Improved Usability of Differential Privacy in Machine Learning: Techniques for Quantifying the Privacy-Accuracy Trade-off

Von der Fakultät 5 (Informatik, Elektrotechnik und Informationstechnik) der Universität Stuttgart zur Erlangung der Würde eines Doktors der Naturwissenschaften (Dr. rer. nat.) genehmigte Abhandlung

Vorgelegt von

Daniel Bernau

aus Hildesheim

Hauptberichter:	Prof. Ralf Küsters
Mitberichter:	Prof. Florian Kerschbaum, University of Waterloo, Kanada
Mitprüfer:	Prof. Michael Sedlmair
Vorsitzender:	Prof. Marco Aiello
Tag der mündlichen Prüfung:	07.10.2022

Institut für Informationssicherheit (SEC) der Universität Stuttgart

2022

Acknowledgements

I was fortunate enough to have worked with exceptionally talented individuals over the last years. Foremost, I thank my advisor Florian Kerschbaum for his patience, rigor, and the many valuable discussions. In addition, I thank Ralf Küsters for accepting and encouraging me as an external PhD student. I am truly grateful for having had the chance to work together with both of you. Through you, I have developed a set of skills that will accompany me well throughout my further career.

I would like to thank my colleagues and fellow PhD candidates at SAP Security Research. In particular, I thank Hannah Keller, Jonas Robl, Martin Härterich, Philip-William Grassal, Steffen Schneider and Tom Ganz for their trust and collaboration over the years. Also, I would like to express my gratitude to my former SAP managers Mathias Kohler, Torsten Zube and Thomas Kaschwig. Without your backing and flexibility this endeavour could not have been concluded.

In addition, I would like to thank the European Union/European Commission for supporting my research through grant agreement 825333 under the Horizon 2020 research and innovation programme. Furthermore, I thank Pierangela Samarati and the team around her for their excellent management of the consortium.

Last but by no means least, I would like to thank my parents, my wife Julia and my son Theo for their outstanding continuous encouragement and support. I'm lost without you.

Contents

List of Figures

List of Tables

List of Algorithms

List of Abbreviations

Adv Advantage

AP Average Precision

API Application Programming Interface

BB Black-box

CDP Central Differential Privacy

CIFAR Canadian Institute for Advanced Research

CNN Convolutional Neural Network

CRISP-DM Cross Industry Standard Process for Data Mining

DI Differential Identifiability

DO Data Owner

DOM Domain

DI Differential Identifiability

DP Differential Privacy

DPSGD Differentially Private Stochastic Gradient Descent

DS Data Scientist

Eq Equation

Exp Experiment

FedAVG Federated Averaging

FL Federated Learning

FN False Negative

FP False Positive

FPR False Positive Rate

GAN Generative Adversarial Network

GPU Graphics Processing Unit

GS Global Sensitivity

HARCNN Human Activity Recognition Convolutional Neural Network

HIPAA Health Insurance Portability and Accountability Act

Hz Hertz

LDP Local Differential Privacy

LA Local Algorithm

LFW Labeled Faces in the Wild

LR Local Randomizer

LS Local Sensitivity

LSTM Long Short-Term Memory

MS MotionSense

MSE Mean Squared Error

MI Membership Inference

ML Machine Learning

MNIST Modified National Institute of Standards and Technology database

NN Neural Network

PATE Private Aggregate Teacher Ensembles

PDF Probability Density Function

PR Precision-Recall

RAM Random Access Memory

RDP Rényi Differential Privacy

SGD Stochastic Gradient Descent

SSIM Structural Similarity Index Measure

TP True Positive

TPR True Positive Rate

TN True Negative

VAE Variational Autoencoders

VGG Visual Geometry Group

WB White-box

Abstract

Differential privacy allows bounding the influence that training data records have on a neural network. To use differential privacy in machine learning with neural networks, data scientists must choose privacy parameter ϵ. Choosing meaningful privacy parameters is key since differentially private neural networks that have been trained with weak privacy parameters might result in excessive privacy leakage, while strong privacy parameters might overly degrade model utility. However, privacy parameter values are difficult to choose for two main reasons. First, the theoretical upper bound on privacy loss ϵ might be loose, depending on the chosen sensitivity and data distribution of practical datasets. Second, legal requirements and societal norms for anonymization often refer to individual identifiability, to which ϵ is only indirectly related.

Within this thesis, we address the problem of choosing ϵ from two angles. First, we quantify the empirical lower bound on the privacy loss under empirical membership inference attacks to allow data scientists to compare the empirical privacy-accuracy trade-off between local and central differential privacy. Specifically, we consider federated and non-federated discriminative models, as well as generative models. Second, we transform the privacy loss under differential privacy into an analytical bound on identifiability map legal and societal expectations w.r.t. identifiability to corresponding privacy parameters.

The thesis contributes techniques for quantifying the trade-off between accuracy and privacy over ϵ. The techniques provide information for interpreting differentially private training datasets or models trained with the differentially private stochastic gradient descent to improve usability of differential privacy in machine learning. In particular, we identify preferable ranges for privacy parameter ϵ and compare local and central differential privacy mechanisms for training differentially private neural networks under membership inference adversaries. Furthermore, we contribute an implementable instance of the differential privacy adversary that can be used to audit trained models w.r.t. identifiability.

Kurzzusammenfassung

Anonymisierung mit Differential Privacy ermöglicht es den Einfluss einzelner Trainingsdaten auf das Training eines neuronalen Netzes zu begrenzen. Um Differential Privacy beim Training neuronaler Netze einzusetzen, müssen Datenanalysten den Privatsphäreparameter ϵ setzen. Die Wahl von ϵ ist wichtig, da neuronale Netze, die mit schwachen ϵ trainiert wurden, zu schwacher Anonymisierung und einer damit einhergehenden Identifizierbarkeit führen können. Starke ϵ hingegen können den Nutzen eines neuronalen Netzes signifikant verschlechtern. Die Wahl von ϵ ist jedoch aus zwei Gründen schwierig. Erstens kann die theoretische obere Schranke für Identifizierbarkeit je nach gewählter Sensitivität und Verteilung im Datensatz weit von einer in der Praxis erreichbaren unteren Schranke entfernt sein. Zweitens beziehen sich rechtliche Anforderungen und Normen zur Anonymisierung teils auf individuelle Identifizierbarkeit, die nur indirekt mit ϵ verbunden ist.

Wir adressieren das Problem der Auswahl von ϵ aus zwei Richtungen. Zuerst quantifizieren wir die empirische untere Schranke für Identifizierbarkeit unter Membership Inference Angreifern, um einen Vergleich zwischen lokaler und zentraler Differential Privacy zu ermöglichen. Konkret betrachten wir diskriminative sowie generative neuronale Netze. Darauf folgend transformieren wir den Privatsphäreparameter in eine analytische, obere Schranke für Identifizierbarkeit, um rechtliche und gesellschaftliche Erwartungen für Identifizierbarkeit in entsprechende Privatsphäreparameter übersetzen zu können.

Somit formuliert diese Thesis Techniken zur Quantifizierung des trade-off zwischen Genauigkeit und Privatsphäre. Wir identifizieren bevorzugte Bereiche für Privatsphäreparameter für lokale und zentrale Differential Privacy Mechanismen. Außerdem formulieren wir eine implementierbare Instanz des Differential Privacy Angreifers, welche zur Auditierbarkeit von trainierten neuronalen Netzen im Bezug auf die Identifizierbarkeit genutzt werden kann.

1. Introduction

Neural networks have successfully been applied to a wide range of learning tasks such as image and text classification as well as generation [Dev+19; Goo+14; SZ15], and sequence prediction [Sil+16]. In some learning tasks, data scientists have to handle personally identifiable or confidential data, which results in two challenges. First, legal restrictions might not permit collecting, processing, or publishing original personal data, such as National Health Service data [New17]. Second, membership inference [Hay+19; NSH19; Sho+17] and model inversion attacks [FJR15; Fre+14] are capable of identifying and reconstructing training data based on information leakage from a trained, published neural network model. Potential mitigation to both challenges is offered by anonymized neural network training with differential privacy, effectively limiting the information that is revealed about every record in the training data (i.e., the privacy loss).

The application of differential privacy for neural network training has received considerable attention from the privacy research community, leading to key contributions such as the tight estimation of privacy loss under composition a composition differentially private functions [KOV17; Mir17] and differentially private stochastic gradient descent [Aba+16; BST14; SS15; SCS13] for training neural networks. Still, data scientists must choose privacy parameter ϵ to train a differentially private neural network. Setting large ϵ values will unlikely mitigate privacy attacks such as membership inference, and setting small ϵ values will reduce model accuracy. Balancing the resulting privacy-accuracy trade-off is a challenging problem, particularly for data scientists who are not experts in DP. Furthermore, privacy parameters only formulate a theoretic upper bound on the privacy loss that might not be reached when training an ML model with differentially private stochastic gradient descent on real-world data. Furthermore, a data scientist can choose between two categories of DP mechanisms: local DP [Wan+17] and central DP [Dwo06]. LDP perturbs the training data before any training takes place, whereas CDP perturbs the gradient update steps during training. The degree of perturbation, which affects the accuracy of the trained neural network on test data, is calibrated for both DP categories by adjusting their respective privacy parameter ϵ. However, data scientists might rule out LDP when designing differentially private neural networks due to concerns raised by the comparatively higher privacy parameter ϵ in LDP. Thus, appropriate values for privacy "parameter ϵ for particular contexts, research goals, or datasets is not self-evident but will be developed through trial and error" [Dwo+11]. In consequence, providing methods quantifying the trade-off between utility and identifiability over ϵ can thus provide information for interpreting differentially private datasets or functions.

This thesis addresses two problems in the context of differentially private neural networks. First, quantification of the empirical lower bound on the privacy loss under empirical membership inference attacks to allow data scientists to compare the empirical privacy-accuracy trade-off between local and central differential privacy. Secondly, the transformation of the privacy loss under differential privacy into an analytical bound on identifiability, to connect differential privacy guarantees to social norms and regulation.

1.1. Contributions

Comparison of the privacy-accuracy trade-offs in central and local differential privacy under a white-box membership inference attack. Attacks that aim to identify the training data of neural networks represent a severe threat to the privacy of individuals in the training dataset. Possible protection is offered by anonymization of the training data or training function with differential privacy. Data scientists can choose between local and central differential privacy, and need to select meaningful privacy parameters ϵ. A comparison of local and central differential privacy based on the privacy parameters furthermore potentially leads data scientists to incorrect conclusions, since the privacy parameters are reflecting different types of mechanisms. Instead, we empirically compare the relative privacy-accuracy trade-off of central and local differential privacy mechanisms under a white-box membership inference attack. While membership inference only reflects a lower bound on inference risk and differential privacy formulates an upper bound, our experiments with several datasets show that the privacy-accuracy trade-off is similar for both types of mechanisms despite the large difference in their upper bound. This suggests that the upper bound is far from the practical susceptibility to membership inference.

Thus, small ϵ in central differential privacy and large ϵ in local differential privacy result in similar membership inference risks, and local differential privacy can be a meaningful alternative to central differential privacy for differentially private deep learning besides the comparatively higher privacy parameters.

Extending the privacy-accuracy trade-off to generative networks. Generative networks for the generation of data complement feedforward neural networks for the classification of data. We use a novel membership inference attack that outperforms state-of-the-art against Variational Autoencoders to quantify the privacy-accuracy trade-off for generative models. Our work complements previous work in two aspects. First, we evaluate the strong reconstruction MI attack against Variational Autoencoders under differential privacy. Second, we address the data scientist's challenge of setting privacy parameter ϵ, which steers the differential privacy strength and thus also the privacy-accuracy trade-off. In our experimental study, we consider image and time series data, and three local and central differential privacy mechanisms. We find that the privacy-accuracy trade-offs strongly depend on the dataset and model architecture. We do rarely observe favorable privacy-accuracy trade-off for Variational Autoencoders and identify a case where LDP outperforms CDP.

Identifiability metrics for transforming privacy parameter ϵ into the posterior Bayesian belief and expected advantage of the DP adversary with tight bounds. Differential privacy allows bounding the influence that training data records have on a machine learning model. To use differential privacy in machine learning, data scientists must choose privacy parameters (ϵ, δ). Choosing meaningful privacy parameters is key since models trained with weak privacy parameters might result in excessive privacy leakage, while strong privacy parameters might overly degrade model utility. However, privacy parameter values are difficult to choose for two main reasons. First, the theoretical upper bound on privacy loss (ϵ, δ) might be loose, depending on the chosen sensitivity and data distribution of practical datasets. Second, legal requirements and societal norms for anonymization often refer to individual identifiability, to which (ϵ, δ) are only indirectly related. We transform (ϵ, δ) to a bound on the Bayesian posterior belief of the adversary assumed by differential privacy concerning the presence of any record in the training dataset. The bound holds for multidimensional queries under composition, and we show that it can be tight in practice. Furthermore, we derive an identifiability bound, which relates the adversary assumed in differential privacy to previous work on membership inference adversaries. We formulate an implementation of this differential privacy adversary that allows data scientists to audit model training and compute empirical identifiability scores and empirical (ϵ, δ).

1.2. Publications

This thesis consists of contributions from six peer-reviewed international security and data management conference publications [Ber+22; Ber+21; BRK22; BBK17; Eib+18; HHB19]. An additional contribution is available only as a preprint [Wun+21]. While the author of this thesis substantially contributed to all of these publications, some of the publication content is not part of this thesis. Please note that the publications are not sorted by publication date, but by the order in which the author of this thesis approached the individual problems. While working on the individual problems the author of this thesis proposed and supervised several Bachelor and Master Theses for which an overview is presented in Table A.1 in Appendix A.1.

Privacy-Preserving Outlier Detection for Data Streams (DBSec 2017) [BBK17]. The publication discusses the effect of outliers in the dataset on utility and privacy when using local differential privacy. The publication suggests Relaxed Sensitivity, a notion for choosing the sensitivity s.t. differential privacy is preserved for a subgroup of all possible inputs and weakened for outliers. In addition, the publication formulates a correction algorithm for outlier detection on differentially private data. The author of this thesis was joint contributor to the theory, implementation and writing regarding differentially private algorithms and outlier detection in this publication. The publication is not discussed within this thesis in detail but motivated the subsequent publications to model the privacy-accuracy trade-off and compare ϵ between local and central differential privacy [Ber+21; Eib+18].

The Influence of Differential Privacy on Short Term Electric Load Forecasting (DACH+ Energy Informatics 2018) [Eib+18]. The publication addresses the privacy-accuracy trade-off between energy consumers and energy providers in energy load forecasting. Instead of choosing ϵ directly the energy provider sets a maximum Bayesian posterior belief of the DP adversary which is then transformed to ϵ. While the maximum Bayesian posterior belief holds for all energy consumers, we show that many energy consumers actually enjoy a far smaller factual Bayesian posterior belief since their consumption is much smaller than the technical maximum against which differential privacy protects. The author of this thesis contributed was main contributor to the theory and writing on differential identifiability and interpretation of ϵ in electric load forecasting in the publication. The author was joint contributor to the implementation of the differential privacy mechanism and the differential identifiability calculation under k-fold adaptive composition for achieving a tighter lower bound of the privacy guarantee in electric load forecasting. The author of this thesis supervised one co-author of the publication. This publication is not included in this thesis but motivated a later publication on quantifying identifiability to choose and audit ϵ in differentially private deep learning [Ber+22].

Comparing Local and Central Differential Privacy Using Membership Inference Attacks (DBSec 2021) [Ber+21]. The publication suggests comparing local and central differential privacy under black- and white-box membership inference. Under this measure, CDP mechanisms are not achieving a consistently better privacy-accuracy trade-off on various datasets and reference models. The trade-off, however, depends on the specific dataset and for each dataset, there are ranges where the relative trade-off is greater for protection against MI than accuracy. The author of this thesis designed the study and was main contributor to theory and writing. The author of this thesis was joint contributor to the implementation and supervised several co-authors. The publication is contained in an extended version that also considers federated learning in Chapter 5 within this thesis.

Monte Carlo and Reconstruction Membership Inference Attacks against Generative Models (PETS 2019) [HHB19]. The publication introduces several membership inference attacks against generative machine learning models to allow modeling the privacy-accuracy trade-off in generative machine learning. However, the work does not consider anonymization during differential privacy but solely regularization techniques against overfitting during model training. The author of this thesis was joint contributor to the theory and writing. The author of this thesis co-supervised a co-author of this publication. Chapter 6 in this thesis extends the publication by quantifying the privacy-accuracy trade-off for Variational Autoencoders with local and central differential privacy under the suggested reconstruction attack [BRK22].

Assessing Differentially Private Variational Autoencoders under Membership Inference (DBSec 2022) [BRK22]. This work quantifies the privacy-accuracy trade-off under the reconstruction MI attack against differentially private Variational Autoencoders. The publication extends the comparison of local and central differential privacy w.r.t. the privacy-accuracy trade-off in feedforward neural networks [Ber+21]. The CDP mechanism offered a more consistent decrease in MI attack performance whereas the LDP mechanisms showed varying levels of protection against MI depending on chosen privacy parameter and setting. The relative privacy-accuracy trade-off highlights that protection against MI often comes at a disproportionately high accuracy drop. The author of this thesis motivated and designed the study. The publication is a joint effort to which both the thesis author and the supervised co-author contributed equally to the theory, implementation and writing. This publication is featured in Chapter 6 within this thesis.

Quantifying Identifiability to Choose and Audit ϵ in Differentially Private Deep Learning (VLDB 2022) [Ber+22]. This publication contributes two identifiability scores that can also be transformed into privacy parameter ϵ in central differential privacy. Furthermore, an implementation for the differential privacy adversary in machine learning is suggested. The implementation is used to audit trained machine learning models w.r.t. the factual privacy loss. By this, it becomes possible to check whether the specified DP bound is reached. We furthermore suggest a heuristic that leads to tight bounds by using local sensitivity. The author of this thesis motivated the study and suggested identifiability-based parameter selection with dataset sensitivity and differential identifiability. The author of this thesis was main contributor to the theory and writing. The author of this thesis was joint contributor to the implementation and supervised two co-authors. This publication is the foundation for Chapter 7 in this thesis.

On the Privacy-Accuracy Trade-Off in Differentially Private Hierarchical Text Classification [Wun+21]. This work uses the methods for quantifying the privacy-accuracy trade-off from this thesis to compare several differentially private hierarchical text classification model architectures to select the model architecture with the best privacy-accuracy trade-off. Furthermore, potential improvements to white-box membership inference attacks against hierarchical classification models are discussed. The author of this thesis was main contributor to the theory. The author of this thesis furthermore was joint contributor to the writing and supervised a co-author. The implementation was performed by the supervised co-author. The work is not featured in this thesis, but complementing this thesis by demonstrating the practical benefits of the thesis contributions.

1.3. Thesis Structure

This thesis is structured as follows. We provide preliminaries for differential privacy and membership inference in Chapter 2. Related work w.r.t. the scope of this thesis is presented in Chapter 3. We formalize the problem of choosing and interpreting privacy parameter ϵ to improve usability of differentially private deep learning in Chapter 4. In the three subsequent chapters, we present approaches for addressing the problem of selecting and interpreting privacy parameters in differentially private deep learning. To this end Chapters 5 and 6 compare the achievable privacy-accuracy trade-off between local and central differential privacy in feedforward and generative neural networks under membership inference attacks. Chapter 7 formulates an implementable instance of the DP adversary for differentially private deep learning to choose and audit privacy parameters based on achievable (i.e., tight) identifiability bounds. We conclude the thesis with a summary and outlook on future work in Chapter 8. Selected additional material is provided in Appendix A.

2. Preliminaries

In the following, we provide definitions and formalization for differential privacy (Sections 2.1 and 2.2), neural networks (Section 2.3), and membership inference attacks (Section 2.4). We furthermore provide definitions for performance metrics that are used within this thesis (Section 2.5).

2.1. Central Differential Privacy

Central differential privacy (CDP), introduced by Dwork [Dwo06], is a mathematical definition for the anonymization of data. In contrast to previous anonymization methods based on generalization (e.g., k-anonymity [SS98]) CDP adds noise to the result of a query function $f(\cdot)$ over a data set $\mathcal{D} = d_1, \ldots, d_n$. Assuming that every participant is represented by one element, privacy is provided to participants in the data set $\mathcal{D}$ as their impact of presence (absence) on the query function $f(\cdot)$ becomes bounded. To add differentially private noise to the result of some arbitrary query $f(\cdot)$, mechanisms $\mathcal{M}$ fulfilling Definition 2.1 are used.

Definition 2.1 ((ϵ, δ)-Central Differential Privacy [Dwo+06]). A mechanism $\mathcal{M}$ preserves (ϵ, δ)-differential privacy if for all independently sampled $\mathcal{D}, \mathcal{D}' \subseteq \mathcal{U}$, where $\mathcal{U}$ is a finite set, with $\mathcal{D}$ and $\mathcal{D}'$ differing in at most one element, and all possible mechanism outputs $\mathcal{S}$

$$\Pr(\mathcal{M}(\mathcal{D}) \in \mathcal{R}) \leq e^{\epsilon} \cdot \Pr(\mathcal{M}(\mathcal{D}') \in \mathcal{R}) + \delta \tag{2.1}$$

◇

If the evaluation of a query function $f : \mathcal{U} \rightarrow \mathcal{R}$ on a dataset $\mathcal{D}$ from domain $\mathcal{U}$ yields a result $r \in \mathcal{R}$, r inevitably leaks information about the entries $d \in \mathcal{D}$ (cf. impossibility of Dalenius' desideratum [Dwo06]). In consequence, r could have been produced from dataset $\mathcal{D}$ or some *neighboring* dataset $\mathcal{D}'$. A neighboring dataset $\mathcal{D}'$ either differs from $\mathcal{D}$ in the presence of one additional datapoint (unbounded CDP) or the value of one datapoint when a datapoint from $\mathcal{D}$ is replaced by another datapoint (bounded CDP). In the context of this thesis, we consider w.l.o.g. unbounded CDP where $\mathcal{D}$ contains one datapoint d more than $\mathcal{D}'$ and $\mathcal{D} \setminus \mathcal{D}' = d$. The impact of a single member $d \in \mathcal{D}$ on $f(\cdot)$ is bounded. If this impact is low compared to the noise specified by CDP, plausible deniability is provided to this member of $\mathcal{D}$, even if $\mathcal{D}$ and the members' properties d (and thus also $\mathcal{D}'$) are known. For example, a single individual participating in a private analysis based on a census income dataset such as Adult [Koh96] could therefore plausibly deny census participation and values of personal attributes. CDP provides a strong guarantee since it protects against a strong adversary with knowledge of up to all points in a dataset except one. As Definition 2.1 is an inequality, the privacy parameter ϵ can be interpreted as an upper bound on privacy loss.

Definition 2.2 (Gaussian Mechanism [DR14]). Let $\epsilon \in (0, 1)$ be arbitrary. For $c^2 > 2\ln\left(\frac{1.25}{\delta}\right)$, the Gaussian mechanism with parameter $\sigma \geq c\frac{\Delta f}{\epsilon}$ gives (ϵ, δ)-CDP, adding noise scaled to $\mathcal{N}(0, \sigma^2)$. $\diamond$

The Gaussian mechanism of Definition 2.2 is a CDP mechanism for perturbing the outcome of stochastic gradient descent in machine learning and adds noise independently sampled from a Gaussian distribution centered at zero. Prior work [DR14] has analyzed the tails of the normal distributions and found that bounding the standard deviation as follows fulfills (ϵ, δ)-CDP:

$$\sigma > \Delta f \sqrt{2\ln(1.25/\delta)}/\epsilon \tag{2.2}$$

Rearranged to solve for ϵ, this is:

$$\epsilon > \Delta f \sqrt{2\ln(1.25/\delta)}/\sigma \tag{2.3}$$

σ depends not only on the privacy parameter ϵ, but also on a scaling factor Δf. Δf is commonly referred to as the sensitivity of a query function $f(\cdot)$. DP holds if mechanisms are scaled to GS_f of Definition 2.3, i.e., the maximum contribution of a record in the dataset to the outcome of $f(\cdot)$.

Definition 2.3 (Global Sensitivity). Let $\mathcal{D}$ and $\mathcal{D}'$ be neighboring. For a given finite set $\mathcal{U}$ and function f the global sensitivity GS_f with respect to a distance function is

$$GS_f = \max_{\mathcal{D},\mathcal{D}'} ||f(\mathcal{D}) - f(\mathcal{D}')||$$

◇

For the Gaussian mechanism, we use Definition 2.3 with the global ℓ_2-sensitivity GS_{f_2}. Definition 2.4 [NRS07] introduces local sensitivity LS_f, a notion related to global sensitivity, which fixes dataset $\mathcal{D}$.

Definition 2.4 (Local Sensitivity). Let $\mathcal{D}$ and $\mathcal{D}'$ be neighboring. For a given finite set $\mathcal{U}$, independently sampled dataset $\mathcal{D} \subseteq \mathcal{U}$, and function f, the local sensitivity $LS_f(\mathcal{D})$ concerning a distance function is

$$LS_f(\mathcal{D}) = \max_{\mathcal{D}'} ||f(\mathcal{D}) - f(\mathcal{D}')||$$

◇

Note that GS_f can also be defined relative to local sensitivity by $GS_f = \max_{\mathcal{D}} LS_f(\mathcal{D})$. Compared to using GS_f, less noise may be added when ϵ is held constant and LS_f is used and ϵ may decrease when the noise distribution is held constant.

The most basic form of accounting for multiple data releases is sequential composition, which states that for a sequence of k mechanism executions each providing (ϵ_i, δ_i)-DP, the total privacy guarantee is ($\sum_i \epsilon_i$, $\sum_i \delta_i$)-DP; however, sequential composition adds more noise than necessary [Aba+16; Mir17]. A tighter analysis of composition is provided by Mironov [Mir17]. (α, ϵ_{RDP})-RDP, with $\alpha > 1$ quantifies the difference in distributions $\mathcal{M}(\mathcal{D}), \mathcal{M}(\mathcal{D}')$ by their Rényi divergence [EH10]. For a sequence of k mechanism executions each providing $(\alpha, \epsilon_{RDP,i})$-RDP, the privacy guarantee is $(\alpha, \sum_i \epsilon_{RDP,i})$-RDP. The (α, ϵ_{RDP})-RDP guarantee converts to $\left(\epsilon_{RDP} - \frac{\ln \delta}{\alpha - 1}, \delta\right)$-DP. The Gaussian mechanism provides RDP by:

$$\epsilon_{RDP} = \alpha \cdot \Delta f^2 / 2\sigma^2 \tag{2.4}$$

2.2. Local Differential Privacy

We refer to the perturbation of entries $d \in \mathcal{D}$ as local differential privacy (LDP) [Wan+17]. We adapt the definitions of Kasiviswanathan et al. [Kas+08] to achieve local differential privacy by using local randomizers $\mathcal{LR}$. In LDP experiments within this thesis, we use a local randomizer to perturb each record $d \in \mathcal{D}$ independently. Since a record may contain multiple correlated features (e.g., items in a preference vector) a local randomizer must be applied sequentially which results in a linearly increasing privacy loss. A series of local randomizer executions per record composes a local algorithm according to Definition 2.6. ϵ-local algorithms are ϵ-local differentially private [Kas+08], where ϵ is a summation of all composed local randomizer guarantees. We perturb low domain data (e.g., binary data) with randomized response [War65], a (composed) local randomizer. Randomized response yields $\epsilon = \ln\left(\frac{\rho}{1-\rho}\right)$ LDP for a one-time collection of values from binary domains (e.g., $\{\texttt{yes}, \texttt{no}\}$) with two fair coins [EPK14]. That is, retention of the original value with probability $\rho = 0.5$ and uniform sampling with probability $(1 - \rho)$.

Definition 2.5 (Local Differential Privacy). A local randomizer (mechanism) $\mathcal{LR} : \mathcal{DOM} \to \mathcal{S}$ is ϵ-local differentially private, if $\epsilon \geq 0$ and for all possible inputs $v, v' \in \mathcal{DOM}$ and all possible outcomes $s \in \mathcal{S}$ of $\mathcal{LR}$

$$\Pr[\mathcal{LR}(v) = s] \leq e^{\epsilon} \cdot \Pr[\mathcal{LR}\,(v') = s] \tag{2.5}$$

◇

Definition 2.6 (Local Algorithm). An algorithm is ϵ-local if it accesses the database $\mathcal{D}$ via $\mathcal{LR}$ with the following restriction: for all $i \in \{1, \ldots, |\mathcal{D}|\}$, if $\mathcal{LR}_1(i), \ldots, \mathcal{LR}_k(i)$ are the algorithms invocations of $\mathcal{LR}$ on index i, where each $\mathcal{LR}_j$ is an ϵ_j-local randomizer, then $\epsilon_1 + \ldots + \epsilon_k \leq \epsilon$. ◇

Definition 2.7 (Laplace Mechanism [DR14]). Given a numerical query function $f : DOM \to \mathbb{R}^k$, the Laplace mechanism with $\lambda = \frac{\Delta_f}{\epsilon}$ is an ϵ-differentially private mechanism, adding noise scaled to $Lap(\lambda, \mu = 0)$. ◇

In this thesis, we also evaluate image data for which we rely on the local randomizer by Fan [Fan18] for LDP image pixelization. The randomizer applies the Laplace mechanism of Definition 2.7 with scale $\lambda = \frac{255 \cdot o}{b^2 \cdot \epsilon}$ to each pixel, thus fulfilling Definition 2.5. Parameter o represents the neighborhood in which LDP is provided. Full neighborhood for an image dataset would require that any picture can become any other picture. In general, providing DP or LDP within a large neighborhood will require high ϵ values to retain meaningful image structure. High privacy will result in random black and white images.

We furthermore use a domain-independent LDP mechanism specifically for Variational Autoencoders which we refer to as VAE-LDP. VAE-LDP by Weggenmann et al. [Weg+22] allows a data scientist to use Variational Autoencoders (cf. Section 2.3.1) as an LDP mechanism to perturb data. This is achieved by limiting the encoder's mean and adding noise to the encoder's standard deviation before sampling the latent z during training. After training, the resulting Variational Autoencoder is used to perturb records with $\epsilon = \frac{\Delta f \sqrt{2 \log(1.25/\delta)}}{\sigma}$. We limit the resulting mean of the encoder to $[-3, 3]$ by using the Hyperbolic tangent activation function. Furthermore, we introduce noise according to noise bound σ by enforcing a lower bound v on the standard deviation of the encoder, i.e., setting the standard deviation to $\max(\sigma, v)$.

2.3. Machine Learning with Neural Networks

The term Machine Learning (ML) is referring to the concept of "giving computers the ability to learn without being explicitly programmed" [Sam59] by using "algorithms that improve their performance at some task through experience" [Mit97, p. 2]. To this end, an ML algorithm creates a model h that encodes rules inferred from training data for later use on test data. Within this thesis, we consider two general types of models for which we restate and adapt the descriptions provided by Ng and Jordan [NJ01]:

- discriminative models for classification of records X to categories Y after learning the conditional probability $P(Y|X = x)$,
- generative models for generating records X for categories Y after learning the distribution $P(X|Y = y)$.

We use feedforward neural networks (NN) to train discriminative and generative models. Table 2.1 states the set of notations for machine learning with neural networks that

Table 2.1.: Notations and context

Symbol	Description
X	Set of vectors $\mathbf{x_1}, \dots, \mathbf{x_j}$ where $x_j^1, \dots, x_j^i$ denote attribute values (*features*) of $\mathbf{x_j}$.
Y	Set of target variables in $\mathbf{y}$ (*labels*).
m	$\lvert Y \rvert$.
$\mathbf{y}$	Vector of target variables (*labels*) where variable $y_j \in \mathbf{y}$ represents the label for $\mathbf{x_j} \in X$.
$h(\cdot)$	Model function (e.g., classifier).
$\hat{y}$	Predicted target variable, i.e., $\hat{y} = h(\mathbf{d})$.
$p(\mathbf{x})$	Softmax confidence vector for $\mathbf{x}$.
$\mathcal{D}$	Join of X and Y s.t. $\mathcal{D}_{\vert} := (\mathbf{x_j}, y_j)$.
d	A record $d \in \mathcal{D}$, where $d := (\mathbf{x}, y)$.
n	$\lvert \mathcal{D}^{train} \rvert$.
$L(h(x;\theta), y)$	Losses of the model function with learned weights θ on record d with true label y.
$\frac{\delta L}{\delta \theta}$	Gradients of the losses w.r.t. the weights.

is used within this thesis. Neural networks represent a category of algorithms that were originally introduced to formalize and mimic human networks of neurons in the brain [Heb49; MP43; WH60]. Feedforward neural networks consist of neurons (nodes), layers (sets of neurons), and weights θ (edges between neurons of consecutive layers). The term feedforward is due to the characteristics that input is being passed through the model to produce output without using any feedback loops from outputs of the model as inputs [GBC16, p. 164]. We refer to the first and the final layer of a neural network as the input and output layer. Any layer between the input and output layer is referred to as a hidden layer. For each neuron in a hidden layer, an activation function receives the sum of all input weights to compute a scalar output weight for each connection to the next layer. It has been theoretically demonstrated that neural networks can approximate any continuous function on any subset of $\mathbb{R}^n$ arbitrarily close already with one hidden layer and a finite number of neurons for nonlinear activation functions (cf. Universal Approximation Theorem [Cyb89; Hor91]). However, in practice neurons are commonly distributed over several layers, and for several image datasets it has been demonstrated that rearranging a given number of neurons into networks with more layers and fewer neurons per layer, and vice versa, can for example achieve similar overall model accuracy while accuracies per category differ [NRK21].

Within this thesis, we train neural networks with gradient descent optimizers such as Stochastic Gradient Descent (SGD) [KW52] and Adam [KB15], which are dominantly used in practice [GBC16, pp. 151, 294]. Such optimizers identify weights that minimize the error of predictions $\hat{y} = h(x; \theta)$ on a training dataset by calculating the vector of partial derivatives of the loss L w.r.t. weights θ. We quantify the loss L by using Cross-Entropy as loss function for classification and the evidence lower bound (ELBO; also variational lower bound) [KW14] for generative models in this thesis. Weight updates are multiplied with learning rate $\eta > 0$ to allow for fine-tuning convergence towards a minimum. For training we split a training dataset $\mathcal{D}^{train} \subseteq \mathcal{D}$ into batches, where each of the datapoints $(x, y) \in \mathcal{D}$ consists of the i features $x^1, \ldots, x^i$ and the label y. We optimize the neural network for each batch (i.e., batch gradient descent). We refer to one pass over the training dataset as one epoch. Within this thesis NNs are commonly trained for multiple epochs until the loss on the training dataset is no longer decreasing (i.e., early stopping [GBC16, p. 241]). A test dataset is used to evaluate the generalization of the trained model.

We use a variety of layer types for neural networks that we will briefly restate for convenience in the following:

- Dense layers: A layer in which all neurons are connected to all neurons of the preceding layer (also called *fully connected layer*).

- Convolutional layer: A layer that convolves the preceding layer output with a convolution kernel and passes the convolution result to the next layer. Convolutional layers have been introduced in the context of image processing to condense input information for better detection of image features [LB98].

- Max Pooling layer: Max Pooling layers receive the output of a convolutional layer as input and use a maximum pooling operation that reduces the input dimensions by a factor of 2 [Ran+07].

- Dropout layer: A layer that randomly that sets inputs to 0 at each batch gradient descent weights update during training according to a dropout probability. The sum of the layer inputs remains unchanged by scaling non-zero inputs by $\frac{1}{1-\text{dropout rate}}$ [Sri+14]. Used to avoid overfitting and foster generalization.

We provide an illustration of an exemplary neural network for image classification with convolutional and dense layers in Figure 2.1. Goodfellow notes that “in modern neural networks, the default recommendation is to use the rectified linear unit or ReLU” [GBC16, p. 174] activation function for training. We follow this advice throughout this thesis when defining neural network architectures, but deviate in cases where we reuse or build upon a previously introduced, established neural network architecture with preset activation functions. For discriminative classification model output layers, we commonly use softmax [GBC16, p. 180] as an activation function which normalizes the outputs of the preceding layer to a probability distribution over the categories Y [GBC16, p. 184].

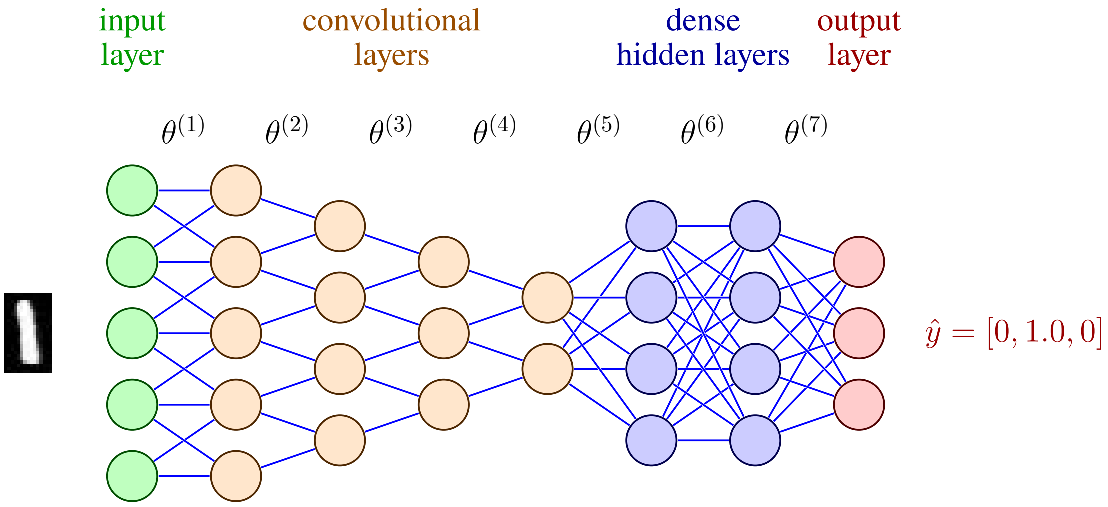


Figure 2.1.: Exemplary illustration of a feedforward neural network with input layer for attributes $x^1, \ldots, x^5$, convolutional layer, dense layers, output layer with categories y_1, y_2, y_3 and softmax vector $\hat{y}$, and weights θ

2.3.1. Variational Autoencoders

Generative models are trained to learn the joint probability distribution $P(X, Y)$ of features X and labels Y of a training dataset $\mathcal{D}^{train}$. We focus on Variational Autoencoders (VAE) [KW14] as a generative model. VAE consist of two neural networks:

encoder E and decoder D. During training a record d is given to the encoder which outputs the mean $E_\mu(x)$ and variance $E_\sigma(x)$ of a Gaussian distribution. A latent variable z is then sampled from the Gaussian distribution $N(E_\mu(x), E_\sigma(x))$ and fed into the decoder D. After successful training the reconstruction $D(z)$ should be close to x. We provide an illustration of a VAE in Figure 2.2. During the training a weighted sum of two terms is minimized. First, the *reconstruction error* $\|D(z) - x\|$. Second, the *Kullback-Leibler divergence KL*$(N(E_\mu(x), E_\sigma(x))||N(0,1))$ between the distribution of latent variables z and the unit Gaussian. The KL divergence term prevents the network from only memorizing certain latent variables since the distribution should be similar to the unit Gaussian. Kingma et al. [KW14] motivate the training objective as a lower bound on the log-likelihood and suggest training E and D for a training objective by using the *reparameterization trick*. Samples $D(z)$ are generated from the VAE by sampling a latent variable $z = \epsilon_V \sigma_x + \mu_x$, where $\epsilon_V \sim N(0, I)$, and passing z through D. Similar to GAN conditional VAE generate samples for a specific label by utilizing a condition c as input to E and D.

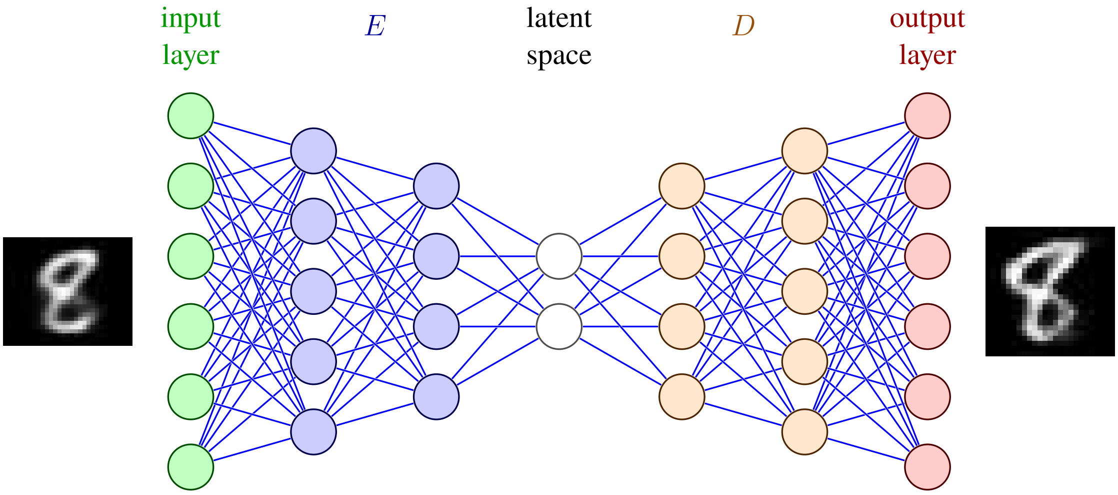


Figure 2.2.: Exemplary illustration of a VAE with input layer for attributes $\hat{x}^1, \ldots, \hat{x}^6$, encoder E for inference, latent space with parameters (μ, σ), decoder D for generation, and output layer with generated data $D(z) = \hat{x}^1, \ldots, \hat{x}^6$

2.3.2. Differentially Private Stochastic Gradient Descent

Within this thesis, we use the differentially private stochastic gradient descent to train neural networks with CDP [Aba+16]. In particular, we use differentially private versions of the plain SGD and Adam optimizer to which we refer as DPSGD and DP-Adam[1]. DPSGD and DP-Adam represent a differentially private neural network training mechanism $\mathcal{M}_{nn}$ that updates the weights θ_i per training step $i \in \{1, \ldots, k\}$ with $\theta_i \leftarrow \theta_{i-1} - \eta \cdot \tilde{g}_i$, where $\tilde{g} = \mathcal{M}_{nn}\left(\partial L/\partial\theta_{i-1}\right)$ denotes a Gaussian perturbed gradient and η is some scaling function on $\tilde{g}$ to compute an update, i.e., learning rate or running moment estimations. After k update steps, $\mathcal{M}_{nn}$ outputs a differentially private weight matrix θ which is used by the prediction function $h(\cdot)$ of a neural network. To limit the sensitivity Δf, the length of each per-example gradient is limited to the clipping norm $\texttt{C}$ before perturbation, and the Gaussian perturbation is proportional to Δf (cf. Equation (2.2)). A data scientist has to specify privacy parameter ϵ and clipping norm $\texttt{C}$ for the training independent of the optimizer. For a dataset $\mathcal{D}$, the unperturbed gradient vector g is analogue to the output of function f in DP and differential privacy is achieved by perturbing the gradient g for $\mathcal{D}$ at any epoch i, i.e., $\tilde{g}_i = \mathcal{M}_i(\mathcal{D})$.

