser. CHI '17, 2017, pp. 2228–2232.

[62] J. Andreoni and B. D. Bernheim, "Social image and the 50-50 norm: A theoretical and experimental analysis of audience effects," *Econometrica*, vol. 77, no. 5, pp. 1607–1636, 2009.

[63] D. Ariely, A. Bracha, and S. Meier, "Doing good or doing well? Image motivation and monetary incentives in behaving prosocially," *American Economic Review*, vol. 99, no. 1, pp. 544–555, 2009.

[64] E. Hoffman, K. McCabe, and V. Smith, "Social distance and other-regarding behavior in dictator games," *American Economic Review*, vol. 86, no. 3, pp. 653–660, 1996.

[65] E. Ostrom, J. Walker, and R. Gardner, "Covenants with and without a sword: Self-governance is possible," *The American Political Science Review*, vol. 86, no. 2, pp. 404–417, 1992.

[66] M. Rege and K. Telle, "The impact of social approval and framing on cooperation in public good situations," *Journal of Public Economics*, vol. 88, no. 7, pp. 1625–1644, 2004.

[67] M. Abramson and S. Gore, "Associative patterns of web browsing behavior.," in *AAAI Fall Symposia*, 2013.

[68] R. Cummings, S. Krehbiel, Y. Mei, R. Tuo, and W. Zhang, "Differentially private change-point detection," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, ser. NeurIPS '18, 2018, pp. 10 848–10 857.

[69] C. L. Canonne, G. Kamath, A. McMillan, A. Smith, and J. Ullman, "The structure of optimal private tests for simple hypotheses," *arXiv preprint arXiv:1811.11148*, 2018.

[70] E. Parzen, "On estimation of a probability density function and mode," *The annals of mathematical statistics*, vol. 33, no. 3, pp. 1065–1076, 1962.

[71] M. Rosenblatt, "Remarks on some nonparametric estimates of a density function," *The Annals of Mathematical Statistics*, pp. 832–837, 1956.

[72] A. Azzalini, A. W. Bowman, and W. Härdle, "On the use of nonparametric regression for model checking," *Biometrika*, vol. 76, no. 1, pp. 1–11, 1989.

[73] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics bulletin*, vol. 1, no. 6, pp. 80–83, 1945.

[74] H. B. Mann and D. R. Whitney, "On a test of whether one of two random variables is stochastically larger than the other," *The annals of mathematical statistics*, pp. 50–60, 1947.

[75] S. W. Roberts, "A comparison of some control chart procedures," *Technometrics*, vol. 8, no. 3, pp. 411–430, 1966.

[76] G. Lorden, "Procedures for reacting to a change in distribution," *The Annals of Mathematical Statistics*, vol. 42, no. 6, pp. 1897–1908, 1971.

[77] M. Pollak, "Optimal detection of a change in distribution," *The Annals of Statistics*, vol. 13, no. 1, pp. 206–227, 1985.

[78] G. V. Moustakides, "Optimal stopping times for detecting changes in distributions," *The Annals of Statistics*, vol. 14, no. 4, pp. 1379–1387, 1986.

[79] T. L. Lai, "Sequential changepoint detection in quality control and dynamical systems," *Journal of the Royal Statistical Society, Series B*, vol. 57, no. 4, pp. 613–658, 1995.

[80] ——, "Sequential analysis: some classical problems and new challenges," *Statistica Sinica*, vol. 11, no. 2, pp. 303–408, 2001.

[81] M. Kulldorff, "Prospective time periodic geographical disease surveillance using a scan statistic," *Journal of the Royal Statistical Society, Series A*, vol. 164, no. 1, pp. 61–72, 2001.

[82] Y. Mei, "Sequential change-point detection when unknown parameters are present in the pre-change distribution," *The Annals of Statistics*, vol. 34, no. 1, pp. 92–122, 2006.

[83] ——, "Efficient scalable schemes for monitoring a large number of data streams," *Biometrika*, vol. 97, no. 2, pp. 419–433, 2010.

[84] H. P. Chan, "Optimal sequential detection in multi-stream data," *The Annals of Statistics*, vol. 45, no. 6, pp. 2736–2763, 2017.

[85] E. Carlstein, "Nonparametric change-point estimation," *The Annals of Statistics*, vol. 16, no. 1, pp. 188–197, 1988.

[86] G. Bhattacharyya and R. A. Johnson, "Nonparametric tests for shift at an unknown time point," *The Annals of Mathematical Statistics*, vol. 39, no. 5, pp. 1731–1743, 1968.

[87] J. D. Gibbons and S. Chakraborti, *Nonparametric statistical inference*. Springer, 2011.

[88] A. Wald and J. Wolfowitz, "On a test whether two samples are from the same population," *The Annals of Mathematical Statistics*, vol. 11, no. 2, pp. 147–162, 1940.

[89] H. W. Lilliefors, "On the kolmogorov-smirnov test for normality with mean and variance unknown," *Journal of the American statistical Association*, vol. 62, no. 318, pp. 399–402, 1967.

[90] S. Couch, Z. Kazan, K. Shi, A. Bray, and A. Groce, "A differentially private wilcoxon signed-rank test," arXiv pre-print 1809.01635, 2018.

[91] ——, "Differentially private nonparametric hypothesis testing," arXiv pre-print 1903.09364, 2019.

[92] C. McDiarmid, "On the method of bounded differences," in *Surveys in Combinatorics*, Cambridger University Press, 1989, pp. 148–188.

[93] Y. Cao, Y. Xie, and N. Gebraeel, "Multi-sensor slope change detection," *Annals of Operations Research*, vol. 263, no. 1-2, pp. 163–189, 2018.

[94] P. Analytics, *Twitter study – august 2009*, `http://pearanalytics.com/wp-content/uploads/2012/12/Twitter-Study-August-2009.pdf`, (accessed on 2022-08-24), 2009.

[95] R. Singh, T. Ahmed, A. Kumar, A. K. Singh, A. K. Pandey, and S. K. Singh, "Imbalanced breast cancer classification using transfer learning," *IEEE/ACM transactions on computational biology and bioinformatics*, vol. 18, no. 1, pp. 83–93, 2020.

[96] X. Yuan, L. Xie, and M. Abouelenien, "A regularized ensemble framework of deep learning for cancer detection from multi-class, imbalanced training data," *Pattern Recognition*, vol. 77, pp. 160–172, 2018.

[97] S. Makki, Z. Assaghir, Y. Taher, R. Haque, M.-S. Hacid, and H. Zeineddine, "An experimental study with imbalanced classification approaches for credit card fraud detection," *IEEE Access*, vol. 7, pp. 93 010–93 022, 2019.

[98] R. A. Mohammed, K.-W. Wong, M. F. Shiratuddin, and X. Wang, "Scalable machine learning techniques for highly imbalanced credit card fraud detection: A comparative study," in *Pacific Rim International Conference on Artificial Intelligence*, Springer, 2018, pp. 237–246.

[99] E. Bagdasaryan, O. Poursaeed, and V. Shmatikov, "Differential privacy has disparate impact on model accuracy," *Advances in neural information processing systems*, vol. 32, 2019.

[100] M. Du, R. Jia, and D. Song, "Robust anomaly detection and backdoor attack detection via differential privacy," *arXiv preprint arXiv:1911.07116*, 2019.

[101] M. Jaiswal and E. M. Provost, "Privacy enhanced multimodal neural representations for emotion recognition," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, 2020, pp. 7985–7993.

[102] M. Nasr, S. Songi, A. Thakurta, N. Papemoti, and N. Carlin, "Adversary instantiation: Lower bounds for differentially private machine learning," in *2021 IEEE Symposium on Security and Privacy (SP)*, IEEE, 2021, pp. 866–882.

[103] C. Tran, M. H. Dinh, and F. Fioretto, "Differentially private deep learning under the fairness lens," *arXiv preprint arXiv:2106.02674*, 2021.

[104] D. Xu, W. Du, and X. Wu, "Removing disparate impact of differentially private stochastic gradient descent on model accuracy," *arXiv preprint arXiv:2003.03699*, 2020.

[105] D. Pujol, R. McKenna, S. Kuppam, M. Hay, A. Machanavajjhala, and G. Miklau, "Fair decision making using privacy-protected data," in *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, 2020, pp. 189–199.

[106] G. E. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *ACM SIGKDD explorations newsletter*, vol. 6, no. 1, pp. 20–29, 2004.

[107] G. Kovács, "An empirical comparison and evaluation of minority oversampling techniques on a large number of imbalanced datasets," *Applied Soft Computing*, vol. 83, p. 105 662, 2019, (IF-2019=4.873).

[108] D. Su, J. Cao, N. Li, E. Bertino, and H. Jin, "Differentially private k-means clustering," in *Proceedings of the sixth ACM conference on data and application security and privacy*, 2016, pp. 26–37.

[109] W. Qardaji, W. Yang, and N. Li, "Differentially private grids for geospatial data," in *2013 IEEE 29th international conference on data engineering (ICDE)*, IEEE, 2013, pp. 757–768.

[110] C. Sun, J. van Soest, and M. Dumontier, "Improving correlation capture in generating imbalanced data using differentially private conditional gans," *arXiv preprint arXiv:2206.13787*, 2022.

[111] R. Torkzadehmahani, P. Kairouz, and B. Paten, "Dp-cgan: Differentially private synthetic data and label generation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 0–0.

[112] D. McClure and J. P. Reiter, "Differential privacy and statistical disclosure risk measures: An investigation with binary synthetic data.," *Trans. Data Priv.*, vol. 5, no. 3, pp. 535–552, 2012.

[113] J. Zhang, G. Cormode, C. M. Procopiuc, D. Srivastava, and X. Xiao, "Privbayes: Private data release via bayesian networks," *ACM Transactions on Database Systems (TODS)*, vol. 42, no. 4, pp. 1–41, 2017.

[114] H. Li, L. Xiong, and X. Jiang, "Differentially private synthesization of multi-dimensional data using copula functions," in *Advances in database technology: proceedings. International conference on extending database technology*, NIH Public Access, vol. 2014, 2014, p. 475.

[115] C. M. Bowen and F. Liu, "Comparative study of differentially private data synthesis methods," *Statistical Science*, vol. 35, no. 2, pp. 280–307, 2020.

[116] T. Farrand, F. Mireshghallah, S. Singh, and A. Trask, "Neither private nor fair: Impact of data imbalance on utility and fairness in differential privacy," in *Proceedings of the 2020 Workshop on Privacy-Preserving Machine Learning in Practice*, 2020, pp. 15–19.

[117] N. Mohammed, R. Chen, B. C. Fung, and P. S. Yu, "Differentially private data release for data mining," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2011, pp. 493–501.

[118] H. He, Y. Bai, E. A. Garcia, and S. Li, "Adasyn: Adaptive synthetic sampling approach for imbalanced learning," in *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)*, IEEE, 2008, pp. 1322–1328.

[119] D. Dua and C. Graff, *UCI machine learning repository*, 2017.

[120] A. Estabrooks, T. Jo, and N. Japkowicz, "A multiple resampling method for learning from imbalanced data sets," *Computational intelligence*, vol. 20, no. 1, pp. 18–36, 2004.

[121] P. Jeatrakul, K. W. Wong, and C. C. Fung, "Classification of imbalanced data by combining the complementary neural network and smote algorithm," in *International Conference on Neural Information Processing*, Springer, 2010, pp. 152–159.

[122] K. Chaudhuri and S. A. Vinterbo, "A stability-based validation procedure for differentially private machine learning," *Advances in Neural Information Processing Systems*, vol. 26, 2013.

[123] F. Yu, M. Rybar, C. Uhler, and S. E. Fienberg, "Differentially-private logistic regression for detecting multiple-snp association in gwas databases," in *International Conference on Privacy in Statistical Databases*, Springer, 2014, pp. 170–184.

[124] N. Holohan, S. Braghin, P. Mac Aonghusa, and K. Levacher, "Diffprivlib: The ibm differential privacy library," *arXiv preprint arXiv:1907.02444*, 2019.

[125] T. Fawcett, "Roc graphs: Notes and practical considerations for researchers," *Machine learning*, vol. 31, no. 1, pp. 1–38, 2004.

[126] F Provost and T Fawcett, "Analysis and visualization of classifier performance: Comparison under imprecise class and cost distributions in: Proc of the 3rd international conference on knowledge discovery and data mining," 1997.

[127] A. D. Wyner, "Capabilities of bounded discrepancy decoding," *The Bell System Technical Journal*, vol. 44, no. 6, pp. 1061–1122, 1965.

[128] P. Boyvalenkov, S. Dodunekov, and O. R. Musin, "A survey on the kissing numbers," *arXiv preprint arXiv:1507.03631*, 2015.

[129] H. Han, W.-Y. Wang, and B.-H. Mao, "Borderline-smote: A new over-sampling method in imbalanced data sets learning," in *International conference on intelligent computing*, Springer, 2005, pp. 878–887.

[130] C. Bunkhumpornpat, K. Sinapiromsaran, and C. Lursinsap, "Safe-level-smote: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem," in *Pacific-Asia conference on knowledge discovery and data mining*, Springer, 2009, pp. 475–482.