2.4. Membership Inference

Membership inference (MI) is a threat model for quantifying how accurately an honest-but-curious membership inference adversary $\mathcal{A}_{\texttt{MI}}$ can identify members of the training dataset in machine learning. MI attacks are of particular interest for members of the training dataset when the nature of the training dataset is revealing sensitive information. For example, a medical training dataset containing patients with different types of cancer, or a training dataset that is used to predict the week of pregnancy based on the shopping cart [Hil12]. A related attack building upon MI is attribute inference [Yeo+18] where records in the training dataset are partially known and specific attribute values shall be inferred. In this thesis, we solely consider MI since protection against MI offers protection against attribute inference. Yeom et al. [Yeo+18] formalize MI in Experiment 2.1 on which we will build upon in parts of this thesis.

[1] We use Tensorflow Privacy optimizers throughout this thesis: `https://github.com/tensorflow/privacy`

Experiment 2.1. *(Membership Inference* $\mathrm{Exp}^{\mathrm{MI}}$*) Let* $\mathcal{A}_{\mathtt{MI}}$ *be an MI adversary,* $\mathcal{M}$ *be a differentially private learning algorithm,* n *be a positive integer, and* $\mathtt{Dist}$ *be a distribution over datapoints* (x, y)*. Sample* $\mathcal{D} \sim \mathtt{Dist}^n$ *and let* $\mathbf{r} = \mathcal{M}(\mathcal{D})$*. The membership experiment proceeds as follows:*

1. *Sample* $z_{\mathcal{D}}$ *uniformly from* $\mathcal{D}$ *and* $z_{\mathtt{Dist}}$ *from* $\mathtt{Dist}$
2. *Choose* $b \leftarrow \{0, 1\}$ *uniformly at random*
3. *Let*
$$z = \begin{cases} z_{\mathcal{D}} & \text{if } b\text{=}1 \\ z_{\mathtt{Dist}} & \text{if } b\text{=}0 \end{cases}$$
4. $\mathcal{A}_{\mathtt{MI}}$ *outputs* $b' = \mathcal{A}_{\mathtt{MI}}(\mathbf{r}, z, \mathtt{Dist}, n, \mathcal{M}) \in \{0, 1\}$*. If* $b' = b$*,* $\mathcal{A}_{\mathtt{MI}}$ *succeeds and the output of the experiment is 1. It is 0 otherwise.*

In specific, we use black-box and white-box membership inference attacks which differ in the knowledge that $\mathcal{A}_{\mathtt{MI}}$ is assumed to possess about an ML model. In black-box MI $\mathcal{A}_{\mathtt{MI}}$ is limited to external features of a machine learning model such as the loss or prediction confidence during inference. In contrast, in white-box MI $\mathcal{A}_{\mathtt{MI}}$ possesses also internal features of the model such as gradients. In particular, we consider MI attacks against central learning by Shokri et al. [SS15], Yeom et al. [Yeo+18], and the MI attack against both central and federated neural networks by Nasr et al. [NSH19]. MI attacks assume that $\mathcal{A}_{\mathtt{MI}}$ has access to a trained prediction function $h(\cdot)$, knowledge about the hyperparameters and DPs mechanisms that were used for training. We refer to the trained prediction function as *target model* and the training data as $\mathcal{D}^{\mathtt{train}}_{\mathtt{target}}$. Given this accessible information $\mathcal{A}_{\mathtt{MI}}$ learns a binary classifier, the *attack model*, that allows to classify data into members and non-members w.r.t. the target model training dataset with high accuracy. The accuracy of an attack model is evaluated on a balanced dataset including all members (target model training data) and an equal number of non-members (target model test data), which simulates the worst case where $\mathcal{A}_{\mathtt{MI}}$ tests membership for all training records. MI exploits that an ML classifier such as a neural network (NN) tends to classify a record $d = (x, y)$ from its training dataset $\mathcal{D}^{\mathtt{train}}_{\mathtt{target}}$ with different confidence $p(\mathbf{x})$ given $h(\mathbf{x})$ for features $\mathbf{x}$ and true label y than a record $d \notin \mathcal{D}^{\mathtt{train}}_{\mathtt{target}}$.

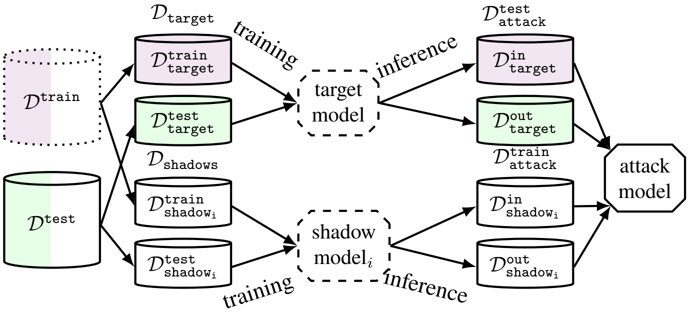

$\mathcal{D}_{\mathtt{target}}$
training
$\mathcal{D}^{\mathtt{train}}_{\mathtt{target}}$
$\mathcal{D}^{\mathtt{test}}_{\mathtt{target}}$
target model
inference
$\mathcal{D}^{\mathtt{test}}_{\mathtt{attack}}$
$\mathcal{D}^{\mathtt{in}}_{\mathtt{target}}$
$\mathcal{D}^{\mathtt{out}}_{\mathtt{target}}$
$\mathcal{D}^{\mathtt{train}}$
$\mathcal{D}^{\mathtt{test}}$
$\mathcal{D}_{\mathtt{shadows}}$
$\mathcal{D}^{\mathtt{train}}_{\mathtt{attack}}$
attack model
$\mathcal{D}^{\mathtt{train}}_{\mathtt{shadow_i}}$
$\mathcal{D}^{\mathtt{test}}_{\mathtt{shadow_i}}$
shadow model$_i$
$\mathcal{D}^{\mathtt{in}}_{\mathtt{shadow_i}}$
$\mathcal{D}^{\mathtt{out}}_{\mathtt{shadow_i}}$
training
inference

Figure 2.3.: Black-box MI with attack features $(y^*, p(\mathbf{x}))$. LDP perturbation on $\mathcal{D}^{train}$ (dotted) and CDP on target model and shadow model training (dashed). Data that was used during target model training is colored: training (violet) and validation (green)

2.4.1. Black-Box MI Attack

The black-box (BB) MI attack of Shokri et al. [SS15] is limited to external features of a trained machine learning model. This is for example the case when a model is exposed through an API. Black-box MI exploits that an ML classifier such as a neural network (NN) tends to classify a record d from its training dataset $\mathcal{D}^{\mathtt{train}}_{\mathtt{target}}$ with different high softmax confidence $p(\mathbf{x})$ given $h(\mathbf{x})$ at its true y in than a record $d \notin \mathcal{D}^{\mathtt{train}}_{\mathtt{target}}$. Therefore, $\mathcal{A}_{\mathtt{MI}}$ follows two steps. First, $\mathcal{A}_{\mathtt{MI}}$ trains copies of the target model w.r.t. structure and hyperparameters, so called *shadow models*, on data statistically similar to $\mathcal{D}^{\mathtt{train}}_{\mathtt{target}}$ and $\mathcal{D}^{\mathtt{test}}_{\mathtt{target}}$. It applies that $|\mathcal{D}^{\mathtt{train}}_{\mathtt{shadow_i}}| = |\mathcal{D}^{\mathtt{test}}_{\mathtt{shadow_i}}| \wedge \mathcal{D}^{\mathtt{train}}_{\mathtt{shadow_i}} \cap \mathcal{D}^{\mathtt{test}}_{\mathtt{shadow_i}} = \varnothing \wedge |\mathcal{D}_{\mathtt{shadow_i}} \cap \mathcal{D}_{\mathtt{shadow_j}}| \geq 0$ for any $i \neq j$. After training, each shadow model is invoked by $\mathcal{A}_{\mathtt{MI}}$ to classify all respective training data (member records) and test data (non-member records), i.e., $p(\mathbf{x}), \forall\, d \in \mathcal{D}^{\mathtt{train}}_{\mathtt{shadow_i}} \cup \mathcal{D}^{\mathtt{test}}_{\mathtt{shadow_i}}$. Since $\mathcal{A}_{\mathtt{MI}}$ has full control over $\mathcal{D}^{\mathtt{train}}_{\mathtt{shadow_i}}$ and $\mathcal{D}^{\mathtt{test}}_{\mathtt{shadow_i}}$, each shadow model's output $(p(\mathbf{x}), y)$ is appended with label "`in`" if the corresponding record $d \in \mathcal{D}^{\mathtt{train}}_{\mathtt{shadow_i}}$. Otherwise, its label is "`out`". Second, $\mathcal{A}_{\mathtt{MI}}$ trains a binary classification attack model per target variable $y \in Y$ to map $p(\mathbf{x})$ to the indicator "`in`" or "`out`". The triples $(p(\mathbf{x}), y, \mathtt{in/out})$ serve as the attack model training dataset, i.e., $\mathcal{D}^{\mathtt{train}}_{\mathtt{attack}}$. Thus, the attack model exploits the imbalance between predictions on $d \in \mathcal{D}^{\mathtt{train}}_{\mathtt{target}}$ and $d \notin \mathcal{D}^{\mathtt{train}}_{\mathtt{target}}$. An illustration of the BB MI attack is given in Figure 2.3. The features for attack model training are generated by passing shadow model training and test data again through the trained shadow model. The attack model AP is computed on features similarly extracted from the target model.

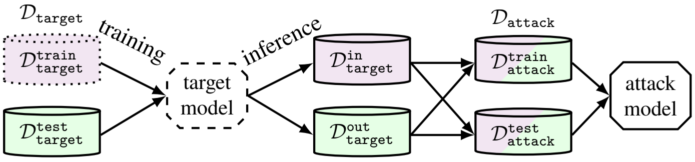

$\mathcal{D}_{\texttt{target}}$
$\mathcal{D}^{\texttt{train}}_{\texttt{target}}$
$\mathcal{D}^{\texttt{test}}_{\texttt{target}}$
training
target model
inference
$\mathcal{D}^{\texttt{in}}_{\texttt{target}}$
$\mathcal{D}^{\texttt{out}}_{\texttt{target}}$
$\mathcal{D}_{\texttt{attack}}$
$\mathcal{D}^{\texttt{train}}_{\texttt{attack}}$
$\mathcal{D}^{\texttt{test}}_{\texttt{attack}}$
attack model

Figure 2.4.: White-box MI with attack features $\big(y^*, p(\mathbf{x}), L(h(x;\theta), y), \frac{\delta L}{\delta \theta}\big)$. LDP perturbation on $\mathcal{D}^{\texttt{train}}_{\texttt{target}}$ (dotted) and DP on target model training (dashed). Target model training is colored: training (violet) and validation (green)

2.4.2. White-Box MI Attack

White-box MI [NSH19] makes two assumptions about $\mathcal{A}_{\texttt{MI}}$. First, $\mathcal{A}_{\texttt{MI}}$ can observe internal features of the ML model in addition to external features (i.e., model outputs). The internal features comprise observed losses $L(h(x;\theta))$, gradients $\frac{\delta L}{\delta \theta}$ and the learned weights θ of $h(\cdot)$. Second, $\mathcal{A}_{\texttt{MI}}$ is aware of a portion of $\mathcal{D}^{\texttt{train}}_{\texttt{target}}$ and $\mathcal{D}^{\texttt{test}}_{\texttt{target}}$. These portions were set to 50% by Nasr et al. [NSH19] and are the same within this thesis to allow comparison. Second, $\mathcal{A}_{\texttt{MI}}$ extracts internal and external features of a balanced set of confirmed members and non-members. An illustration of the white-box MI attack is given in Figure 2.4. Again, $\mathcal{A}_{\texttt{MI}}$ is assumed to know a portion of $\mathcal{D}^{\texttt{train}}_{\texttt{target}}$ and $\mathcal{D}^{\texttt{test}}_{\texttt{target}}$ and generates attack features by passing these records through the trained target model. $\mathcal{A}_{\texttt{MI}}$ trains a binary classification attack model per target variable $y \in Y$ to map $p(\mathbf{x})$ to the indicator "$\texttt{in}$" or "$\texttt{out}$". The set $\big(L(h(x;\theta)), \frac{\delta L}{\delta \theta}, p(\mathbf{x}), y, \texttt{in/out}\big)$ serves as attack model training dataset, i.e., $\mathcal{D}^{\texttt{train}}_{\texttt{attack}}$. Thus, the MI attack model exploits the imbalance between predictions on $d \in \mathcal{D}^{\texttt{train}}_{\texttt{target}}$ and $d \notin \mathcal{D}^{\texttt{train}}_{\texttt{target}}$. Attack model accuracy is computed on features extracted from the target model likewise.

When a central target model is trained by aggregating models from different parties over their respective data we speak of federated learning. In federated learning, the white-box MI attack generally is performed as in central learning setting. However, since the trained central target model is shared with all parties, $\mathcal{A}_{\texttt{MI}}$ can perform the MI attack as a participating party to learn about the training dataset of other parties (local MI adversary $\mathcal{A}_{\texttt{MI-L}}$), or as the central aggregator (global MI adversary $\mathcal{A}_{\texttt{MI-G}}$). $\mathcal{A}_{\texttt{MI-L}}$ receives a copy of the central target model after aggregation by the data scientist. $\mathcal{A}_{\texttt{MI-G}}$ leverages the model parameters that have been received by the aggregator from the other parties.

2.5. Performance Metrics

Within this thesis, we mostly use performance metrics for neural network classification tasks. We consider membership inference a binary classification task which is classifying records into training dataset members and non-members. It follows that *True Positives* (TP) are represented by records that were labeled as members and belong to the training dataset. *True Negatives* (TN) are records from test dataset that were not labeled as members of the training dataset. *False Positives* (FP) and *False Negatives* (FN) are represented by records from the test and training dataset that are incorrectly classified as members and non-members. We measure *Accuracy* according to Definition 2.8 and refer to the accuracy on train and test data as train and test accuracy. In addition, we use *Precision* and *Recall* to specifically evaluate the relevancy of the classifications. Correct identification of training data members is measured by *Precision* in Definition 2.9 and complete identification is quantified by *Recall* as of Definition 2.10.

Definition 2.8 (Accuracy [SW10]).

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

◇

Definition 2.9 (Precision [Tin10a]).

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

◇

Definition 2.10 (Recall [Tin10a]).

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

◇

Additionally, we provide definitions for the *True Positive Rate* (TPR) and *False Positive Rate* (FPR) in Definition 2.11 and 2.12 for completeness. We, however, only use TPR and

FPR indirectly for finally calculating two MI accuracy measures: *Average Precision* (AP) under the *Precision-Recall Curve* to capture in Definition 2.13 the inference accuracy w.r.t. members and the membership advantage over random guessing w.r.t. members and non-members in Definition 2.14. Yeom [Yeo+18] demonstrate that membership advantage is bound by $Adv < e^{\epsilon} - 1$.

Definition 2.11 (True Positive Rate [Tin10b]).

$$\text{True Positive Rate} = Sensitivity = \frac{TP}{TP + FN}$$

◇

Definition 2.12 (False Positive Rate [Tin10b]).

$$\text{False Positive Rate} = Specificity = \frac{FP}{FP + TN}$$

◇

Definition 2.13 (Average Precision under the Precision-Recall Curve [sci]).

$$AP = \sum_{n} (\mathit{Recall}_n - \mathit{Recall}_{n-\mathit{1}}) \cdot \mathit{Precision}_n,$$

for classification thresholds n. ◇

Definition 2.14 (Membership Advantage).

$$\mathit{Adv} = \mathit{TPR} - \mathit{FPR} = \mathit{Recall} - \mathit{FPR}.$$

◇

3. Related Work

We present related work in this thesis in two steps. First, we discuss related work w.r.t. to the underlying motivation of this thesis and to the identified research problems (Section 4.1) that this thesis is addressing. Second, we provide related work for each approach that we formulate to address the research problems in Chapters 5, 6, and 7.

This thesis is motivated by the work of Abadi et al. [Aba+16] on differentially private stochastic gradient descent for neural networks. Their work demonstrates two implications when using differential privacy in machine learning with neural networks: on the one hand that the privacy loss analysis with sequential composition is not tight, and instead, advanced composition theorems need to be applied to calculate the privacy loss in machine learning with neural networks. On the other hand, their work demonstrates that differentially private neural networks can yield test accuracies quite close to the test accuracies of non-DP neural networks for the same learning tasks (e.g., MNIST and CIFAR-10 image classification). Both implications led us to question the factual privacy loss that a participant in the dataset is facing at a given upper bound ϵ, which is not addressed by Abadi et al.: *By what means could the factual privacy loss be quantified, aside from the theoretic upper bound on the privacy loss? Are differentially private models with a test accuracy close to the original test accuracy also offering meaningful anonymization?* Rocher et al. [RHM19] raise a similar question for supposedly anonymized, published datasets by focusing on the quantification of identifiability for assessing whether the datasets should be considered pseudonymous or truly anonymous. However, their work does not consider differential privacy in machine learning with neural networks, but instead focuses on techniques for generalization and sampling.

In close time proximity the implementation of differential privacy for several user recommendation features in their operating systems iOS and MacOS was announced [WWD16]. Since ϵ was not communicated by Apple in the beginning Tang et al. [Tan+17] performed an analysis of the implementation and find the ϵ per datapoint to be up to 2, and for a system to be up to 16 per day. McSherry notes in this context that “Apple has put some kind of handcuffs on in how they interact with your data. It just turns out those handcuffs are made out of tissue paper”, and that “Anything much bigger than one is not a very reassuring guarantee. Using an ϵ value of 14 per day strikes me as relatively pointless” [Gre17]. This thesis strives to gather evidence for claims such as raised by McSherry by quantifying the factual privacy loss.

Complementary to and mostly decoupled from the discussion about ϵ in differentially private machine learning, threat models for the confidentiality of training data in machine learning with neural networks received significant attention. While the threat models generally assume an honest-but-curious adversary with access to the trained model, they part in their assumption about background knowledge of the training data. Model inversion attacks assume that the adversary strives for the identification of records that are highly likely for a given classification label (i.e., not necessarily individual records of the training dataset, but an idealistic representation of training dataset records). In contrast, model inference attacks assume an adversary who strives to identify training data membership or non-membership for specific records [Hay+19; HHB19; NSH18; NSH19; SS15; Sho+17]. In this thesis we use membership inference attacks for comparison of ϵ between different mechanisms and model architectures for differentially private training of neural networks. In contrast, related work [Hay+19; JE19; Yeo+18] in membership inference attacks considered differential privacy as a means for mitigating membership inference attacks. We want to note that there are more efficient means available for the goal of mitigating membership inference attacks such as adversarial regularization [NSH18] and generalization [LOK21].

Several works also approached the question of differences in lower and upper bounds for the privacy loss in differentially private machine learning with neural networks under specific threat models. Yeom et al. [Yeo+18] suggest capturing the lower and upper bound on privacy loss under a membership inference adversary in a bound called *membership advantage*. However, their membership advantage bound is not tight as Jayaraman and Evans [JE19] demonstrate for the differentially private stochastic gradient descent. Humphries et al. [Hum+20] derive a bound for membership advantage that is tighter than the original bound of Yeom et al. [Yeo+18] by analyzing an adversary with additional information and giving up the i.i.d. assumption. We use the precision and recall of the membership inference adversary to quantify and compare the lower bound on the privacy loss between local and central differential privacy, arguing that membership in the training data is capturing the privacy loss (i.e., the sensitive information to be protected). In addition, we formulate an implementation of the strong DP adversary and transformations of ϵ into identifiability scores. The implementation and score allow a tight estimation of the privacy loss and auditing of trained models w.r.t. privacy loss. Nasr et al. [Nas+21] also formulate an implementation of the strong DP adversary in close time proximity to a similar result of this thesis [Ber+22]. While making similar observations, Nasr et al. do not consider identifiability bounds and do not suggest a heuristics for closing the gap between lower and upper bounds on the privacy loss.

Lastly several other works also remark that ϵ is not absolute but rather relative in the context of specific datasets, functions and mechanisms. Dwork et al. [Dwo+11] highlight that ϵ is measuring the "cumulative privacy loss suffered by an individual in a given database", and note that more research is required on the value of ϵ. To this end Dwork et al. [DKM19] suggest a public ϵ registry after having performed several interviews in which "no clear consensus on how to choose ϵ or even how to approach this" [DKM19] question of choosing ϵ could be identified. The difficulty of choosing ϵ is further underlined by Garfinkel et al. [GAP18] in the context of releasing differentially private US Census statistics, since several US states objected to the use of differential privacy due to potential side effects on redistricting [Wan21]. We provide insights into the differences in ϵ between LDP and CDP, for different datasets and model architectures, and to the privacy loss that is encountered under a membership inference adversary and the strong adversary considered by differential privacy.

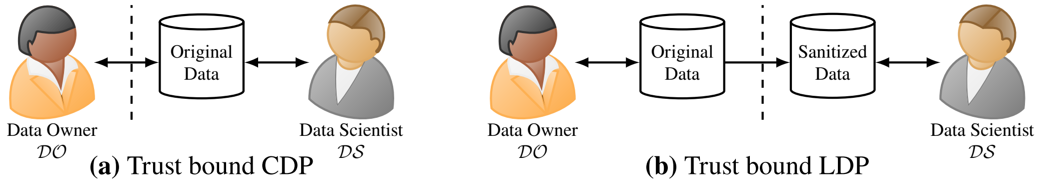
4. Problem Definition

This chapter states the research problem addressed in this thesis in Section 4.1. The research problem is discussed from two aspects, namely quantification of the privacy-accuracy trade-off in the presence of a machine learning adversary and quantification of identifiability in the presence of a differential privacy adversary, in Sections 4.2 and 4.3.

4.1. Interpreting Privacy Parameter ϵ

This thesis addresses the question of how to choose privacy parameter ϵ in differentially private machine learning of neural networks. Within this thesis, we consider the use of differential privacy during the training of a neural network along a generic data science process (e.g., CRISP-DM [WH00]). In a data science process a dataset $\mathcal{D}$ of a data owner $\mathcal{DO}$ is (i) transformed (i.e., pre-processed), (ii) used by a data scientist $\mathcal{DS}$ to learn a prediction function $h(\cdot)$ (e.g., classification of a record d to a feature y with a neural network), (iii) exposed by $\mathcal{DS}$ for use by third parties after training (e.g., through an API). We consider two trust models in this thesis which we depict in Figure 4.1. In the first trust model, the data owner $\mathcal{DO}$ trusts $\mathcal{DS}$ w.r.t. obeying the differential privacy parameters and mechanisms, and that the provided data will only be used for the agreed purpose (e.g., differentially private training of a neural network). In this trust model, depicted in Figure 4.1a, CDP can be used while learning $h(\cdot)$. In addition, $\mathcal{DS}$ and $\mathcal{DO}$ can choose to perturb each record $d \in \mathcal{D}$ during the data science process transformation step by choosing LDP instead of CDP. In the second trust model $\mathcal{DO}$ does not trust $\mathcal{DS}$ w.r.t. enforcing differential privacy. In this case, depicted in Figure 4.1b $\mathcal{DO}$ and $\mathcal{DS}$ are restricted to LDP mechanisms.

Notwithstanding whether local or central differential privacy is used $\mathcal{DS}$ must be able to justify that meaningful anonymization and thus privacy is provided to records in the dataset [Kel20; PE16]. $\mathcal{DS}$ steers the anonymization strength and implicitly also



(a) Trust bound CDP

(b) Trust bound LDP

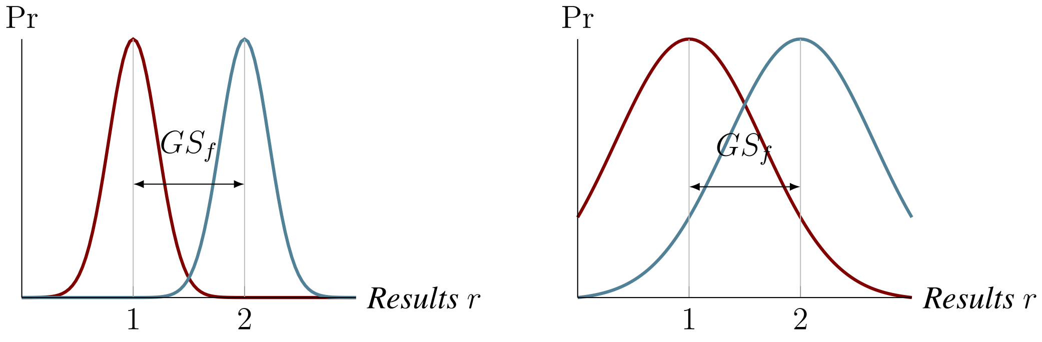
Figure 4.1.: Trust boundaries in CDP and LDP (dotted lines)

the utility by the privacy parameter ϵ. Rearranging Equations (2.1) and (2.5) for ϵ, we can make two immediate observations. First, ϵ is formulating an upper bound on the likely ratio between neighboring datasets from which the result of differentially private mechanism potentially was produced:

$$e^{\epsilon} \geq \begin{cases} \frac{\Pr[\mathcal{M}(\mathcal{D})\in\mathcal{R})]}{\Pr[\mathcal{M}(\mathcal{D}')\in\mathcal{R})]+\delta} & \text{CDP,} \\ \frac{\Pr[\mathcal{LR}(v)=s]}{\Pr[\mathcal{LR}(v')=s]} & \text{LDP} \end{cases}$$

Taking into consideration that neighboring datasets differ in one entry we refer to the upper bound as privacy bound w.r.t. an individual in the dataset. Furthermore, we refer to e^{ϵ} as *privacy loss* in the rest of this thesis. Second, the upper bound on the privacy loss will not be reached if the sensitivity between datasets $\mathcal{D}, \mathcal{D}'$ is smaller than the assumed GS_f. In consequence, ϵ is “typically set after a societal-technical process during which the mathematical-theoretical understandings are matched with practical real-life experience, and then reviewed and adjusted periodically” [Nis+18]. To illustrate the effect of smaller and larger values for privacy parameter ϵ Figure 4.2 provides two simplified illustrations of the Gaussian mechanism for $\epsilon = 3$ and $\epsilon = 0.1$ (cf. Equation (2.2)). Note that the upper bound on the privacy loss, i.e., the difference between the red and blue curve at any point on the x Axis is only reached when the difference between mechanism outputs is close to GS_f.

In addition to setting privacy parameter ϵ, a metric by which the performance of differentially private neural networks is measured needs to be set (e.g., F1-Score, Mean Squared Error, or Accuracy). The choice of performance metric depends on the learning task. In classification tasks and also for evaluation of regularization techniques that apply noise to the training data to foster generalization, the accuracy of the trained model over test data is commonly used to evaluate performance [GBC16; GC95; Mat92]. The privacy loss and the performance metric allow to quantify and compare privacy-accuracy



(a) $\epsilon = 3$ **(b)** $\epsilon = 0.1$

Figure 4.2.: Gaussian mechanism for $f(\mathcal{D}) = 1$, $f = (\mathcal{D}') = 2$, $GS_f = 1$ and $\epsilon \in \{0.1, 3\}$

trade-offs. In summary, this thesis addresses the following research problems.

Problem 1. *Quantifying the empirical lower bound on the privacy loss under empirical attacks to allow data scientists to compare the empirical privacy-accuracy trade-off between local and central differential privacy. The problem is approached separately for discriminative and generative models.*

(i) Comparing the empirical privacy-accuracy trade-off between local and central differential privacy for central discriminative neural networks, where training data is provided to the data scientist by the data owners.

(ii) Extending Problem 1.(i) to federated discriminative neural network where data owners train local models over their training data and solely share model parameters (e.g., weights) with data scientists.

(iii) Comparison of the privacy-accuracy trade-off between local and central differential privacy for central generative models.

Problem 2. *Transforming the privacy loss under differential privacy into an analytical bound on identifiability.*

4.2. Trade-Off Quantification Based on Privacy Loss Under a Machine Learning Adversary

When $\mathcal{DO}$ trusts $\mathcal{DS}$ differentially private neural networks can be realized by either perturbing the training data with LDP or the training optimizer with CDP. However, the upper bound on the privacy loss in LDP and CDP cannot be directly compared between LDP and CDP (cf. Definition 2.1 and 2.5). Instead, the privacy loss in LDP and CDP can be compared empirically by implementing an adversary who has access to a trained neural network and strives for the identification of the training data. The success of such an adversary then yields a lower bound on the privacy loss and allows to compare LDP and CDP by their privacy-accuracy trade-off. A desirable trade-off should lower model test accuracy less than it lowers the success of the adversary.

However, in differentially private machine learning $\mathcal{DS}$ can choose to either implement the strong adversary with unconstrained auxiliary knowledge that differential privacy is defined to hold against [Dwo06], or to implement potentially inferior adversaries with constrained auxiliary knowledge. The implementation of the DP adversary with unconstrained auxiliary knowledge is based on several strong assumptions such as knowledge of up to all but one record in the original dataset [Dwo06; LC11]. In contrast, several implementable adversaries with fewer assumptions and general applicability to machine learning have been suggested by related work [Car+19; FJR15; Fre+14; Hay+19; HHB19; NSH18; NSH19; SS15; Sho+17; Tra+16; YFJ17; Yeo+18]. In particular membership inference [Hay+19; HHB19; NSH19; SS15; Sho+17; Yeo+18] and model inversion [FJR15; Fre+14; Tra+16]. Model inversion adversaries aim for the reconstruction of training data records based on the output of a trained machine learning model (e.g., prediction confidences). Membership inference adversaries aim for identifying specific records based on the output of a trained machine learning model. In this thesis, we focus on membership inference adversaries to quantify the privacy loss.

An empirical membership adversary $\mathcal{A}_{\mathtt{MI}}$ allows data scientists $\mathcal{DS}$ to quantify a lower bound on the privacy loss with evaluation metrics such as precision, recall, membership advantage, and accuracy (cf. Definitions 2.9, 2.10, and 2.14). Prior work considered lower and upper bounds on the privacy loss for CDP with the differentially private stochastic gradient descent [JE19], and neglected precision and recall in favor of accuracy and F1-

score [Rah+18; SS15] which are limited in case of imbalanced train and test data, or the membership advantage of $\mathcal{A}_{\mathtt{MI}}$ [Hay+19; YFJ17; Yeo+18]. The specific metric that $\mathcal{DS}$ chooses for quantifying the empirical privacy loss depends on whether membership or non-membership in the training dataset is considered sensitive, and whether the empirical lower bound should be compared against the theoretic upper bound (e.g., comparing empirical membership advantage with upper bound on membership advantage).

Empirical lower bounds on the privacy loss are facing limitations w.r.t. conclusions that can be drawn. On the one hand, the difference in lower and upper bound will depend on the dataset and the learning task for which machine learning is performed, especially if training data is unbalanced which benefits overfitting by leading to poor generalization of the machine learning model from training to test data [JS02]. On the other hand, the model architecture might affect the difference in lower and upper bounds since some model architectures will also introduce regularization effects that limit overfitting and thus weaken the membership inference in addition to differential privacy (e.g., federated learning with model averaging is yielding regularization similar to dropout [McM+17]). We use multiple reference datasets from several domains (e.g., images, preference matrices) and multiple reference model architectures to address concerns w.r.t. limitations of insights and conclusions. Chapters 5 and 6 use membership inference attacks to choose ϵ based on the privacy-accuracy trade-off. We consider feedforward neural networks in central and federated learning, and generative networks.

4.3. Bounds on Identifiability

The previous section discussed the opportunities and limitations of using membership inference adversaries to interpret the privacy-accuracy trade-off for choices of ϵ a posteriori. However, membership inference adversaries are inferior to the strong adversary with auxiliary information about the training dataset against which differential privacy is destined to hold [Dwo06]. This is because the differential privacy adversary is assumed to know the alternative dataset $\mathcal{D}'$ instead of only the distribution from which $\mathcal{D}'$ was chosen. Thus, the lower and upper bound on the privacy loss under membership inference adversaries may not be close [JE19]. An implementation of the differential privacy adversary with arbitrary auxiliary knowledge overcomes this limitation and allows audit of the empirical privacy loss when training a differentially private machine learning model.

A complementary problem to auditing trained models w.r.t. the empirical privacy loss is the actual specification of an upper bound on the privacy loss by setting a target ϵ a priori. This is of particular interest to fulfill societal expectations [Nis16] or legal requirements [HS10; Kel20; PE16] on identifiability when anonymizing data or machine learning models. A transformation of privacy parameter ϵ into probabilistic identifiability scores such as the reidentification likelihood has been demonstrated for one-dimensional, non-iterative statistical aggregate queries with CDP [LC11; LC12]. However, differentially private machine learning with CDP involves multidimensional, iterative queries. In addition, the specified overall ϵ for the training of a neural network should be close to the factual ϵ after training, since otherwise model accuracy is overly decreased by excessive noise. To this end, we extend and complement prior work with auditing and transformation means for ϵ.

Chapter 7 addresses the challenges of transforming privacy parameter ϵ to scores that quantify identifiability and formulates an implementation of the differential privacy to audit trained neural networks to audit the empirical privacy loss after training. We again use reference datasets from two domains (images, preference matrices) and two reference model architectures.

5. Assessing Differential Privacy under Membership Inference Attacks

Within this chapter, we compare the empirical privacy protection under the white-box MI attack of Nasr et al. [NSH19] against LDP and CDP mechanisms for learning problems from diverse domains: consumer preferences, face recognition and health data. The MI attack indicates a lower bound on the inference risk whereas DP formulates an upper bound [JE19; YFJ17; Yeo+18]. However, in practice even high privacy parameters in LDP may already offer protection against attacks such as membership inference. Depending on whether a neural network is directly trained by a data scientist on a central server based on the data of all data owners or indirectly trained by a data scientist based on the input from several servers that belong to separate data owners, neural networks are trained central and federated manner. In federated neural network training (i.e., federated training) the data scientist's central neural network is trained iteratively and collaboratively, by first having data owners train a local neural network on their respective data and second aggregating the individual neural networks at the central server of the data scientist [Li+20a]. In consequence, federated learning supports data owners in keeping the sovereignty of their sensitive or personal training data by sharing neural network weights that have been computed over a batch of the data owner's data instead of individual data with the data scientist [McM+17]. The threat model in federated learning is thus differing from the threat model in central learning, and we will thus consider federated learning in addition to central learning within this chapter. This chapter makes the following contributions:

- Comparing LDP and CDP by the average precision of their MI precision-recall curve as privacy measure, and show that under this measure LDP and CDP have similar privacy-accuracy trade-offs despite vastly different ϵ.

- Showing that CDP mechanisms are not achieving a consistently better privacy-accuracy trade-off on various datasets and reference models. The trade-off rather depends on the specific dataset.

- Analyzing the relative privacy-accuracy trade-off and showing that it is not constant over ϵ, but that for each data set there are ranges where the relative trade-off is greater for protection against MI than accuracy.

- By comparing federated learning and central learning we observe a regularization effect that leads to DP parameters with larger ϵ in federated learning.

Section 5.1 formulates our approach for comparing LDP and CDP under membership inference and Section 5.2 discusses the relative privacy-accuracy trade-off. Section 5.3 extends the approach to federated learning. We describe evaluation datasets in Section 5.4. Findings are presented in Section 5.5. Related work and summary are provided in Section 5.6 and Section 5.7.

5.1. Evaluating CDP and LDP under MI

DP has been shown to formulate a theoretical upper bound on the accuracy of MI adversaries [Yeo+18], and thus the use of DP should impact the classification accuracy of $\mathcal{A}_{\texttt{MI}}$. To illustrate the effect of the privacy parameter ϵ on the MI attack we focus on two questions related to the identifiability of training data within this work: “How many records predicted as `in` are truly contained in the training dataset?” (precision), and “How many truly contained records are predicted as `in`?” (recall). For analysis we use precision-recall curves which depict the precision and recall for various classification thresholds, and thus reflect the possible MI attack accuracies of $\mathcal{A}_{\texttt{MI}}$. We compare the precision-recall curves by their average precision (AP) to assess the overall effect of DP on MI. The AP approximates the integral under the precision-recall curve as a weighted mean of the precision P per threshold t and the increase in recall R from the previous threshold, i.e.: $AP = \sum_t (R_t - R_{t-1}) \cdot P_t$. We prefer this non-interpolated technique over interpolated calculations of the area under the curve, since the precision-recall curve is not guaranteed to decline monotonically and thus the linear trapezoidal interpolation might yield an overoptimistic representation [DG06; Eve+10]. Good MI attack models will realize an AP of close to 1 while poor MI attack models will be close to the baseline of uniform random guessing, hence $AP = 0.5$. We will commonly refer to the AP for MI as MI AP within this thesis. $\mathcal{DO}$ and $\mathcal{DS}$ have two options to apply DP against MI within the data science process introduced in Chapter 4. Either in the form of LDP by applying a local randomizer to the training data and using the resulting $\mathcal{LR}\left(\mathcal{D}_{\texttt{target}}^{\texttt{train}}\right)$ for training, or CDP with a differentially private optimizer on $\mathcal{D}_{\texttt{target}}^{\texttt{train}}$. A discussion and comparison of LDP and CDP purely based on the privacy parameter ϵ likely falls short and potentially leads data scientists to incorrect conclusions, since the privacy parameters are reflecting different types of mechanisms. Furthermore, data scientists give up flexibility w.r.t. applicable learning algorithms, if ruling out the use of LDP due to comparatively greater ϵ and instead solely investigating CDP (e.g., DPSGD). We suggest comparing LDP and CDP by their concrete effect on the MI AP and the resulting privacy-accuracy trade-off. While we consider a specific MI attack our methodology is applicable to other MI attacks as well. Models that use CDP are represented by dashed lines in Figure 2.4. In the LDP setup, the target model is trained with perturbed records from a local randomizer, i.e., $\mathcal{LR}\left(\mathcal{D}_{\texttt{target}}^{\texttt{train}}\right)$. However, to increase his attack accuracy $\mathcal{A}_{\texttt{MI}}$ needs to learn attack models with high accuracy on the original data from which the perturbed records stem, i.e., $\mathcal{D}_{\texttt{target}}^{\texttt{train}}$. Perturbation with LDP is represented by dotted lines in Figure 2.4.

5.2. Relative Privacy-Accuracy Trade-off

We calculate the relative privacy-accuracy trade-off φ below in Equation (5.1) for LDP and CDP as the relative difference between $\mathcal{A}_{\texttt{MI}}$'s change in MI AP to $\mathcal{DS}$'s change in test accuracy. Let AP_{orig}, AP_{ϵ} be the MI APs, and ACC_{orig}, ACC_{ϵ} be the test accuracies for the original and DP target model. Furthermore, let ACC_{base} be the baseline test accuracy of uniform random guessing $1/m$, where m denotes the number of labels in the dataset, and AP_{base} be the baseline MI AP at 0.5. We fix ACC_{base}, AP_{base} since $\mathcal{A}_{\texttt{MI}}$ and $\mathcal{DS}$ would perform worse than uniform random guessing at lower values. Starting from φ''

we rearrange and bound the cases where MI AP and ACC increase over ϵ to obtain φ':

$$\begin{aligned}
\varphi'' &= \frac{(AP_{orig} - AP_{\epsilon})/(AP_{orig} - AP_{base})}{(ACC_{orig} - ACC_{\epsilon})/(ACC_{orig} - ACC_{base})} \\
\varphi' &= \frac{\max(0, (AP_{orig} - AP_{\epsilon}) \cdot (ACC_{orig} - ACC_{base}))}{\max(0, (ACC_{orig} - ACC_{\epsilon}) \cdot (AP_{orig} - AP_{base}))} \\
\varphi &= \min\left(2, \frac{\max(0, (AP_{orig} - AP_{\epsilon}) \cdot (ACC_{orig} - ACC_{base}))}{\max(0, (ACC_{orig} - ACC_{\epsilon}) \cdot (AP_{orig} - AP_{base}))}\right)
\end{aligned} \tag{5.1}$$

To prevent φ' from approaching infinitely large values when the accuracy remains stable while AP decreases significantly, and the undefined case of $ACC_{orig} \leq ACC_{\epsilon}$, we bound by 2 and finally obtain φ. In consequence, when the relative gain in privacy (lower AP) exceeds the relative loss in accuracy, it applies that $1 < \varphi \leq 2$, and $0 \leq \varphi < 1$ when the loss in test accuracy exceeds the gain in privacy. Hence, φ quantifies the relative loss in accuracy and the relative gains in privacy for a given privacy parameter ϵ and captures the relative privacy-accuracy trade-off as a ratio that we seek to maximize.

5.3. Central and Federated learning

In a data science process such as the previously mentioned CRISP-DM, a data scientist centrally gathers data from data owners, trains and evaluates a machine learning model (e.g., a neural network). In federated learning (FL) data owners first train individual models over their respective training data, secondly share their respective weight updates with the data scientist who then aggregates the weight updates for training a central model, and thirdly distributes the aggregated model to all participating data owners [McM+17]. By training a local model for each data owner and only sharing local models with the data scientist instead of individual training data, federated learning with model aggregation provides data owners sovereignty over their data and offers data scientists scalable training for central models.

To train local models data owners receive an initialized neural network architecture from the data scientist. Afterward each data owner sends their neural network parameters, for example the respective gradients or model weights, in specific interval (e.g., every other epoch) to the central data scientist for aggregation. The central model is updated with the aggregated parameters received from all data owners. Data owners replace their local model once a new global model is propagated from the data scientist after aggregation. In contrast to CRISP-DM, there is no single point in time where a model is being deployed for production, rather the models are constantly updated. Furthermore, due to the localization constraints, the data scientist cannot preprocess the training data. Preprocessing can only occur at each data owner.

Several FL algorithms can be used to aggregate local models. We use the *Federated Averaging Algorithm* for distributed SGD [Aga+18], specified in Definition 5.1. In the Federated Averaging Algorithm a global model $F(w)$ is comprised of the averaged local models of participating data owners K. Furthermore, a communication period represents one iteration in which data owners train their local models for k epochs [McM+17]. The fraction of data owners that train in parallel for one communication period is set by $0 \leq C \leq 1$. Inverse proportional weighting of the share of data that each data owner possesses in a given communication period ensures that data owners with few samples still have impact to the training of the global model.

Definition 5.1 (Federated Averaging Algorithm [Aga+18]). Let $F(w) : \mathcal{R}^d \longrightarrow \mathcal{R}$ be of the form

$$F(w) = \frac{1}{C \cdot |K|} \sum\nolimits_{i=0}^{C \cdot |K|} h_i(W)$$

where each $f_i(x; W)$ resides at the i-th data owner. ◇

5.4. Datasets and Learning Tasks

We consider four datasets for experiments. The datasets have been used in related work on MI and face recognition. Each dataset is also summarized in Table 5.1. The reference datasets are mostly unbalanced w.r.t. the amount of training data per training label, a characteristic that we found to benefit MI attacks. For example, Texas Hospitals Stays and Purchases Shopping Carts provided by Shokri et al. [SS15] are unbalanced in terms of records per label, as shown in Figures 5.1 and 5.2.

Texas Hospital Stays. The Texas Hospital Stays dataset [Sho+17] is an unbalanced dataset and consists of high dimensional binary vectors representing patient health features. Each record within the dataset is labeled with a procedure. The learning task is to train a fully connected neural network for classification of patient features to a procedure and

Table 5.1.: Overview of datasets considered in the evaluation

Dataset	**Model**	**LDP**
Texas Hospital Stays [Sho+17]	Fully connected NN with three layers ($512 \times 128 \times m$) [Sho+17].	$19,125$ – 638 ($6382 \times \epsilon_i$)
Purchases Shopping Carts [Sho+17]	Fully connected NN with two layers ($128 \times m$) [Sho+17] (i.e., logistic regression).	1800 – 60 ($600 \times \epsilon_i$)
Labeled Faces in the Wild [Hua+07]	VGG-Very-Deep-16 CNN [PVZ15]	62.5×10^6 – $6,250$ ($250 \times 250 \times \epsilon_i$)
Skewed Purchases	Fully connected NN with two layers ($128 \times m$) [Sho+17] (i.e., logistic regression).	$1,800$ – 60 ($600 \times \epsilon_i$)

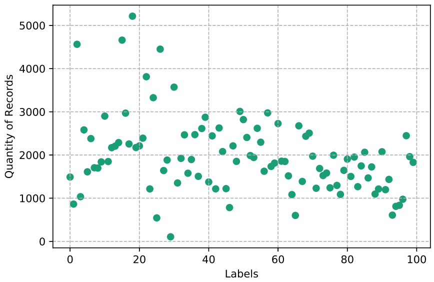
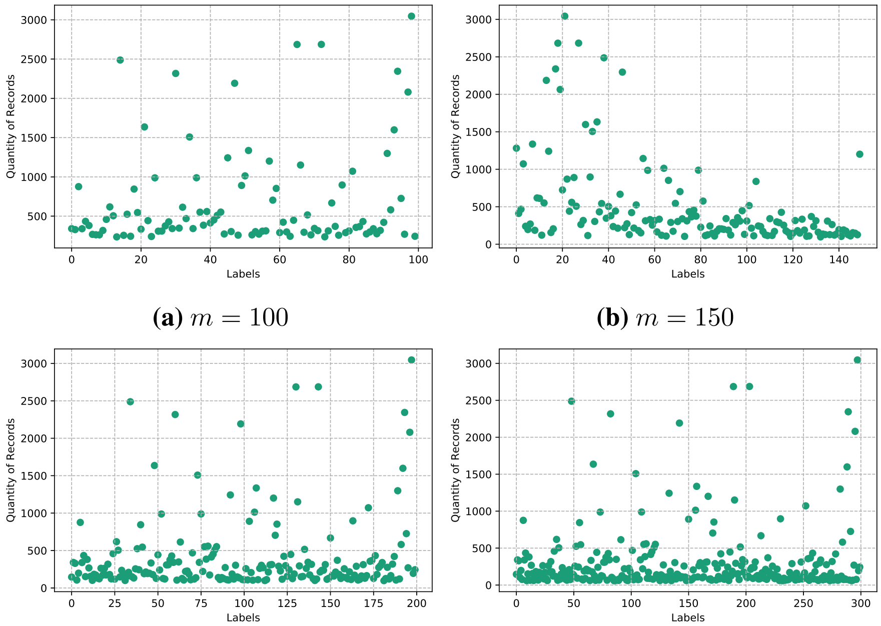


Figure 5.1.: Quantity of records per label for Purchases Shopping Carts

we do not try to re-identify a known individual, and fully comply with the data use agreement for the original public use data file. We train and evaluate models for a set of most common procedures $m \in \{100, 150, 200, 300\}$. Depending on the number of procedures the dataset comprises 67330 – 89815 records and 6170 – 6382 features. To allow comparison with related work [NSH19; Sho+17], we train and test the target model on $n = 10000$ records respectively.

Purchases Shopping Carts. This dataset is also unbalanced and consists of binary vectors with 600 features that represent customer shopping carts [Sho+17]. However, a significant difference to the Texas Hospital Stays dataset is that the number of features is almost 90% lower. Each vector is labeled with a customer group. The learning task is to classify shopping carts to customer groups by using a fully connected neural network. The dataset is provided in four variations with varying numbers of labels $m \in \{10, 20, 50, 100\}$ and comprises 38551 – 197324 records. We sample $n = 8000$



(a) $m = 100$

(b) $m = 150$

(c) $m = 200$

(d) $m = 300$

Figure 5.2.: The Quantity of records per Label for the Texas Hospital Stays Dataset

records each for training and testing the target model. Again, this methodology ensures comparability with related work [NSH19; Sho+17].

Labeled Faces in the Wild. The Labeled Faces in the Wild (LFW) dataset contains labeled images each depicting a specific person with a resolution of 250×250 pixels (i.e., features) [Hua+12; Hua+07]. The dataset has a long distribution tail w.r.t. to the number of images per label with a minimum of 6 and a maximum of 530 pictures. We thus focus on learning the topmost labels $m \in \{20, 50, 100\}$ with 1906, 2773 and 3651 overall records respectively. We start our comparison of LDP and CDP from a pre-trained VGG-Very-Deep-16 CNN faces model [PVZ15] by keeping the convolutional core, exchanging the dense layer at the end of the model and training for LFW grayscale faces. For LDP, we apply differentially private image pixelization within the neighborhood $o = \sqrt{250 \times 250}$ and avoid coarsening by setting $b = 1$. We transform all images to grayscale before LDP and CDP training.

Skewed Purchases. We specifically crafted this balanced dataset[1] to mimic a transfer learning task, i.e., the application of a trained model to novel data that is similar to the training data w.r.t. format but following a different distribution. This situation arises for Purchases Shopping Carts, if for example not enough high-quality shopping cart data for a specific retailer are available yet. Thus, only few high-quality data (e.g., manually crafted examples) can be used for testing and large amounts of low quality data from potentially different distributions for training (e.g., from other retailers). In effect the distribution between train and test data varies for this dataset. Similar to Purchases Shopping Carts the dataset consists of 200000 records with 600 features and is analyzed for $m \in \{10, 20, 50, 100\}$ labels. However, each vector x in the training dataset X is generated by using two independent random coins to sample a value from $\{0, 1\}$ per position $i = 1, \ldots, 600$. The first coin steers the probability $\Pr[x_i = 1]$ for a fraction of 600 positions per record x. We refer to these positions as indicator bits (*ind*) which indicate products frequently purchased together. The second coin steers the probability $\Pr[x_i = 1]$ for a fraction of $600 - \left(\frac{600}{|m|}\right)$ positions per record. We refer to these positions as noise bits (*noise*) that introduce scatter in addition to *ind*. We let $\Pr_{ind}[x_i = 1] = 0.8 \wedge \Pr_{noise}[x_i = 1] = 0.2$, $\forall x \in X_{train}$ and $\Pr_{ind}[x_i = 1] = 0.8 \wedge Pr_{noise}[x_i = 1] = 0.5, \wedge x \in X_{test}, 1 \leq i \leq 600$. The difference in information entropy between test and train data is ≈ 0.3.

5.5. Experiments

We perform an experiment that compares the privacy-accuracy trade-off for LDP and CDP by MI AP instead of privacy parameter ϵ per dataset. We first compare the privacy-accuracy trade-off for central learning and secondly for federated learning.

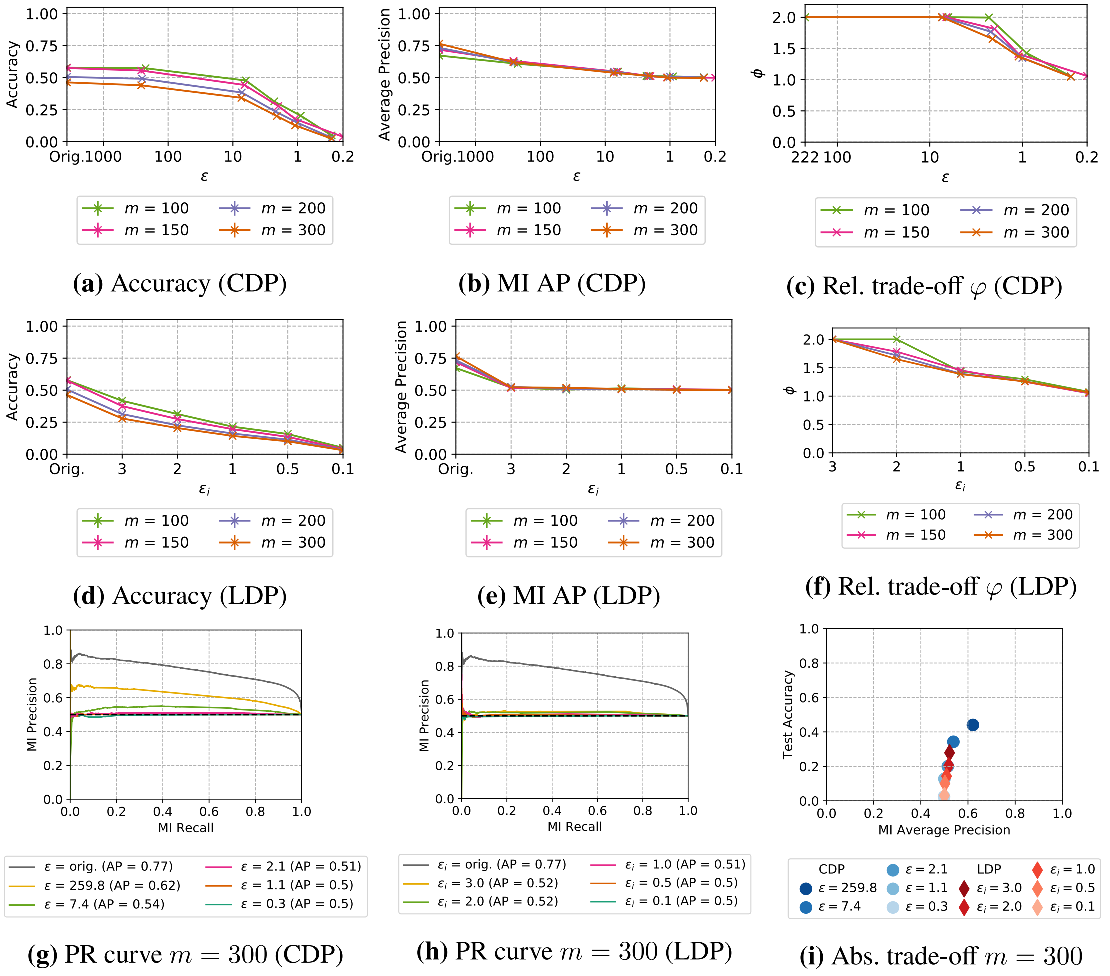
The results of each experiment are visualized by three sets of figures. First, we compare the relative privacy-accuracy trade-off φ resulting from test accuracy and MI AP over ϵ. We present this information for CDP per dataset in Figures 5.3 to 5.9 a,b,c and for LDP in Figures 5.3 to 5.9 d,e,f. The obtained information serves as a basis to identify privacy parameters at which the MI AP is converging towards the baseline. Second, we state the precision-recall curves from which MI AP was calculated to illustrate the slope with which

[1] We provide this dataset along with all evaluation code on GitHub: `https://github.com/SAP-samples/security-research-membership-inference-and-differential-privacy`

precision and recall are diverging from the baseline for LDP and CDP in Figures 5.3 to 5.9 g,h. Third, we compare the absolute privacy-accuracy trade-offs per dataset for both LDP and CDP in a scatterplot. We present this information in Figures 5.3 to 5.9i. For each dataset the model training stops once the test data loss is stagnating (i.e., early stopping) or a maximum number of epochs is reached. This design avoids excessive overfitting and increases real-world relevance. For all executions of the experiment CDP noise is sampled from a Gaussian distribution (cf. Equation (2.2)) with scale $\sigma = \textit{noise multiplier } z \times \textit{clipping norm } \mathtt{C}$. We evaluate increasing noise regimes per dataset by evaluating noise multipliers $z \in \{0.5, 2, 4, 6, 16\}$ and calculate the resulting ϵ at a fixed $\delta = \frac{1}{n}$. However, since the batch size, dataset size and number of epochs are also influencing the Rényi differential privacy accounting a fixed z will inevitably result in different overall ϵ for different datasets. For LDP we use the same hyperparameters as in the original training and evaluate two local randomizers, namely Randomized Response and LDP Image Pixelization with the Laplace mechanism. For each randomizer we state the individual ϵ_i per invocation (i.e., per anonymized value). We apply Randomized Response to all datasets except LFW with a range of privacy parameter values $\epsilon_i \in \{0.1, 0.5, 1, 2, 3\}$ that reflect retention probabilities ρ from $5\% - 90\%$ (cf. Section 2.2). For LFW each pixel is perturbed with Laplace noise, and also investigate a wide range of resulting noise regimes by varying ϵ_i. For federated learning we assume $K = 4$ data owners and a participation rate of $C = 1$ to align with the experimental setup from Nasr et al. [NSH19]. Furthermore, data owners train for one epoch before propagating their parameters and the data scientist aggregates the weight matrices after each communication round. We limit our federated learning experiment to the relative privacy-accuracy trade-off w.r.t. $\mathcal{A}_{\mathtt{MI-G}}$ (cf. Section 2.4.2). In addition, we omit the precision-recall curves and do not evaluate the Skewed Purchases dataset, since highly imbalanced data was incompatible with our experimental framework[2].

For sake of completeness we provide the resulting overall privacy parameters ϵ, z, hyperparameters and train accuracies for all datasets for LDP and CDP in Table 5.1 and Table A.2 in Appendix A.2. The experiment is repeated five times per dataset to stabilize measurements and we report mean values with error bars unless otherwise stated. Precision-recall curves depict all experiment data.

[2] All code for federated learning experiments is provided under the following link: `https://github.com/SAP-samples/security-research-fed-dp-mia`



(a) Accuracy (CDP) **(b)** MI AP (CDP) **(c)** Rel. trade-off φ (CDP)

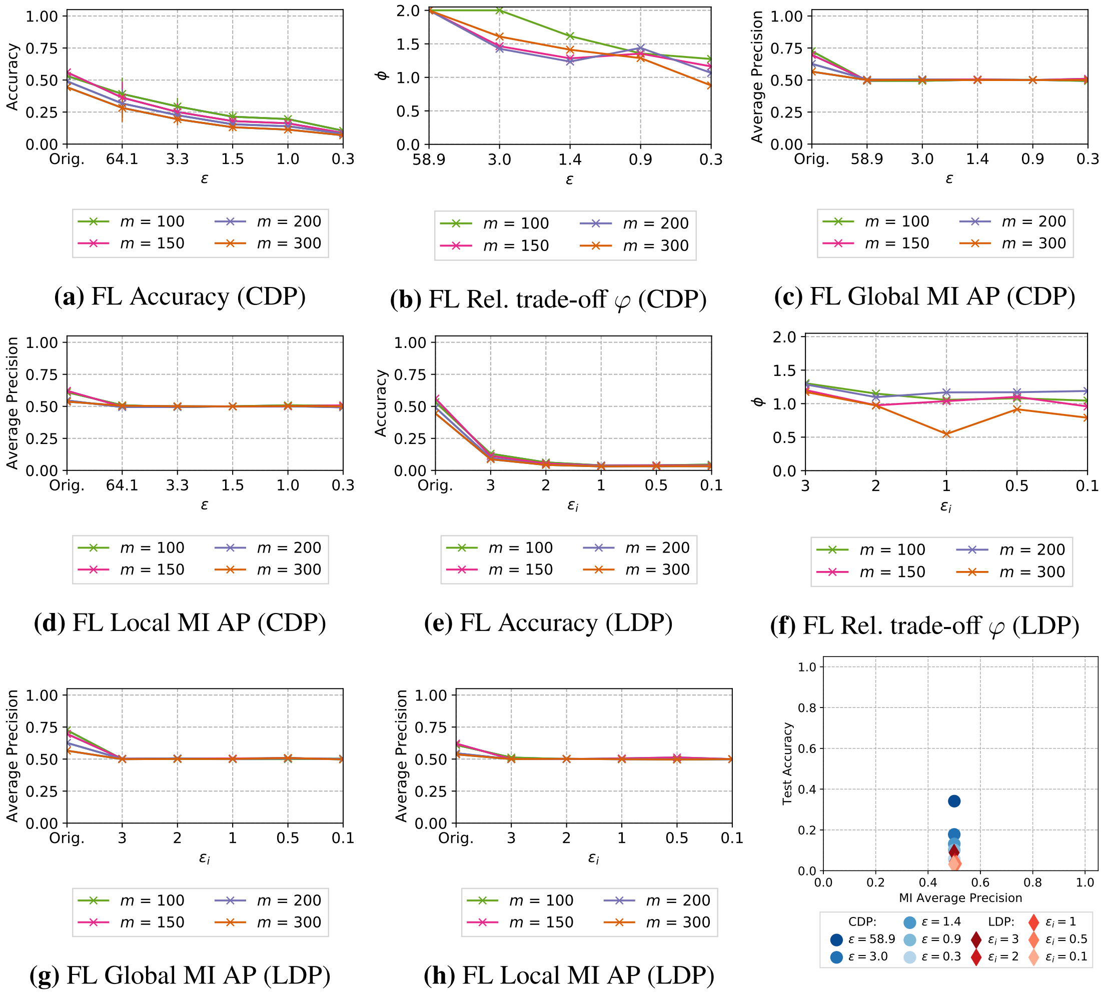
(d) Accuracy (LDP) **(e)** MI AP (LDP) **(f)** Rel. trade-off φ (LDP)

(g) PR curve $m = 300$ (CDP) **(h)** PR curve $m = 300$ (LDP) **(i)** Abs. trade-off $m = 300$

Figure 5.3.: Texas Hospital Stays accuracy and privacy (error bars lie within points)

5.5.1. Texas Hospital Stays

Central learning. For Texas Hospital Stays we observe that LDP and CDP are achieving very similar privacy-accuracy trade-offs under MI. The main difference between LDP and CDP is observable in a smoother decrease of target model test accuracy for CDP in contrast to LDP, which are depicted in Figures 5.3a and 5.3d. The smoother decay also manifests in a slower drop of MI AP for CDP in comparison to LDP as stated in Figures 5.3b and 5.3e. Texas Hospital Stays represents an unbalanced high dimensional dataset and both factors foster MI. However, the increase in dataset imbalance by increasing m is negligible w.r.t. MI AP. The relative privacy-accuracy trade-off for LDP and CDP is also



(a) FL Accuracy (CDP)

(b) FL Rel. trade-off φ (CDP)

(c) FL Global MI AP (CDP)

(d) FL Local MI AP (CDP)

(e) FL Accuracy (LDP)

(f) FL Rel. trade-off φ (LDP)

(g) FL Global MI AP (LDP)

(h) FL Local MI AP (LDP)

(i) FL Abs. trade-off $m = 300$

Figure 5.4.: FL Texas Hospital Stays accuracy and privacy (error bars lie within points)

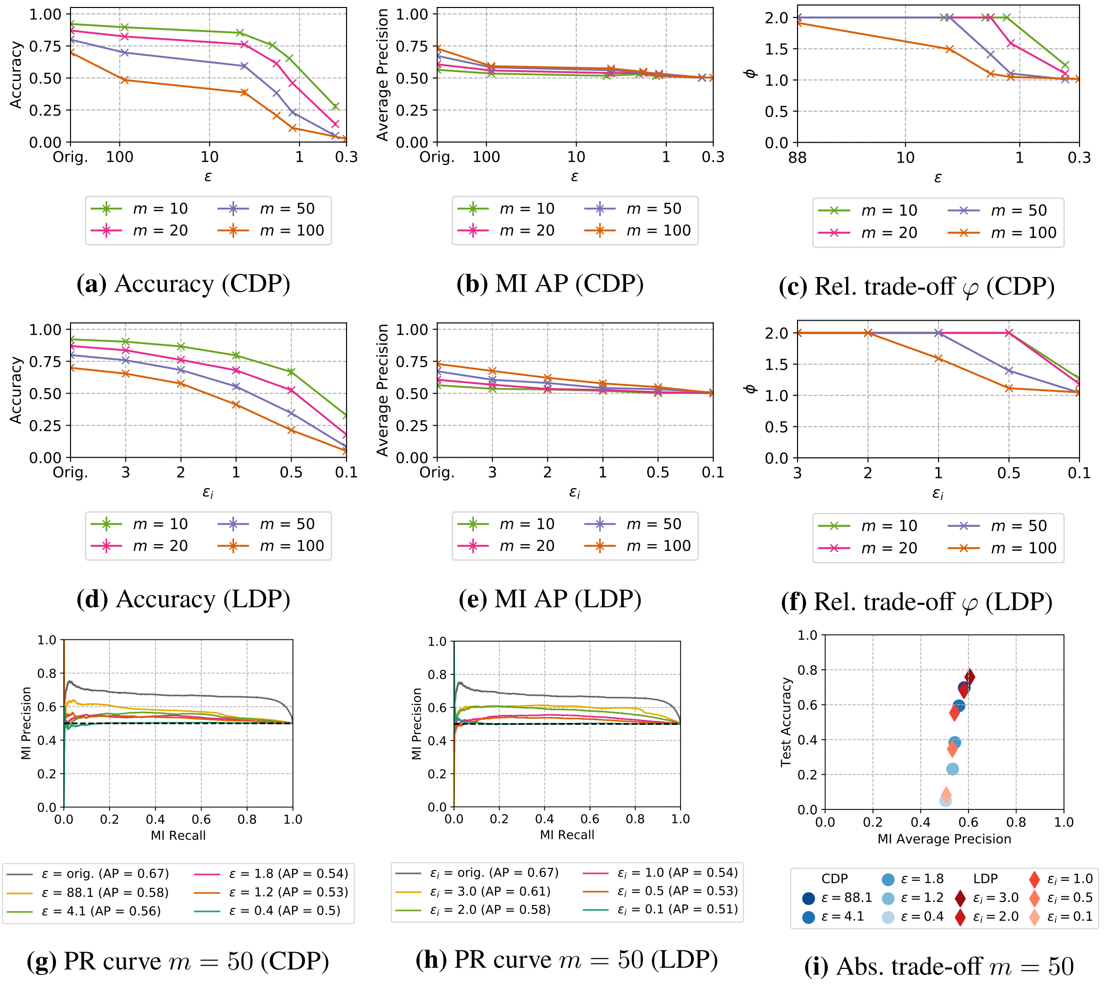
close and for example the baseline MI AP of 0.5 is reached at $\varphi \approx 1.5$, as depicted in Figures 5.3c and 5.3f. In the example case of $m = 300$ $\mathcal{DS}$ might prefer to use CDP, since the space of achievable MI APs in LDP is narrow while CDP also yields MI APs between original and baseline as illustrated in the precision-recall curves in Figures 5.3g and 5.3h, and the scatterplot in Figure 5.3i. This observation is similar, though weaker, for all other m.

Federated learning. In comparison to central learning, we can observe a faster drop in target model test accuracy both for LDP in Figure 5.4a and CDP in Figure 5.4e. However, the decline in accuracy still remains smoother for CDP. Noticeable, the MI AP drops to baseline already after adding the slightest noise for CDP and LDP as can be seen for $\mathcal{A}_{\texttt{MI-L}}$ and $\mathcal{A}_{\texttt{MI-G}}$ in Figures 5.4c, 5.4d, 5.4g and 5.4h. $\mathcal{A}_{\texttt{MI-G}}$ has a stronger initial MI AP than $\mathcal{A}_{\texttt{MI-L}}$, and larger number of classes m slightly improves the MI AP. The relative trade-off for LDP and CDP in Figures 5.4b and 5.4f suggest that CDP achieves a strictly better relative trade-off with a maximum of $\varphi \approx 2$ for $\epsilon \approx 59$ awhile LDP achieves a maximum of $\varphi \approx 1.25$ at $\epsilon_i \approx 3$. The absolute trade-off in Figure 5.4i underlines this impression. For all other m the absolute trade-off is similar.

5.5.2. Purchases Shopping Carts

Central learning. CDP and LDP are achieving similar target model test accuracies on the Purchases dataset as depicted in Figures 5.5a and 5.5d. However, LDP is allowing a slightly smoother decrease in test accuracy over ϵ. Figure 5.5b illustrates that the CDP MI AP is somewhat resistant to noise and remains above 0.5 until a small $\epsilon \approx 1$. The LDP MI APs are significantly higher and decrease slower than the baseline as depicted by Figure 5.5e. A comparison of the relative privacy-accuracy trade-offs φ in Figures 5.5c and 5.5f underlines that CDP and LDP achieve similar trade-offs and LDP allows for smoother drops in the MI AP in contrast to CDP. Thus, LDP is the preferred choice for this dataset, if $\mathcal{DS}$ desires to lower the MI AP to a level *between* original and baseline. This is illustrated for example for $m = 50$ in the precision-recall curves in Figures 5.5g, 5.5h and the scatterplots in Figure 5.5i. It is noticeable that while the overall ϵ for LDP and CDP differs by a magnitude of up to 10 times the relative and absolute privacy-accuracy trade-offs are close to each other. The observations also hold for other m.

Federated learning. The target model test accuracy gradually decays for CDP and LDP as depicted in Figures 5.6a and 5.6f. Similarly the relative privacy-accuracy trade-off is favorable for LDP and CDP for all $m \neq 50$ with $\phi > 1$, and CDP provides a sharp initial decline in MI AP whereas LDP gradually decreases MI AP when comparing Figures 5.6b and 5.6e. In summary, the data suggests $\mathcal{DO}$ to choose CDP due to the more favorable privacy-accuracy trade-off as for example depicted in Figure 5.6i for $m = 10$. However, the LDP trade-off is only slightly inferior for this dataset in federated learning similar to central learning.



(a) Accuracy (CDP) **(b)** MI AP (CDP) **(c)** Rel. trade-off φ (CDP)

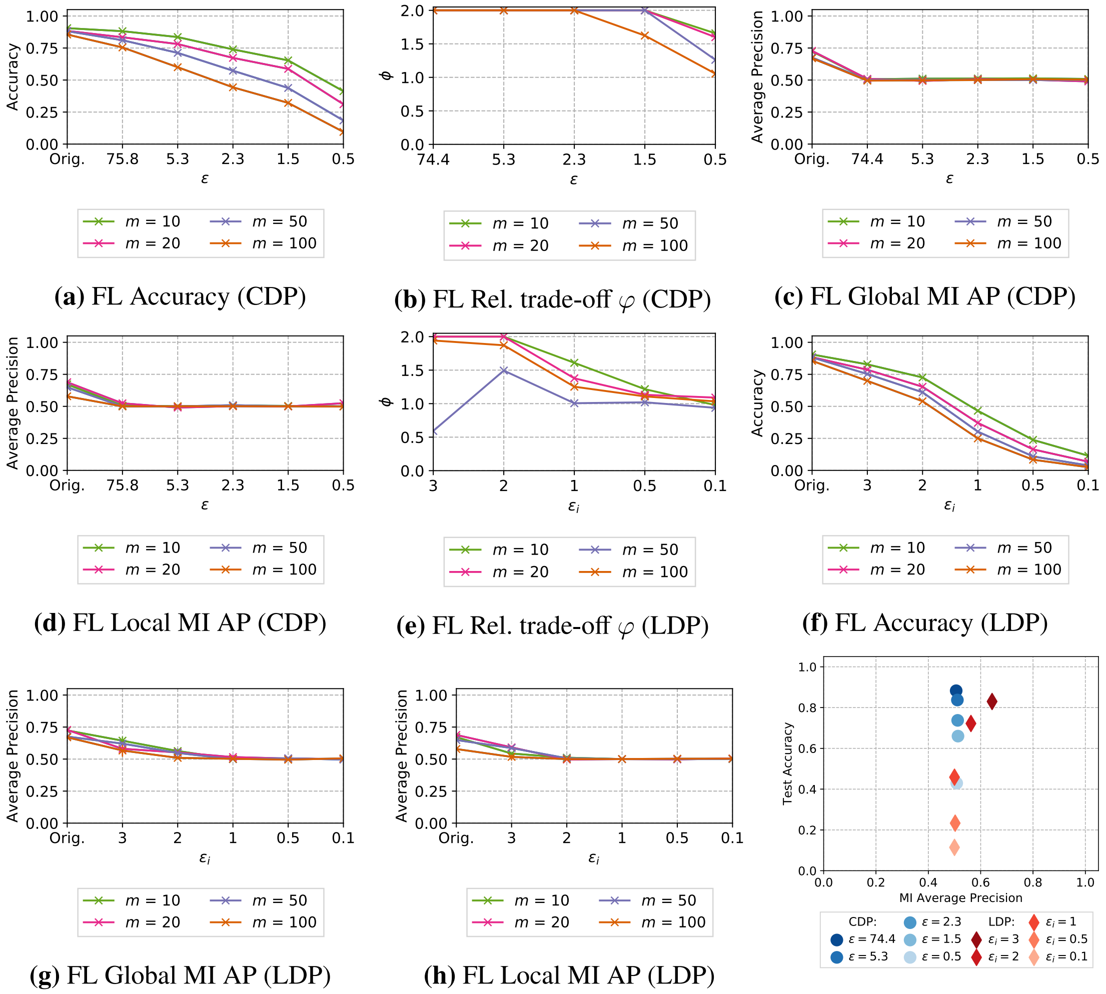
(d) Accuracy (LDP) **(e)** MI AP (LDP) **(f)** Rel. trade-off φ (LDP)

(g) PR curve $m = 50$ (CDP) **(h)** PR curve $m = 50$ (LDP) **(i)** Abs. trade-off $m = 50$

Figure 5.5.: Purchases accuracy and privacy (error bars lie within points)

5.5.3. LFW

Central learning. For LFW the target model reference architecture converges for both CDP and LDP towards the same test accuracy, which is reflecting the majority class. However, the target model test accuracy decay over ϵ is much smoother for CDP when comparing Figures 5.7a and 5.7d. Furthermore, the structural changes caused by LDP Image Pixelization seem to lead to quicker losses in test accuracy. W.r.t. the relative privacy-accuracy trade-off φ in Figures 5.7c and 5.7f CDP outperforms LDP. At MI AP $= 0.5$ CDP achieves $\varphi \approx 1.5$ for all m while LDP yields $\varphi \approx 1.1$ for all m. The $\varphi = 0$ observed at $\epsilon_i = 10000$ for $m = 100$ is due to an actual increase in MI AP that is



(a) FL Accuracy (CDP)

(b) FL Rel. trade-off φ (CDP)

(c) FL Global MI AP (CDP)

(d) FL Local MI AP (CDP)

(e) FL Rel. trade-off φ (LDP)

(f) FL Accuracy (LDP)

(g) FL Global MI AP (LDP)

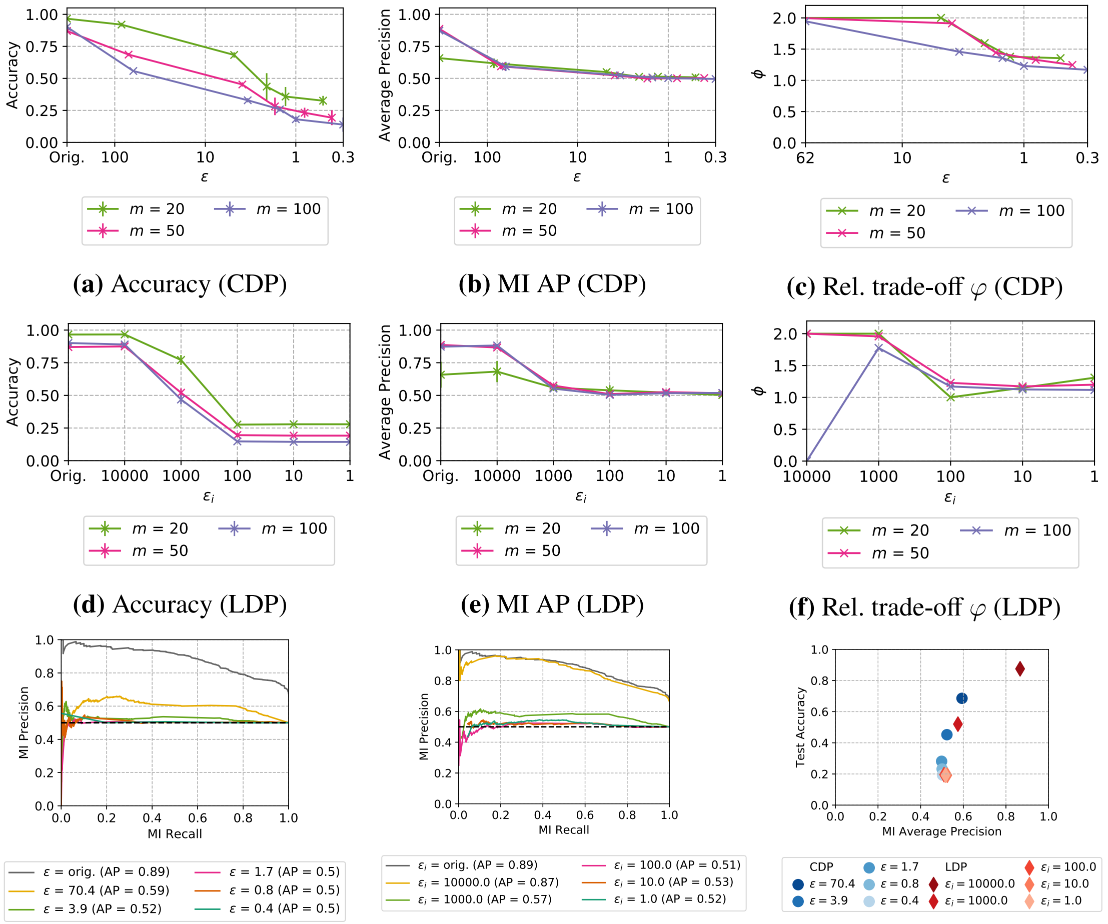
(h) FL Local MI AP (LDP)

(i) FL Abs. trade-off $m = 10$

Figure 5.6.: FL Purchases accuracy and privacy (error bars lie within points)

comparatively larger than the decrease in test accuracy. The exemplary precision-recall curves for $m = 50$ in Figures 5.7g and 5.7h furthermore illustrate that CDP can already have a large effect on MI AP at high ϵ. In addition, we observe from Figure 5.7i that CDP realizes a strictly better absolute privacy-accuracy trade-off under MI for $m = 50$.

Federated learning. We do not state results for $m = 100$ in the federated learning setting since the dataset was too resource-intensive for our differentially private federated learning framework. Interestingly the target model test accuracy plots show a similarly shaped decline for LDP and CDP in Figures 5.8a and 5.8e. We can furthermore see that $\mathcal{A}_{\texttt{MI-G}}$ constitutes a much stronger attack than $\mathcal{A}_{\texttt{MI-L}}$ when comparing the MI AP between Figure 5.8c and 5.8d for CDP, as well as Figure 5.8g and 5.8h for LDP. The best relative trade-offs with $\varphi \approx 2$ are yield for vastly different overall ϵ considering CDP at $\epsilon \approx 5.6$ and LDP $\epsilon_i = 1000$ for $m = 20$. For $m = 50$ CDP achieves superior trade-offs to LDP. The comparable performance of CDP and LDP for $m = 50$ is underlined by the overlapping absolute trade-off in Figure 5.8i, which however favorably separates CDP from LDP for $m = 20$. All in all, the observed MI APs are lower for federated learning compared to central learning at similar target model test accuracies.