[131] E. Hargittai, "Digital natives? Variation in internet skills and uses among members of the "net generation"," *Sociological Inquiry*, vol. 80, no. 1, pp. 92–113, 2010.

[132] N. B. Ellison, C. Steinfield, and C. Lampe, "The benefits of Facebook "friends:" Social capital and college students' use of online social network sites," *Journal of Computer-Mediated Communication*, vol. 12, no. 4, pp. 1143–1168, 2007.

[133] P. Calafiore and D. S. Damianov, "The effect of time spent online on student achievement in online economics and finance courses," *The Journal of Economic Education*, vol. 42, no. 3, pp. 209–223, 2011.

[134] Symantec, *Webpulse site review request*, `https://sitereview.bluecoat.com/`, Accessed: 9/30/2020.

[135] Amazon, *Alexa top sites by category*, `https://www.alexa.com/topsites/categories`, Accessed: 6/29/2018.

[136] B. Efron and R. Tibshirani, "Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy," *Statistical science*, pp. 54–75, 1986.

[137] R. M. Groves, F. J. Fowler Jr, M. P. Couper, J. M. Lepkowski, E. Singer, and R. Tourangeau, *Survey Methodology*. John Wiley & Sons, 2011, vol. 561.

[138] K. Thomas, C. Grier, J. Ma, V. Paxson, and D. Song, "Design and evaluation of a real-time URL spam filtering service," in *Proceedings of the 2011 IEEE Symposium on Security and Privacy*, ser. S&P '11, 2011, pp. 447–462.

[139] C. Cao and J. Caverlee, "Behavioral detection of spam URL sharing: Posting patterns versus click patterns," in *Proceedings of the 2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, ser. ASONAM '14, 2014, pp. 138–141.

[140] ——, "Detecting spam URLs in social media via behavioral analysis," in *Advances in Information Retrieval*, Springer International Publishing, 2015, pp. 703–714.

[141] X. Dong, J. A. Clark, and J. L. Jacob, "User behaviour based phishing websites detection," in *Proceedings of the 2008 International Multiconference on Computer Science and Information Technology*, 2008, pp. 783–790.

[142] BroadbandSearch, *Mobile vs. desktop internet usage*, `https://www.broadbandsearch.net/blog/mobile-desktop-internet-usage-statistics`, Accessed: 5/4/2021, 2021.

[143] B. E. Duffy and N. K. Chan, ""You never really know who's looking": Imagined surveillance across social media platforms," *New Media & Society*, vol. 21, no. 1, pp. 119–138, 2019.

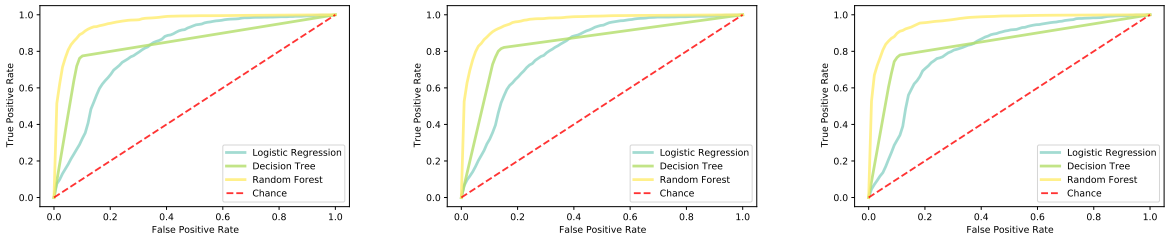
Appendix A: Additional Figures for Chapter 3

A.1 Experimental results on other datasets

A.1.1 Datasets

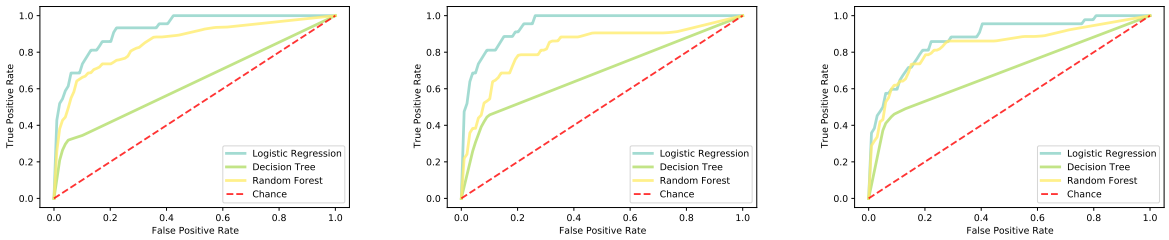
Phoneme dataset We use the Phoneme dataset [missing citation] that aims to distinguish between nasal (class 0) and oral sounds (class 1). The dataset has 5 numeric features and 5404 instances (3818 or 70.65% belonging to class 0 and 1586 or 29.35% belonging to class 1). The features were chosen to characterize each vowel. We apply standard transformations such as normalization to training samples and normalization with the same parameters to test samples.

A.1.2 Additional figures



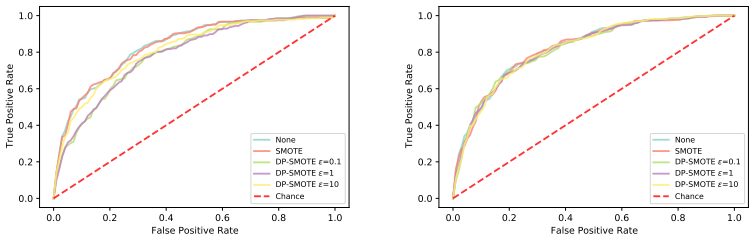
(a) No pre-processing (b) SMOTE pre-processing (c) DP-SMOTE pre-processing

Figure A.1: ROC Curves for multiple classifiers on phoneme datasets with varying preprocessing techniques.



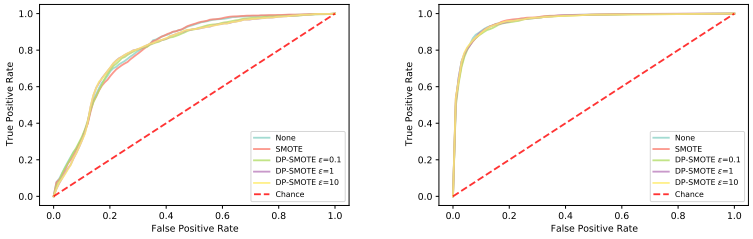
(a) No pre-processing (b) SMOTE pre-processing (c) DP-SMOTE pre-processing

Figure A.2: ROC Curves for multiple classifiers on abalone datasets with varying preprocessing techniques.