(a) Accuracy (CDP)

(b) MI AP (CDP)

(c) Rel. trade-off φ (CDP)

(d) Accuracy (LDP)

(e) MI AP (LDP)

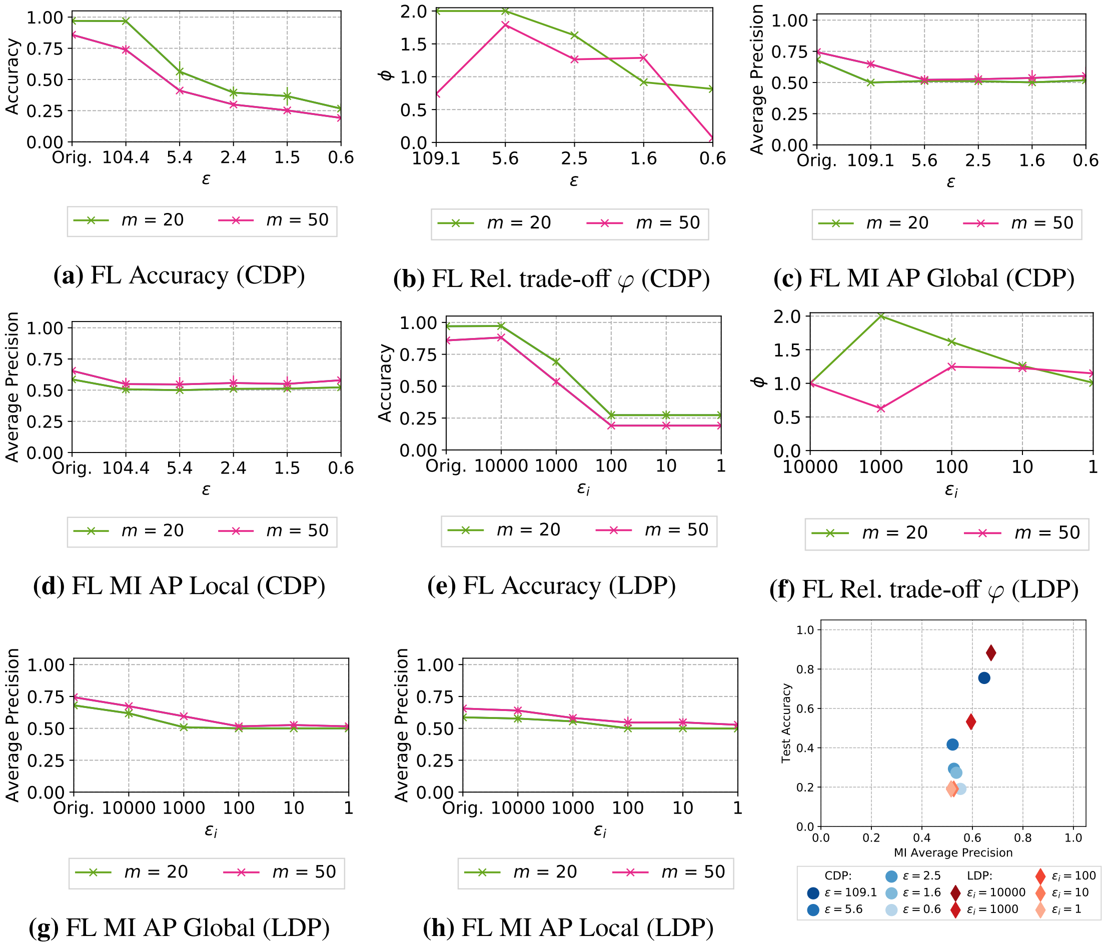
(f) Rel. trade-off φ (LDP)

(g) PR curve $m = 50$ (CDP)

(h) PR curve $m = 50$ (LDP)

(i) Abs. trade-off $m = 50$

Figure 5.7.: LFW accuracy and privacy (error bars lie within points)



(a) FL Accuracy (CDP)

(b) FL Rel. trade-off φ (CDP)

(c) FL MI AP Global (CDP)

(d) FL MI AP Local (CDP)

(e) FL Accuracy (LDP)

(f) FL Rel. trade-off φ (LDP)

(g) FL MI AP Global (LDP)

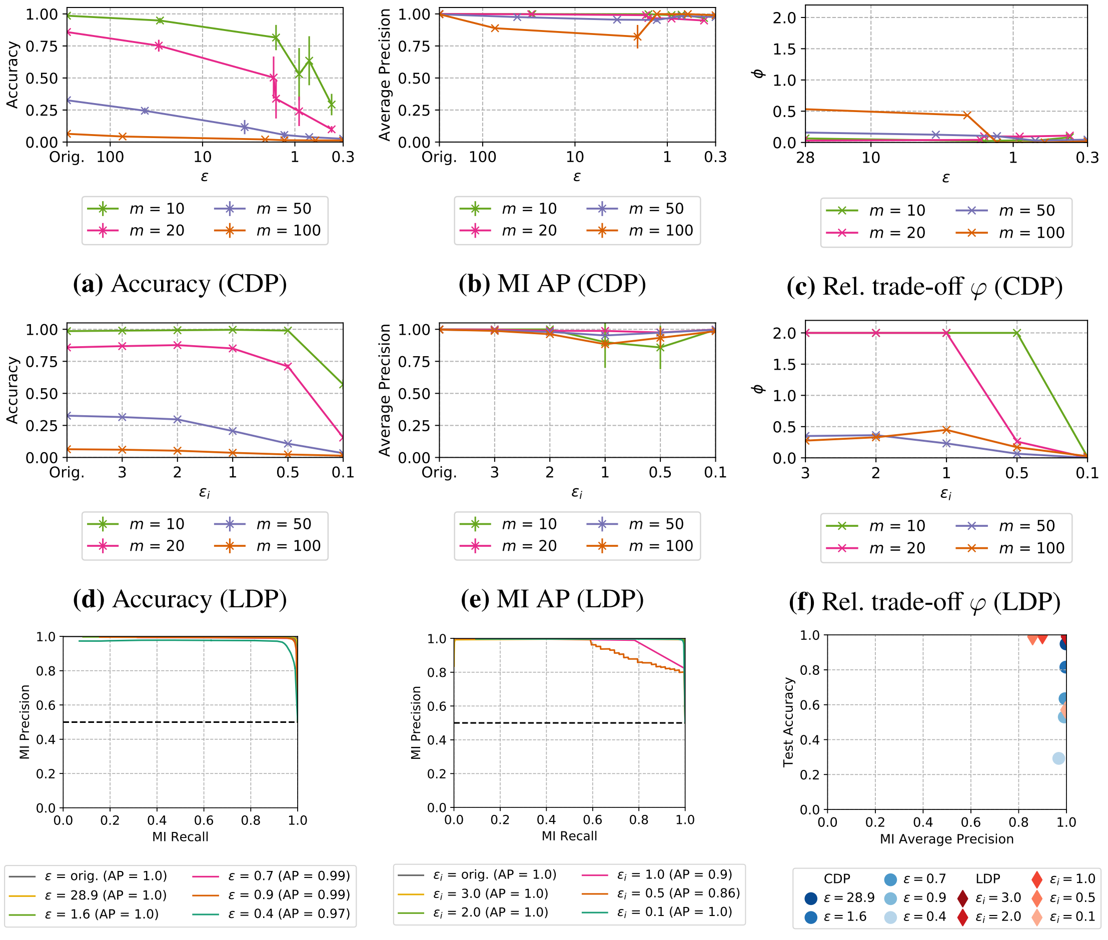
(h) FL MI AP Local (LDP)

(i) FL Abs. trade-off $m = 50$

Figure 5.8.: FL LFW accuracy and privacy(error bars lie within points)

5.5.4. Skewed Purchases

The effects of dimensionality and imbalance of a dataset on MI have been addressed by related work [NSH19; Sho+17]. However, the effect of a domain gap between training and test data which is found in transfer learning when insufficient high-quality data for training is initially available and reference data that potentially follows a different distribution has not been addressed. For this task, we consider the Skewed Purchases dataset. Figures 5.9a and 5.9d show that the LDP test accuracy is in fact only decreasing at very small ϵ_i whereas CDP again gradually decreases over ϵ. This leads to consistently higher test accuracy in comparison to CDP. W.r.t. the relative privacy-accuracy trade-off LDP outperforms CDP



(a) Accuracy (CDP) **(b)** MI AP (CDP) **(c)** Rel. trade-off φ (CDP)

(d) Accuracy (LDP) **(e)** MI AP (LDP) **(f)** Rel. trade-off φ (LDP)

(g) PR curve $m = 10$ (CDP) **(h)** PR curve $m = 10$ (LDP) **(i)** Abs. trade-off $m = 10$

Figure 5.9.: Skewed Purchases accuracy and privacy (error bars lie within points)

as depicted by φ in Figures 5.9c and 5.9f. However, we observe several outliers. Most notably for CDP, the MI AP decreases for $m = 100$ and large ϵ values but increases for small ϵ as shown in Figure 5.9b. This is a consequence of the target model resorting to random guessing for test records. Similarly, for LDP the MI AP for $m \in \{10, 100\}$ first decreases before recovering again as depicted in Figure 5.9e. We reason about the cause of these outliers by analyzing the target model's decisive confidence values. LDP generalizes the training data towards the test data, however, at $\epsilon_i = 1.0$ LDP leads to nearly indistinguishable test and train distributions. Thus, the decisive softmax confidence of the target model increases in comparison to smaller and larger ϵ_i. For $m = 10$ the absolute privacy-accuracy trade-off is also favorable for LDP as depicted in Figure 5.9i.

5.6. Related Work

Our work is related to DP in central and federated learning with neural networks, attacks against the confidentiality of training data, and performance benchmarking of neural networks.

CDP is a common means to realize differentially private neural networks by adding noise to the gradients during model training. Fundamental approaches for CDP perturbation with the differentially private gradient descent during model training were provided by Song et al. [SCS13], Bassily et al. [BST14], and Shokri et al. [SS15]. Abadi et al. [Aba+16] formulated the DPSGD that was used in this chapter. Mironov [Mir17] introduces Rényi DP for measuring the DPSGD privacy loss over composition. Iyengar et al. [Iye+19] suggest a hyperparameter-free algorithm for differentially private convex optimization for standard optimizers. Alternatives to the FedAVG algorithm are Matched Averaging (FedMA) [Wan+20] and FedProx [Li+20b] which focus more on *heterogeneous* statistical distributions within the datasets of the data owners (i.e., federated learning clients). While FedMA computes a *layer-wise* average, FedAVG a *coordinate-wise* average, and FedProx modifies the federated SGD by adding a proximity term. We did not find any related work in the context of privacy and anonymization for FedMA and FedProx. A stand-alone differentially private protocol for federated learning (i.e., not modifying the SGD of an existing federated learning protocol) is represented by *PATE* [Pap+18]. PATE considers to each data owner as an individual *Teacher* providing input to a central *aggregate teacher* in the form of prediction votes. The aggregate teacher adds Laplacian noise to the vote histogram and yields the prediction with the highest noisy vote from the ensemble (*max-of-Laplacian* mechanism). The last step in training with PATE involves a student model. The student model uses unlabeled and non-sensitive data and combines them with a prediction label by the aggregate teacher. PATE relies on the assumption that an adversary has no means to directly interact with the aggregate teacher or the teacher ensemble, but only interacts with the student model. In contrast, the DPSGD in FedAVG used in this chapter has no such assumption and instead allows to consider $\mathcal{A}_{\texttt{MI-L}}$ and $\mathcal{A}_{\texttt{MI-G}}$. Agarwal et al. [Aga+18] formulate a CDP Binomial mechanism for federated learning. While their work focuses on identifying favorable trade-offs between privacy and communication efficiency we focus on identifying favorable privacy-accuracy trade-offs to choose ϵ.

Fredrikson et al. [FJR15; Fre+14] formulate model inversion attacks that use target model softmax confidence values to reconstruct training data per label. In contrast, MI attacks are addressing the threat of identifying individual records in a dataset [Bac+16; San+09]. Yeom et al. [Yeo+18] have demonstrated that the upper bound on MI risk for CDP can be converted into an expected bound for MI advantage. We state MI precision and recall, arguing that `in` is the sensitive information. Jayaraman and Evans [JE19] showed that the theoretic MI upper bound and the achievable MI lower bound are far apart in CDP. We observe, that LDP can be an alternative to CDP as the upper and lower bounds are even farther apart from each other. Shokri et al. [NSH18] formulate optimal mitigation against their MI attack [Sho+17] by using adversarial regularization. By applying the MI attack gain as a regularization term to the objective function of the target model, a non-leaking behavior is enforced w.r.t. MI. While their approach protects against their MI adversary, DP mitigates any adversary with arbitrary background information. Carlini et al. [Car+19] suggest *exposure* as a metric to measure the extent to which neural networks memorize sensitive information. Similar to our work, they apply DP for mitigation. We focus on attacks against machine learning models targeting the identification of members of the training dataset. Abowd and Schmutte [AS19] describe an economic social choice framework to choose privacy parameter ϵ. We compare LDP and CDP mechanisms aside from ϵ. Rahman et al. [Rah+18] applied a black-box MI attack against DPSGD models on CIFAR-10 and MNIST. They evaluate the severity of MI attack by the F1-score which results in numerically higher scores but assumes `out` labels to be sensitive.

MLPERF [Web18] and DPBench [Hay+16] are frameworks for machine learning performance measurements and evaluation of DP. We focus on comparing the privacy-utility trade-off and apply the core principles of both benchmarks.

5.7. Summary

Privacy parameter ϵ alone is unsuited to compare and select and compare DP mechanisms. We consistently observed that while the theoretic upper bound on inference risk reflected by ϵ in LDP is higher by a factor of hundreds or even thousands in comparison to CDP, the practical protection against a white-box MI attack is actually not considerably weaker at similar model accuracy. For Texas Hospital Stays LDP mitigates white-box MI at an overall $\epsilon = 6382$ whereas CDP lies between $\epsilon = 0.9$ for $m = 100$ and $\epsilon = 0.3$ for $m = 300$. This observation at the baseline MI AP also holds for Purchases Shopping Carts where LDP $\epsilon = 60$ and CDP is between $\epsilon = 0.4$ for $m = 10$ and $\epsilon = 0.3$ for $m = 100$, and LFW (LDP $\epsilon = 62.5 \times 10^2$, CDP $\epsilon = 2.1$ to $\epsilon = 1.5$). Thus, we note that assessing privacy solely based on ϵ falls short. Given the results of the previous sections, we rather encourage data scientists to also quantify privacy under an empirical attack such as white-box MI in addition to ϵ.

LDP and CDP result in similar privacy-accuracy curves. A wide range of privacy regimes in CDP and LDP can be compared with our methodology under MI. We observed for most datasets that similar privacy-accuracy combinations are obtained for well generalizing models (i.e., use of early stopping against excessive overfitting) that were trained with LDP or CDP. We also ran the experiments with black-box MI (i.e., only model outputs) and observed that the additional assumptions made by white-box MI (e.g., access to internal gradient and loss information) only yield a small increase in MI AP (3 – 5%). For sake of completeness we provide plots for these additional black-box MI experiments in Appendix A.3, but will not discuss them further within this thesis. The privacy-accuracy scatterplots depict that LDP and CDP formulate very similar privacy-accuracy trade-offs for Purchases Shopping Carts, LFW, and Texas Hospital Stays. At two occasions on the smaller classification tasks Purchases Shopping Carts $m = \{10, 20\}$ and Skewed Purchases $m = \{10, 20\}$ LDP realizes a strictly better privacy-accuracy trade-off w.r.t. the practical inference risk. These observations lead us to conclude that LDP is an alternative to CDP for differentially private deep learning on binary and image data since the privacy-accuracy trade-off is often similar at the same model accuracy despite the significantly larger ϵ. Thus, data scientists should consider using LDP especially when required to use optimizers without CDP implementations or when training ensembles (i.e., multiple models over one dataset), since the privacy loss will accumulate overall ensemble target models when assuming that training data is reused between ensemble models. Here, we see one architectural benefit of LDP: flexibility. LDP training data can be used for all ensemble models without increasing the privacy loss in contrast to CDP.

The relative privacy-accuracy trade-off is favorable within a small interval. We observed that the privacy-accuracy trade-off as visualized in the scatterplots throughout this work allows identifying whether CDP or LDP achieve better test accuracy at similar APs. However, the scatterplots do not reflect whether the target model test accuracy is decreasing slower, similar, or stronger than MI AP decreases over the privacy parameter ϵ. For this purpose, we introduced φ. We found that φ allows us identifying ϵ intervals in which the MI AP loss is stronger than the test accuracy loss for all datasets. On the high dimensional datasets, Texas Hospital Stays and LFW CDP consistently achieves higher φ than LDP. In contrast, φ values are similar for LDP and CDP on Purchases and superior for LDP on Skewed Purchases.

Federated Learning adversary localization. We observe that $\mathcal{A}_{\texttt{MI}-\texttt{G}}$ is strictly stronger than $\mathcal{A}_{\texttt{MI}-\texttt{L}}$ in federated learning with FedAVG. To assess the lower bound on the privacy loss in MI we thus recommend implementing $\mathcal{A}_{\texttt{MI}-\texttt{G}}$. However, in case the data scientist is trusted by all participants and the data owners do not trust each other $\mathcal{A}_{\texttt{MI}-\texttt{L}}$ formulates a lower bound on the privacy loss.

Federated Learning provides an inherent privacy gain. We observed a slight gain in privacy for federated learning when we compare the MI AP between the central and federated learning experiments. For instance, the MI adversary $\mathcal{A}_{\texttt{MI}}$ in central learning on LFW has an average precision close to 0.9 in Figure 5.7b, while $\mathcal{A}_{\texttt{MI}-\texttt{G}}$ in federated learning achieves only about 0.75 in Figure 5.8c. We assume that the lower MI attack performance in federated learning is coupled with the generalization effect of FedAVG. By averaging the weights a similar effect as dropout can be achieved, which minimizes the risk of overfitting and to that end limits membership inference [McM+16]. However, our experiments do not allow to conclude that federated learning generally prevents MI attacks, since $\mathcal{A}_{\texttt{MI}-\texttt{G}}$ and $\mathcal{A}_{\texttt{MI}-\texttt{L}}$ achieve MI precisions well above the baseline in all experiments. Thus, the use of DPSGD in FedAVG is slightly improving protection against MI adversaries in comparison to central learning. Our experiment results show that federated learning achieves similar privacy-accuracy trade-offs to central learning for larger ϵ. Thus the gap between the upper and lower bound on the privacy loss is actually larger in federated learning. While this thesis generally does not discuss the use of cryptographic techniques, we want to note that the membership inference attack against federated learning would remain feasible even if secure aggregation is used to protect individual updates at the aggregator (e.g., Segal et al. [Seg+17]). However, without the use of cryptographic techniques stronger attacks than MI are feasible (such as the

Differential Identifiability attack described in Chapter 7 of this thesis).

In conclusion, this chapter addressed Problem 1.(i) and 1.(ii). The chapter quantified the lower bound on the privacy loss in LDP and CDP for central and federated differentially private deep learning under a white-box MI attack. The lower bound was quantified as the average precision of the MI precision-recall curve, a metric that particularly quantifies the privacy of members in the training data. The accuracy was measured by the target model test accuracy. Taken together, MI AP and target model test accuracy support data scientists in choosing among available DP mechanisms and selecting privacy parameter ϵ. Our experiments for diverse learning tasks and datasets show that neither LDP nor CDP yields a consistently better privacy-accuracy trade-off. While MI only yields a lower bound on the privacy loss whereas ϵ in DP formulates an upper bound on the privacy loss, we observed that the lower bounds for LDP and CDP are close at similar model accuracy despite the large differences in their upper bound. This suggests that the upper bound is far from the practical susceptibility to MI attacks and that data scientists should also consider applying LDP despite the large privacy parameter values. Especially, since LDP does not require privacy accounting when training multiple models and offers flexibility w.r.t. optimizers. We consider the relative privacy-accuracy trade-off for LDP and CDP as the ratio of losses in accuracy and privacy over ϵ, and show that it is only favorable within a small interval. We find that federated learning decreases the performance of MI adversaries due to a generalization effect. When using DPSGD in federated learning the gap between the upper and lower bound on the privacy loss for both LDP and CDP is larger than for central learning.

6. Assessing Differential Privacy under Membership Inference for Variational Autoencoders

Generative machine learning models such as Variational Autoencoders (VAE) and Generative Adversarial Networks (GAN) infer rules about the distribution of training data to generate new images, tables, or numeric datasets that follow the training data distribution. The decision whether to use GAN or VAE depends on the learning task and dataset. However, similar to machine learning models for classification [Car+19; FJR15; NSH19; Sho+17; ZLH19] trained generative models leak information about individual training data records [Che+20; Hay+19; HHB19]. Anonymization of the training data or a training optimizer with differential privacy (DP) can reduce such leakage by limiting the privacy loss that an individual in the training would encounter when contributing their data [Aba+16; Ber+21; JE19]. Depending on the privacy parameter ϵ differential privacy has a significant impact on the accuracy of the generative model since the perturbation affects how closely generated samples follow the training data distribution. Balancing privacy and accuracy for differentially private generative models is a challenging task for data scientists since privacy parameter ϵ states an upper bound on the privacy loss. In contrast, quantifying the privacy loss under a concrete attack such as membership inference allows quantifying and comparing the accuracy-privacy trade-off between differentially private generative models. This chapter compares the privacy-accuracy trade-off for differentially private VAE. This is motivated by previous work that has identified VAE are more prone to membership inference attacks than GAN [HHB19]. Hence, data scientists may want to particularly consider the use of differential privacy when training VAE. In particular, we formulate an experimental study to validate whether our methodology allows identifying sweet spots w.r.t. the privacy-accuracy trade-off in

VAE. We conduct experiments for two datasets covering image and activity data, and for three different local and central differential privacy mechanisms. We make the following contributions.

- Quantifying the privacy-accuracy trade-off under membership inference attacks for differentially private VAE.
- Comparing local and central differential privacy w.r.t. the privacy-accuracy trade-off for image and motion data VAE.

This chapter is structured as follows. The reconstruction attack against Variational Auto Encoders is introduced in Section 6.1. We formulate our approach for quantifying and comparing the privacy-accuracy trade-off for differentially private VAE in Section 6.2. Section 6.3 introduces reference datasets and learning tasks. Section 6.4 presents the evaluation. We introduce related work in Section 6.5 and Section 6.6 provides a summary. A preliminary version of the results of this chapter was published in a master thesis that the author of this thesis proposed and supervised [Rob21].

6.1. Reconstruction Attack Against Variational Autoencoders

The reconstruction attack is solely applicable to Variational Autoencoders. During training, reconstructions $D(z)$ close to the current training data record x are rewarded. Hence, for training data more precise reconstructions of the VAE can be expected. However, the outputs $D(z)$ are not deterministic. They depend on the latent variable z which is sampled from the distribution $N\left(E_\mu(x), E_\sigma(x)\right)$ whose parameters are the output of the encoder network E. Hence, we repeat this process n times and set

$$\hat{f}_{\text{rec}}(x) = -\frac{1}{n}\sum_{i=1}^{n}\|D\left(z_i\right) - x\| \tag{6.1}$$

where z_i $(i = 1, \ldots, n)$ are samples from the distribution $N\left(E_\mu(x), E_\sigma(x)\right)$. This term is frequently used in practice as part of the loss function of VAE. One of the contributions of a paper that was published during this thesis [HHB19] is to apply this loss to the problem of membership inference. Specifically, the function $\hat{f}_{\text{rec}}(x)$ is applied in the attack types as the discriminating function $\hat{f}(x)$. This induces the reconstruction attack. Note that this attack considers the white-box MI adversary $\mathcal{A}_{\texttt{MI}}$ with access to the VAE model. The reconstruction MI attack assumes that a reconstructed training record will have a smaller reconstruction loss than a reconstructed test record and repeatedly computes the reconstruction $\hat{x} = D(z)$ for a record x by drawing the latent variable z. The mean reconstruction distance for $N = 300$ samples is then calculated with Equation (6.1). Furthermore, the reconstruction MI attack depends on the availability of a distance measure d. In this chapter, we use the generic Mean Squared Error (MSE) and the image domain-specific Structural Similarity Index Measure (SSIM) as distance measures. A record x is likely a training record in case of small mean reconstruction distances for MSE or a similarity close to 1 for SSIM.

6.2. Accuracy and Privacy for Variational Autoencoders

We compare the privacy-accuracy trade-off for differentially private VAE to support a data scientist $\mathcal{DS}$ in choosing privacy parameters ϵ. For this, we formulate a framework to quantify privacy and accuracy as well as the privacy-accuracy trade-off for differentially private VAE with local or central differential privacy. The framework is depicted in Figure 6.1. The framework first splits a dataset $\mathcal{D}$ into three distinct subsets: training data $\mathcal{D}^{train}$, validation data $\mathcal{D}^{val}$ and test data $\mathcal{D}^{test}$. The *target model* VAE is trained on $\mathcal{D}^{train}$ and optimized on $\mathcal{D}^{val}$. After training, we use the target model to generate a new dataset $\mathcal{D}^{gen}$ with the same distribution as $\mathcal{D}^{train}$. We use $\mathcal{D}^{gen}$ as input for the *target classifier*, a feedforward neural network for classification, to quantify the accuracy of the target model by the target classifier accuracy on $\mathcal{D}^{test}$. Our framework quantifies privacy using a MI adversary $\mathcal{A}_{\texttt{MI}}$ performing a MI attack (cf. Section 2.3.1). The MI attack dataset $\mathcal{D}^{atk}$ for training and evaluating the MI attack model is sampled equally from $\mathcal{D}^{train}$ and $\mathcal{D}^{test}$. We use the framework to calculate the baseline trade-off, as well as CDP and LDP trade-off. The baseline trade-off is calculated from the baseline target classifier test accuracy and the MI attack without any DP mechanism. For the CDP trade-off the target model is trained with DP-Adam (cf. Section 2.1).

The LDP trade-off can be computed in three settings to which we refer as LDP-Train, LDP-Full, and VAE-LDP. In LDP-Train a LDP mechanism is applied solely to $\mathcal{D}^{train}$,

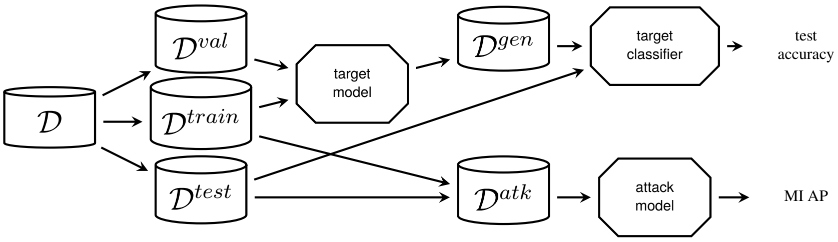


Figure 6.1.: Data flow for the framework

but not $\mathcal{D}^{val}$ and $\mathcal{D}^{test}$. This scheme is similar to Denoising Autoencoders [Vin+10]. However, we evaluated the LDP-Train setting and observed it to be mostly impractical for VAE since it introduces a transfer learning task. In particular, working on two different data distributions for $\mathcal{D}^{train}$ and $\mathcal{D}^{test}$ leads to distant latent representations and contrasting reconstructions. This neither benefits the target classifier test accuracy nor reduces MI attack performance in comparison to perturbing both training and test data. Hence, we only mention LDP-Train for sake of completeness but will not discuss LDP-Train in the rest of this chapter. In LDP-Full, $\mathcal{D}$ is perturbed and the training objective of the target model and the target classifier is changed implicitly (i.e., performance on perturbed data). VAE-LDP perturbs generated data $\mathcal{D}^{gen}$ by training a perturbation model that follows the target model architecture to enforce LDP.

The use of LDP also leads to MI attack variations. In particular, the MI attack can either be evaluated against perturbed or unperturbed records in $\mathcal{D}^{atk}$. We argue that in the LDP-Full setting the MI attack performance against unperturbed records is particularly relevant from the viewpoint of $\mathcal{DS}$, since the unperturbed records represent the actual sensitive information and otherwise the attack model would solely learn the to differentiate two distributions by the perturbation skew. Hence, within this chapter for the LDP settings, we exclusively consider the MI attack performance against unperturbed records from $\mathcal{D}^{train}$.

We evaluate the accuracy of the VAE target model based on the performance of a subsequent target classifier on $\mathcal{D}^{test}$ after training on $\mathcal{D}^{gen}$. This is a common approach to evaluate the accuracy of generative models [Fri+19; JYS19; TKP19]. To evaluate the accuracy of the MI attack we again use the Average Precision of the Precision-Recall curve (MI AP) which considers membership as sensitive information (i.e., neglecting non-membership; cf. Section 5.1). The MI AP quantifies the integral under the precision-recall curve as a weighted mean of the precision P per threshold and the increase in recall R from the previous threshold. Using the accuracy of such a curve instead of a singular value allows us to measure the MI attack performance under optimal conditions. For example, the MI adversary $\mathcal{A}_{\texttt{MI}}$ could decide to increase the assumed certainty by raising the threshold closer to 1. Independently on the target model accuracy, $\mathcal{DS}$ might be interested in lowering MI AP below a predefined threshold that is motivated by legislation (similar to the HIPAA requirement on group sizes [HS10]). We again quantify the relative trade-off between accuracy and privacy by φ which considers the relative difference between the change in test accuracy for $\mathcal{DS}$ and the change in MI AP for $\mathcal{A}_{\texttt{MI}}$.

6.3. Datasets and Learning Tasks

Within this chapter, we use two reference datasets for image and activity data.

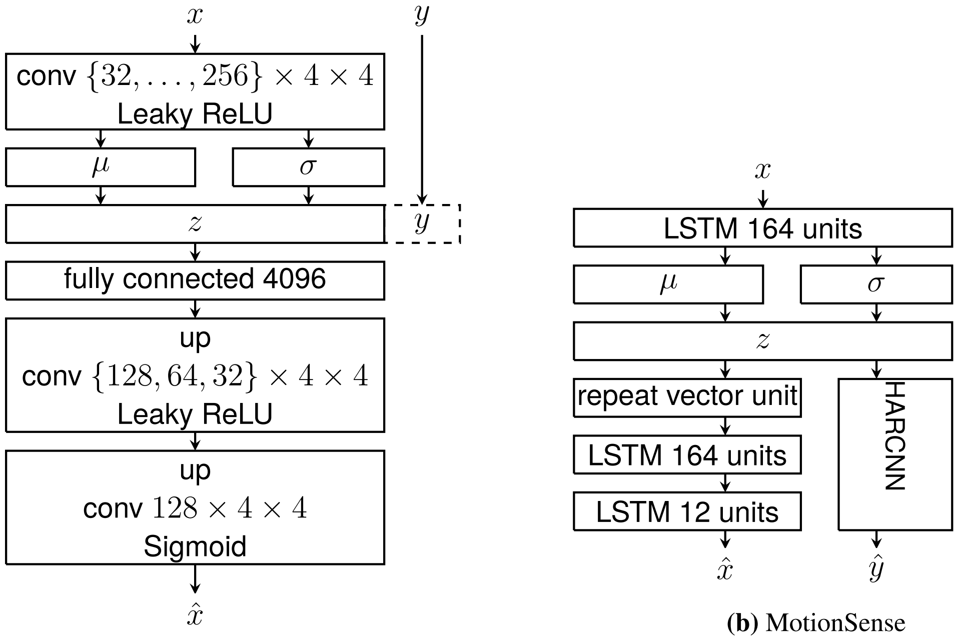
Labeled Faces in the Wild (LFW). LFW is a reference dataset for image classification that we already introduced and used in Section 5.4 in the previous chapter for a classification task. In this chapter, we pursue generative models and will limit our description to information that is relevant to the generative task in this chapter. We resize the 250×250 images to 64×64 by using a bilinear filter and normalizing pixels to $[0, 1]$ for improved accuracy. We consider the most frequent 20 and 50 labels to which we again refer as LFW20 and LFW50. 50% of the data is allocated to $\mathcal{D}^{train}$, 20% to $\mathcal{D}^{val}$, and 30% to $\mathcal{D}^{test}$. Our VAE target model is an extension of the architecture by Hou et al. [Hou+17] and is depicted in Figure 6.2a. E consists of four convolutional layers with 4×4 kernels, a stride of two, and Leaky ReLU as an activation function. D comprises a dense layer followed by four convolutional layers with 3×3 kernels, a stride of one and Leaky ReLU as an activation function. Before each convolutional layer, we perform upsampling on a scale of two with the nearest neighbor method. New data is generated by randomly drawing z from a multivariate Gaussian distribution which is passed through the decoder to create a new record. The target classifier is built upon a pre-trained VGG-Very-Deep-16 (VGG16) model [SZ15]. The first part of VGG16 consists of multiple blocks of convolutional layers and max-pooling layers for feature extraction. The second part of VGG16 is a fully-connected network for classification. After loading the pre-trained weights[1] we keep the convolutional core and train the classification part.

MotionSense (MS). MS is a reference dataset for human activity recognition with 70610 accelerometer and gyroscope sensor measurements [Mal+18]. Each measurement consists of twelve datapoints. Measurements are labeled with activities such as walking downstairs, jogging, and sitting. The associated learning task is to label a time series of measurements collected at 50 Hz with the corresponding activity. The VAE target model shall reconstruct such a time series. We normalize the data to $[-1, 1]$ and group the measurements into series of 10 seconds. 10% of the data is allocated to $\mathcal{D}^{train}$ and $\mathcal{D}^{val}$ each, and the remaining 80% is allocated to $\mathcal{D}^{test}$. Using 10% of data for training is in line with previous work on MI against generative models [Che+20; Hay+19; HHB19]. For the target model, we use a multitask approach in which E consists of a simple LSTM layer with 164 cells followed by two dense layers for μ and σ. D starts with a repeat vector unit for z. This allows us to create sequences and pass z to an LSTM layer. Furthermore, a second LSTM layer with twelve units is used to output sequences for each sensor. To support the reconstruction task we input μ to a classifier. Figure 6.2b shows the target model architecture. New data is generated by passing training records of a given label through E to obtain z, and subsequently passing z through D. We have to sample z from the label-specific latent distribution since the latent space is clustered as a consequence of the multitask classifier. The overall loss is balanced with $\lambda_1 = 0.01$, $\lambda_2 = 50$, $\lambda_3 = 0.5$ for KL-loss, reconstruction loss and classifier loss respectively. The target classifier is based on the Human Activity Recognition Convolutional Neural Network (HARCNN) architecture for time series data by Saeed [Sae16]. In HARCNN each convolutional layer is followed by a dropout layer which we set to 0.3 to learn a more general representation of the data. The final two fully-connected layers are used for classification.

6.4. Evaluation

Instead of comparing privacy parameter ϵ we designed and performed an experiment to compare the privacy-accuracy trade-off in different DP settings. The experiment is somewhat similar to the Experiment performed in Section 5.5 for discriminative models. quantifies the target classifier test accuracy and MI AP by using the framework depicted

[1] https://github.com/rcmalli/keras-vggface



(a) LFW

(b) MotionSense

Figure 6.2.: VAE target model architectures

in Figure 6.1 (cf. Section 6.2). We discuss the experiment for each dataset in four parts. First, we state the *baseline* test accuracy of the target classifier on non-generated data to provide information on the general drop in test accuracy between generated and non-generated data. Second and Third, we discuss CDP and LDP results. Fourth, the results for VAE-LDP are presented. For CDP, LDP, and VAE-LDP the experiment results are depicted in two figures each, stating target classifier accuracy over ϵ and MI AP over ϵ. In each figure, we also state the original target classifier test accuracy and MI AP for unperturbed data.

6.4.1. Setup

For each dataset, the target model is trained for 1000 epochs after which the target model test loss did not decrease significantly while the target classifier accuracy did not increase anymore. The target classifier is trained on generated samples from the VAE until the target classifier test data loss is stagnating (i.e., early stopping). This experiment design avoids overfitting and increases the real-world relevance of our results. For CDP we again use DP-Adam which samples noise from a Gaussian distribution (cf. Definition 2.2) with scale $\sigma =$ noise multiplier $z \times$ clipping norm $\mathtt{C}$. We use the heuristic of Abadi et al. [Aba+16] and set $\mathtt{C}$ as the median of norms of the unclipped gradients throughout 100 training epochs. We evaluate increasing CDP noise regimes for the target model by evaluating noise multipliers $z \in \{0.001, 0.01, 0.1, 0.5, 1\}$ and state the resulting ϵ values. The noise levels cover a wide range from baseline accuracy to naive majority vote. Similar to related work and previous experiments in this thesis we set $\delta = \frac{1}{|\mathcal{D}|}$ in our experiments [Aba+16; Ber+21]. Due to the varying LDP mechanisms we again state the privacy parameter ϵ_i for a single mechanism execution for feature i per dataset in the next sections. VAE-LDP perturbation models are trained with various noise bounds $\sigma \in \{0.1, 1, 10, 100, 1000\}$. The overall (ϵ, δ) values for CDP, LDP, and VAE-LDP are presented in Table A.4 in Appendix A.4. For the MI attack, we randomly draw 1000 records both from $\mathcal{D}^{train}$ and $\mathcal{D}^{test}$ for $\mathcal{D}^{atk}$. The experiments were run on Amazon Web Services Elastic Compute Cloud instances of type “p2.xlarge”[2] with 64 GiB RAM. This instance type is optimized for GPU computing. We implemented[3] our experiments in Python 3.8 and use TensorFlow Privacy[4]. We identify hyperparameter values for batch size, epochs, and learning rate for all target classifiers with Bayesian optimization and provide an overview of the used parameters in Table A.3 in Appendix A.4.

6.4.2. LFW

On non-generated baseline images, the target classifier achieves baseline test accuracies of 0.78 and 0.66 for LFW20 and LFW50. For generated images, we provide two accuracy metrics. Namely, the SSIM of the images generated by the target model and the test accuracy of the target classifier. Figure 6.3a states the accuracy metrics for unperturbed and CDP perturbed VAE. The figure illustrates that the unperturbed VAE does not generate images in close proximity to the baseline images. However, the images still suffice to produce target classifier test accuracies well above majority voting. Shapes of the head, hair, and some facial expressions as well as the background can be observed for reconstructed images in Figure A.6 in Appendix A.4. We also use SSIM as a domain-specific distance metric for the reconstruction MI attack. Figure 6.3b illustrates that the

[2] `https://aws.amazon.com/ec2`

[3] We provide code: `https://github.com/SAP-samples/security-research-vae-dp-mia`

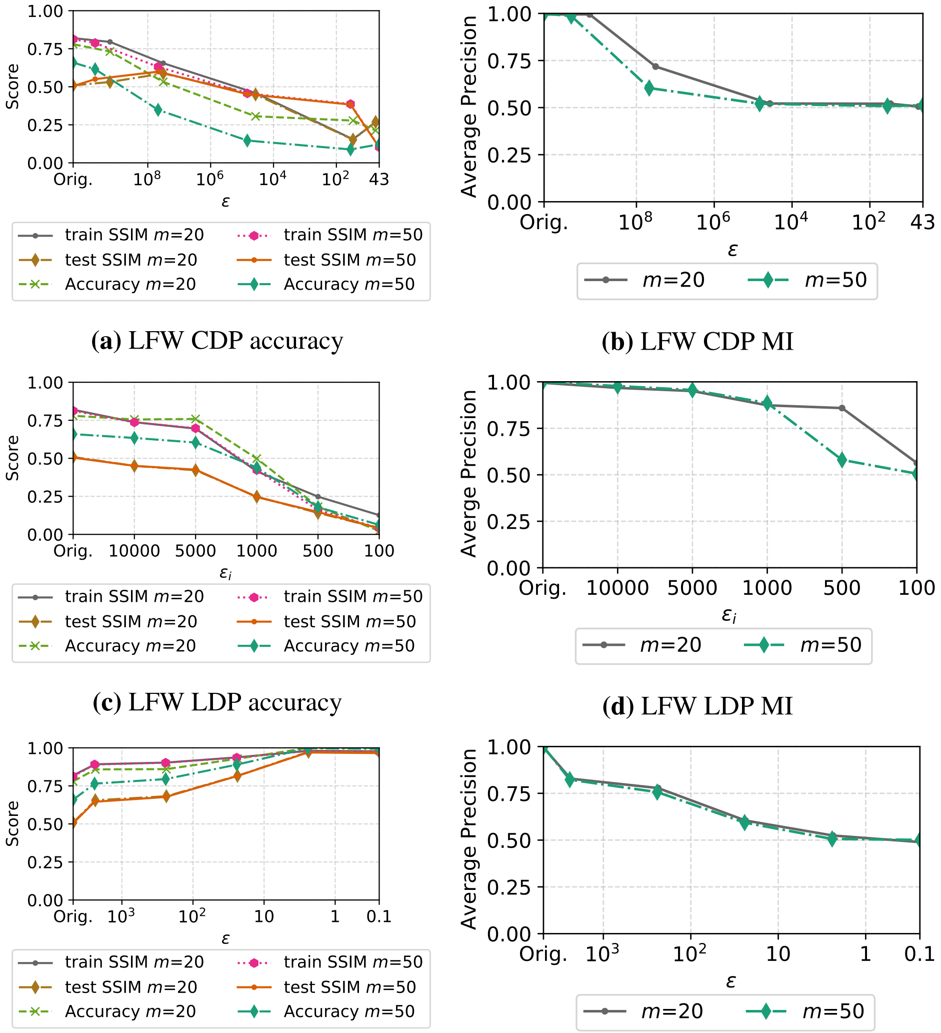
[4] `https://github.com/tensorflow/privacy`

reconstruction MI attack yields a perfect MI AP of 1 for unperturbed VAE. This high MI AP is due to the large gap between train and test SSIM.

Figure 6.3a states CDP test accuracy over ϵ. The steady accuracy decrease is due to the closing target model train-test gap, which we state in Table A.4 in Appendix A.4. The resulting regularization also lowers the SSIM of the generated images. A particular sharp drop in SSIM is observable for $z = 0.5$ ($\epsilon \approx 350$). For this datapoint posterior collapse occurs when E produces noisy μ and σ leading to unstable latent codes z which in turn are ignored by D. In consequence, D produces reconstructions independently of z leading to an increased reconstruction loss, while μ and σ become constant and minimize the KL-loss [Luc+19]. As a consequence, the target classifier resorts to a majority vote. The CDP MI AP over ϵ is stated in Figure 6.3b. The increased regularization caused by CDP is at the same time lowering MI AP. In addition, due to the inherent label imbalance in LFW, the VAE reconstruction of loosely populated labels is worse than the reconstruction for labels with more records. Still, the resulting privacy-accuracy trade-off leaves space for compromise. When $\mathcal{DS}$ would for example be willing to accept an MI AP of up to 0.6 this would require setting $z \leq 0.1$ ($\epsilon \approx 10^5$). $z = 0.1$ leads to target classifier test accuracy of 0.31. However, if $\mathcal{DS}$ its their threshold to 0.75 this would allow for $z = 0.01$ ($\epsilon \approx 10^8$) and a target classifier test accuracy of 0.52.

For LDP we use differentially private image pixelization (cf. Section 2.2) to create LDP training and test datasets within neighborhood $o = \sqrt{64 \times 64}$. Figure 6.3c presents the LDP test accuracy and SSIM over ϵ_i. In contrast to the CDP experiments the target classifier test accuracy and the target model SSIM do not show a regularization effect caused by the introduced noise for LDP. The train-test gap narrows only slightly and the random noise introduced in the dataset makes the reconstruction task for the VAE more difficult. Thus, the reconstruction attack MI AP in Figure 6.3d remains nearly unchanged until $\epsilon_i <= 500$ at which point the target model SSIM and the target classifier test accuracy are already at poor levels and little room for compromise is existing.

VAE-LDP accuracy over ϵ is presented in Figure 6.3e. Counterintuitively, the test accuracy even rises over ϵ and the train-test gap and SSIM gap narrow. This is due to the VAE-LDP perturbation model which reconstructs only essential facial features and leaves the background grey when faced with small ϵ. Hence the learning task for the target classifier and the reconstruction task for the VAE are simplified. Figure A.6 in Appendix A.4 underlines this observation by showing the same image for VAE-LDP



(a) LFW CDP accuracy

(b) LFW CDP MI

(c) LFW LDP accuracy

(d) LFW LDP MI

(e) LFW VAE-LDP accuracy

(f) LFW VAE-LDP MI

Figure 6.3.: LFW accuracy and privacy

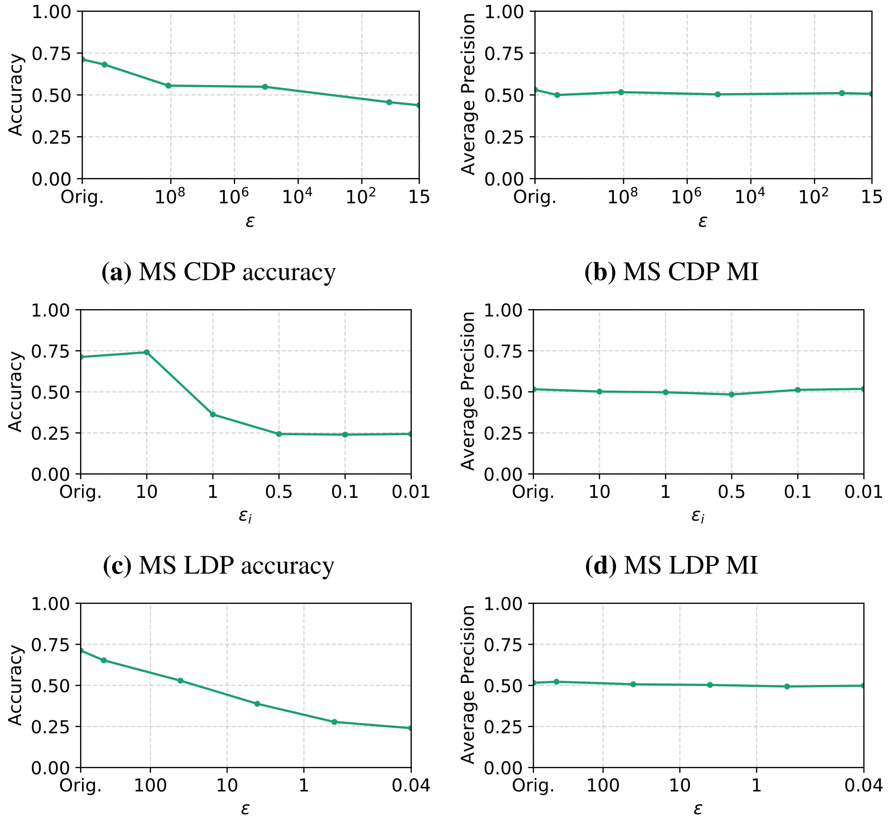
with increasing noise. The reconstruction attack against VAE-LDP in Figure 6.3f also decreases as the SSIM gap closes. All in all, the results point towards an advantage of the VAE-LDP mechanism over the LDP image pixelization mechanism. The main disadvantage of the VAE-LDP mechanism over image pixelization is the increased effort to optimize perturbation model hyperparameters.

6.4.3. MotionSense

Due to the absence of a domain specific accuracy metric, we solely consider test accuracy as an accuracy metric for this dataset. The target classifier for MS achieves a baseline test accuracy of 0.99 for non-generated data. Figure 6.4a states the test accuracy for original and CDP perturbed data over ϵ. The test accuracy is dropping to 0.71 for generated data, which is due to the target model being unable to reconstruct time series for all activities equally well. The reconstruction MI attack has not been used for a time series data in previous work and we suggest to use MSE as the reconstruction MI attack distance metric. The original MI attack performance is depicted in Figure 6.4b and achieves an MI AP 0.52. We see three main reasons for the low MI AP in comparison to LFW. First, MS is more balanced in comparison to LFW. Second, there are significantly more records in MS than in LFW and thus more records per label allow to learn a more general representation. Third, sensor measurements exhibit ambiguities and thus the target model tends to learn more general trends instead of absolute values.

The CDP target classifier test accuracy only slightly worsens with increasing noise as illustrated in Figure 6.4a. This is mostly due to the target classifier resorting to a majority vote for particular activities with increasing noise. Figure A.7 in Appendix A.4 shows the confusion matrix for the target classifier at $z = 1$ ($\epsilon \approx 16$). The target classifier resorts to a majority vote for labels 0 to 3 which represent different types of movements, but is still able to distinguish labels 4 and 5 which represent standing and sitting. The latter two activities are of different nature than the movements and remain distinguishable under noise. The MI AP illustrated in Figure 6.4b shows again the ineffectiveness of the reconstruction MI attack against the MS time series data.

For LDP we use the Laplace mechanism to perturb each measurement (cf. Section 2.2) and specify the sensitivity per sensor as the maximum of all corresponding observed values to create differentially private time series. Figure 6.4c shows the target classifier accuracy over ϵ_i. Notably, the target classifier test accuracy increases slightly before dropping sharply over ϵ_i. Here, small noise levels are actually positively influencing the target model training and hence also allow the target classifier to better distinguish between different labels. In general, the simple LDP mechanism used within this experiment seems to prevent the target model to infer structural information and in turn limits the reconstruction and meaningful generation of records. Figure 6.4d presents the



(a) MS CDP accuracy

(b) MS CDP MI

(c) MS LDP accuracy

(d) MS LDP MI

(e) MS VAE-LDP accuracy

(f) MS VAE-LDP MI

Figure 6.4.: MS accuracy and privacy

MI attack performance. The MI AP decreases to 0.5 already at the largest ϵ_i and remains close to the baseline for all further ϵ_i.

VAE-LDP test accuracy over ϵ is depicted in Figure 6.4e. In comparison to LFW the MS perturbation models do not focus on the essential features of the data and in turn, the target classifier cannot benefit from increased perturbation. Due to this the predictions also shift to a majority vote for label 5 and lower the test accuracy significantly. The VAE-LDP MI AP over ϵ is illustrated in Figure 6.4f. Note that at $\sigma = 0.1$ ($\epsilon \approx 40$) an outlier is present where the target model did not learn a continuous latent space and thus the reconstruction of records from $\mathcal{D}^{test}$ suffered. However, the VAE-LDP results show similar trends as the above LDP results.

6.5. Related Work

We discuss related work from three categories. First, we briefly discuss generative models and accuracy metrics for generative models. Second, we provide background on differential privacy in generative models. Third, we introduce related work on membership inference attacks against generative models.

Generative Adversarial Networks by Goodfellow et al. [Goo+14] represent an alternative to VAE. We focus on VAE since VAE in comparison to GAN were observed to be more prone to MI attacks [HHB19]. Salimans et al. [Sal+16] introduce the Inception Score to automatically evaluate the utility of sampled images from generative models. The main advantage of the Inception Score over other metrics such as SSIM is the correlation with human judgments. However, Barrat et al. [BS18] point out that the Inception Score is foremost meaningful for the ImageNet dataset due to pre-training. Therefore, we consider the test accuracy of a target classifier to evaluate the VAE accuracy.

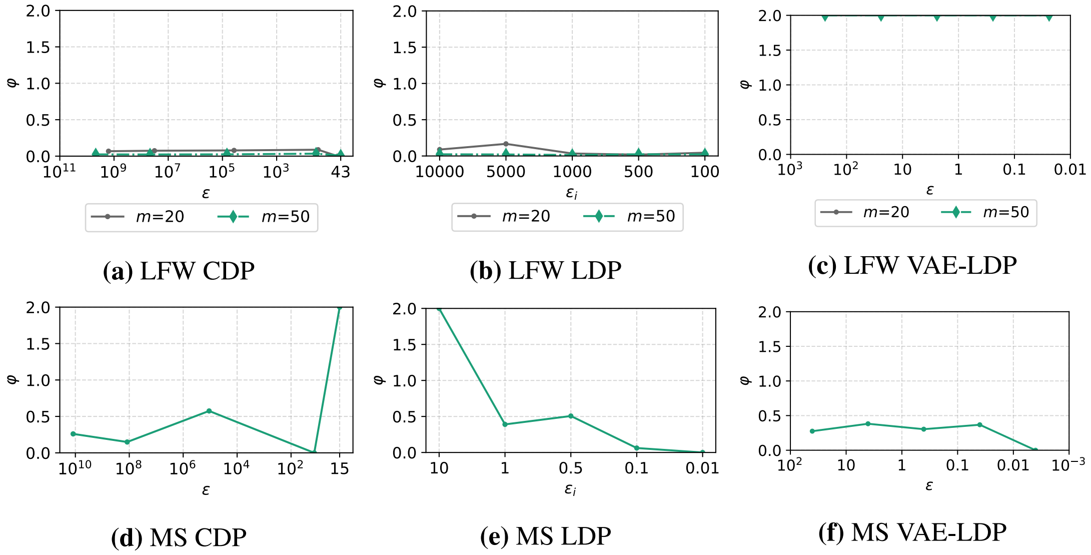
Torkzadehmahani et al. [TKP19] propose the DP-cGAN framework to generate differentially private data and labels. Similar to our work they train target classifiers on the generated data to evaluate model accuracy. We consider VAE with LDP and CDP. Jordon et al. [JYS19] extend the differentially private federated learning architecture PATE [Pap+18] to GAN. Similar to us, they analyze the accuracy of a target classifier for various privacy parameters, yet Jordon et al. do not discuss privacy aside from privacy parameter ϵ. Frigerio et al. [Fri+19] evaluate a CDP GAN for time series data also w.r.t. MI attacks. We also consider LDP and quantify the trade-off between privacy and accuracy. Takahashi [Tak+20] proposes an enhanced version of the DP-SGD for VAE by adjusting the noise that is injected to the loss terms. We use DP-Adam where their improvement is not applicable.

Hayes et al. [Hay+19] propose the LOGAN framework for MI attacks against GAN under various assumptions for the knowledge of $\mathcal{A}_{\mathtt{MI}}$. For their black-box attacks, they train a separate discriminator model to distinguish between members and non-members. In contrast, we consider statistical MI attack models, allowing for MI attacks against generative models without the need to train a separate attack model. Hilprecht et al. [HHB19] propose Monte-Carlo MI attacks against GAN and VAE. We use their reconstruction MI attack and are the first to consider this attack under differential privacy. Chen et al. [Che+20] extend the reconstruction MI attack to a partial black-box setting where $\mathcal{A}_{\mathtt{MI}}$ solely has access to the latent space z but not the internal parameters of the generative model. Their attack composes different losses targeting various aspects of a model and takes the reconstruction as well as the latent representation into consideration. We ran all experiments within this chapter also for their attack and the consideration of latent representation did lead to strictly weaker MI AP. The gradient matching attack of Zhu et al. [ZLH19] strives for the reconstruction of training data from publicly available gradients. In contrast, we focus on the identification of training data.

6.6. Summary

Image data yields higher MI attack performance than time-series data. The reconstruction MI attack has been shown effective for image data in prior work [Che+20; HHB19], despite being fairly simple and only takes one metric for disparate behaviour of the target model into consideration. This is in line with the identified gap in image reconstruction for LFW and the gap was exploited by using SSIM as a distance measure for the reconstruction MI attack. For MS we were not able to identify a measure that provides equal success. Since activity measurements exhibit many ambiguities the target model learns to reconstruct relative trends instead of concrete measurements that represent a specific movement. Therefore, the target model generalizes more and is less prone to MI attacks. Additionally, previous research [NSH19; Sho+17] has shown that large datasets with few labels are generally less vulnerable to MI attacks.

Small noise yields a favorable relative privacy-accuracy trade-off for image data. For CDP and image data we recommend using as little noise as possible. The relative accuracy drop for $\mathcal{DS}$ largely exceeds the performance loss for $\mathcal{A}_{\mathtt{MI}}$ throughout the CDP experiments for LFW. This trend is illustrated in Figure 6.5a which highlights that the drop in target classifier test accuracy is always larger than the privacy gain by reduced MI AP. For MS the reconstruction MI attack only achieves a performance close to random guessing already against original data. Hence, small DP noise is already sufficient to push the MI AP to random guessing. This is reflected in Figure 6.5d, where we see an optimal φ already for $z = 0.001$. Similarly for LDP Figures 6.5b and 6.5e show only a few favorable φ for both datasets. These few favorable trade-offs again indicate that differentially private image pixelization and the Laplace mechanism disproportionately



(a) LFW CDP **(b)** LFW LDP **(c)** LFW VAE-LDP

(d) MS CDP **(e)** MS LDP **(f)** MS VAE-LDP

Figure 6.5.: φ for CDP, LDP and VAE-LDP, for LFW and MotionSense

harm model accuracy over protecting privacy. Compared to CDP, LDP shows better trade-offs for small privacy parameter. However, $\mathcal{DS}$ generally gives up more accuracy compared to the gain in privacy.

VAE-LDP outperforms LDP and CDP w.r.t. the relative privacy-accuracy trade-off. In our experiments, the VAE-LDP yielded the best trade-off between target classifier test accuracy and MI AP. This finding is supported by φ depicted in Figures 6.5c and 6.5f. We identified the interaction between the perturbation models, that retain essential image features, and the targeted classification task as the primary reason for the superior trade-off. φ for the VAE-LDP experiments highlight that small noise bounds are protecting from the reconstruction MI attack. For larger noise bounds however, φ only offers limited informative value since the MI AP pivots around random guessing while the target classifier test accuracy is bound by the overall classification baseline.

VAE are highly susceptible to noise introduced during training. Our results indicate that CDP leads to a regularization effect and directly addresses a key driver for MI AP. However, CDP also required additional hyperparameter optimization and increases computational cost. LDP mechanisms consume information within the data to foster protection and hence the test accuracy decreases heavily depend on how the LDP mechanism alters the training data. For example, differentially private image pixelization damages the structures of images to preserve privacy. The more information consumed by the LDP mechanism, the worse the target classifier test accuracy becomes. This effect is clearly visible in the MS dataset, where the decrease in target classifier accuracy is similar to the overall classification baseline. When this characteristic is present MI is affected mostly as a consequence of diminishing model performance. This is facilitated by the lack of regularization effect which keeps a present relative gap for the MI attacks to exploit. The VAE-LDP mechanism preserves essential features of the LFW dataset during perturbation. The preservation of essential features are beneficial to the overall classification task as the test accuracy remains high while the MI AP decreases.

In summary, within this chapter, we addressed Problem 1.(iii) by formulating a validation framework for quantifying the relative privacy-accuracy trade-off for VAE. We used the framework to evaluate and compare two LDP and one CDP mechanism for image and time series data w.r.t. their privacy-accuracy trade-off. In particular, the LFW image recognition dataset was very susceptible to the reconstruction MI attack whereas the MotionSense activity recognition dataset with more records and fewer labels was mostly resistant to MI. The CDP mechanism offered a more consistent decrease in MI attack performance whereas the LDP mechanisms showed varying levels of protection depending on chosen privacy parameter and setting. The relative privacy-accuracy trade-off highlights that protection often comes at a disproportionately high accuracy cost.

7. Quantifying Identifiability to Choose and Audit ϵ

Several privacy regulations [HS10; PE16] consider individual identifiability to gauge anonymization strength. Therefore, scores that quantify reidentification risk to individuals can strongly affect the widespread implementation of anonymization techniques [Nis16]. In consequence, if DP shall be used to comply with privacy regulations and find widespread adoption [NW18; Par14], quantifying the resulting identifiability from privacy parameters (ϵ, δ) is required [CT13; NW18]. Multiple approaches for choosing privacy parameters have been introduced, yet they do not reflect identifiability [AS19; Hsu+14], part from the original DP definition [Ber+21; LC12; Rah+18; Yeo+18], or lack applicability to common DP mechanisms for ML [LC11]. Especially in ML, practical membership inference attacks have been used to measure identifiability [Ber+21; Che+20; Hay+19; JE19; Jay+20; Rah+18; Sho+17; Yeo+18]. However, $\mathcal{A}_{\texttt{MI}}$ is not assumed to have auxiliary information about the members of datasets that they aim to differentiate, which DP adversaries are assumed to possess. MI attacks thus offer intuition about the outcome of practical attacks; nonetheless, bounds on MI attacks in terms of ϵ are not tight [JE19], and consequently MI can only represent an empirical lower bound on identifiability.

Rather than analyzing $\mathcal{A}_{\texttt{MI}}$, we consider a DP adversary with arbitrary auxiliary knowledge and derive maximum *Bayesian posterior belief* ρ_β as an identifiability bound related to (ϵ, δ), which bounds the adversary's certainty in identifying a member of the training data. Furthermore, we define the complementary score *expected membership advantage* ρ_α, which is related to the probability of success in a Bernoulli trial over the posterior beliefs. ρ_α depends on the entire distribution of observed posterior beliefs, not solely the worst-case posterior belief, and allows direct comparison with the membership advantage bound of Yeom et al. [Yeo+18] for $\mathcal{A}_{\texttt{MI}}$. We will show that the DP adversary achieves a greater membership advantage than $\mathcal{A}_{\texttt{MI}}$, implying that while both adversaries can be used to evaluate the protection of DP in machine learning, our implementable instance of the DP adversary comes closer to DP bounds.

A subsequent question is whether our identifiability bounds are tight in practice since the factual guarantee (ϵ, δ) depends on the difference between possible input datasets [NRS07]. In differentially private stochastic gradient descent, noise is scaled to global sensitivity, the maximum change that any single record in the training dataset is assumed to cause on the gradient during any training step. However, since all training data records are likely to be within the same domain (e.g., pictures of cars vs. pictures of nature scenes), global sensitivity might far exceed the difference between gradients over all training steps. We propose scaling the sensitivity to the difference between the gradients of a fixed dataset and any neighboring dataset and show for three reference datasets that we can indeed achieve tight bounds. Our main contributions are:

- Identifiability bounds for the posterior belief and expected membership advantage that are mathematical transformations of privacy parameters (ϵ, δ) and used in conjunction with RDP composition.

- The practical implementation of an adversary that meets all assumptions on worst-case adversaries against DP and allows us to audit DPSGD model instances w.r.t. to the empirical privacy loss besides enabling comparison with membership inference adversaries.

- A heuristic for scaling sensitivity in differentially private stochastic gradient descent. This heuristic leads to tight bounds on identifiability.

This chapter is structured as follows. We discuss the relation of differential identifiability to differential privacy in Section 7.1. In Sections 7.2 and 7.3 we formulate identifiability scores and provide upper bounds on them. Section 7.4 specifies the application of these scores for a deep learning scenario, and we evaluate the scores for three deep learning reference datasets in Section 7.5. We present related work in Section 7.6. Section 7.7 discusses the practical relevance of our findings and summarizes the chapter.

7.1. Differential Identifiability and the Relation to the DP Adversary

Lee et al. [LC11; LC12] introduce differential identifiability (DI) as a strong inference threat model. DI assumes that the adversary calculates the likelihood of all possible input datasets, so-called *possible worlds* in a set Ψ, given a mechanism output r. Li et al. [Li+13] show that the DI threat model maps to the worst-case against which bounded DP protects when $|\Psi| = 2$, since DP considers two neighboring datasets $\mathcal{D}$, $\mathcal{D}'$ by definition. The DI experiment $\mathrm{Exp}^{\mathrm{DI}}$ is similar to $\mathrm{Exp}^{\mathrm{MI}}$ (cf. Experiment 2.1) since the adversary must decide whether the dataset contains the member that differs between the known $\mathcal{D}'$ and $\mathcal{D}$, or not. For comparison, we reformulate DI as a cryptographic experiment in Experiment 7.1.

Experiment 7.1. *(Differential Identifiability* $\mathrm{Exp}^{\mathrm{DI}}$*) Let* $\mathcal{A}_{\mathtt{DI}}$ *be an adversary,* $\mathcal{M}$ *be a differentially private learning algorithm,* $\mathcal{D}$ *and* $\mathcal{D}'$ *be neighboring datasets drawn mutually independently from distribution* `Dist`*, using either bounded or unbounded definitions. The differential identifiability experiment* $\mathrm{Exp}^{\mathrm{DI}}$ *proceeds as follows:*

1. *Set* $\mathbf{r}_D := \mathcal{M}(\mathcal{D})$ *and* $\mathbf{r}_{\mathcal{D}'} := \mathcal{M}(\mathcal{D}')$
2. *Choose* $b \leftarrow \{0, 1\}$ *uniformly at random*
3. *Let*
$$\mathbf{r} = \begin{cases} \mathbf{r}_{\mathcal{D}}, & \text{if } b = 1 \\ \mathbf{r}_{\mathcal{D}'}, & \text{if } b = 0 \end{cases}$$
4. $\mathcal{A}_{\mathtt{DI}}$ *outputs* $b' = \mathcal{A}_{\mathtt{DI}}(\mathbf{r}, \mathcal{D}, \mathcal{D}', \mathcal{M}, \mathtt{Dist}) \in \{0, 1\}$*. If* $b' = b$*,* $\mathcal{A}_{\mathtt{DI}}$ *succeeds and the output of the experiment is 1. It is 0 otherwise.*

Since Experiment 7.1 precisely defines an adversary with access to arbitrary background knowledge of up to all but one record in $\mathcal{D}$ and $\mathcal{D}'$, $\mathcal{A}_{\mathtt{DI}}$ is an implementable instance of the DP adversary [DR16]. Compared to the MI adversary, the DI adversary is stronger, since $\mathcal{A}_{\mathtt{DI}}$ knows the alternative dataset $\mathcal{D}'$ instead of only the distribution `Dist` from which $\mathcal{D}'$ was chosen. The experiment defined above is general and applies to deep learning using gradient descent as follows: the knowledge of the mechanism $\mathcal{M}$

implies knowledge about the architecture of the NN and the learning parameters $\eta, \mathtt{C}$, as well as the number of iterations k. The experiment is formulated s.t. it could be applied for a single iteration, and the output $\mathbf{r}$ of the mechanism is the perturbed gradient $\tilde{g}_i$ from iteration i of the NN training. However, after the entire learning process, consisting of k iterations, $\mathcal{A}_{\mathtt{DI}}$ has more information $R_k = (\mathbf{r}_0, \mathbf{r}_1, \ldots, \mathbf{r}_k)$ and therefore a higher chance to win Experiment 7.1. In this case, the same value of b is chosen in every round, since the training data is kept constant over all learning steps. This is the standard case considered in this chapter and motivates the need for composition theorems. According to Experiment 7.1, the DI adversary could know almost all of the training data from a public dataset of census data, for example, and observe the NN gradient updates at every training step. The assumption that $\mathcal{A}_{\mathtt{DI}}$ has access to all gradients during learning may seem overly strong; however, this setting is of theoretical interest, since the bounds that we prove for the DI adversary $\mathcal{A}_{\mathtt{DI}}$ will also hold for weaker adversaries. Furthermore, the assumptions can be fulfilled in federated learning, for example. In federated learning, multiple data owners jointly train a global model by sharing gradients for their individual subsets of training data with a central aggregator. The aggregator combines the gradients and shares the aggregated update with all data owners. If $\mathcal{A}_{\mathtt{DI}}$ participates as a data owner, $\mathcal{A}_{\mathtt{DI}}$ can observe the joint model updates.

7.2. Identifiability Scores for DP

We will prove in Section 7.2.1 that if $\mathrm{Adv}^{\mathrm{DI}}$ is bounded, then this bound also holds for $\mathrm{Adv}^{\mathrm{MI}}$. Equivalently, we prove that $\mathcal{A}_{\mathtt{DI}}$ is stronger than $\mathcal{A}_{\mathtt{MI}}$ due to additional available auxiliary information. In addition, we formulate two scores for the identifiability of individual training records when releasing a differentially private NN. The scores are compatible with DP under multidimensional queries and composition. First, we define *posterior belief* β, which quantifies identifiability for iterative mechanisms in Section 7.2.2. Second, we define *membership advantage* $\mathrm{Adv}^{\mathrm{DI}}$ for $\mathcal{A}_{\mathtt{DI}}$ in Section 7.2.3, which is a complementary identifiability score offering a scaled quantification of the adversary's probability of success.

7.2.1. Relation of Membership Inference and Differential Identifiability

$\mathcal{A}_{\texttt{DI}}$ knows both neighboring datasets $\mathcal{D}$ and $\mathcal{D}'$ instead of only receiving one value d and the size n of the dataset from which the datapoints are drawn.

Proposition 7.1. *DI implies MI: if* $\mathcal{A}_{\texttt{MI}}$ *wins* $\mathrm{Exp}^{\mathrm{MI}}$*, then one can construct* $\mathcal{A}_{\texttt{DI}}$ *that wins* $\mathrm{Exp}^{\mathrm{DI}}$.

Proof. We prove the proposition by contradiction: assume that the mechanism $\mathcal{M}$ successfully protects against $\mathcal{A}_{\texttt{DI}}$, but that there exists an adversary $\mathcal{A}_{\texttt{MI}}$ that wins $\mathrm{Exp}^{\mathrm{MI}}$. Again, we assume w.l.o.g. that $\mathcal{D} \setminus \mathcal{D}' \neq \{\}$. We construct an adversary $\mathcal{A}_{\texttt{DI}}$ that also wins $\mathrm{Exp}^{\mathrm{DI}}$ as follows:

1. On inputs $\mathcal{D}, \mathcal{D}', \mathcal{M}, \mathbf{r}, \texttt{Dist}$, $\mathcal{A}_{\texttt{DI}}$ calculates $n = |\mathcal{D}|$ and let $d = \mathcal{D} \setminus \mathcal{D}'$.
2. $\mathcal{A}_{\texttt{DI}}$ gives $(d, \mathbf{r}, n, \texttt{Dist})$ to $\mathcal{A}_{\texttt{MI}}$.
3. $\mathcal{A}_{\texttt{MI}}$ gives $b'' = \mathcal{A}_{\texttt{MI}}(d, \mathbf{r}, n, \texttt{Dist})$ to $\mathcal{A}_{\texttt{DI}}$ in response.
4. $\mathcal{A}_{\texttt{DI}}$ outputs $b' = b''$.

By the definition of $\mathrm{Exp}^{\mathrm{DI}}$, $\mathcal{A}_{\texttt{DI}}$ wins if $b' = b$, and thus succeeds in the following cases:
Case 1: $b = 1$, which means $\mathbf{r} = \mathcal{M}(\mathcal{D})$. Since $d \in \mathcal{D}$, this is exactly the case where $\mathcal{A}_{\texttt{MI}}$ correctly outputs $b'' = 1$. Therefore $b' = b$.
Case 2: $b = 0$, which means $\mathbf{r} = \mathcal{M}(\mathcal{D}')$. Since $d \notin \mathcal{D}'$, this is exactly the case where $\mathcal{A}_{\texttt{MI}}$ correctly outputs $b'' = 0$. Therefore $b' = b$. For both cases $\mathcal{A}_{\texttt{DI}}$ wins ($b' = b$), which contradicts the assumption that the mechanism $\mathcal{M}$ successfully protects against $\mathcal{A}_{\texttt{DI}}$. It is at least as difficult for a mechanism to protect against $\mathrm{Exp}^{\mathrm{DI}}$ as against $\mathrm{Exp}^{\mathrm{MI}}$, which is equivalent to the statement that if $\mathcal{A}_{\texttt{MI}}$ wins $\mathrm{Exp}^{\mathrm{MI}}$, then $\mathcal{A}_{\texttt{DI}}$ wins $\mathrm{Exp}^{\mathrm{DI}}$ as well. □

7.2.2. Posterior Belief in Identifying the Training Dataset

To quantify individual identifiability from privacy parameters (ϵ, δ), we use the Bayesian posterior belief. After having observed gradients R_k, the adversary $\mathcal{A}_{\texttt{DI}}$ can update the probabilities for both the training dataset $\mathcal{D}$ and the alternate dataset $\mathcal{D}'$, which differs from $\mathcal{D}$ in an individual record $d = \mathcal{D} \setminus \mathcal{D}'$. The posterior belief quantifies the certainty with which $\mathcal{A}_{\mathtt{DI}}$ can identify the training dataset used by a NN and consequently the presence of the individual record d. This belief is formulated as a conditional probability depending on observations R_k during training in Definition 7.1. For a census dataset such as Adult, the posterior belief measures the probability that a particular individual d participated in the census after observing training using data $\mathcal{D}$. Since this belief has an upper bound for each possible member d of the dataset, no member of $\mathcal{D}$ can be identified. Posterior belief, therefore, relates theoretical DP privacy guarantees to privacy regulations and societal norms through its identifiability formulation, since the noise, and therefore the posterior belief, depends on (ϵ, δ).

Definition 7.1 (Posterior Belief). Consider the setting of Experiment 7.1 and denote $R_k = (\mathbf{r}_0, \mathbf{r}_1, \ldots, \mathbf{r}_k)$ as the result matrix, comprising k multidimensional mechanism results. The posterior belief in the correct dataset $\mathcal{D}$ is defined as the probability conditioned on all the information observed during the adaptive computations

$$\beta_k := \Pr(\mathcal{D}|R_k) = \frac{\Pr(\mathcal{D}, R_k)}{\Pr(\mathcal{D}, R_k) + \Pr(\mathcal{D}', R_k)}$$

where the probability $\Pr\left(\mathcal{D}|R_k\right)$ is over the random iterative choices of the mechanisms up to step k. ◇

Each β_k can be computed from the previous β_{k-1}. The final belief can be computed using Lemma 7.1, which we will use to further analyze the strongest possible adversary $\mathcal{A}_{\mathtt{DI}}$ of Experiment 7.1.

Lemma 7.1 (Calculation of the posterior belief). Assuming uniform priors and independent mechanisms $\mathcal{M}_i$ (more precisely, the noise of the mechanisms must be sampled independently), the posterior belief on dataset $\mathcal{D}$ can be computed as

$$\begin{aligned}\beta_k &= \frac{\prod_{i=1}^{k} \Pr(\mathcal{M}_i(\mathcal{D}) = \mathbf{r}_i)}{\prod_{i=1}^{k} \Pr(\mathcal{M}_i(\mathcal{D}) = \mathbf{r}_i) + \prod_{i=1}^{k} \Pr(\mathcal{M}_i(\mathcal{D}') = \mathbf{r}_i)} \\ &= \frac{1}{1 + \frac{\prod_{i=1}^{k} \Pr(\mathcal{M}_i(\mathcal{D}') = \mathbf{r}_i)}{\prod_{i=1}^{k} \Pr(\mathcal{M}_i(\mathcal{D}) = \mathbf{r}_i)}}\end{aligned}$$

Proof. We prove the lemma by iteration over k.
$k = 1$: We assume the attacker starts with uniform priors $\Pr(\mathcal{D}) = \Pr(\mathcal{D}') = \frac{1}{2}$. Thus, $\beta_1(\mathcal{D}|R_1)$ can be directly calculated by dividing both numerator and denominator of β by the numerator:

$$\begin{aligned}\beta_1(\mathcal{D}|R_1) &= \frac{\Pr(\mathcal{M}_1(\mathcal{D}) = \mathbf{r}_1)}{\Pr(\mathcal{M}_1(\mathcal{D}) = \mathbf{r}_1) + \Pr(\mathcal{M}_1(\mathcal{D}') = \mathbf{r}_1)} \\ &= \frac{1}{1 + \frac{\Pr(\mathcal{M}_1(\mathcal{D}')=\mathbf{r}_1)}{\Pr(\mathcal{M}_1(\mathcal{D})=\mathbf{r}_1)}}\end{aligned}$$

$k-1 \to k$: In the second step $\beta_{k-1}(\mathcal{D}|R_{k-1})$ is used as the prior, using the shorthand notations $\beta_k := \beta_k(\mathcal{D}|R_k)$, and in the last step $p_k := \Pr(\mathcal{M}_k(\mathcal{D}) = \mathbf{r}_k)$ and $p'_k := \Pr(\mathcal{M}_k(\mathcal{D}') = \mathbf{r}_k)$ the calculation of $\beta_k(\mathcal{D}|R_k)$ starts as for the induction start $k = 1$

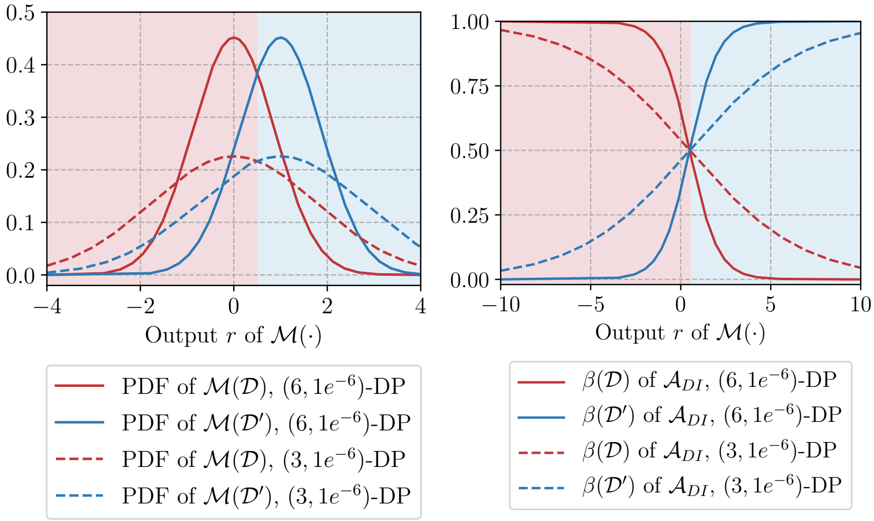
$$\begin{aligned}\beta_k &= \frac{\Pr(\mathcal{M}_k(\mathcal{D}) = \mathbf{r}_k) \cdot \beta_{k-1}}{\Pr(\mathcal{M}_k(\mathcal{D}) = \mathbf{r}_k) \cdot \beta_{k-1} + \Pr(\mathcal{M}_k(\mathcal{D}') = \mathbf{r}_k) \cdot (1 - \beta_{k-1})} \\ &= \frac{1}{1 + \frac{\Pr(\mathcal{M}_k(\mathcal{D}')=\mathbf{r}_k) - \Pr(\mathcal{M}_k(\mathcal{D}')=\mathbf{r}_k)\cdot\beta_{k-1}}{\Pr(\mathcal{M}_k(\mathcal{D})=\mathbf{r}_k)\cdot\beta_{k-1}}} = \frac{1}{1 + \frac{p'_k - p'_k\beta_{k-1}}{p_k\beta_{k-1}}}\end{aligned}$$

Now the induction assumption can be substituted for the right term of the denominator and then multiplying the numerator and denominator with $\prod_{i=1}^{k-1} p_i + \prod_{i=1}^{k-1} p'_i$ leads to

$$\begin{aligned}\frac{p'_k - p'_k\beta_{k-1}}{p_k\beta_{k-1}} &= \frac{p'_k - p'_k \frac{\prod_{i=1}^{k-1} p_i}{\prod_{i=1}^{k-1} p_i + \prod_{i=1}^{k-1} p'_i}}{p_k \frac{\prod_{i=1}^{k-1} p_i}{\prod_{i=1}^{k-1} p_i + \prod_{i=1}^{k-1} p'_i}} \\ &= \frac{p'_k \left(\prod_{i=1}^{k-1} p_i + \prod_{i=1}^{k-1} p'_i\right) - p'_k \prod_{i=1}^{k-1} p_i}{p_k \prod_{i=1}^{k-1} p_i} \\ &= \frac{\prod_{i=1}^{k} p'_i}{\prod_{i=1}^{k} p_i}\end{aligned}$$

where in the last step the first and the third term in the denominator cancel and can be inserted back into the last form of β_k above. □

In our analysis, $\mathcal{A}_{\mathtt{DI}}$ is a binary classifier that chooses the label with the highest posterior probability β_k. If prior beliefs are uniform, this decision process can be simplified. Consider $X_1 := \mathcal{M}(\mathcal{D})$ and $X_0 := \mathcal{M}(\mathcal{D}')$. Since $\mathcal{A}_{\mathtt{DI}}$ knows $\mathcal{D}, \mathcal{D}'$ and $\mathcal{M}$,



(a) Probability density functions

(b) Posterior belief

Figure 7.1.: The decision boundary of $\mathcal{A}_{\texttt{DI}}$

$\mathcal{A}_{\texttt{DI}}$ also knows the corresponding probability densities g_{X_1} and g_{X_0}. The densities are identical and defined by $\mathcal{M}$, but are centered at the different results $f(\mathcal{D})$ and $f(\mathcal{D}')$, respectively, as visualized in Figure 7.1a with $f(\mathcal{D}) = 0, f(\mathcal{D}') = 1$. When $\mathcal{A}_{\texttt{DI}}$ has equal prior beliefs, $\mathcal{A}_{\texttt{DI}}$ decides whether R_k is more likely to stem from X_1 or X_0 and therefore chooses

$$\mathcal{A}_{\texttt{DI}}(R_k, \mathcal{D}, \mathcal{D}', \mathcal{M}, \texttt{Dist}) = \underset{D \in \{\mathcal{D}, \mathcal{D}'\}}{\arg\max}\, \beta(D|R_k) = \underset{b \in \{0,1\}}{\arg\max}\, g_{X_b}(R_k) \tag{7.1}$$

$\beta(\mathcal{D})$ and $\beta(\mathcal{D}')$ for our example are visualized in Figure 7.1b. $\mathcal{A}_{\texttt{DI}}$ acts like a naive Bayes classifier whose decision is depicted by the background color. The input features are the perturbed results R_k, and the exact probability distribution of each label is known. The distributions are entirely defined by $\mathcal{D}$, $\mathcal{D}'$, and $\mathcal{M}$, so $\mathcal{A}_{\texttt{DI}}$ does not use the knowledge of `Dist`. The posterior belief quantifies the probability of R_k; however, in another instance, R_k could differ. In Section 7.3.1, we will therefore define an upper bound on $\beta(\mathcal{D})$.

7.2.3. Advantage in Identifying the Training Dataset

The posterior belief β_k quantifies the probability of inferring membership of a single record d. For example, when β_k is low for a census dataset, the individual d can plausibly deny presence in $\mathcal{D}$, and thus presence in the census. In practice, it is also important to know how often $\mathcal{A}_{\texttt{DI}}$ makes a correct guess, which only occurs when $\beta_k > 0.5$. This is quantified by the advantage Adv, which is the success rate normalized to the range $[-1, 1]$, where $Adv = 0$ corresponds to random guessing. Membership advantage was introduced to quantify the success of $\mathcal{A}_{\texttt{MI}}$ [Yeo+18]; however, Adv can also be used for $\mathcal{A}_{\texttt{DI}}$ of $\mathrm{Exp}^{\mathrm{DI}}$ by generalizing Definition 2.14 to Definition 7.2.

Definition 7.2 (Advantage). Given an experiment Exp the advantage is defined as

$$Adv = 2\Pr(Exp = 1) - 1$$

where the probability is over the random iterative choices of the mechanisms up to step k. The advantage in $\mathrm{Exp}^{\mathrm{DI}}$ is denoted $\mathrm{Adv}^{\mathrm{DI}}$, while the advantage in $\mathrm{Exp}^{\mathrm{MI}}$ is $\mathrm{Adv}^{\mathrm{MI}}$. $\diamond$

7.3. Derivation of Upper Bounds

Within this section, we use the DP guarantee to derive upper bounds for *posterior belief* and *advantage* in Sections 7.3.1 and 7.3.2. In Section 7.3.3, we define the expected membership advantage for the Gaussian mechanism, since the original bound is loose.

7.3.1. Upper Bound for the Posterior Belief

We formulate a generic bound on the Bayesian posterior belief that is independent of datasets $\mathcal{D}$ and $\mathcal{D}'$, the mechanism $\mathcal{M}$, and the result matrix $R = (\mathbf{r}_0, \mathbf{r}_1, \ldots, \mathbf{r}_k)$ comprising k multidimensional mechanism outputs. The proposed bound solely assumes that the DP bound holds and makes no further simplifications, which results in an identifiability-based interpretation of DP guarantees. Theorem 7.1 shows that $\mathcal{A}_{\texttt{DI}}$ operates under the sequential composition theorem, for both ϵ-DP and (ϵ, δ)-DP.

Theorem 7.1 (Bounds for the Adaptive Posterior Belief). Consider experiment $\mathrm{Exp}^{\mathrm{DI}}$ with neighboring datasets $\mathcal{D}$ and $\mathcal{D}'$. Let $\mathcal{M}_1, \ldots, \mathcal{M}_k$ be a sequence of arbitrary but

independent differentially private learning algorithms.
(i) Each $\mathcal{M}_i$ provides $\epsilon_1, \ldots, \epsilon_k$-DP to functions f_i with multidimensional output. Then the posterior belief of $\mathcal{A}_{\mathtt{DI}}$ is bounded by

$$\beta_k(\mathcal{D}|R_k) \leq \rho_\beta = \frac{1}{1 + e^{-\sum_{i=1}^{k} \epsilon_i}}$$

(ii) Each $\mathcal{M}_i$ provides (ϵ_i, δ_i)-DP to multidimensional functions f_i. Then the same bound as above holds with probability $1 - \sum_{i=1}^{k} \delta_i$.

Proof. (i) The adversary with unbiased prior (i.e., 0.5) has a maximum posterior belief of $1/(1 + e^{-\epsilon})$ when the ϵ-differentially private Laplace mechanism is applied to a function with a scalar output [LC12]. This upper bound holds also for arbitrary ϵ-differentially private learning algorithms with multidimensional output. We bound the general belief calculation by the inequality of Definition 2.1. Analogously, $\Pr(\mathcal{M}(\mathcal{D}) = \mathbf{r}) \leq e^{\epsilon} \Pr(\mathcal{M}(\mathcal{D}') = \mathbf{r}) + \delta$. Assuming equal priors, the posterior belief can be calculated as follows:

$$\begin{aligned}\beta(\mathcal{D}|R) &= \frac{1}{1 + \frac{\prod_{i=1}^{k} \Pr(\mathcal{M}_i(\mathcal{D}')=\mathbf{r}_i)}{\prod_{i=1}^{k} \Pr(\mathcal{M}_i(\mathcal{D})=\mathbf{r}_i)}} \\ &\leq \frac{1}{1 + \frac{\prod_{i=1}^{k} \Pr(\mathcal{M}_i(\mathcal{D}')=\mathbf{r}_i)}{\prod_{i=1}^{k} e^{\epsilon_i} \Pr(\mathcal{M}_i(\mathcal{D}')=\mathbf{r}_i)+\delta_i}}\end{aligned}$$

For $\delta = 0$, the last equation simplifies to:

$$\begin{aligned}\beta(\mathcal{D}|R) &\leq \frac{1}{1 + \frac{\prod_{i=1}^{k} \Pr(\mathcal{M}_i(\mathcal{D}')=\mathbf{r}_i)}{\prod_{i=1}^{k} e^{\epsilon_i} \Pr(\mathcal{M}_i(\mathcal{D}')=\mathbf{r}_i)}} \\ &= \frac{1}{1 + \prod_{i=1}^{k} e^{-\epsilon_i}} = \frac{1}{1 + e^{-\sum_{i=1}^{k} \epsilon_i}} = \rho_\beta\end{aligned}$$

(ii) We use properties of RDP to prove the posterior belief bound for multidimensional

(ϵ_i, δ_i)-differentially private mechanisms.

$$\beta_k(\mathcal{D}|R) = \frac{1}{1 + \prod_{i=1}^{k} \frac{\Pr(\mathcal{M}_i(\mathcal{D}')=\mathbf{r}_i)}{\Pr(\mathcal{M}_i(\mathcal{D})=\mathbf{r}_i)}} \tag{7.2}$$

$$= \frac{1}{1 + \prod_{i=1}^{k} \frac{\Pr(\mathcal{M}_i(\mathcal{D}')=\mathbf{r}_i)}{\left(e^{\epsilon_{RDP,i}} \cdot \Pr(\mathcal{M}_i(\mathcal{D}')=\mathbf{r}_i)\right)^{1-1/\alpha}}} \tag{7.3}$$

$$= \frac{1}{1 + \prod_{i=1}^{k} e^{-\epsilon_{RDP,i}(1-1/\alpha)} \cdot \Pr(\mathcal{M}_i(\mathcal{D}') = \mathbf{r}_i)^{1/\alpha}} \tag{7.4}$$

In the step from Equation (7.2) to Equation (7.3), we use the probability preservation property, $\Pr(\mathcal{M}(\mathcal{D}) = \mathbf{r}) \leq (e^{\epsilon_{RDP}} \Pr(\mathcal{M}(\mathcal{D}') = \mathbf{r}))^{1-1/\alpha}$, which appears in Langlois et al. [LSS14] and generalizes Lyubashevsky et al. [LPR13]. This same property was used by Mironov [Mir17] to prove that RDP guarantees can be converted to (ϵ, δ) guarantees. In the context of this proof, Mironov also implies that ϵ-DP holds when $e^{\epsilon_{RDP}} \Pr(\mathcal{M}(\mathcal{D}') = \mathbf{r}) > \delta^{\alpha/(\alpha-1)}$, since otherwise $\Pr(\mathcal{M}(\mathcal{D}) = \mathbf{r}) \leq \delta$. We, therefore, assume $e^{\epsilon_{RDP}} \Pr(\mathcal{M}(\mathcal{D}') = \mathbf{r}) > \delta^{\alpha/(\alpha-1)}$, which occurs in at least $1 - \delta$ cases. We continue from Equation (7.4):

$$\beta_k(\mathcal{D}) \leq \frac{1}{1 + \prod_{i=1}^{k} e^{-\epsilon_{RDP,i}(1-1/\alpha)} \cdot \left(\delta_i^{\alpha/(\alpha-1)} \cdot e^{-\epsilon_{RDP,i}}\right)^{1/\alpha}}$$

$$= \frac{1}{1 + \prod_{i=1}^{k} e^{-\epsilon_{RDP,i}} \cdot \delta_i^{1/(\alpha-1)}}$$

$$= \frac{1}{1 + \prod_{i=1}^{k} e^{-\epsilon_{RDP,i}} \cdot e^{-1/(\alpha-1)\ln(1/\delta_i)}}$$

$$= \frac{1}{1 + \prod_{i=1}^{k} e^{-(\epsilon_{RDP,i} + (\alpha-1)^{-1}\ln(1/\delta_i))}} \tag{7.5}$$

$$= \frac{1}{1 + \prod_{i=1}^{k} e^{-\epsilon_i}} = \frac{1}{1 + e^{-\sum_{i=1}^{k} \epsilon_i}} = \rho_\beta \tag{7.6}$$

□

Note that we use the conversion from RDP to DP in the step from Equation 7.5 to Equation 7.6 (cf. Section 2.1). Equivalently one can specify a desired posterior belief and calculate the overall ϵ, which can be spent on composition of differentially private queries:

$$\epsilon = \ln\left(\frac{\rho_\beta}{1-\rho_\beta}\right) \tag{7.7}$$

The value for δ can be chosen independently according to the recommendation that $\delta \ll \frac{1}{N}$ with N points in the input dataset [DR14].