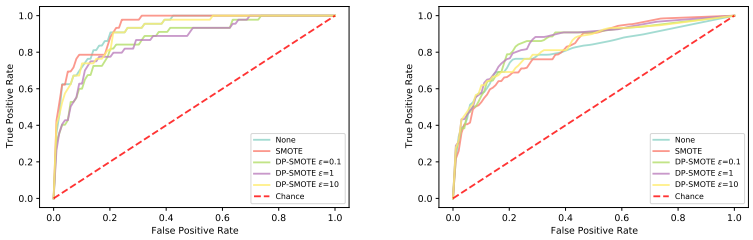
(a) Logistic Regression (b) Random Forest

Figure A.3: ROC Curves for (a) Logistic Regression, (b) Random Forest classifiers varying preprocessing techniques: None, SMOTE, and DP-SMOTE on diabetes dataset.



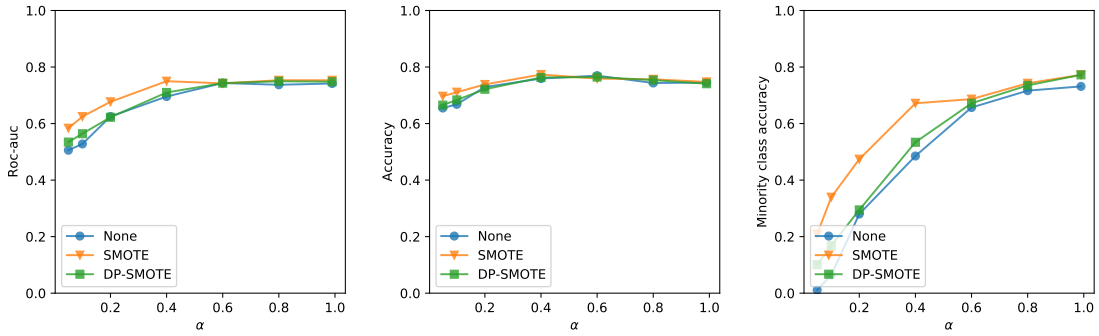
(a) Logistic Regression (b) Random Forest

Figure A.4: ROC Curves for (a) Logistic Regression, (b) Random Forest classifiers varying preprocessing techniques: None, SMOTE, and DP-SMOTE on phoneme dataset.



(a) Logistic Regression (b) Random Forest

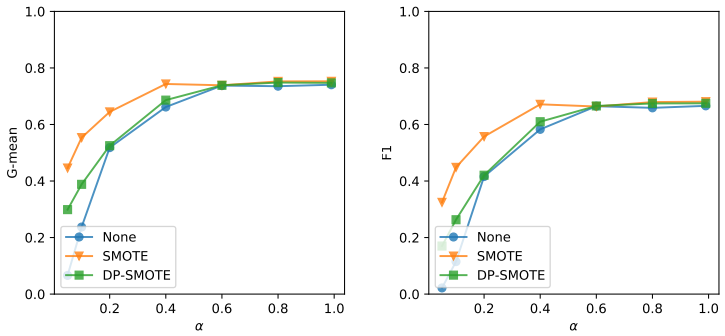
Figure A.5: ROC Curves for (a) Logistic Regression, (b) Random Forest classifiers varying pre-processing techniques: None, SMOTE, and DP-SMOTE on abalone dataset.



(a) Random Forest, ROC-AUC

(b) Random Forest, Accuracy

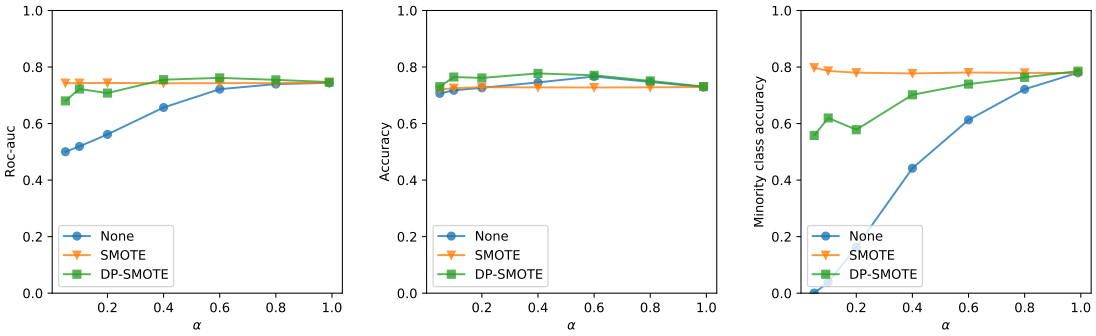
(c) Random Forest, Minority class accuracy



(d) Random Forest, G-Mean

(e) Random Forest, F1

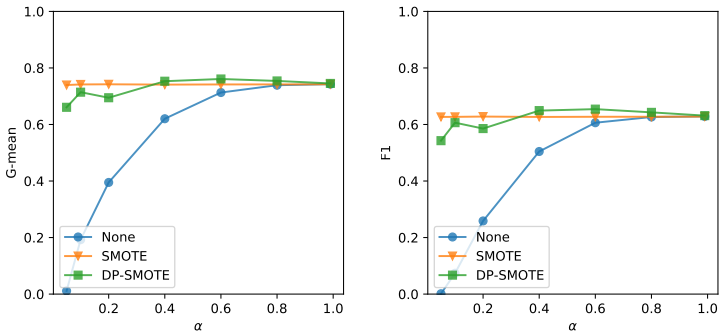
Figure A.6: Performance metrics for diabetes dataset for no pre-processing, SMOTE and DP-SMOTE under varying values of balance parameter β with random forest classifier and $\alpha = 1, \epsilon = 1$.



(a) Logistic Regression, ROC-AUC

(b) Logistic Regression, Accuracy

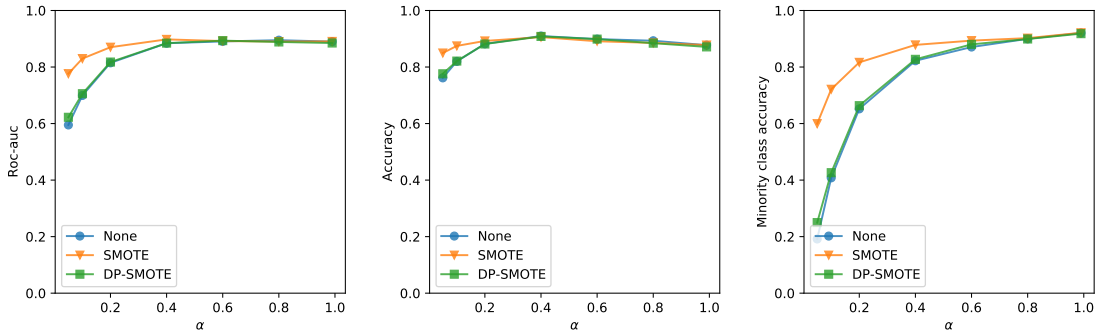
(c) Logistic Regression, Minority class accuracy