7.3.2. Upper Bound for the Advantage in Identifying the Training Dataset for General Mechanisms

We now formulate an upper bound for the advantage $\mathrm{Adv}^{\mathrm{DI}}$ of $\mathcal{A}_{\mathtt{DI}}$ in Proposition 7.2. The membership advantage of $\mathcal{A}_{\mathtt{MI}}$ has been bounded in terms of ϵ and defines $\mathcal{A}_{\mathtt{MI}}$'s success [Yeo+18]. The general bound for $\mathcal{A}_{\mathtt{MI}}$ also holds for $\mathcal{A}_{\mathtt{DI}}$ based on Proposition 7.1.

Proposition 7.2 (Bound on the Expected Membership Advantage for $\mathcal{A}_{\mathtt{DI}}$)**.** *For any ϵ-DP mechanism the identification advantage of $\mathcal{A}_{\mathtt{DI}}$ in experiment $\mathrm{Exp}^{\mathrm{DI}}$ can be bounded as*

$$\mathrm{Adv}^{\mathrm{DI}} \leq (e^{\epsilon}-1)\Pr(\mathcal{A}_{\mathtt{DI}} = 1 | b = 0)$$

Proof. First, the definition is rewritten by separating true positives and true negatives. Then using that both datasets are equally likely to be chosen by the adversary ($\Pr(b = 1) = \Pr(b = 0) = 0.5$). We substitute $\Pr(b' = 0\ |b = 0)$ by the probability of the complementary event $1 - \Pr(b' = 1\ |b = 0))$ and $b' = 1$ by $\mathcal{A}_{\mathtt{DI}} = 1$, which leads to Equation (7.8), of Yeom et al. [Yeo+18]

$$\begin{aligned}\mathrm{Adv}^{\mathrm{DI}} &= 2(\Pr(b=1)\Pr(b'=1\ |b=1) \\ &\quad + \Pr(b=0)\Pr(b'=0\ |b=0)) - 1 \\ &= \Pr(\mathcal{A}_{\mathtt{DI}} = 1|b=1) - \Pr(\mathcal{A}_{\mathtt{DI}} = 1|b=0) \end{aligned} \tag{7.8}$$

which is the difference between the probability for detecting $\mathcal{D}$ and the probability of incorrectly choosing $\mathcal{D}$. Now we use the fact that the mechanism $\mathcal{M}$ turns r into random variables $X_1 := \mathcal{M}(\mathcal{D})$ and $X_0 := \mathcal{M}(\mathcal{D}')$ for the cases $b = 1$ and $b = 0$, respectively.

We formulate the probability density functions as g_{X_1} and g_{X_0}. Additionally, $A(r)$ is introduced as a shorthand for $\mathcal{A}_{\mathtt{DI}}(\mathbf{r}, \mathcal{D}, \mathcal{D}', \mathcal{M}, \mathtt{Dist})$

$$\begin{aligned}
\mathrm{Adv}^{\mathrm{DI}} &= \Pr(\mathcal{A}_{\mathtt{DI}} = 1 | r = \mathcal{M}(\mathcal{D})) - \Pr(\mathcal{A}_{\mathtt{DI}} = 1 | r = \mathcal{M}(\mathcal{D}')) \\
&= \mathbb{E}_{r=\mathcal{M}(\mathcal{D})}(\mathcal{A}_{\mathtt{DI}}(\mathbf{r}, \mathcal{D}, \mathcal{D}', \mathcal{M}, \mathtt{Dist})) - \\
&\quad \mathbb{E}_{r=\mathcal{M}(\mathcal{D}')}(\mathcal{A}_{\mathtt{DI}}(\mathbf{r}, \mathcal{D}, \mathcal{D}', \mathcal{M}, \mathtt{Dist})) \\
&= \int g_{X_1}(\mathbf{r}) A(\mathbf{r}) \mathrm{d}\mathbf{r} - \int g_{X_0}(\mathbf{r}) A(\mathbf{r}) \mathrm{d}\mathbf{r} \qquad (7.9) \\
&= \int (g_{X_1}(\mathbf{r}) - g_{X_0}(\mathbf{r})) A(\mathbf{r}) \mathrm{d}\mathbf{r} \qquad (7.10)
\end{aligned}$$

Since ϵ-DP formulated as $Pr(\mathcal{M}(\mathcal{D}) \in S) \leq e^{\epsilon} Pr(\mathcal{M}(\mathcal{D}') \in S)$ holds for all S it yields the same inequality $g_{X_1} \leq e^{\epsilon} g_{X_0}$ for the densities at each point

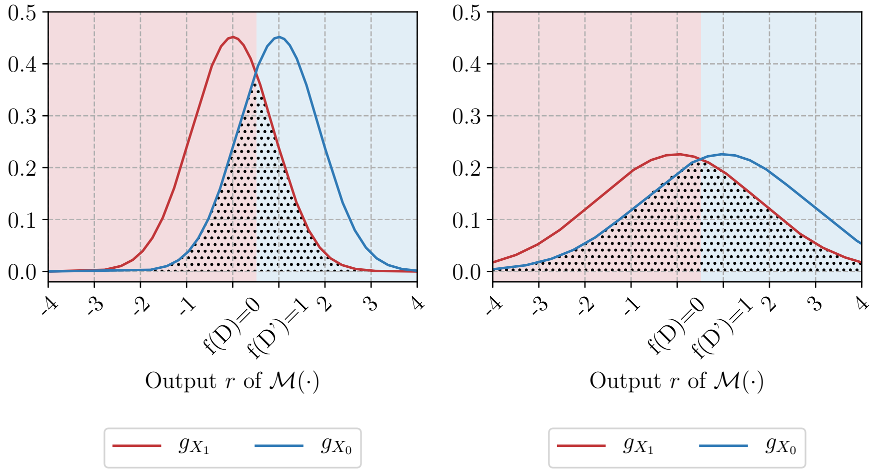
$$\begin{aligned}
\mathrm{Adv}^{\mathrm{DI}} &\leq (e^{\epsilon} - 1) \int g_{X_0}(\mathbf{r}) A(r) \mathrm{d}\mathbf{r} \\
&= (e^{\epsilon} - 1) \Pr(\mathcal{A}_{\mathtt{DI}} = 1 | b = 0) \\
&\leq e^{\epsilon} - 1
\end{aligned}$$

□

Bounding $\Pr(\mathcal{A}_{\mathtt{DI}} = 1 | b = 0)$ by 1 results in $\mathrm{Adv}^{\mathrm{DI}} \leq e^{\epsilon} - 1$. When $\mathcal{A}_{\mathtt{DI}}$ acts like a naive Bayes classifier, only a complete lack of utility from infinite noise results in $\Pr(\mathcal{A}_{\mathtt{DI}} = 1 | b = 0) = 0.5$. Otherwise, $\Pr(\mathcal{A}_{\mathtt{DI}} = 1 | b = 0) \ll 0.5$; therefore, the membership advantage bound is usually not tight. This is in line with Jayaraman et al. [JE19] who expect that this would be the case for MI.

7.3.3. Upper Bound for the Advantage in Identifying the Training Dataset for Gaussian Mechanisms

In practice, $\mathcal{A}_{\mathtt{DI}}$ will be faced with a specific DP mechanism, and we focus on the mechanism used in DPSGD to find a tighter bound than the generic bound described in the previous section. We use the notation $\mathcal{A}_{\mathtt{DI,Gau}}$ and $\mathrm{Adv}^{\mathrm{DI,Gau}}$ to specify the adversary and advantage of an instantiation of $\mathcal{A}_{\mathtt{DI}}$ against the Gaussian mechanism with (ϵ, δ)-DP.



(a) Error regions $(6, 1e^{-6})$-DP **(b)** Error regions $(3, 1e^{-6})$-DP

Figure 7.2.: Error regions for varying ϵ, $\mathcal{M}_{Gau}$

We now derive a tighter bound ρ_α on $\mathrm{Adv}^{\mathrm{DI,Gau}}$ and continue from Equation (7.10). Note that under the assumption of equal priors, the strongest possible adversary of Equation (7.1) maximizes Equation (7.10) by choosing $b = 1$ if $(g_{X_1}(\mathbf{r}) - g_{X_0}(\mathbf{r})) > 0$ and $b = 0$ otherwise. The resulting bound on $\mathrm{Adv}^{\mathrm{DI,Gau}}$ is constructed from $\mathcal{A}_{\texttt{DI,Gau}}$'s strategy; however, the bound holds for all weaker adversaries, including $\mathcal{A}_{\texttt{MI}}$. Since we argue that $\mathcal{A}_{\texttt{DI,Gau}}$ precisely represents the assumptions of DP, the bound should hold for other possible attacks in the realm of DP and the Gaussian mechanism under the i.i.d. assumption.

Since $\mathcal{A}_{\texttt{DI,Gau}}$ is a naive Bayes classifier with known probability distributions, we use the properties of normal distributions (we refer to Tumer et al. [TG96] for full details). We find that the decision boundary does not change under $\mathcal{M}_{Gau}$ with different (ϵ, δ) guarantees as long as the probability density functions (PDF) are symmetric. Holding $\mathcal{M}(\mathcal{D}) = r$ constant and reducing (ϵ, δ) solely affects the posterior belief of $\mathcal{A}_{\texttt{DI,Gau}}$, not the choice of $\mathcal{D}$ or $\mathcal{D}'$. For illustration, consider the example in Figure 7.2. If a $(6, 10^{-6})$-DP $\mathcal{M}_{Gau}$ is used for perturbation, $\mathcal{A}_{\texttt{DI,Gau}}$ has to choose between the two PDFs in Figure 7.2a. Increasing the privacy guarantee to $(3, 10^{-6})$-DP in Figure 7.2b squeezes the PDFs and belief curves. The corresponding regions of error are shaded in Figures 7.2a and 7.2b, where we see that a stronger guarantee reduces $\mathrm{Adv}^{\mathrm{DI,Gau}}$.

We assume throughout this chapter that $\mathcal{A}_{\texttt{DI,Gau}}$ has uniform prior beliefs on the possible databases $\mathcal{D}$ and $\mathcal{D}'$. This distribution is iteratively updated based on the posterior resulting from the mechanism output r. If $\mathcal{M}_{Gau}$ is used to achieve (ϵ, δ)-DP, we can determine the expected membership advantage of the practical attacker $\mathcal{A}_{\mathtt{DI},\mathtt{Gau}}$ analytically by the overlap of the resulting Gaussian distributions [MKB79, p. 321]. We thus consider two multidimensional Gaussian PDFs (i.e., $\mathcal{M}(\mathcal{D})$, $\mathcal{M}(\mathcal{D}')$) with covariance matrix Σ and means (without noise) $\mu_{\mathbf{1}} = f(\mathcal{D}), \mu_{\mathbf{2}} = f(\mathcal{D}')$. This leads us to Theorem 7.2.

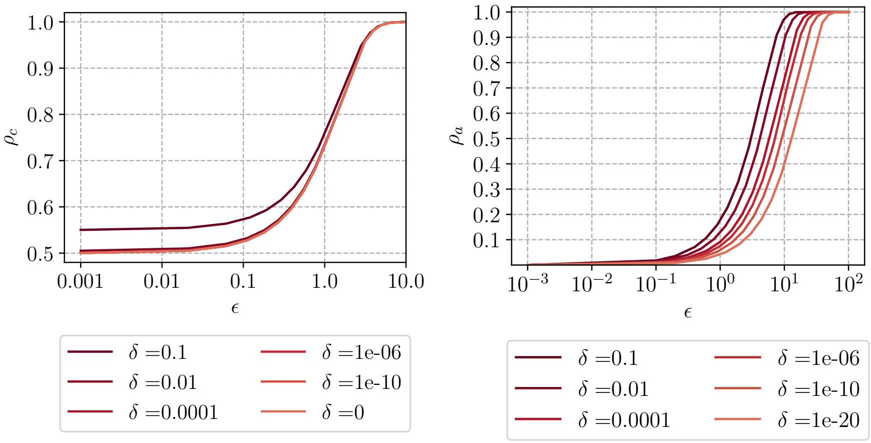
Theorem 7.2 (Tight Bound on the Expected Adversarial Membership Advantage). For the (ϵ, δ)-differentially private Gaussian mechanism, the expected membership advantage of the strong probabilistic adversary on either dataset $\mathcal{D}, \mathcal{D}'$.

$$\mathrm{Adv}^{\mathrm{DI}} \leq \rho_\alpha = 2\Phi\left(\frac{\epsilon}{2\sqrt{2\ln(1.25/\delta)}}\right) - 1$$

where Φ is the cumulative density function of the standard normal distribution.

Proof. We start from Equation (7.9) where the Gauss distributions are g_{X_1} and g_{X_0}. Since both distributions arise from the same mechanism they have the same Σ but different means $\mu_1 = f(\mathcal{D})$ and $\mu_0 = f(\mathcal{D}')$. Since the strongest adversary is the Bayes adversary that chooses according to Equation (7.1) and we assume equal priors, the decision boundary between $\mathcal{D}$ and $\mathcal{D}'$ is the point of intersection of the densities (see Figure 7.2a for the 1D-case). We use linear discriminant analysis where the boundary is a hyperplane halfway between μ_1 and μ_0. This plane is halfway $(\Delta/2)$ between the two centers, where Δ is the Mahalanobis distance $\Delta = \sqrt{(\mu_{\mathbf{1}} - \mu_{\mathbf{2}})^T \Sigma^{-1} (\mu_{\mathbf{1}} - \mu_{\mathbf{2}})}$ [Mah36]. Notably, the decision boundary between $\mathcal{D}$ and $\mathcal{D}'$ does not depend on Σ, but on the possible distance between μ_1 and μ_0 (i.e., sensitivity). As we add independent noise in all dimensions $\Sigma = \sigma^2 \mathbb{I}$, we simplify all calculations from Equation (7.9) to the one-dimensional case and simplify $\Delta = \frac{\|\mu_{\mathbf{1}} - \mu_{\mathbf{2}}\|_2}{\sigma}$. Thus,

$$\begin{aligned}\mathrm{Adv}^{\mathrm{DI,Gau}} &= \Phi(\Delta/2) - \Phi(-\Delta/2) = 2\Phi(\Delta/2) - 1 \\ &= 2\Phi\left(\frac{\|\mu_{\mathbf{1}} - \mu_{\mathbf{2}}\|_2}{2\sigma}\right) - 1 \qquad (7.11)\end{aligned}$$



(a) ρ_β

(b) ρ_α

Figure 7.3.: ρ_β and ρ_α for various (ϵ, δ) when using $\mathcal{M}_{Gau}$

Inserting the standard deviation needed for (ϵ, δ)-DP from Equation (2.2) then yields

$$\begin{aligned}\mathrm{Adv}^{\mathrm{DI,Gau}} &= 2\Phi\left(\frac{\|\boldsymbol{\mu_1} - \boldsymbol{\mu_2}\|_2}{2GS_{f_2}(\sqrt{2\ln(1.25/\delta)}/\epsilon)}\right) - 1 \\ &\leq 2\Phi\left(\frac{\epsilon}{2(\sqrt{2\ln(1.25/\delta)})}\right) - 1 = \rho_\alpha\end{aligned}$$

□

We can calculate ϵ from a chosen maximum expected advantage

$$\epsilon = \sqrt{2\ln(1.25/\delta)}\,\Phi^{-1}\left(\frac{\rho_\alpha + 1}{2}\right) \tag{7.12}$$

(ϵ, δ) guarantees with $\delta > 0$ can be expressed via a scalar value ρ_α. In summary, we now have complementary interpretability scores, where ρ_β represents a bound on individual deniability and ρ_α relates to the expected probability of reidentification. While ρ_β holds for all mechanisms, ρ_α was derived solely for the Gaussian mechanism. We provide example plots of ρ_β and ρ_α for different (ϵ, δ) in Figure 7.3. To compute both scores, we use Theorems 7.1 and 7.2. We set $f(\mathcal{D}) = (0_1, 0_2, \dots, 0_k)$ and $f(\mathcal{D}') = (1_1, 1_2, \dots, 1_k)$ for all dimensions k, so $GS_{f_2} = \sqrt{k}$. Figure 7.3a illustrates that there is no significant difference for ρ_β between ϵ-DP and (ϵ, δ)-DP. In contrast, ρ_α strongly depends on the choice of δ.

7.3.4. RDP Instead of Sequential Composition

In iterative settings, such as NN training, the data scientist will have to perform multiple mechanism executions, which necessitates the use of composition theorems to split the total guarantee into guarantees per iteration (ϵ_i, δ_i). Sequential composition only offers loose bounds in practice [DRV10; KOV17]; we suggest using RDP composition, which allows a tight analysis of the privacy loss over a series of mechanisms. Therefore, we adapt both ρ_β and ρ_α to RDP.

We first demonstrate that RDP composition results in stronger (ϵ, δ) guarantees than sequential composition for a fixed bound ρ_β. We start from Equation (7.5):

$$\begin{aligned}
\beta_k(\mathcal{D}|R) &\leq \frac{1}{1+\prod_{i=1}^{k} e^{-(\epsilon_{RDP,i}+(\alpha-1)^{-1}\ln(1/\delta_i))}} \\
&= \frac{1}{1+e^{k(\alpha-1)^{-1}\ln(\delta_i)-\sum_{i=1}^{k}\epsilon_{RDP,i}}} \qquad (7.13)\\
&= \frac{1}{1+e^{(\alpha-1)^{-1}\ln(\delta_i^k)-\sum_{i=1}^{k}\epsilon_{RDP,i}}} \\
&= \frac{1}{1+e^{-(\sum_{i=1}^{k}\epsilon_{RDP,i}-(\alpha-1)^{-1}\ln(\delta_i^k))}} = \rho_\beta \qquad (7.14)
\end{aligned}$$

We assume the same value of δ_i is used during every execution and can therefore remove it from the sum in Equation (7.13). Equation (7.14) and the conversion (α, ϵ_{RDP})-RDP to $\left(\epsilon_{RDP} - \frac{\ln\delta}{\alpha-1}, \delta\right)$-DP imply that an RDP-composed bound can be achieved with a composed δ equal to δ_i^k. We know that sequential composition results in a composed δ value equal to $k\delta_i$. Since $\delta^k < k\delta$, RDP offers a stronger (ϵ, δ) guarantee for the same ρ_β, and results in a tighter bound for ρ_β under composition. This behavior can also be interpreted as the fact that holding the composed (ϵ, δ) guarantee constant, the value of ρ_β is greater when the sequential composition is used compared to RDP.

A similar analysis of the expected membership advantage under composition is required when considering a series of mechanisms $\mathcal{M}$. We restrict our elucidations to the Gaussian mechanism. The k-fold composition of $\mathcal{M}_{Gau_i}$, each step guaranteeing $(\alpha, \epsilon_{RDP,i})$-RDP, can be represented by a single execution of $\mathcal{M}_{Gau}$ with k-dimensional output guaranteeing $(\alpha, \epsilon_{RDP} = k\epsilon_{RDP,i})$-RDP. We start from Equation (7.11), and use Equation (2.4) and the fact that GS_{f_2} bounds $\|\mu_{1,i} - \mu_{2,i}\|$.

$$\begin{aligned}
\mathrm{Adv}^{\mathrm{DI,Gau}} &= 2\Phi\left(\frac{\|\boldsymbol{\mu_1} - \boldsymbol{\mu_2}\|_2}{2\sigma_i}\right) - 1 = 2\Phi\left(\frac{\sqrt{k}\|\mu_{1,i} - \mu_{2,i}\|_2}{2GS_{f_2}\sqrt{\alpha/(2\epsilon_{RDP,i})}}\right) - 1 \\
&\leq 2\Phi\left(\frac{\sqrt{k}}{2\sqrt{\alpha/(2\epsilon_{RDP,i})}}\right) - 1 = 2\Phi\left(\sqrt{\frac{k\epsilon_{RDP,i}}{2\alpha}}\right) - 1 \\
&= 2\Phi\left(\sqrt{\frac{\epsilon_{RDP}}{2\alpha}}\right) - 1 = \rho_\alpha
\end{aligned}$$

The result shows that $\mathcal{A}_{\mathtt{DI,Gau}}$ fully takes advantage of the RDP composition properties of $\epsilon_{RDP,i}$ and α; as expected, ρ_α takes on the same value, regardless of whether k composition steps with $\epsilon_{RDP,i}$ or a single composition step with ϵ_{RDP} is carried out. Therefore, we can calculate the final ρ_α for functions with multiple iterations, such as the training of deep learning models, and ρ_α can be decomposed into a privacy guarantee per composition step with RDP.

7.4. Application to Deep Learning

In DPSGD, the stochastic gradient descent optimizer adds Gaussian noise with standard deviation σ to the computed gradients. The added noise ensures that the learned NN is (ϵ, δ) differentially private w.r.t. the training dataset. This section illustrates our method for choosing DPSGD privacy parameters. Data scientists may first choose upper bounds for the posterior belief, from which ϵ is obtained using Equation (7.7). From ϵ and the sensitivity, the standard deviation σ of the Gaussian noise is determined.

We discuss a heuristic for estimating the local sensitivity in Section 7.4.1. Then, Section 7.4.2 formulates an algorithm for implementing $\mathcal{A}_{\mathtt{DI,Gau}}$, and discusses how this algorithm is used to empirically quantify the posterior belief and the advantage. Finally, using the implemented adversary $\mathcal{A}_{\mathtt{DI,Gau}}$ a method for auditing the privacy loss ϵ and the bounds derived in Section 7.3 is provided in Section 7.4.3.

7.4.1. Setting Privacy Parameters and Determining the Sensitivity

Based on the recommendation to set `C` to the median of the norms of unclipped gradients [Aba+16] we set $\mathtt{C} = 3$ in all our experiments. In the following, we describe how to set up the system in order to determine the standard deviation of Gaussian noise σ. We want to limit $\mathcal{A}_{\mathtt{DI,Gau}}$'s belief of distinguishing a training dataset differing in any chosen person by setting the upper bound for the posterior belief ρ_β. We then transform ρ_β to an overall ϵ for the k update steps in DPSGD using Equation (7.7), which in turn leads to σ for the DPSGD using Equation (2.2). In Equation (2.2) two parameters need to be set: Δf and δ. While we set δ to $1/|\mathcal{D}|$ for all experiments, the choice of Δf is more challenging. The upper bound for the privacy loss ϵ can only be reached when Δf is set specifically to the sensitivity of the dataset at hand. We can calculate the local sensitivity for bounded DP as

$$LS_{\hat{g}_i}(\mathcal{D}) = n \cdot ||\hat{g}_i(\mathcal{D}') - \hat{g}_i(\mathcal{D})||,$$

and for unbounded DP as

$$LS_{\hat{g}_i}(\mathcal{D}) = ||(n-1) \cdot \hat{g}_i(\mathcal{D}') - n \cdot \hat{g}_i(\mathcal{D})||,$$

where $\hat{g}_i(\mathcal{D})$ and $\hat{g}_i(\mathcal{D}')$ represent the average of all clipped, unperturbed per-example gradients $\bar{g}_i(d) \forall d \in \mathcal{D}$ and $d \in \mathcal{D}'$, respectively.

Since clipping is done before perturbation, the global sensitivity GS_f in DPSGD is set to the clipping norm for unbounded DP, i.e., $GS_f = \mathtt{C}$. The sensitivity bounds the impact of a datapoint on the total gradient, equivalent to the difference between the gradients differing between $\mathcal{D}$ and $\mathcal{D}'$, which is artificially bounded by C for unbounded DP. For bounded DP where one record is instead replaced with another in $\mathcal{D}'$, the lengths of the clipped gradients of these two records could each be C and point in opposite directions resulting in $n \cdot ||\hat{g}_i(\mathcal{D}') - \hat{g}_i(\mathcal{D})||_2 \leq 2\mathtt{C}$.

Although C bounds the influence of a single training record on the gradient, C may well be loose, since C does not necessarily reflect the factual difference between the training dataset and possible neighboring datasets. When C is loose, the DP bound on privacy loss ϵ is not reached, and the identifiability metrics ρ_α and ρ_β will not be reached either. Nissim et al. [NRS07] proposed local sensitivity LS_f to specifically scale noise to the input data. The use of LS_f decreases the noise scale by narrowing the DP guarantee from protection against inference on any possible adjacent datasets to inference on the original dataset and any adjacent dataset. In ML projects training and test data are often sampled from a static holdout, where all datapoints stem from a domain of similar data. If the holdout is a very large dataset, only the specific neighboring datasets possible in this domain need to be protected under DP. To reach the DP bound, we suggest fixation of the training dataset $\mathcal{D}$ and considering only neighboring datasets $\mathcal{D}'$ adjacent to $\mathcal{D}$.

However, approximating $LS_{\hat{g}_i}$ for NN training is difficult because the gradient function output depends not only on $\mathcal{D}$ and $\mathcal{D}'$ but also on the architecture and current weights of the network. To ease this dilemma, we propose *dataset sensitivity* in Definition 7.3. Dataset sensitivity is a heuristic with which we strive to consider the neighboring dataset $\hat{\mathcal{D}}'$ with the largest difference to $\mathcal{D}$ within the overall ML dataset $\mathcal{U}$ in an effort to approximate $LS_{\hat{g}_i}$. We assume that similar datapoints will result in similar gradients. While this assumption does not necessarily hold under crafted adversarial examples [GSS15], for which privacy protection cannot be guaranteed, the malicious intent renders the necessity for their protection debatable.

Definition 7.3 (Dataset Sensitivity). Consider a given dataset $\mathcal{U}$, a training dataset $\mathcal{D} \subseteq \mathcal{U}$, all neighboring datasets $\mathcal{D}' \subseteq \mathcal{U}$ and a dissimilarity measure γ. The dataset sensitivity $DS(\mathcal{D})$ w.r.t. dissimilarity measure γ is then defined as

$$DS(\mathcal{D}) = \max_{\mathcal{D}'} \gamma(\mathcal{D}, \mathcal{D}')$$

and consequently

$$\hat{\mathcal{D}}' := \arg\max_{\mathcal{D}'} \gamma(\mathcal{D}, \mathcal{D}')$$

◇

In Definition 7.3 the dissimilarity measure of specific datasets is not further specified. In practice, if a dissimilarity or distance measure γ of individual datapoints is available, it can be used to find the most dissimilar neighboring dataset $\hat{\mathcal{D}}'$ that maximizes the dataset sensitivity. The computation of $\mathcal{D}'$ depends on the neighboring datasets and is different for unbounded and bounded DP. More precisely, for unbounded DP one forms $\hat{\mathcal{D}}' = \mathcal{D} \setminus \{d'\}$ by removing the most dissimilar datapoint $\hat{d}$ from the training data:

$$\hat{d} = \arg\max_{d_1 \in \mathcal{D}} \sum_{d_2 \in \mathcal{D} \setminus d_1} \gamma(d_1, d_2). \tag{7.15}$$

The dataset $\hat{\mathcal{D}}'$ is then used to approximate the local sensitivity $LS_{\hat{g}_i}$ by

$$LS_{\hat{g}_i}(\mathcal{D}) \approx \hat{LS}_{\hat{g}_i}(\mathcal{D}) := \|\bar{g}_i(\hat{d})\|, \tag{7.16}$$

where $\bar{g}_i(d)$ is the clipped gradient of datapoint d in iteration i. The simplification from $LS_{\hat{g}_i}$ to DS allows us to bypass the complex gradient calculations to identify dissimilar $\mathcal{D}$ and $\mathcal{D}'$. The computational complexity of computing the dataset sensitivity only depends on the dataset size n, but not the number of iterations k, as the local sensitivity does. For bounded DP where a neighboring dataset is formed by replacing an element $\{d\} \in \mathcal{D}$ with an element $d' \in \mathcal{U} \setminus \mathcal{D}$ one searches for

$$(\hat{d}, \hat{d}') = \underset{d \in \mathcal{D}, d' \in \mathcal{U} \setminus \mathcal{D}}{\arg\max}\ \gamma(d, d'), \tag{7.17}$$

and approximates the local sensitivity as

$$LS_{\hat{g}_i}(\mathcal{D}) \approx \hat{LS}_{\hat{g}_i}(\mathcal{D}) := \|\bar{g}_i(\hat{d}) - \bar{g}_i(\hat{d}')\|. \tag{7.18}$$

7.4.2. Empirical Quantification of Posterior Beliefs and Advantages

In Section 7.4.1 the noise scale σ limits the upper bound for the posterior belief of $\mathcal{A}_{\texttt{DI}}$ on the original dataset $\mathcal{D}$. According to Theorem 7.1 this upper bound holds with probability $1 - \delta$. For a given dataset, the posterior belief might be much smaller than the bound, so it is desirable to determine the empirical posterior belief on $\mathcal{D}$. The same holds for the advantage $\mathrm{Adv}^{\mathrm{DI}}$ and the upper bound ρ_α from Theorem 7.2 w.r.t. identifying dataset $\mathcal{D}$. We formulate an implementation of the adversary $\mathcal{A}_{\texttt{DI,Gau}}$ which allows us to assess the empirical posterior belief β and membership advantage $\mathrm{Adv}^{\mathrm{DI}}$, and thus the empirical privacy loss of specifically trained models. The adversary $\mathcal{A}_{\texttt{DI,Gau}}$ strives to identify the training dataset, having the choice between neighboring datasets $\mathcal{D}$ and $\mathcal{D}'$. In addition to $\mathcal{D}$ and $\mathcal{D}'$, $\mathcal{A}_{\texttt{DI,Gau}}$ is assumed to have knowledge of the NN learning parameters and updates after every training step $i \leq k$: learning rate η, weights θ_i, perturbed gradients $\tilde{g}_i$, privacy mechanism $\mathcal{M}_i$, parameters (ϵ, δ), $\texttt{C}$, the resulting standard deviation σ of the Gaussian distribution and the prior beliefs. The implementation of $\mathcal{A}_{\texttt{DI}}$ for DPSGD is provided in Algorithm 7.1.

Algorithm 7.1: $\mathcal{A}_{\texttt{DI,Gau}}$ in Deep Learning for Unbounded DP

```
Require: Neighboring datasets $\mathcal{D}$,$\mathcal{D}'$ with $n$,$n'$ records, respectively, $k$, $\theta_0$, $\eta$, $\tilde{g}_i$ per training
    step $i \leq k$, $\mathcal{M}_i$, $(\epsilon_i, \delta_i)$, prior beliefs $\beta_0(\mathcal{D}) = \beta_0(\mathcal{D}') = 0.5$,
 1: for $i \in [k]$ do
 2:     Calculate clipped Batch gradients
 3:     $\hat{g}_i(\mathcal{D}) \leftarrow \mathcal{M}_i(\mathcal{D}, \sigma = 0)$
 4:     $\hat{g}_i(\mathcal{D}') \leftarrow \mathcal{M}_i(\mathcal{D}', \sigma = 0)$
 5:     Calculate Sensitivity and $\sigma$
 6:     $\Delta f \leftarrow GS_{\hat{g}} = \texttt{C}$
 7:     $\sigma_i = \Delta f \sqrt{2 \ln(1.25/\delta_i)}/\epsilon_i$
 8:     Calculate Belief
 9:     $\beta_{i+1}(\mathcal{D}) \leftarrow \frac{\beta_i(\mathcal{D}) \cdot \Pr[\mathcal{M}_i(\mathcal{D}, \sigma=\sigma_i) = \tilde{g}_i]}{\beta_i(\mathcal{D}) \cdot \Pr[\mathcal{M}_i(\mathcal{D}, \sigma=\sigma_i) = \tilde{g}_i] + \beta_i(\mathcal{D}') \cdot \Pr[\mathcal{M}_i(\mathcal{D}') = \tilde{g}_i]}$
10:     $\beta_{i+1}(\mathcal{D}') \leftarrow 1 - \beta_{i+1}(\mathcal{D})$
11:     Compute weights
12:     $\theta_{i+1} \leftarrow \theta_i - \eta \tilde{g}_i$
13: end for
14: Output $\mathcal{D}$ if $\beta_k(\mathcal{D}) > \beta_k(\mathcal{D}')$, $\mathcal{D}'$ otherwise
```

In each learning step $\mathcal{A}_{\texttt{DI}}$ first computes the unperturbed, clipped batch gradients for both datasets based on the resulting weights from the previous step of the perturbed learning algorithm (Steps 3 and 4). Then $\mathcal{A}_{\texttt{DI,Gau}}$ calculates the sensitivity. The ϵ_i and δ_i for each iteration are calculated using RDP composition (cf. Equation (2.4)). Consequently, the Gaussian mechanism scale σ is calculated from (ϵ, δ) and Δf using Equation (2.2). Using the standard deviation σ, the posterior belief β_i is updated in Step 9 based on the observed perturbed clipped gradient $\tilde{g}_i$ and the unperturbed gradients from Steps 3 and 4. The calculation is based on Lemma 7.1. After the training finished, $\mathcal{A}_{\texttt{DI,Gau}}$ tries to identify the used dataset based on the final posterior beliefs β_k on the two datasets. $\mathcal{A}_{\texttt{DI,Gau}}$ wins the identification game, if $\mathcal{A}_{\texttt{DI,Gau}}$ chooses the used dataset $\mathcal{D}$. The advantage to win the experiment is statistically estimated from several identical repetitions of the experiment. $\mathrm{Adv}^{\mathrm{DI,Gau}}$ and δ are empirically calculated by counting the cases in which β_k for $\mathcal{D}$ exceeds 0.5 and ρ_β, respectively.

One pass over all records in $\mathcal{D}$ (i.e., one epoch), can comprise multiple update steps. In mini-batch gradient descent, a number of b records from $\mathcal{D}$ is sampled for calculating an update, and one epoch results in $|\mathcal{D}|/b$ update steps. In batch gradient descent, all records in $\mathcal{D}$ are used within one update step, and one epoch consists of a single update step. We operate with batch gradient descent since it reflects the auxiliary side knowledge

Table 7.1.: Time complexity for DS, β and Adv

Algorithm	Time complexity	Comment
DS	$O(n^2)$	One-time effort for training dataset.
β	$O(nk)$	Computing belief from clipped Batch gradients.
Adv	$O(1)$	Computing Adv for individual training (cf. 14 in Algorithm 7.1)

of $\mathcal{A}_{\mathtt{DI}}$; thus k denotes the overall number of epochs and training iterations. In some of the following experiments we will set $\Delta f = LS_{\hat{g}_i}(\mathcal{D})$ in Step 6 by calculating the local sensitivity $LS_{\hat{g}_i}$ for the clipped gradients $\hat{g}_i$ (cf. Definition 2.4). These assumptions are similar to those of white-box MI attacks against federated learning [NSH19].

The time complexities for calculating dataset sensitivity, posterior belief, and advantage are stated in Table 7.1. Note that the calculation effort will either lie with $\mathcal{A}_{\mathtt{DI}}$ or the data scientist, depending on whether an audit or an actual attack is performed. The calculation of dataset sensitivity is well parallelizable for the considered dissimilarity measures.

7.4.3. Method for Auditing ϵ

In this section, we introduce a method to empirically determine the privacy loss ϵ. This empirical loss is denoted ϵ' and is relevant for data scientists. If ϵ' is close to ϵ, the DP perturbation does not add more noise than necessary. However, if ϵ' is far below ϵ, too much noise is added, and utility is unnecessarily lost. We repeat the training process multiple times and use the set of results to calculate ϵ'. The empirical loss ϵ' can be calculated from observed $LS_{\hat{g}}$, β_k, and $\mathrm{Adv}^{\mathrm{DI,Gau}}$ during model training:

- From $LS_{\hat{g}_1}, \ldots, LS_{\hat{g}_k}$, the empirical ϵ' is calculated as follows: (i) calculate $\sigma_1, \ldots, \sigma_k$ as $\sigma_i = 2\mathtt{C}/LS_{\hat{g}_i} \cdot \sigma$ (cf. Equation (2.3)) for each repetition of the experiment, (ii) calculate ϵ' with RDP composition with target δ, epochs k, and σ using Tensorflow privacy accountant[1], and (iii) choose the maximum value $\epsilon'^{\max}$ over all repetitions of the experiment.

[1] https://github.com/tensorflow/privacy/blob/master/tensorflow_privacy/privacy/analysis/rdp_accountant.py

- From posterior beliefs β, ϵ' is calculated by (i) choosing the maximum final posterior belief $\beta_k^{\max}$ for all experiments and (ii) setting $\epsilon' = \beta_k^{\max}/(1 - \beta_k^{\max})$ using Equation (7.7).

- From $\mathrm{Adv}^{\mathrm{DI,Gau}}$: (i) counting the number of wins n_{win}, i.e., how often $\beta_k > 0.5$ over all n_{Exp} experiments, (ii) estimate $\mathrm{Adv}^{\mathrm{DI,Gau}} = 2n_{win}/n_{Exp} - 1$, and (iii) calculate
$\epsilon' = \sqrt{2\ln(1.25/\delta)}\,\Phi^{-1}\left(\frac{\mathrm{Adv}^{\mathrm{DI,Gau}}+1}{2}\right)$ using Equation (7.12).

This empirical loss ϵ' will only be close to ϵ if noise is added according to the sensitivity of the dataset. Of the three variants above, the calculation from the sensitivities is the most direct method. The calculation from the posterior belief is less direct. Since the identification advantage ignores the size of the belief it is expected to be the least accurate way to estimate ϵ.

Furthermore, we also implement the MI adversary $\mathcal{A}_{\texttt{MI}}$ defined by Yeom et al. [Yeo+18] and compare the resulting advantage to the advantage achieved by $\mathcal{A}_{\texttt{DI,Gau}}$. This instance of $\mathcal{A}_{\texttt{MI}}$ uses the loss L of a neural network prediction in an approach similar to $\mathcal{A}_{\texttt{DI,Gau}}$, who analyzes the gradient updates instead.

7.5. Evaluation

We empirically show that we can train models that yield an empirical privacy loss ϵ' close to the specified privacy loss bound ϵ. We achieve an advantage equal to ρ_α and tightly bound posterior belief ρ_β when the sensitivity is set to $LS_{\hat{g}_i}$ for the clipped batch gradients at every update step i. Privacy is specified by setting the upper bound for the belief, e.g., to $\rho_\beta = 0.9$. Together with the sensitivity (cf. Section 7.4.1), this determines the noise of the Gaussian mechanism and yields ϵ. The posterior belief β and the advantage $\mathrm{Adv}^{\mathrm{DI,Gau}}$ are then empirically determined using the implemented adversary[2] $\mathcal{A}_{\texttt{DI,Gau}}$ as described in Section 7.4.2. The empirical privacy loss ϵ' is determined as described in Section 7.4.3. We evaluate $\mathcal{A}_{\texttt{DI,Gau}}$ for three ML datasets: the MNIST image dataset[3], the Purchase-100 customer preference dataset [Sho+17], and the Adult census income

[2] We provide code and data for this chapter: `https://github.com/SAP-samples/security-research-identifiability-in-dpdl`.