(d) Logistic Regression, G-Mean

(e) Logistic Regression, F1

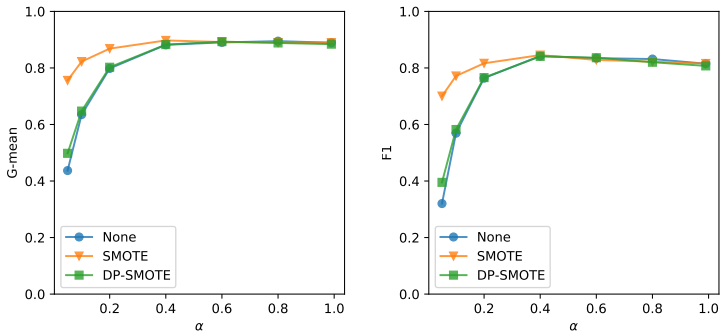
Figure A.7: Performance metrics for phoneme dataset for no pre-processing, SMOTE and DP-SMOTE under varying values of balance parameter β with logistic regression classifier and $\alpha = 1, \epsilon = 1$.



(a) Random Forest, ROC-AUC

(b) Random Forest, Accuracy

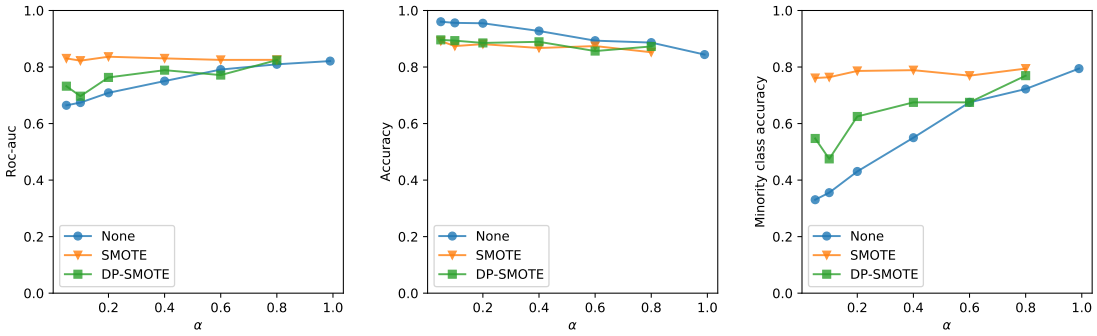
(c) Random Forest, Minority class accuracy



(d) Random Forest, G-Mean

(e) Random Forest, F1

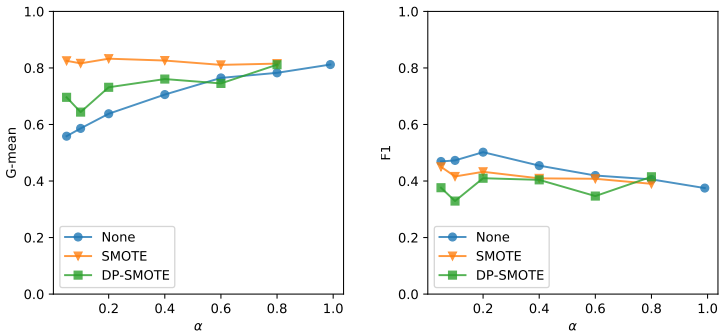
Figure A.8: Performance metrics for phoneme dataset for no pre-processing, SMOTE and DP-SMOTE under varying values of balance parameter β with random forest classifier and $\alpha = 1, \epsilon = 1$.



(a) Logistic Regression, ROC-AUC

(b) Logistic Regression, Accuracy

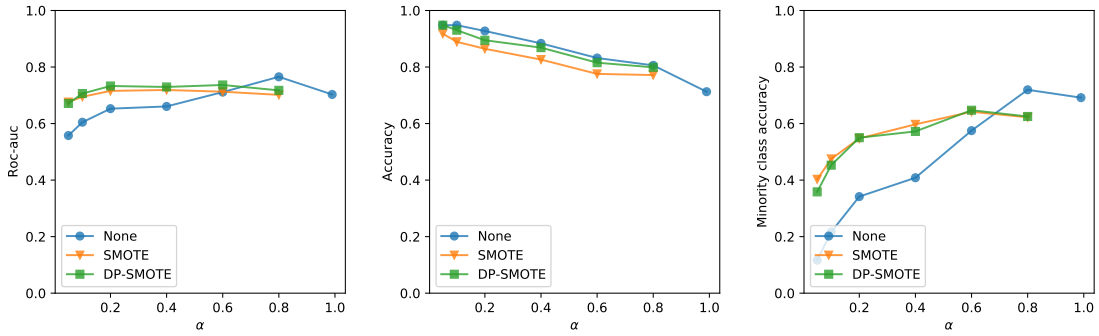
(c) Logistic Regression, Minority class accuracy



(d) Logistic Regression, G-Mean

(e) Logistic Regression, F1

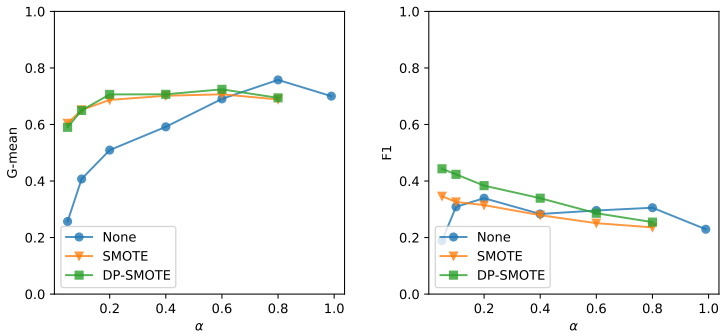
Figure A.9: Performance metrics for abalone dataset for no pre-processing, SMOTE and DP-SMOTE under varying values of balance parameter β with logistic regression classifier and $\alpha = 1, \epsilon = 1$.



(a) Random Forest, ROC-AUC

(b) Random Forest, Accuracy

(c) Random Forest, Minority class accuracy



(d) Random Forest, G-Mean

(e) Random Forest, F1

Figure A.10: Performance metrics for abalone dataset for no pre-processing, SMOTE and DP-SMOTE under varying values of balance parameter β with random forest classifier and $\alpha = 1, \epsilon = 1$.

Appendix B: Additional Results for Chapter 4

B.1 Additional figures and tables

Figure B.1 shows the distribution of browsing actions and website category for the top 100 most browsed websites (as measured by number of browsing actions) by all users during the study. Each website is color-coded to indicate the category of that website. We observe that the top nine websites have high levels of activity, and that the level of activity drops off quickly in the distribution to leave a long tail.

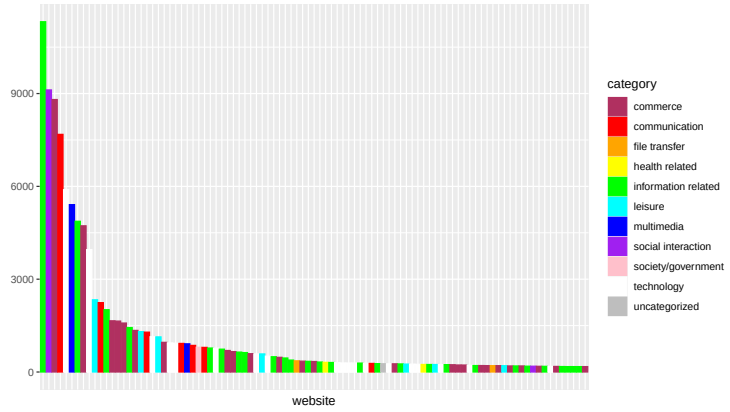


Figure B.1: Distribution of browsing actions performed on the 100 most browsed websites in the study (as measured by number of browsing actions), color-coded by category.

Table B.1 supports the analysis of RQ1 in Section 4.4.2, and presents the p-values of pairwise tests for differences in browsing behavior across website categories. Specifically, it shows the p-values for Pearson's χ^2-test for homogeneity of the distribution of browsing actions within each category, as illustrated in Figure 4.2, across all pairs of website categories.

Figure B.2 supports the analysis of RQ2 in Section 4.5.1 by providing a more detailed visualization of the δ_is in Figure 4.3a at a per-participant level. Each subfigure corresponds to a

	Technology	Information Related	Communication	Society/ Government	Social Interaction	Multimedia	Leisure	Health Related	File Transfer	Adult Related	Security Threats	Liability Concerns	Security Concerns
Commerce	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.07	0.95	0.41	1.00	$< 10^{-5}$
Technology		1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.24	1.00	1.00	1.00	$< 10^{-5}$
Information Related			1.00	1.00	1.00	0.12	1.00	1.00	1.00	1.00	1.00	1.00	$< 10^{-5}$
Communication				1.00	1.00	1.00	1.00	1.00	0.22	1.00	0.74	1.00	$< 10^{-5}$
Society/ Government					1.00	1.00	1.00	1.00	0.09	1.00	0.41	1.00	$< 10^{-5}$
Social Interaction						1.00	1.00	1.00	1.00	1.00	1.00	1.00	$< 10^{-5}$
Multimedia							1.00	1.00	**0.04**	1.00	**0.03**	1.00	$< 10^{-5}$
Leisure								1.00	1.00	1.00	1.00	1.00	$< 10^{-5}$
Health Related									1.00	1.00	1.00	1.00	$< 10^{-5}$
File Transfer										1.00	1.00	1.00	$< 10^{-5}$
Adult Related											1.00	1.00	$< 10^{-5}$
Security Threats												1.00	$< 10^{-5}$
Liability Concerns													$< 10^{-5}$

Table B.1: p-values for Pearson's χ^2-test for homogeneity based on distribution of browsing actions within websites. Tests were performed pairwise for all website categories.

single participant, with their δ_i shown for each day of the study. Recall that $\delta_i = 0$ corresponds to perfectly accurate perceptions of time spent browsing, $\delta_i < 0$ (resp. $\delta_i > 0$) corresponds to an overestimation (resp. underestimation) of browsing time. The red line in each subfigure illustrates the participant's average δ_i across all days in the study. We see that most users overestimate their time spent browsing, some by small amounts and some by large amounts.

Table B.2 supports the analysis of RQ2 in Section 4.5.2 by showing the number of participants who selected each category in the pre-study survey as one of their most browsed categories, and the number of participants who were observed to have each website category as one of their most browsed categories. Recall that in the pre-study survey, participants could select as many categories as they wished. For participant i who selected k_i categories in the pre-study survey, we included their k_i most browsed categories in the latter evaluation.

category	number of participants who chose each category	number of participants for whom each category was among their top k_i
Shopping	17	6
Reference	16	14
Social Network	23	20
Entertainment	27	21
Business	13	8
Search	27	24
News	6	1
Banking	12	2
Blogging	4	1

Table B.2: Alexa Top Websites [135] categories offered in the pre-study survey, along with number of participants who named each category as among their most frequently browsed and number of participants for whom each category was among their observed top categories of browsing during the study.

B.2 Alternative Methodologies for RQ2

In this section, we consider two alternative methodologies for measuring the difference between participants' perceived and actual time spent browsing.

Using 5 minutes of inactivity as a cutoff. We first consider using 5 minutes of inactivity as a cutoff to end an active browsing session, rather than 30 minutes as in Section 4.5.1. Intuitively, this will shorten each browsing session by 25 minutes, as participants will be considered inactive sooner after their last browsing action. This alternative methodology gives a more accurate measure of browsing activities that involve the actions listed in Table 4.1, but may be less likely to capture passive browsing experiences, such as watching a video or reading a long article.

Similar to the findings in Section 4.5.1, we find that the majority of participants (29 out of 31, 93.55%) overestimate their daily browsing time. Figure B.3 is analogous to Figure 4.3, as it visualizes the relationship between participants' actual time spent browsing and their perceived time spent browsing. Figure B.3a shows a scatter plot of the observed daily average browsing

time versus the perceived (self-reported) number of hours spent browsing per day, with one point corresponding to each participant. For ease of comparison with the results of Section 4.5.1, the orange dots correspond to analysis with 5 minutes as a cutoff time, and the blue dots correspond to analysis with 30 minutes as a cutoff time. The red line $x = y$ corresponds to perfectly accurate perceptions. Figure B.3b shows the distribution of error δ_i in hours among participants. Recall that δ_i is the difference between participant i's observed daily average browsing time (now, as measured using a 5 minute cutoff for inactivity) and the number of hours per day they reported to spend browsing in the pre-study survey. Using a t-test, we find that the mean of the δ_is among participants is significantly different from 0 ($t = -6.497$, $p < 10^{-7}$), which implies that participants still do not have accurate perceptions of their active browsing time, even under this alternative analysis method.

We additionally investigate whether the biases in participants' perceptions differ among demographic groups. We find no significant difference between δ_is for different genders ($t = -0.595$, $p = 0.505$), age groups ($t = -.724$, $p = 0.487$; $t = 0.220$, $p = 0.607$; $t = 0.394$, $p = 0.531$), and races ($t = 0.223$, $p = 0.794$; $t = -0.803$, $p = 0.537$; $t = -0.794$, $p = 0.428$). These results are presented in Table B.3.