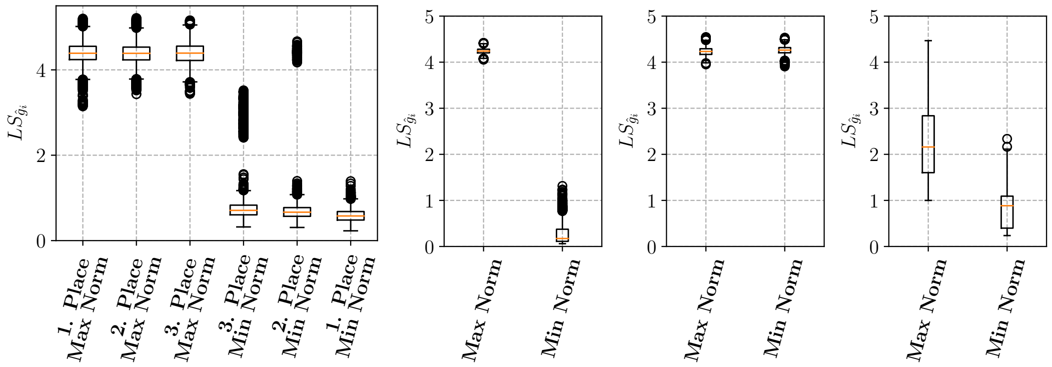
[3] Dataset and detailed description available at: `http://yann.lecun.com/exdb/mnist/`

dataset [Koh96]. To improve training speed in our experiments, we set training dataset $\mathcal{D}$ to a randomly sampled subset of size 100 for MNIST and 1000 for both Purchase-100 and Adult. Multiple trainings and perturbations are evaluated on the sampled $\mathcal{D}$.

The MNIST NN consists of two convolutional layers with kernel size $(3, 3)$ each, batch normalization, and max pooling with pool size $(2, 2)$, and a 10-neuron softmax output layer. For Purchase-100, the NN comprises a 600-neuron input layer, a 128-neuron hidden layer, and a 100-neuron output layer. Our NN for Adult consists of a 104-neuron input layer due to the use of dummy variables for categorical attributes, two 6-neuron hidden layers, and a 2-neuron output layer. We used ReLU and softmax activation functions for the hidden layers, and the output layer. For all experiments, we chose the learning rate $\eta = 0.005$ and set the number of iterations $k = 30$ which led to converging models. Preprocessing comprised the removal of incomplete records, and data normalization.

7.5.1. Evaluation of Sensitivities

While local sensitivity is favored when striving to reach the privacy bound, we evaluate and compute both $\Delta f = \hat{LS}_{\hat{g}_i}(\mathcal{D})$ and $\Delta f = GS_{\hat{g}}$, as described in Section 7.4.1. In addition, we consider bounded and unbounded DP in our experiments. In order to find the most dissimilar datapoint for the construction of $\hat{\mathcal{D}'}$ in Equations (7.15) and (7.17) we require a dissimilarity measure. We considered domain-specific candidates for the dissimilarity measures: the negative structural similarity index measure (SSIM) and Euclidean distance for MNIST, and the Hamming, Euclidean, Manhattan, and Cosine distance for the datasets Purchase-100 and Adult. We chose these metrics because we expect them to contain information relevant to the gradients of datapoints. However, for example, we quickly noticed for the Euclidean distance on MNIST image data that it does not capture the meaning or shapes pictured and thus falls short. Instead, the SSIM captures structure in images, and images with a small SSIM dissimilarity values resulted in similar gradients, while images with greater dissimilarity resulted in very different gradients. This observation supports the hypothesis that an appropriate domain-specific measure can be used to estimate local sensitivity $LS_{\hat{g}_i}$ from dataset sensitivity DS. For Purchases-100 the Hamming distance was clearly superior to the Cosine distance as illustrated in Figures 7.4b and 7.4c. The Manhattan distance fit best for the Adult dataset. For the sensitivity experiments, the bound for the posterior belief is set to $\rho_\beta = 0.9$.



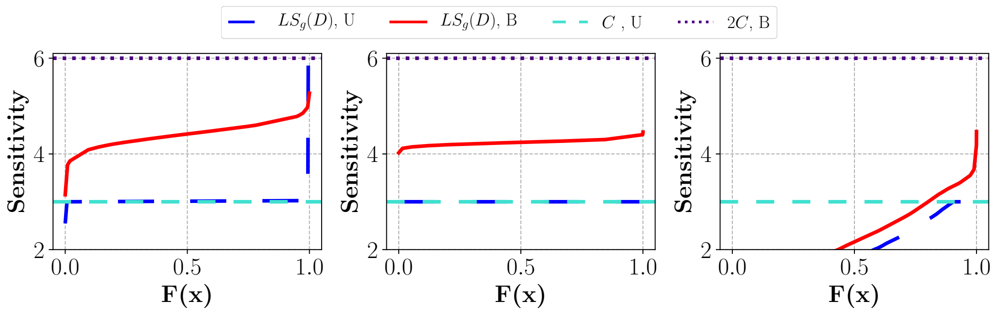
(a) MNIST: SSIM distance

(b) Purchase-100: Hamming distance

(c) Purchase-100: Cosine distance

(d) Adult: Manhattan distance

Figure 7.4.: Distribution of the local sensitivity $LS_{\hat{g}_i}(\mathcal{D})$ computed by $\mathcal{A}_{\mathtt{DI,Gau}}$ using Equation (7.15) from max to min difference in $\mathcal{D}$ and $\mathcal{D}'$ for $k = 30$ and 250 experiment repetitions



(a) MNIST

(b) Purchase-100

(c) Adult

Figure 7.5.: Sensitivities over the course of the training for $\rho_\beta = 0.9$ ($\epsilon = 2.2$) and $\mathtt{C} = 3$

To confirm that maximizing dataset sensitivity from Definition 7.3 allows us to approximate $LS_{\hat{g}_i}$, we train with several differing $\mathcal{D}'$ and evaluate the sensitivities for all $k = 30$ iterations. For the MNIST dataset, the top three choices of $\mathcal{D}'$ that maximize DS and the three choices that minimize DS are used. As expected, the resulting local sensitivities $LS_{\hat{g}_i}$ shown in Figure 7.4a are clearly larger for the three top choices. The outliers for the second and third smallest dataset sensitivities only account for 1.6% and 5.2% of the 7500 overall observed sensitivity norms. More importantly, no far outliers occur for the largest and smallest sensitivities. The same general trend holds for Purchase-100 and Adult in Figures 7.4b and 7.4d, which we limit to the maximum and minimum DS due to space constraints. If the chosen global sensitivity is too large compared to the local sensitivity of a specific dataset too much noise will be added when using $GS_{\hat{g}}$, as described in Section 7.4.1. Global sensitivity $GS_{\hat{g}}$ and local sensitivity $LS_{\hat{g}_i}$ are determined for bounded and unbounded DP over 1000 repetitions for $\rho_\beta = 0.9$ ($\epsilon = 2.2$) according to Equation (7.16) and Equation (7.18). Both can be compared in Figure 7.5.

7.5.2. Quantification of Identifiability for DPSGD

For each of the 1000 experiment repetitions, the posterior belief β_k and the membership advantage $\mathrm{Adv}^{\mathrm{DI,Gau}}$ are experimentally determined using the implementation of $\mathcal{A}_{\mathtt{DI,Gau}}$ for DPSGD. We set $\rho_\beta = 0.9$ ($\epsilon = 2.2$) and compare bounded and unbounded DP. Table 7.2 shows the analytically obtained values for privacy loss ϵ, and the bound ρ_α for the advantage. The parameters ϵ, δ, and ρ_α for $\rho_\beta = 0.9$ can be read from Table 7.3; ϵ is determined from Equation (7.7), whereas ρ_α is calculated from ϵ from Theorem 7.2.

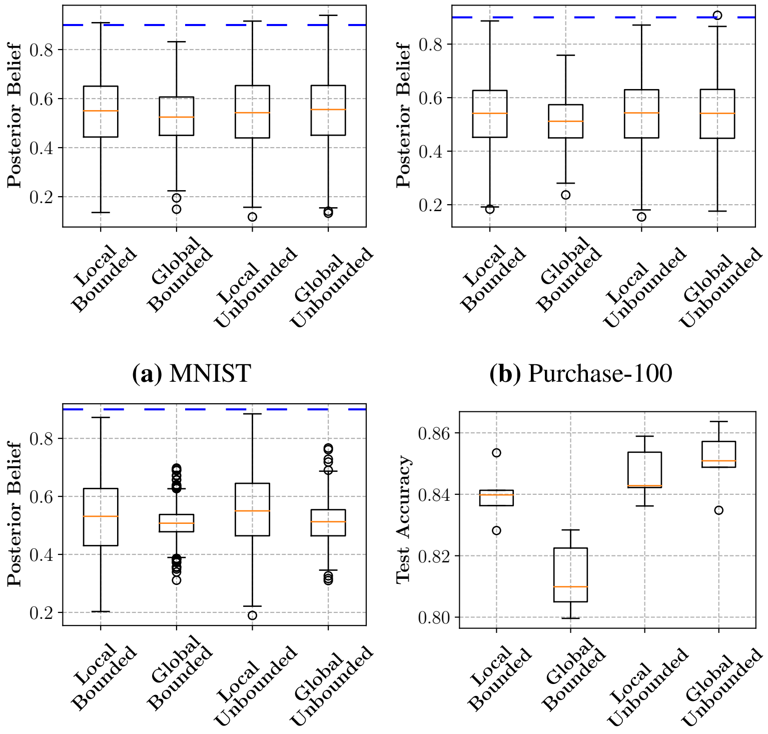
Table 7.2.: Empirical $\mathrm{Adv}^{\mathrm{DI,Gau}}$ and δ' for $\rho_\beta = 0.9$ using $LS_{\hat{g}_i}$ and $GS_{\hat{g}}$ with bounded (B) and unbounded (U) DP

	MNIST		Purchase-100		Adult	
	$\mathrm{Adv}^{\mathrm{DI,Gau}}$	δ'	$\mathrm{Adv}^{\mathrm{DI,Gau}}$	δ'	$\mathrm{Adv}^{\mathrm{DI,Gau}}$	δ'
LS B	0.24	2e-3	0.25	0	0.17	0
LS U	0.23	2e-3	0.23	0	0.22	0
GS B	0.18	0	0.1	0	0.13	0
GS U	0.27	4e-3	0.24	1e-3	0.18	0

Table 7.3.: Experiment setting for posterior belief ρ_β and δ with analytically determined privacy loss ϵ and advantage bound ρ_α

	MNIST				Purchase-100				Adult			
ρ_β	0.52	0.75	0.9	0.99	0.53	0.75	0.9	0.99	0.53	0.75	0.9	0.99
δ	0.01				0.001				0.001			
ϵ	0.08	1.1	2.2	4.6	0.12	1.1	2.2	4.6	0.12	1.1	2.2	4.6
ρ_α	0.01	0.14	0.28	0.54	0.01	0.12	0.23	0.46	0.01	0.12	0.23	0.46

First, we verify that the upper bound ρ_β on the posterior belief holds. The posterior beliefs β_k of these experiments are described in Figures 7.6a, 7.6b, and 7.6c. For a single experiment, the posterior belief on the training dataset $\mathcal{D}$ is on average only slightly above 0.5. While for most cases the posterior belief is far below the bound of 0.9 (specified by



(a) MNIST **(b)** Purchase-100

(c) Adult **(d)** MNIST test accuracy

Figure 7.6.: Distribution of empirical posterior beliefs β_k (panels a to c) and an example for test accuracy after training with $\rho_\beta = 0.9$ ($\epsilon = 2.2$) (panel d)

the blue, dashed line), the upper bound is violated with a small probability. The relative frequency of these violations is denoted as δ'. Since the DP bound, and thus ρ_β, only holds with probability $1 - \delta$ according to Theorem 7.1 violations are acceptable as long as $\delta' \leq \delta$. Indeed, the experimentally obtained δ' for $\rho_\beta = 0.9$ in Table 7.2 is always smaller than the corresponding δ in Table 7.3. Similarly, the advantage should be close to the estimate ρ_α stated in Table 7.3. The advantage is experimentally estimated as the relative frequency of experiments where the implemented adversary $\mathcal{A}_{\texttt{DI,Gau}}$ correctly chooses $\mathcal{D}$ and is stated in Table 7.2.

Figure 7.6 illustrates the influence of sensitivity in the bounded and unbounded DP settings. In Figures 7.6a, 7.6b and 7.6c, the chosen upper bound $\rho_\beta = 0.9$ (blue line) is clearly not reached for the bounded case when global sensitivities are used. Similarly, the advantage of $\mathcal{A}_{\texttt{DI,Gau}}$ in Table 7.2 is smaller when the global sensitivity is used. Here it holds that $LS_{\hat{g}_i}(\mathcal{D}) < 2\texttt{C} = \Delta f$, which implies that the examples differing between $\mathcal{D}'$

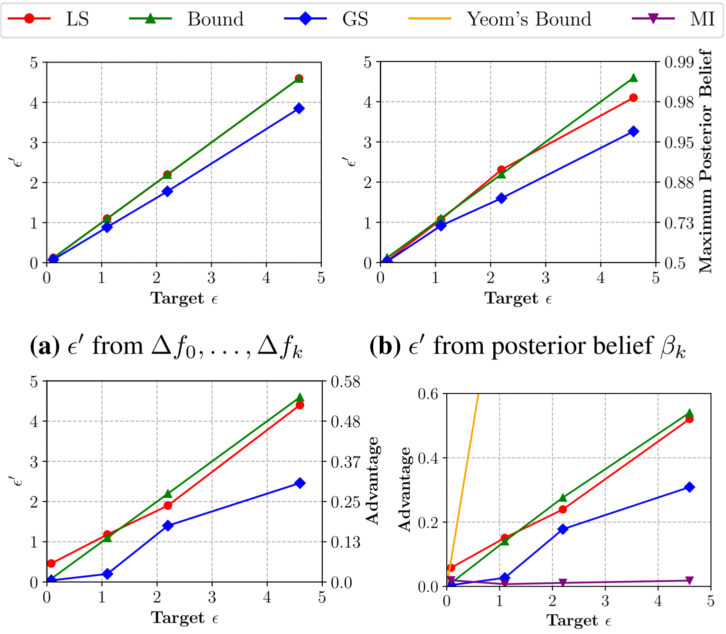
and $\mathcal{D}$ do not point in opposite directions in the bounded setting. For the unbounded DP case, this effect is not observed with the MNIST and Purchase-100 datasets. Instead, the use of local and global sensitivity leads to the same distribution of posterior beliefs and approximately the same advantage. This result stems from the fact that the per-example gradients throughout all epochs were close to or greater than $\mathtt{C} = 3$, i.e., the differentiating example in $\mathcal{D}$ must have the gradient magnitude $\mathtt{C} = 3$. However, in the Adult dataset, $LS_{\hat{g}_i}(\mathcal{D}) < \mathtt{C} = 3$, so too much noise is added using $GS_{\hat{g}}$ in the unbounded DP setting as well.

From a practical standpoint, these observations are critical, since unnecessary noise degrades the utility of the model when the global sensitivity is too large, as shown in Figure 7.6d. While all experiments were done with $\mathtt{C} = 3$, we expect a similar relationship between $LS_{\hat{g}_i}$ and $GS_{\hat{g}}$ for different values of $\mathtt{C}$, since we observed the unclipped gradients to usually be greater than $\mathtt{C} = 3$.

7.5.3. Auditing DPSGD

This section details the audit of ϵ. As shown in Section 7.4.3, the calculation of the empirical loss ϵ' can be based on (i) the local sensitivity, (ii) the posterior beliefs β_k or (iii) on the advantage $\mathrm{Adv}^{\mathrm{DI,Gau}}$. To validate that the empirical loss ϵ' is close to the target privacy loss ϵ we use the setting described in Section 7.5.2 and Table 7.3.

The resulting empirical loss ϵ' is compared to the target privacy loss ϵ for the bounded case in Figures 7.7 to 7.9. As expected Figures 7.7a, 7.8a, and 7.9a support that the privacy loss ϵ can be best estimated from the local sensitivity: the red curve lies on the ideal green curve. The estimation from the posterior beliefs is less precise as shown in Figures 7.7b, 7.8b, and 7.9b. The estimation is worst from the advantage in Figures 7.7c, 7.8c and 7.9c, where the red curve deviates most from the ideal green curve for all datasets. It is evident that the use of global sensitivity (blue lines) results in an underestimation of ϵ for all datasets. When local sensitivity is used, the small deviation from the ideal curve confirms that $\mathcal{A}_{\mathtt{DI,Gau}}$ comes close to the theoretical privacy guarantees offered by DP. A data scientist who specifies ϵ via the identifiability bounds ρ_α and ρ_β can audit ϵ using the implementation of $\mathcal{A}_{\mathtt{DI,Gau}}$. We see that in some cases $\epsilon' > \epsilon$, or equivalently $\beta_k(\mathcal{D}) > \rho_\beta$. These variations are due to the probabilistic nature of the estimation and the bound only holds with probability 1-δ. Furthermore, we observe on some occasions that $\mathrm{Adv}^{\mathrm{DI,Gau}} > \rho_\alpha$ which stems from the fact that $\mathrm{Adv}^{\mathrm{DI,Gau}}$ is an expected value for a series of experiments, which falls within a confidence interval around ρ_α.

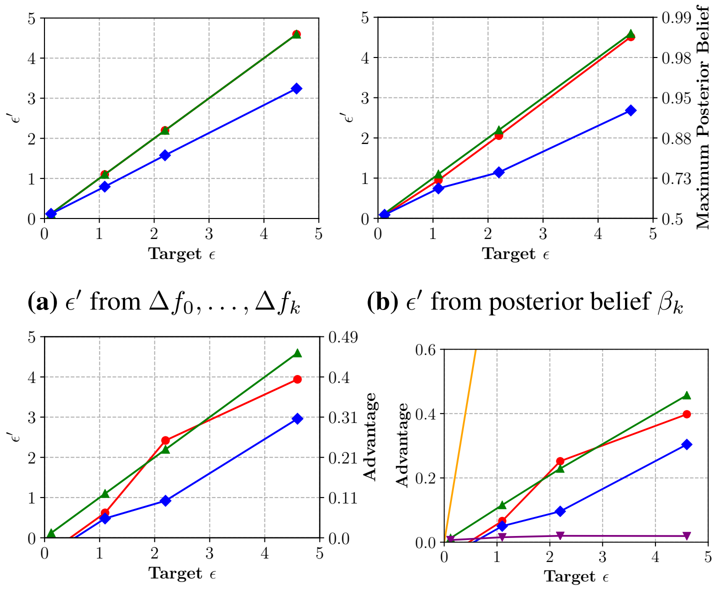


(a) ϵ' from $\Delta f_0, \ldots, \Delta f_k$ **(b)** ϵ' from posterior belief β_k

(c) ϵ' from advantage $\mathrm{Adv}^{\mathrm{DI,Gau}}$ **(d)** Comparison to MI

Figure 7.7.: Audit of ϵ (a-c) and comparison with $\mathcal{A}_{\texttt{MI}}$ (d) for MNIST data (bounded case)

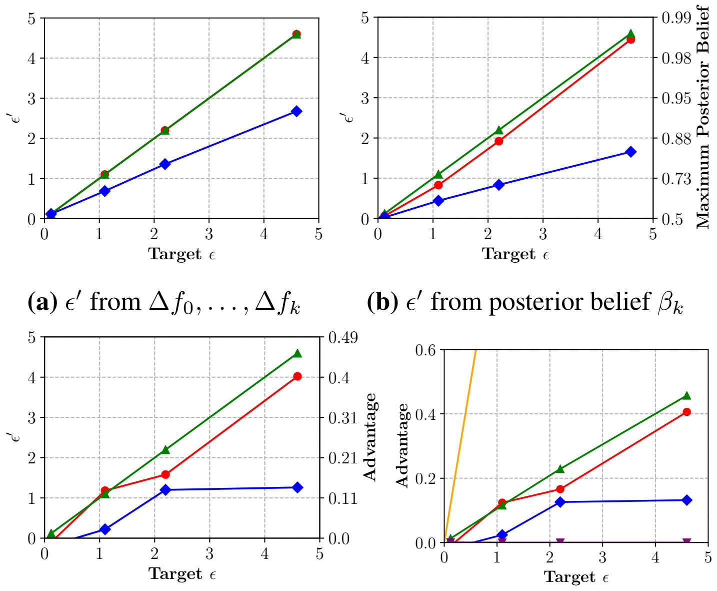
To enable comparison with membership inference we implemented $\mathcal{A}_{\texttt{MI}}$ by expanding the implementation of Jayaraman and Evans [JE19], which implements the attack suggested by Yeom et al. [Yeo+18]. Figures 7.7d, 7.8d, and 7.9d visualize the advantage resulting from both $\mathcal{A}_{\texttt{DI,Gau}}$ and $\mathcal{A}_{\texttt{MI}}$ for our setting, as well as the bounds provided by the DP guarantee and the MI bound of Yeom et al. [Yeo+18]. We see that the MI bound is very loose for all evaluated datasets, as previously noted by Jayaraman and Evans [JE19]. Furthermore, we see that our implementation of $\mathcal{A}_{\texttt{DI,Gau}}$ significantly outperforms $\mathcal{A}_{\texttt{MI}}$ on all datasets and values of ϵ.



(a) ϵ' from $\Delta f_0, \ldots, \Delta f_k$ **(b)** ϵ' from posterior belief β_k

(c) ϵ' from advantage $\mathrm{Adv}^{\mathrm{DI,Gau}}$ **(d)** Comparison to MI

Figure 7.8.: Audit of ϵ (a-c) and comparison with $\mathcal{A}_{\mathtt{MI}}$ (d) for Purchase-100 data (bounded case)



(a) ϵ' from $\Delta f_0, \ldots, \Delta f_k$ **(b)** ϵ' from posterior belief β_k

(c) ϵ' from advantage $\mathrm{Adv}^{\mathrm{DI,Gau}}$ **(d)** Comparison to MI

Figure 7.9.: Audit of ϵ (a-c) and comparison with $\mathcal{A}_{\mathtt{MI}}$ (d) for Adult data (bounded case)

7.6. Related Work

Choosing and interpreting DP privacy parameters has been addressed from several directions.

Lee and Clifton [LC11; LC12] proposed DI as a Bayesian privacy notion that quantifies ϵ w.r.t. an adversary's maximum posterior belief ρ_β on a finite set of possible input datasets. Yet, both papers focus on the scalar ϵ Laplace mechanism without composition, while we consider the (ϵ, δ) multidimensional Gauss mechanism under RDP composition. Li et al. [Li+13] demonstrate that DI matches the DP definition when an adversary decides between two neighboring datasets $\mathcal{D}, \mathcal{D}'$. Kasiviswanathan et al. [KS14] also provide a Bayesian interpretation of DP. While they also formulate posterior belief bounds and discuss local sensitivity, they do not cover expected advantage and implementation aspects such as dataset sensitivity.

The choice of privacy parameter ϵ has been tied to economic consequences. Hsu et al. [Hsu+14] derive a value for ϵ from a probability distribution over a set of negative events and the cost for compensation of affected participants. Our approach avoids the ambiguity of selecting bad events. Abowd and Schmutte [AS19] describe a social choice framework for choosing ϵ, which uses the production possibility frontier of the model and the social willingness to accept privacy and accuracy loss. We part from their work by choosing ϵ w.r.t. the advantage of the strong DP adversary. Eibl et al. [Eib+18] propose a scheme that allows energy providers and energy consumers to negotiate DP parameters by fixing a tolerable noise scale of the Laplace mechanism. The noise scale is then transformed into the individual posterior belief of the DP adversary per energy consumer. We part from their individual posterior belief analysis and suggest using the local sensitivity between two datasets that are chosen by the dataset sensitivity heuristic.

The evaluation of DP in a deep learning setting has largely focused on MI attacks [Ber+21; Che+20; Hay+19; JE19; Jay+20; Rah+18; Sho+17]. From Yeom et al. [Yeo+18] we take the idea of bounding membership advantage in terms of DP privacy parameter ϵ. However, while MI attacks evaluate the DP privacy parameters in practice, DP is defined to offer protection from far stronger adversaries, as Jayaraman et al. [JE19] empirically validated. Humphries et al. [Hum+20] derive a bound for membership advantage that is tighter than the bound derived by Yeom et al. [Yeo+18] by analyzing an adversary with additional information. Furthermore, they analyze the impact of giving up the i.i.d. assumption. Their work does not suggest an implementation of the strong DP adversary, whereas our work suggests a DP adversary implementation.

Jagielski et al. [JUO20] estimate empirical privacy guarantees based on Monte Carlo approximations. While they use active poisoning attacks to construct datasets $\mathcal{D}$ and $\mathcal{D}'$ that result in maximally different gradients under gradient clipping, we define dataset sensitivity, which does not require the introduction of malicious samples.

7.7. Summary

This chapter presented an implementation of the differential privacy adversary $\mathcal{A}_{\mathtt{DI}}$ that allows data scientists to audit ϵ when training a differentially private neural network. Furthermore, we present a transformation of the privacy parameter ϵ to identifiability bounds. $\mathcal{A}_{\mathtt{DI}}$ diverges from other attacks against DP or neural networks, such as membership inference, which necessitates a discussion of $\mathcal{A}_{\mathtt{DI}}$'s properties in relation to alternative approaches. Our goal is to construct an adversary that most closely challenges DP, and can be connected to societal norms and legislation via identifiability scores. To this end, $\mathcal{A}_{\mathtt{DI}}$ knows all but one element of the training data and the gradients at every update step. Since the DP guarantee must hold in the presence of all auxiliary information, both of these assumptions relate the attack model $\mathcal{A}_{\mathtt{DI}}$ directly to the DP guarantee. Since $\mathcal{A}_{\mathtt{DI}}$ has knowledge of all but one element instead of only the distribution, $\mathcal{A}_{\mathtt{DI}}$ possesses significantly more information than $\mathcal{A}_{\mathtt{MI}}$. A natural question arises w.r.t. $\mathcal{A}_{\mathtt{DI}}$'s practical relevance. We see this work relevant for the federated learning setting in which $\mathcal{A}_{\mathtt{DI}}$ receives the gradients during every update step as a participating data owner. If the data owner would not receive frequent gradient updates but solely a trained model, the attack by $\mathcal{A}_{\mathtt{DI}}$ would be mitigated. Furthermore, $\mathcal{A}_{\mathtt{DI}}$ could realistically obtain knowledge of a significant portion of the training data, since public reference data is often used in training datasets and only extended with some custom training data records, necessitating the notion of DP in general.

To further comment on the utility that can be achieved from a differentially private model, we note that the optimal choice for $\mathtt{C}$ may stray from the original recommendation of Abadi et al. [Aba+16]. We follow this recommendation and set $\mathtt{C} = 3$, which limits the utility loss that results when $\mathtt{C}$ is too large (unnecessary noise addition) and too small (loss of information about the gradient). Since this balance holds for unbounded

DP and does not consider the notion of local sensitivity, we expect that a different $\mathtt{C}$ may yield better utility than what we report. Varying $\mathtt{C}$ may also change the balance between local sensitivity and global sensitivity from Figures 7.7 to 7.9. Furthermore, since gradients change throughout training, the optimal value of $\mathtt{C}$ at the beginning of training may no longer be optimal toward the end of training according to McMahan et al. [McM+18]. Setting the clipping norm adaptively as suggested by Thakkar et al. [TAM19] may improve utility by changing $\mathtt{C}$ as training progresses. We expect that doing so might bring ϵ' closer to ϵ when auditing the DP guarantee, and achieve similar by using local sensitivity.

In summary, we defined two identifiability bounds for the DP adversary in ML with DPSGD: maximum posterior belief ρ_β and expected membership advantage ρ_α. These bounds can be transformed into privacy parameter ϵ. In consequence, with ρ_α and ρ_β, data owners and data scientists can map legal and societal expectations w.r.t. identifiability to corresponding DP privacy parameters. Furthermore, we implemented an instance of the DP adversary for ML with DPSGD and showed that it allows us to audit parameter ϵ. We evaluated the effect of sensitivity in DPSGD and showed that our upper bounds are reached under multidimensional queries with composition. To reach the bounds, sensitivity must reflect the local sensitivity of the dataset. We approximate the local sensitivity for DPSGD with a heuristic, improving the utility of the differentially private model when compared to the use of global sensitivity. The chapter thus extends our previous work on research Problem 1 in Chapter 5 by considering the DP adversary $\mathcal{A}_{\mathtt{DI}}$ instead of $\mathcal{A}_{\mathtt{MI}}$, and addresses research Problem 2 by introducing the identifiability scores expected membership advantage ρ_α and maximum posterior belief ρ_β.

8. Conclusion

This chapter draws conclusions based on the addressed research problems and observed results. We will furthermore discuss possibilities for future work. This thesis addressed the question of how to balance the privacy-accuracy trade-off in differentially private machine learning with neural networks. In particular, two problems were approached. First, Problem 1 regarding quantification of the empirical lower bound on the privacy loss under membership inference attacks to allow data scientists to compare the privacy-accuracy trade-off between local and central differential privacy. Secondly, Problem 2 regarding the transformation of the privacy loss under differential privacy into an analytical bound on identifiability, to connect differential privacy guarantees to social norms and regulation. We provide the following contributions to ease Problem 1.

- Comparing the lower bound on privacy loss in LDP and CDP by the average precision of their MI precision-recall curve, and showing that under this measure LDP and CDP have similar privacy-accuracy trade-offs despite vastly different ϵ.
- Demonstrating that CDP mechanisms are not achieving a consistently better privacy-accuracy trade-off on various datasets and reference models. The trade-off rather depends on the specific dataset.
- Quantifying the relative privacy-accuracy trade-off and showing that it is not constant over ϵ, but there are dataset specific ranges where the relative trade-off is greater for protection against MI than target model test accuracy.

For Problem 2 on measuring the lower and upper bound on the privacy loss in terms of identifiability, this thesis provides the following contributions.

- Identifiability bounds for the posterior belief and expected membership advantage that are mathematical transformations of privacy parameters (ϵ, δ) and used in conjunction with RDP composition.

- Practical implementation of an adversary that meets all assumptions on worst-case adversaries against DP and allows us to audit DPSGD model instances w.r.t. to the lower bound privacy loss.
- A heuristic for scaling sensitivity in differentially private stochastic gradient descent. This heuristic leads to tight bounds on identifiability.

Overall, we see evidence that methods for quantifying the trade-off between utility and privacy over ϵ provide information for interpreting differentially private datasets or functions, and hence ease the problem of choosing ϵ. However, the findings of this thesis face similar constraints as those found in interpretable machine learning in general. Namely data specific insights and explanations that underlay statistical uncertainty. We reduced the issue by considering datasets from multiple domains and repeating experiments. However, if extending the experiments of this study to more datasets per domain the trade-offs could be validated w.r.t. similarity per domain, and domain-specific implications on the privacy parameter ϵ.

We see three questions arising for further research which we discuss in the following. First, while differential privacy is a technique that offers a data scientist to create anonymized data from a legal perspective [Par14], anonymization might not always be required (i.e., when processing non-personal data). In such cases, mitigations against attacks such as membership inference can also be achieved by using regularization techniques [LOK21; NSH18]. A comparison of the experiments within this thesis for differential privacy and regularization under φ would contribute towards this research direction w.r.t. discriminative and generative models. Second, several model architectures with comparable accuracy are available for some learning tasks (e.g., plain neural networks, deep convolutional neural networks, and transformer based networks for text classification). Similar to the central and federated learning experiments in Chapter 5 we observed these varying model architectures to yield quite different privacy-accuracy trade-offs in a related study that we did not include in this thesis [Wun+21]. Taking into consideration the scale of the training data required for meaningful test accuracy for available models, and the model's robustness towards noise when choosing a model architecture for differentially private machine learning with neural networks can hence lead to smaller privacy losses at comparable accuracy to alternative model architectures. Third, we discuss two particular adversaries in this thesis, namely the membership inference adversary $\mathcal{A}_{\texttt{MI}}$ and the differential identifiability adversary $\mathcal{A}_{\texttt{DI}}$. These adversaries differ significantly in the assumptions they make about adversarial knowledge of the training data. Such differences can also be found between white-box and black-box MI adversaries and between membership inference and model inversion adversaries. We observed at least for membership inference adversaries that the significant increase in assumptions of the adversary on training data only leads to a small increase in membership inference attack performance, at a large increase in computational effort. Hence, recommendations for the practitioner w.r.t. computationally efficient estimations of lower bound would support the use of such simulations in practice.

A. Appendix

A.1. Supervised Theses

The theses stated in Table A.1 have been suggested to thesis candidates and were also supervised by the author of this thesis.

Table A.1.: Proposed and supervised theses

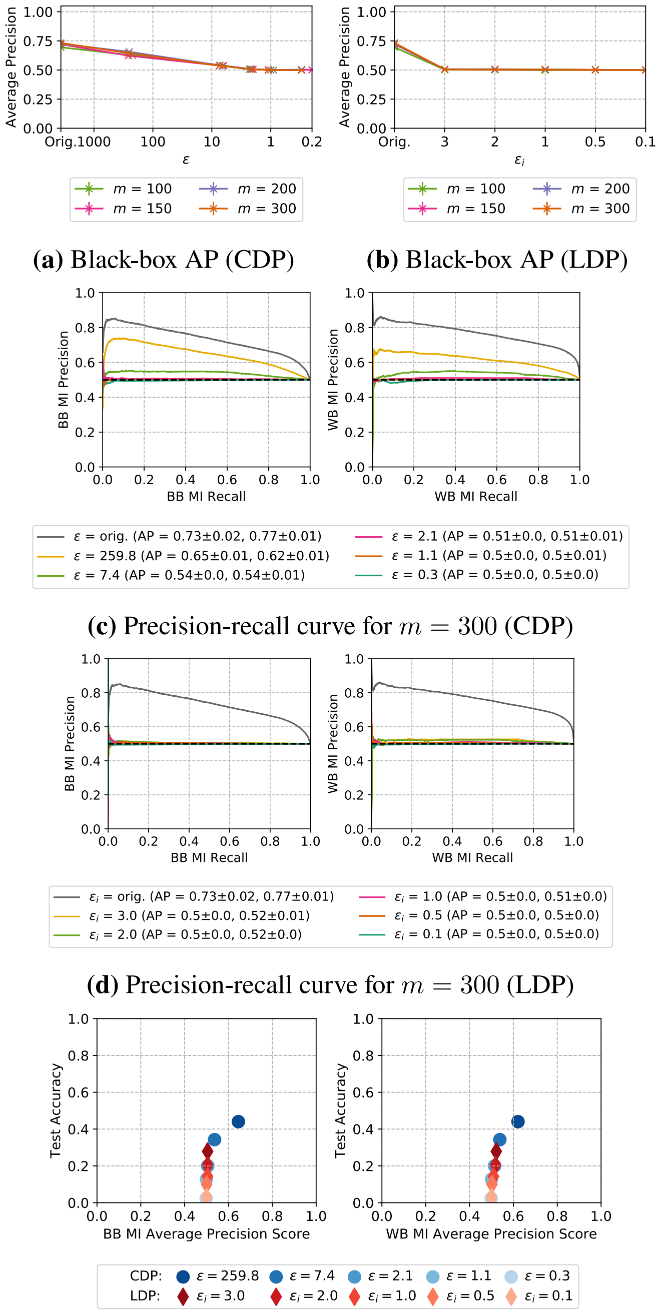
Name	Thesis	Institution	2nd Supervisor at Institution	Year
Wasilij Beskorovajnov	General-Purpose Anonymization with Differential Privacy and Randomized Response	Karlsruhe Institute of Technology	Prof. Dr. Joern Müller-Quade	2017
Philip-William Grassal	Evaluation of an Attacker-Based Approach to Rationally Parameterize Differentially Private Mechanisms	Baden-Württemberg Cooperative State University Karlsruhe	Prof. Dr. Thomas Freytag	2017
Jonas Robl	Evaluating Membership Inference Attacks on Differentially Private Neural Networks	Baden-Württemberg Cooperative State University Karlsruhe	Dr. Svetlana Meissner	2018
Hannah Keller	Risk-based Metrics for Differentially Private Deep Learning	Ludwigshafen University of Applied Sciences	NA	2019
Steffen Schneider	Membership Inference Attacks on Differentially Private Neural Networks	Maastricht University	Prof. Dr. Siamak Mehrkanoon	2020
Dominik Wunderlich	Differentially Private Hierarchical Text Classification	Karlsruhe Institute of Technology	Prof. Dr. Thorsten Strufe	2020
Tom Ganz	Assessing and Selecting ϵ for Differentially Private Federated Learning with Inference Attacks	Karlsruhe University of Applied Sciences	Prof. Dr. Astrid Laubenheimer	2020
Jonas Robl	Balancing Privacy and Utility in Differentially Private Generative Models by Inference Attacks	Heidelberg University	Prof. Dr. Vincent Heuveline	2021

A.2. White-Box MI Experiment Hyperparameters

Table A.2.: Target model training accuracy (from orig. to smallest ϵ), CDP ϵ values (from $z = 0.5$ to $z = 16$) and hyperparameters

		Texas Hospital Stays				Purchases Shopping Carts				LFW			Skewed Purchases			
m		100	150	200	300	10	20	50	100	20	50	100	10	20	50	100
LDP		0.86 1.0 1.0 1.0 0.99 0.82	0.92 1.0 1.0 1.0 0.95 0.71	0.83 1.0 1.0 0.98 0.86 0.59	0.81 1.0 1.0 0.92 0.72 0.53	0.99 0.97 0.88 0.64 0.58 0.44	1.0 0.97 0.85 0.58 0.47 0.38	1.0 0.95 0.86 0.69 0.62 0.49	0.99 0.94 0.90 0.79 0.75 0.51	1.0 1.0 1.0 0.22 0.24 0.25	1.0 1.0 0.96 0.18 0.17 0.17	1.0 1.0 1.0 0.13 0.13 0.13	1.0 1.0 1.0 1.0 0.93 0.52	1.0 1.0 1.0 0.99 0.98 0.55	1.0 1.0 1.0 0.97 0.9 0.71	1.0 0.99 0.97 0.89 0.80 0.45
CDP		0.86 0.74 0.57 0.35 0.22 0.05	0.92 0.75 0.54 0.31 0.19 0.04	0.83 0.69 0.48 0.26 0.16 0.03	0.81 0.62 0.42 0.22 0.13 0.02	1.0 0.95 0.91 0.80 0.69 0.28	1.0 0.91 0.84 0.69 0.51 0.14	1.0 0.82 0.71 0.46 0.28 0.05	0.99 0.63 0.51 0.27 0.14 0.02	1.0 0.99 0.76 0.44 0.36 0.32	1.0 0.87 0.5 0.28 0.23 0.19	1.0 0.79 0.35 0.25 0.18 0.13	1.0 1.0 1.0 0.92 0.89 0.66	1.0 1.0 0.96 0.8 0.64 0.24	1.0 0.97 0.6 0.25 0.12 0.03	1.0 0.58 0.1 0.02 0.02 0.01
ϵ		222.6 6.3 2.3 0.9 0.3	259.8 6.6 2.0 1.1 0.2	251.5 7.3 2.2 1.0 0.3	259.8 7.4 2.1 1.1 0.3	88.1 4.6 2.0 1.3 0.4	88.1 4.1 1.8 1.2 0.4	88.1 4.1 1.8 1.2 0.4	88.1 4.1 1.8 1.2 0.3	84.3 4.8 2.1 1.3 0.5	70.4 3.9 1.7 0.8 0.4	62.4 3.4 1.5 1.0 0.3	28.9 1.6 0.7 0.9 0.4	29.8 1.7 1.6 0.9 0.4	42.2 3.5 1.3 0.7 0.3	73.5 2.1 1.3 0.6 0.3
Learning rate	Orig.	0.01	0.01	0.01	0.01	1e-3	1e-3	1e-3	1e-3	1e-3	1e-3	1e-3	1e-3	1e-3	1e-3	1e-3
	CDP	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	1e-3	8e-4	8e-4	1e-3	1e-3	1e-3	1e-3
	LDP	0.01	0.01	0.01	0.01	1e-3	1e-3	1e-3	1e-3	1e-3	1e-3	1e-3	1e-3	1e-3	1e-3	1e-3
Batch size	Orig.	128	128	128	128	128	128	128	128	32	32	32	100	100	100	100
	CDP	128	128	128	128	128	128	128	128	16	16	16	100	100	100	100
	LDP	128	128	128	128	128	128	128	128	32	32	32	100	100	100	100
Epochs	Orig.	200	200	200	200	200	200	200	200	30	30	30	200	200	200	200
	CDP	1000	1000	1000	1000	200	200	200	200	110	110	110	200	200	200	200
	LDP	200	200	200	200	200	200	200	200	30	30	30	200	200	200	200
Clipping Norm	CDP	4	4	4	4	4	4	4	4	3	3	3	4	4	4	4