Adjustments for desktop versus mobile browsing. Our pre-study survey asked participants about their perceived time spent browsing, without distinguishing between desktop[1] and mobile browsing, but our extension was only able to capture browsing on a desktop or laptop. Recent 2021 data [142] found that 55.9% of users' browsing time is spent on a desktop device. We account for discrepancy by scaling down each participant's self-reported time spent browsing by a factor of 0.559 and repeating the analysis of Section 4.5.1.

Even after the adjustment, most participants still overestimate the amount of time they spend online, relative to our observational measurements (80.6% of without adjustment vs. 77.4% with

[1]Desktop here refers to devices that default to desktop versions of websites, which includes desktop and laptop personal computers, but does not include phones or tablets.

adjustment). However, the mean of adjusted error δ_is is lower (-1.41 hours) than for non-adjusted values (-4.5 hours), suggesting that while participants still overestimate their time spent browsing, they overestimate by a smaller amount, relative to no adjustment. Figure B.4 is analogous to Figure 4.3, showing (a) a scatter plot of participants' observed daily average browsing time versus their adjusted perceived (self-reported) hours of daily browsing, and (b) distribution of errors δ_i.

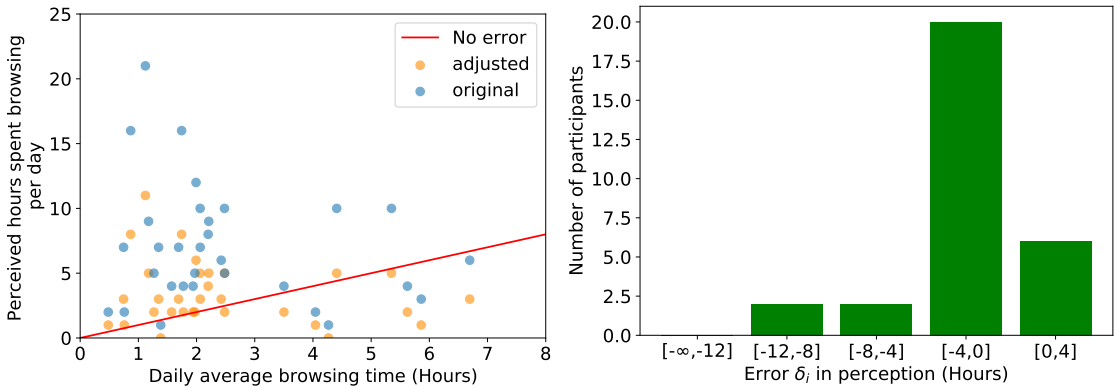
Using a t-test, we find that the mean of the adjusted δ_is is significantly different from 0 ($t = -2.348$, $p = 0.026$). When we look for differences in perception errors across demographic groups, we find no significant difference between adjusted δ_is for different genders ($t = -0.485$, $p = 0.621$), age groups ($t = -0.783$, $p = 0.446$; $t = 0.465$, $p = 0.535$; $t = 0.708$, $p = 0.472$), and races ($t = -0.125$, $p = 0.937$; $t = -0.838$, $p = 0.479$; $t = -0.561$, $p = 0.569$). A complete presentation of these results is given in Table B.4.

Feature	p-value
Gender	
Male vs Female	0.505
Race	
Asian vs Black or African American	0.794
Asian vs White	0.537
Black or African American vs White	0.428
Age	
18-24 vs 25-34	0.487
18-24 vs 35-44	0.607
25-34 vs 35-44	0.531

Table B.3: p-values for pairwise t-test for equality of means of perception errors δ_is across demographic groups using 5 minutes of inactivity as a cutoff.

Feature	p-value
Gender	
Male vs Female	0.621
Race	
Asian vs Black or African American	0.937
Asian vs white	0.479
Black or African American vs white	0.569
Age	
18-24 vs 25-34	0.446
18-24 vs 35-44	0.535
25-34 vs 35-44	0.472

Table B.4: p-values for pairwise t-test for equality of means of perception errors δ_is across demographic groups using adjusted self-reports of browsing activity.



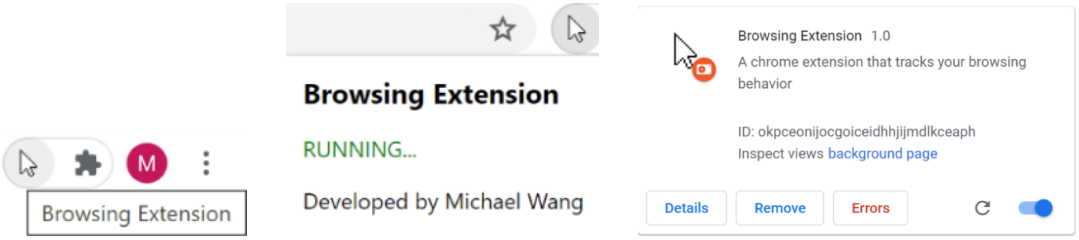
(a) Observed vs perceived browsing time

(b) Distribution of error in perceptions

Figure B.4: (a) Scatter plot illustrating observed daily average browsing time vs. adjusted perceived (self-reported) number of hours spent browsing per day. Each point corresponds to one participant. Adjusted and original (non-adjusted, as in Section 4.5.1) points are shown. Red line $x = y$ corresponds to no error in perceptions. (b) Distribution of error values δ_i in the participant population based on the adjusted perceived values. The average error δ_i is -1.41 hours (SD=3.35), with 77.4% of participants over-estimating their time spent online.

B.3 Screenshots of Study Materials

In this appendix, we show images related to participants' experience during the study. Figure B.5 shows the recruitment flyer advertising the study that was used to recruit participants. Figure B.6(a) shows the extension logo that appeared continuously in the Chrome browser to the participants during the study, Figure B.6(b) shows the extension menu that would appear if the participant clicked on the extension logo, and Figure B.6(c) shows extension information that was viewable on the Chrome Extensions page. These were all designed to look generic and to neither reveal the purpose of the study, nor to remind participants that their browsing behaviors were being collected, to better address RQ3. This drove our design of the logo as simply a mouse cursor and the extension name as simply "Browsing Extension". Since the extension was designed to collect browsing data in the background without interfering with participant browsing, this was the only visual that participants experienced during data collection.



(a) Extension logo (b) Extension menu displayed when logo is clicked (c) Extension information available in the Chrome Extensions page

Figure B.6: Screenshots of the browsing extension in the Chrome browser as seen by the participants during the study.

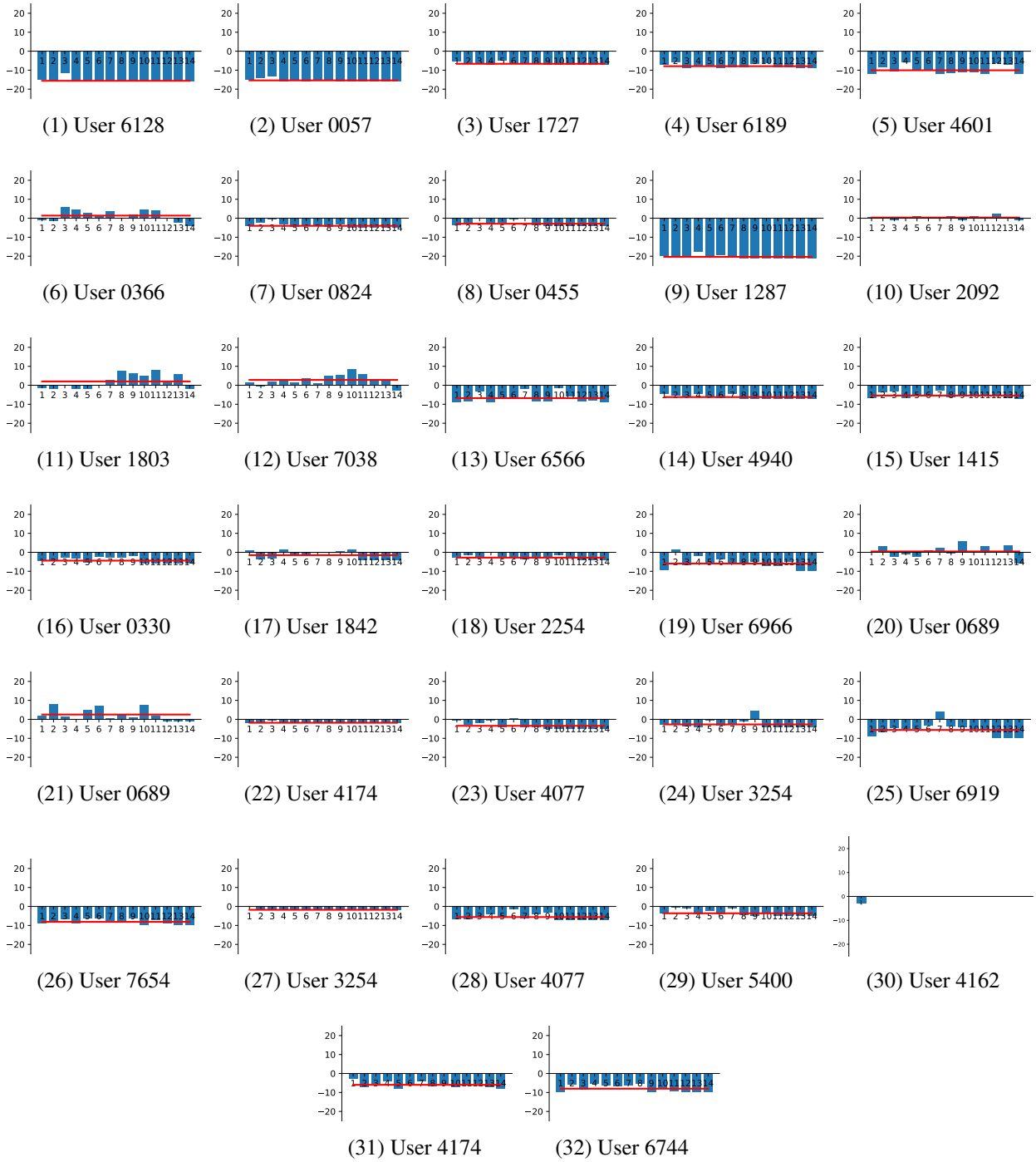
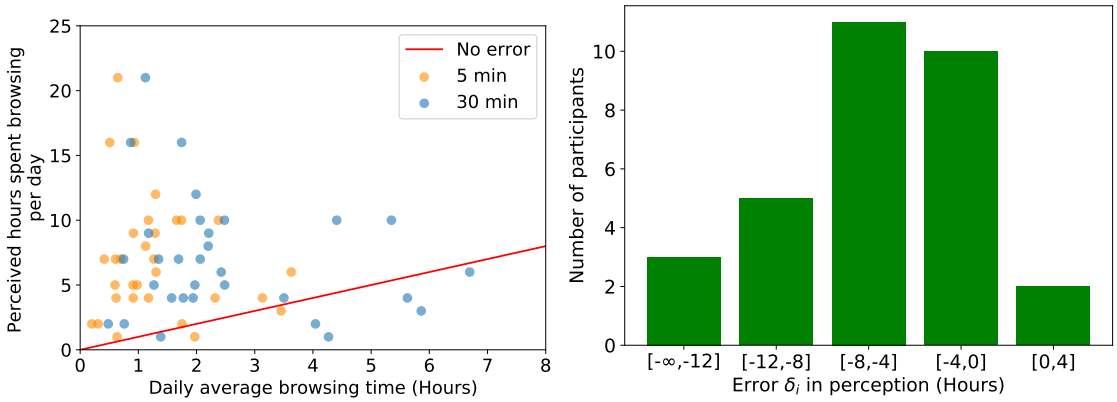


Figure B.2: Differences between the actual number of hours spent browsing per day and self-reported number of browsing hours per day, for each participant in the study. On non-active browsing days, the time spent browsing was set to zero. The x-axis enumerates the day of the experiment. The red horizontal line is a mean of these differences over the days of experiment.



(a) Observed vs perceived browsing time

(b) Distribution of error in perceptions

Figure B.3: (a) Scatter plot illustrating observed daily average browsing time vs. perceived (self-reported) number of hours spent browsing per day. Each point corresponds to one participant. Orange dots correspond to analysis with 5 minutes of inactivity as a cutoff, and the blue dots correspond to analysis with 30 minutes as a cutoff time as in Section 4.5.1. Red line $x = y$ corresponds to no error in perceptions. (b) Distribution of error values δ_i in the participant population using 5 minutes of inactivity as a cutoff.

Internet Browsing Study

Georgia Institute of Technology

Volunteers Wanted for a Research Study

Are you over 18 years old and concerned about data privacy? If so, then you may be eligible to take part in an internet browsing study!

The purpose of this study is to determine how real people interact with the internet so that we can better protect user data.

Participants will receive an incentive payment of up to $200

For information about this study, please email Dr. Rachel Cummings at rachelc@gatech.edu

If interested, please take the screening survey here:
LINK

Figure B.5: Recruitment flyer