A.3. Black-Box MI Experiments



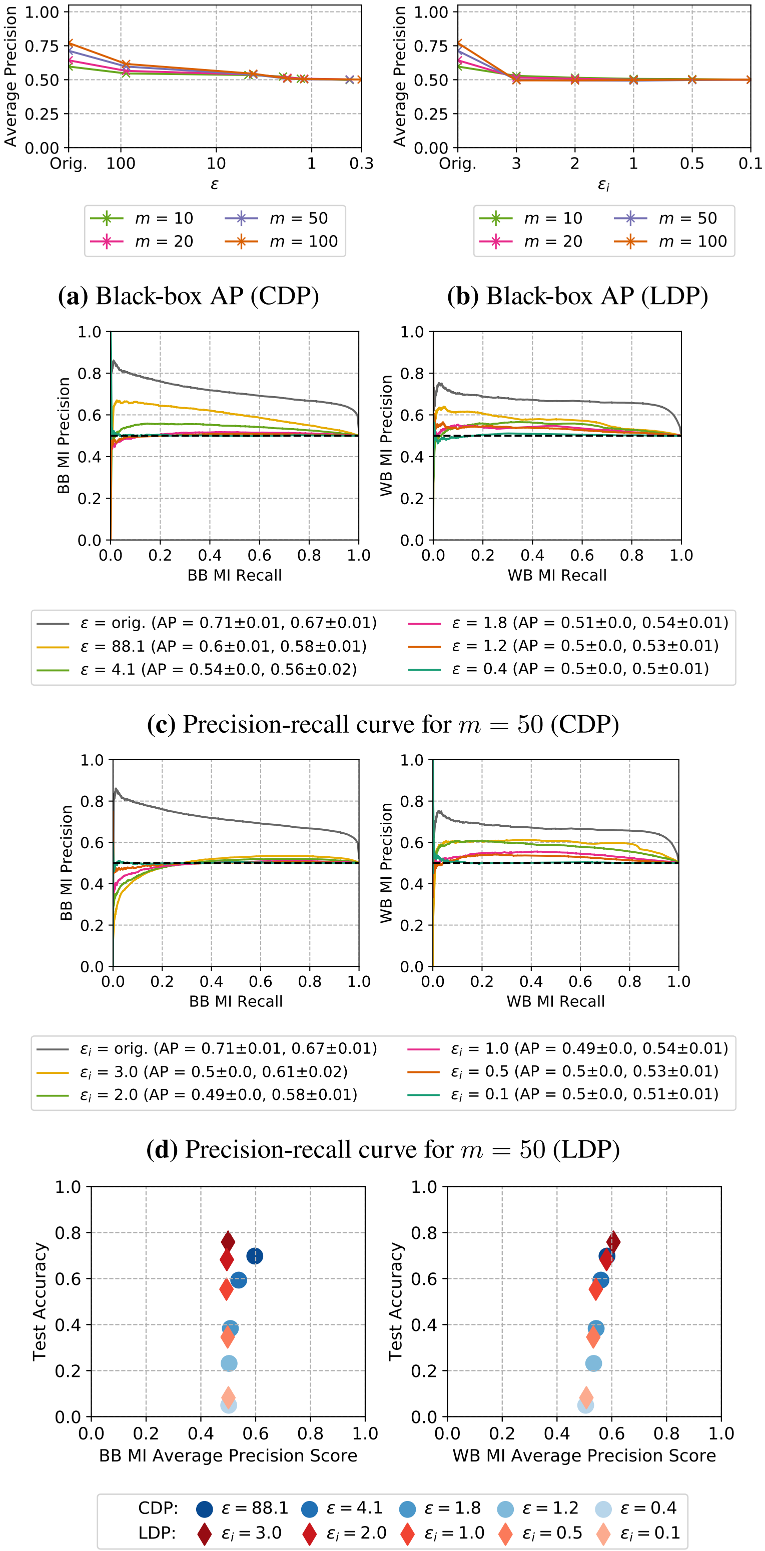
(a) Black-box AP (CDP) **(b)** Black-box AP (LDP)

(c) Precision-recall curve for $m = 300$ (CDP)

(d) Precision-recall curve for $m = 300$ (LDP)

(e) BB and WB privacy-accuracy trade-off for $m = 300$

Figure A.1.: Texas Hospital Stays accuracy and privacy (error bars lie within most points)



(a) Black-box AP (CDP)

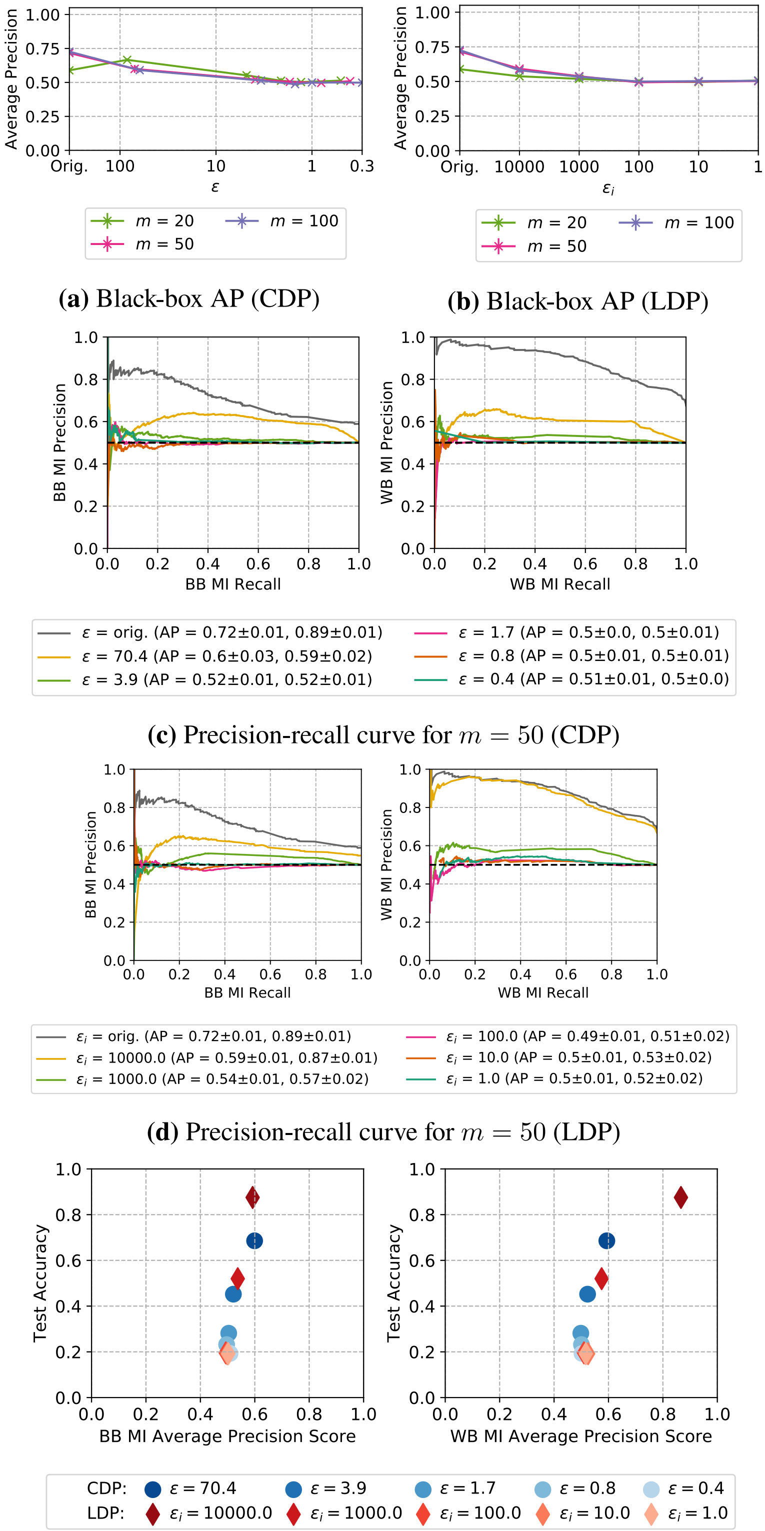
(b) Black-box AP (LDP)

(c) Precision-recall curve for $m = 50$ (CDP)

(d) Precision-recall curve for $m = 50$ (LDP)

(e) BB and WB privacy-accuracy trade-off comparison for $m = 50$

Figure A.2.: Purchases accuracy and privacy (error bars lie within most points)



(a) Black-box AP (CDP)

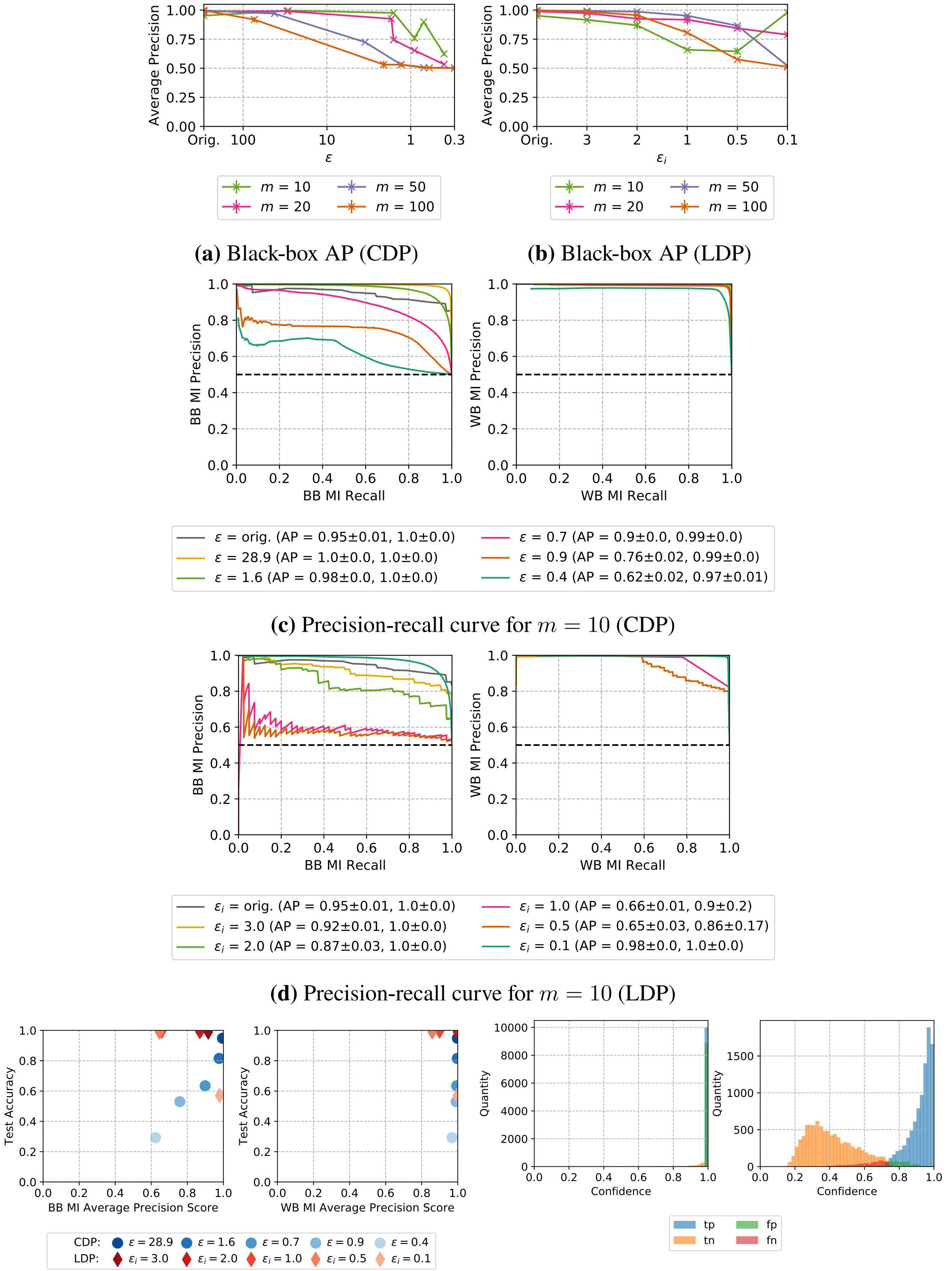
(b) Black-box AP (LDP)

(c) Precision-recall curve for $m = 50$ (CDP)

(d) Precision-recall curve for $m = 50$ (LDP)

(e) BB and WB privacy-accuracy trade-off comparison for $m = 50$

Figure A.3.: LFW accuracy and privacy (error bars lie within most points)



(a) Black-box AP (CDP)

(b) Black-box AP (LDP)

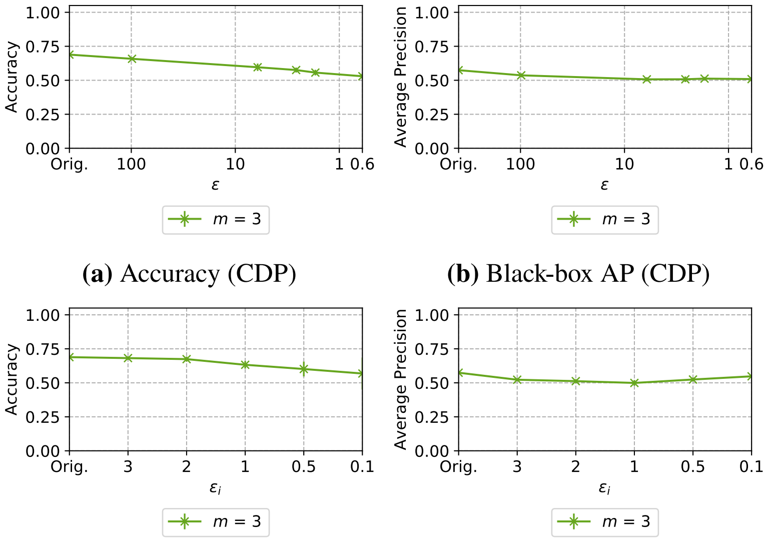
(c) Precision-recall curve for $m = 10$ (CDP)

(d) Precision-recall curve for $m = 10$ (LDP)

(e) BB and WB privacy-accuracy trade-off comparison for $m = 10$

(f) LDP target model decisive confidence according to BB attack model classification for $m = 10$, $\epsilon = 1.0$ (left) and $\epsilon = 0.1$ (right)

Figure A.4.: Skewed Purchases accuracy and privacy (error bars lie within most points)



(a) Accuracy (CDP)

(b) Black-box AP (CDP)

(c) Accuracy (LDP)

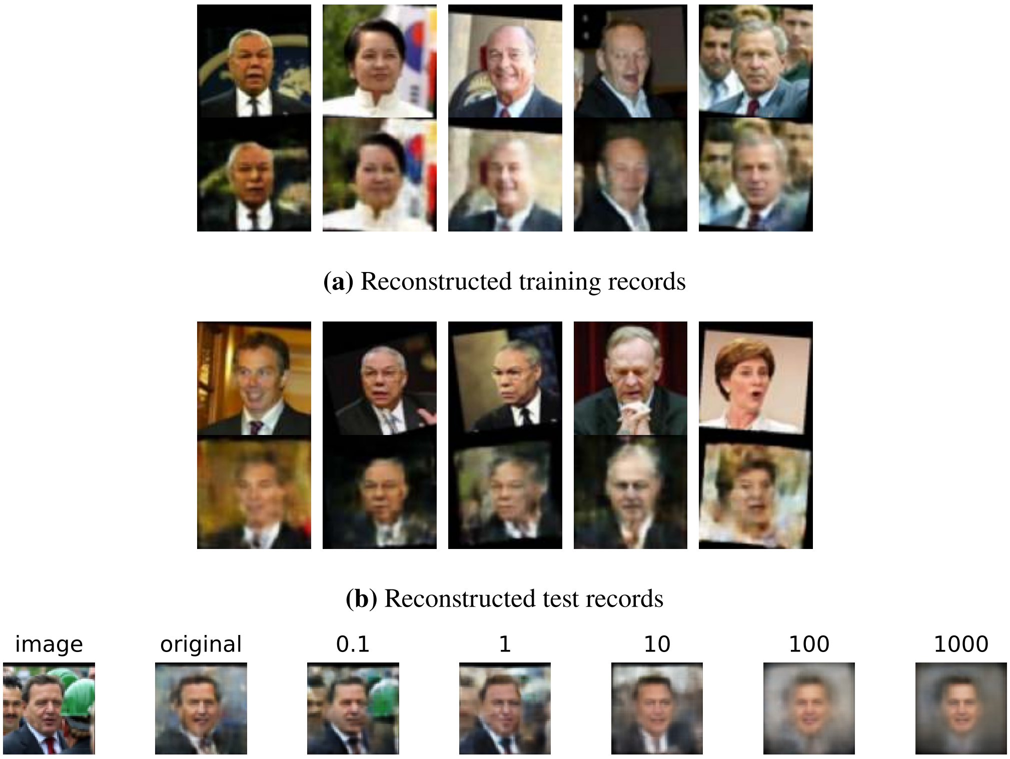
(d) Black-box AP (LDP)

Figure A.5.: COLLAB accuracy and privacy (error bars lie within most points)

A.4. VAE MI Experiments

Table A.3.: Target classifier hyperparameters

		Orig., CDP	LDP					VAE-LDP				
			10000	5000	1000	500	100	0.1	1	10	100	1000
LFW20	learning rate	2.4e-05	2.44e-4	8.58e-05	3.66e-05	2.35e-4	1.43e-05	4.03e-4	1.42e-4	9.34e-05	1.39e-4	1.38e-3
	batch size	16	16	16	16	16	64	16	64	64	16	64
	epochs	33	100	10	97	16	24	49	34	50	45	46
	test accuracy	0.98	0.97	0.97	0.82	0.55	0.28	0.94	0.93	0.98	1	1
LFW50	learning rate	1e-05	3.5e-05	4.41e-4	1.85e-4	1.72e-4	1e-05	9.24e-05	3.29e-05	7.39e-05	9.76e-4	1.27e-4
	batch size	16	16	64	64	64	64	16	16	16	64	32
	epochs	100	96	100	35	90	21	49	37	10	32	20
	test accuracy	0.95	0.94	0.93	0.7	0.41	0.2	0.9	0.91	0.97	1	1
MS	ϵ_i		10	1	0.5	0.1	0.01	0.1	1	10	100	1000
	learning rate	9.8e-4	9.37e-4	7.26e-4	7.72e-4	9.87e-05	1.08e-05	1.09e-3	6.75e-4	1.14e-4	2.48e-3	3.71e-05
	batch size	64	64	64	16	16	16	256	128	32	32	64
	epochs	25	25	25	25	25	6	21	9	23	16	24
	test accuracy	0.99	0.98	0.93	0.8	0.29	0.25	0.68	0.53	0.39	0.3	0.24



(a) Reconstructed training records

(b) Reconstructed test records

(c) VAE-LDP generated samples for LFW20

Figure A.6.: Comparison of reconstructed records and generated samples

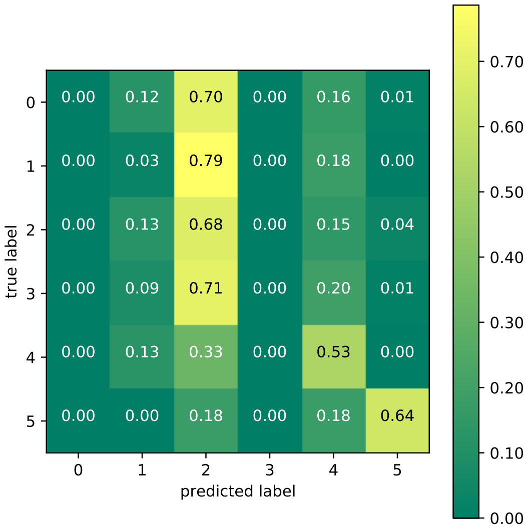


Figure A.7.: Confusion matrix for the target classifier for MotionSense CDP $z = 1$

Table A.4.: Target model hyperparameters, CDP and VAE-LDP (ϵ, δ), and LDP ϵ

		Orig.	CDP				
z			0.001	0.01	0.1	0.5	1
LFW20	learning rate	5.57e-4	4.67e-4	1.82e-4	1.1e-4	5.62e-4	4.01e-05
	batch size	16	32	16	64	16	32
	epochs	1000	1000				
	C, microbatch	-	0.03, 4				
	(ϵ, δ)	-	(15948925900.96, 1.08e-03)	(321853746.70, 1.08e-03)	(372950.22, 1.08e-03)	(295.07, 1.08e-03)	(56.41, 1.08e-03)
	train-test gap	260.3	210.6	62.5	13.8	17.9	10.5
LFW50	learning rate	5.88e-4	2.15e-4	1.04e-4	5.14e-05	1.93e-4	6.86e-4
	batch size	16	16		32		
	epochs	1000	1000				
	C, microbatch	-	0.02, 4				
	(ϵ, δ)	-	(47295786259.73, 7.27e-04)	(468786259.73, 7.27e-04)	(681781.38, 7.27e-04)	(353.74, 7.27e-04)	(43.31, 7.27e-04)
	train-test gap	259.4	195.4	29.4	9.2	7.1	33
MS	learning rate	1e-3					
	batch size	32					
	epochs	1000					
	C, microbatch	-	3.4e-5, 4				
	(ϵ, δ)	-	(120986947509.93, 1.42e-04)	(1196947509.93, 1.42e-04)	(1093201.38, 1.42e-04)	(137.57, 1.42e-04)	(15.73, 1.42e-04)
	train-test gap	0.7	0.4	0.3	0.1	0.1	0
LDP							
ϵ_i			10000	5000	1000	500	100
LFW20	learning rate		9.22e-4	1.52e-4	2.13e-4	1.14e-4	1e-3
	batch size		32	16	64	32	16
	epochs		1000				
	ϵ		5.718e+07	2.859e+07	5.718e+06	2.859e+06	571800
	train-test gap		267	265	224	160	123
LFW50	learning rate		4.61e-4	2.41e-4	4.31e-4	1.19e-05	1e-05
	batch size		16		64		
	epochs		1000				
	ϵ		8.319e+07	4.1595e+07	8.319e+06	4.1595e+06	831900
	train-test gap		272	264	204	21	5
ϵ_i			10	1	0.5	0.1	0.01
MS	learning rate		1e-3				
	batch size		32				
	epochs		1000				
	ϵ		706190	70619	35309.5	7061.9	706.19
	train-test gap		0.7	0.9	2.7	4.4	4.8
VAE-LDP							
σ			0.1	1	10	100	1000
LFW20	learning rate		5.57e-4				
	batch size		16				
	epochs		1000				
	(ϵ, δ)		(2366.15, 5.25e-04)	(236.61, 5.25e-04)	(23.66, 5.25e-04)	(2.37, 5.25e-04)	(0.24, 5.25e-04)
	train-test gap		156	145	64	3	2
LFW50	learning rate		5.88e-4				
	batch size		16				
	epochs		1000				
	(ϵ, δ)		(2422.52, 3.61e-04)	(242.25, 3.61e-04)	(24.23, 3.61e-04)	(2.42, 3.61e-04)	(0.24, 3.61e-04)
	train-test gap		168	158	68	4	3
MS	learning rate		1e-3				
	batch size		32				
	epochs		1000				
	(ϵ, δ)		(404.96, 1.42e-05)	(40.50, 1.42e-05)	(4.05, 1.42e-05)	(0.40, 1.42e-05)	(0.04, 1.42e-05)
	train-test gap		0.1	0	0.2	0.1	0

Bibliography

[Aba+16] Martín Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. "Deep Learning with Differential Privacy". In: *Proceedings of the Conference on Computer and Communications Security*. CCS. New York, NY, USA: ACM Press, 2016.

[AS19] John M. Abowd and Ian M. Schmutte. "An Economic Analysis of Privacy Protection and Statistical Accuracy as Social Choices". In: *American Economic Review* 109.1 (2019).

[Aga+18] Naman Agarwal, Ananda Theertha Suresh, Felix Yu, Sanjiv Kumar, and H. Brendan McMahan. "cpSGD: Communication-Efficient and Differentially-Private Distributed SGD". In: *Advances in Neural Information Processing Systems*. NeurIPS. Red Hook, NY, USA: Curran Associates Inc., 2018.

[Bac+16] Michael Backes, Pascal Berrang, Mathias Humbert, and Praveen Manoharan. "Membership Privacy in MicroRNA-Based Studies". In: *Proceedings of the Conference on Computer and Communications Security*. CCS. New York, NY, USA: ACM Press, 2016.

[BS18] Shane Barratt and Rishi Sharma. *A Note on the Inception Score*. 2018. arXiv: `1801.01973 [stat.ML]`.

[BST14] Raef Bassily, Adam Smith, and Abhradeep Thakurta. "Private Empirical Risk Minimization". In: *Proceedings of the Symposium on Foundations of Computer Science*. SFCS. Piscataway, NJ, USA: IEEE Computer Society, 2014.

[Ber+22] Daniel Bernau, Günther Eibl, Philip W. Grassal, Hannah Keller, and Florian Kerschbaum. "Quantifying Identifiability to Choose and Audit ϵ in

Differentially Private Deep Learning". In: *Proceedings of the Conference on Very Large Databases*. VLDB. 2022.

[Ber+21] Daniel Bernau, Jonas Robl, Philip W. Grassal, Steffen Schneider, and Florian Kerschbaum. "Comparing Local and Central Differential Privacy Using Membership Inference Attacks". In: *Proceedings of the Conference on Data and Applications Security and Privacy*. DBSEC. Cham: Springer International Publishing, 2021.

[BRK22] Daniel Bernau, Jonas Robl, and Florian Kerschbaum. "Assessing Differentially Private Variational Autoencoders under Membership Inference". In: *Proceedings of the Conference on Data and Applications Security and Privacy*. DBSEC. Cham: Springer International Publishing, 2022.

[BBK17] Jonas Böhler, Daniel Bernau, and Florian Kerschbaum. "Privacy-Preserving Outlier Detection for Data Streams". In: *Proceedings of the Conference on Data and Applications Security and Privacy*. DBSEC. Cham: Springer International Publishing, 2017.

[Car+19] Nicholas Carlini, Chang Liu, Jernej Kos, Úlfar Erlingsson, and Dawn Song. "The Secret Sharer: Measuring Unintended Neural Network Memorization and Extracting Secrets". In: *Proceedings of the USENIX Security Symposium*. Berkeley, CA, USA: USENIX Association, 2019.

[Che+20] Dingfan Chen, Ning Yu, Yang Zhang, and Mario Fritz. "GAN-Leaks: A Taxonomy of Membership Inference Attacks against Generative Models". In: *Proceedings of the Conference on Computer and Communications Security*. CCS. New York, NY, USA: ACM Press, 2020.

[CT13] Chris Clifton and Tamir Tassa. "On Syntactic Anonymity and Differential Privacy". In: *Proceedings of the Conference on Data Engineering Workshops*. ICDEW. Piscataway, NJ, USA: IEEE Computer Society, 2013.

[Cyb89] George Cybenko. "Approximation by Superpositions of a Sigmoidal Function". In: *Mathematics of Control, Signals and Systems* 2.4 (1989), pp. 303–314.

[DG06] Jesse Davis and Mark Goadrich. “The Relationship between Precision-Recall and ROC Curves”. In: *Proceedings of Conference on Machine Learning*. ICML. New York, NY, USA: ACM Press, 2006.

[Dev+19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. NAACL-HLT. Association for Computational Linguistics, 2019.

[Dwo06] Cynthia Dwork. “Differential Privacy”. In: *Proceedings of the International Colloquium on Automata, Languages and Programming*. ICALP. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006.

[Dwo+06] Cynthia Dwork, Krishnaram Kenthapadi, Frank D. McSherry, Ilya Mironov, and Moni Naor. “Our Data, Ourselves: Privacy Via Distributed Noise Generation”. In: *Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques*. EUROCRYPT. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006.

[DKM19] Cynthia Dwork, Nitin Kohli, and Deirdre Mulligan. “Differential Privacy in Practice: Expose your Epsilons!” In: *Journal of Privacy and Confidentiality* 9.2 (Oct. 2019).

[Dwo+11] Cynthia Dwork, Frank D. McSherry, Kobbi Nissim, and Adam Smith. “Differential Privacy – A Primer for the Perplexed”. In: *Proceedings of the Conference of European Statisticians*. CES. 2011.

[DR14] Cynthia Dwork and Aaron Roth. “The Algorithmic Foundations of Differential Privacy”. In: *Foundations and Trends in Theoretical Computer Science* 9.3-4 (2014).

[DR16] Cynthia Dwork and Guy N. Rothblum. *Concentrated Differential Privacy*. 2016. arXiv: `1603.01887 [cs.DS]`.

[DRV10] Cynthia Dwork, Guy N. Rothblum, and Salil Vadhan. “Boosting and Differential Privacy”. In: *Proceedings of the Symposium on Foundations of Computer Science*. SFCS. Piscataway, NJ, USA: IEEE Computer Society, 2010.

[Eib+18] Günther Eibl, Kaibin Bao, Philip W. Grassal, Daniel Bernau, and Hartmut Schmeck. “The Influence of Differential Privacy on Short Term Electric Load Forecasting”. In: *Energy Informatics* 1.1 (2018).

[EPK14] Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. “RAPPOR: Randomized Aggregatable Privacy-Preserving Ordinal Response”. In: *Proceedings of the Conference on Computer and Communications Security*. CCS. New York, NY, USA: ACM Press, 2014.

[EH10] Tim van Erven and Peter Harremoës. “Rényi Divergence and Majorization”. In: *Proceedings of the Symposium on Information Theory*. ISIT. Piscataway, NJ, USA: IEEE Computer Society, 2010.

[Eve+10] Mark Everingham, Luc Gool, Christopher K. Williams, John Winn, and Andrew Zisserman. “The Pascal Visual Object Classes Challenge”. In: *International Journal of Computer Vision* 88.2 (2010).

[Fan18] Liyue Fan. “Image Pixelization with Differential Privacy”. In: *Proceedings of the Conference on Data and Applications Security and Privacy*. DBSEC. Cham: Springer International Publishing, 2018.

[FJR15] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. “Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures”. In: *Proceedings of the Conference on Computer and Communications Security*. CCS. New York, NY, USA: ACM Press, 2015.

[Fre+14] Matt Fredrikson, Eric Lantz, Somesh Jha, Simon Lin, David Page, and Thomas Ristenpart. “Privacy in Pharmacogenetics: An End-to-End Case Study of Personalized Warfarin Dosing”. In: *Proceedings of the USENIX Security Symposium*. Berkeley, CA, USA: USENIX Association, 2014.

[Fri+19] Lorenzo Frigerio, Anderson Santana de Oliveira, Laurent Gomez, and Patrick Duverger. “Differentially Private Generative Adversarial Networks for Time Series, Continuous, and Discrete Open Data”. In: *Proceedings of the Conference on ICT Systems Security and Privacy Protection*. Springer, 2019.

[GAP18] Simson L. Garfinkel, John M. Abowd, and Sarah Powazek. “Issues Encountered Deploying Differential Privacy”. In: *Proceedings of the Workshop on Privacy in the Electronic Society*. WPES. New York, NY, USA: ACM Press, 2018.

[GBC16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. `https://www.deeplearningbook.org`. MIT Press, 2016.

[Goo+14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. “Generative Adversarial Nets”. In: *Proceedings of the Conference on Neural Information Processing Systems*. NIPS. Red Hook, NY, USA: Curran Associates Inc., 2014.

[GSS15] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. “Explaining and Harnessing Adversarial Examples”. In: *Proceedings of the Conference on Learning Representations*. ICLR. Piscataway, NJ, USA: IEEE Computer Society, 2015.

[GC95] Yves Grandvalet and Stephane Canu. “Comments on ‘Noise Injection into Inputs in Back Propagation Learning’”. In: *IEEE Transactions on Systems, Man, and Cybernetics* 25.4 (1995).

[Gre17] Andy Greenberg. *How One of Apple’s Key Privacy Safeguards Falls Short*. 2017. URL: `https://www.wired.com/story/apple-differential-privacy-shortcomings/` (visited on 01/19/2022).

[Hay+16] Michael Hay, Ashwin Machanavajjhala, Gerome Miklau, Yan Chen, and Dan Zhang. “Principled Evaluation of Differentially Private Algorithms Using DPBench”. In: *Proceedings of the Conference on Management of Data*. SIGMOD. New York, NY, USA: ACM Press, 2016.

[Hay+19] Jamie Hayes, Luca Melis, George Danezis, and Emiliano De Cristofaro. “LOGAN: Membership Inference Attacks Against Generative Models”. In: *Proceedings of the Privacy Enhancing Technologies Symposium*. PETS. 2019.

[HS10] American Department of Health and Human Services. *Guidance Regarding Methods for De-identification of Protected Health Information in Accordance with the Health Insurance Portability and Accountability Act (HIPAA) Privacy Rule*. 2010. URL: https://www.hhs.gov/hipaa/for-professionals/privacy/special-topics/de-identification/index.html (visited on 06/03/2021).

[Heb49] Donald O. Hebb. *The Organization of Behavior: A Neuropsychological Theory*. New York, NY, USA: Wiley, 1949.

[Hil12] Kashmir Hill. “How Target Figured Out a Teen Girl was Pregnant before her Father did”. In: (2012). URL: https://www.forbes.com/sites/kashmirhill/2012/02/16/how-target-figured-out-a-teen-girl-was-pregnant-before-her-father-did/ (visited on 06/01/2021).

[HHB19] Benjamin Hilprecht, Martin Härterich, and Daniel Bernau. “Monte Carlo and Reconstruction Membership Inference Attacks against Generative Models”. In: *Proceedings of the Privacy Enhancing Technologies Symposium*. PETS. 2019.

[Hor91] Kurt Hornik. “Approximation Capabilities of Multilayer Feedforward Networks”. In: *Neural Networks* 4.2 (1991), pp. 251–257.

[Hou+17] Xianxu Hou, Linlin Shen, Ke Sun, and Guoping Qiu. “Deep Feature Consistent Variational Autoencoder”. In: *Proceedings of the Conference on Applications of Computer Vision*. WACV. Piscataway, NJ, USA: IEEE Computer Society, 2017.

[Hsu+14] Justin Hsu, Marco Gaboardi, Andreas Haeberlen, Sanjeev Khanna, Arjun Narayan, Benjamin Pierce, and Aaron Roth. “Differential Privacy: An Economic Method for Choosing Epsilon”. In: *Proceedings of the Computer Security Foundations Workshop*. CSFW. Piscataway, NJ, USA: IEEE Computer Society, 2014.

[Hua+12] Gary B. Huang, Marwan Mattar, Honglak Lee, and Erik Learned-Miller. "Learning to Align from Scratch". In: *Proceedings of the Conference on Neural Information Processing Systems*. NIPS. Red Hook, NY, USA: Curran Associates Inc., 2012.

[Hua+07] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. *Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments*. Technical Report. University of Massachusetts, 2007.

[Hum+20] Thomas Humphries, Matthew Rafuse, Lindsey Tulloch, Simon Oya, Ian Goldberg, and Florian Kerschbaum. *Differentially Private Learning does not bound Membership Inference*. 2020. arXiv: `2010.12112 [cs.CR]`.

[Iye+19] Roger Iyengar, Joseph P. Near, Dawn Song, Om Dipakbhai Thakkar, Abhradeep Thakurta, and Lun Wang. "Towards Practical Differentially Private Convex Optimization". In: *Proceedings of Symposium on Security and Privacy*. S&P. Piscataway, NJ, USA: IEEE Computer Society, 2019.

[JUO20] Matthew Jagielski, Jonathan Ullman, and Alina Oprea. "Auditing Differentially Private Machine Learning: How Private is Private SGD?" In: *Advances in Neural Information Processing Systems*. NeurIPS. Red Hook, NY, USA: Curran Associates Inc., 2020.

[JS02] Nathalie Japkowicz and Shaju Stephen. "The Class Imbalance Problem: A Systematic Study". In: *Intelligent Data Analysis* 6.5 (Oct. 2002), pp. 429–449.

[JE19] Bargav Jayaraman and David Evans. "Evaluating Differentially Private Machine Learning in Practice". In: *Proceedings of the USENIX Security Symposium*. Berkeley, CA, USA: USENIX Association, 2019.

[Jay+20] Bargav Jayaraman, Lingxiao Wang, Katherine Knipmeyer, Quanquan Gu, and David Evans. *Revisiting Membership Inference Under Realistic Assumptions*. 2020. arXiv: `2005.10881 [cs.CR]`.

[JYS19] James Jordon, Jinsung Yoon, and Mihaela van der Schaar. “PATE-GAN: Generating Synthetic Data with Differential Privacy Guarantees”. In: *Proceedings of the Conference on Learning Representations*. ICLR. Piscataway, NJ, USA: IEEE Computer Society, 2019.

[KOV17] Peter Kairouz, Sewoong Oh, and Pramod Viswanath. “The Composition Theorem for Differential Privacy”. In: *IEEE Transactions on Information Theory* 63.6 (2017).

[Kas+08] Shiva P. Kasiviswanathan, Homin K. Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. “What can we Learn Privately?” In: *SIAM Journal on Computing* 40.3 (2008).

[KS14] Shiva P. Kasiviswanathan and Adam Smith. “On the Semantics of Differential Privacy: A Bayesian Formulation”. In: *Journal on Privacy and Confidentiality* 6 (1 2014).

[Kel20] Ulrich Kelber. *Positionspapier zur Anonymisierung unter der DSGVO unter besonderer Berücksichtigung der TK-Branche*. Technical Report. Der Beauftragte für den Datenschutz und Informationsfreiheit, 2020.

[KW52] Jack Kiefer and Jacob Wolfowitz. “Stochastic Estimation of the Maximum of a Regression Function”. In: *The Annals of Mathematical Statistics* 23.3 (1952), pp. 462–466.

[KB15] Diederik P. Kingma and Jimmy Ba. “Adam: A Method for Stochastic Optimization”. In: *Proceedings of the Conference on Learning Representations*. ICLR. Piscataway, NJ, USA: IEEE Computer Society, 2015.

[KW14] Diederik P. Kingma and Max Welling. “Auto-Encoding Variational Bayes”. In: *Proceedings of the Conference on Learning Representations*. ICLR. Piscataway, NJ, USA: IEEE Computer Society, 2014.

[Koh96] Ron Kohavi. “Scaling Up the Accuracy of Naive-Bayes Classifiers: A Decision-Tree Hybrid”. In: *Proceedings of the Conference on Knowledge Discovery and Data Mining*. KDD. Palo Alto, CA, USA: AAAI Press, 1996.

[LSS14] Adeline Langlois, Damien Stehlé, and Ron Steinfeld. “GGHLite: More Efficient Multilinear Maps from Ideal Lattices”. In: *Proceedings of the Conference on the Theory and Applications of Cryptographic Techniques*. EUROCRYPT. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014.

[LB98] Yann LeCun and Yoshua Bengio. “Convolutional Networks for Images, Speech, and Time Series”. In: *The Handbook of Brain Theory and Neural Networks*. Cambridge, MA, USA: MIT Press, 1998, pp. 255–258.

[LC11] Jaewoo Lee and Chris Clifton. “How Much is Enough? Choosing Epsilon for Differential Privacy”. In: *Proceedings of the Conference on Information Security*. ISC. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011.

[LC12] Jaewoo Lee and Chris Clifton. “Differential Identifiability”. In: *Proceedings of the Conference on Knowledge Discovery and Data Mining*. KDD. New York, NY, USA: ACM Press, 2012.

[Li+13] Ninghui Li, Wahbeh Qardaji, Dong Su, Yi Wu, and Weining Yang. “Membership Privacy: A Unifying Framework for Privacy Definitions”. In: *Proceedings of the Conference on Computer and Communications Security*. CCS. New York, NY, USA: ACM Press, 2013.

[Li+20a] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. “Federated Learning: Challenges, Methods, and Future Directions”. In: *IEEE Signal Processing Magazine* 37.3 (2020).

[Li+20b] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. “Federated Optimization in Heterogeneous Networks”. In: *Proceedings of Machine Learning and Systems*. MLSys. 2020.

[LOK21] Jiaxiang Liu, Simon Oya, and Florian Kerschbaum. *Generalization Techniques Empirically Outperform Differential Privacy against Membership Inference*. 2021. arXiv: `2110.05524 [cs.CR]`.

[Luc+19] James Lucas, George Tucker, Roger B. Grosse, and Mohammad Norouzi. “Don’t Blame the ELBO! A Linear VAE Perspective on Posterior Collapse”. In: *Proceedings of the Conference on Neural Information Processing Systems*. NIPS. Red Hook, NY, USA: Curran Associates Inc., 2019.

[LPR13] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. “On Ideal Lattices and Learning with Errors over Rings”. In: *Journal of the ACM* 60.6 (Nov. 2013).

[Mah36] Prasanta C. Mahalanobis. “On the Generalised Distance in Statistics”. In: *Proceedings of the National Institute of Science of India* 2.1 (1936).

[Mal+18] Mohammad Malekzadeh, Richard G. Clegg, Andrea Cavallaro, and Hamed Haddadi. “Protecting Sensory Data against Sensitive Inferences”. In: *Proceedings of the Workshop on Privacy by Design in Distributed Systems*. W-P2ds. New York, NY, USA: ACM Press, 2018.

[MKB79] Kantilal Vardichand Mardia, John T. Kent, and John M. Bibby. *Multivariate Analysis*. New York, NY, USA: Academic Press, 1979.

[Mat92] Kiyotoshi Matsuoka. “Noise Injection into Inputs in Back-Propagation Learning”. In: *IEEE Transactions on Systems, Man, and Cybernetics* 22.3 (1992).

[MP43] Warren S. McCulloch and Walter Pitts. “A Logical Calculus of the Ideas Immanent in Nervous Activity”. In: *The Bulletin of Mathematical Biophysics*. JOUR 5.4 (1943), pp. 115–133.

[McM+16] H. Brendan McMahan, Eider Moore, Daniel Ramage, and Blaise Agüera y Arcas. *Federated Learning of Deep Networks using Model Averaging*. 2016. arXiv: `1602.05629 [cs.CR]`.

[McM+17] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Ag´uera y Arcas. “Communication-Efficient Learning of Deep Networks from Decentralized Data”. In: *Proceedings of the Conference on Artificial Intelligence and Statistics*. AISTATS. JMLR W&CP, 2017.

[McM+18] H. Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. “Learning Differentially Private Recurrent Language Models”. In: *Proceedings of the Conference on Learning Representations*. ICLR. Piscataway, NJ, USA: IEEE Computer Society, 2018.

[Mir17] Ilya Mironov. “Rényi Differential Privacy”. In: *Proceedings of the Computer Security Foundations Symposium*. CSF. Piscataway, NJ, USA: IEEE Computer Society, 2017.

[Mit97] Tom Mitchell. *Machine Learning*. `https://www.cs.cmu.edu/~tom/mlbook.html`. McGraw Hill, 1997.

[NSH18] Milad Nasr, Reza Shokri, and Amir Houmansadr. “Machine Learning with Membership Privacy using Adversarial Regularization”. In: *Proceedings of the Conference on Computer and Communications Security*. CCS. New York, NY, USA: ACM Press, 2018.

[NSH19] Milad Nasr, Reza Shokri, and Amir Houmansadr. “Comprehensive Privacy Analysis of Deep Learning: Stand-alone and Federated Learning under Passive and Active White-box Inference Attacks”. In: *Proceedings of the Symposium on Security and Privacy*. S&P. Piscataway, NJ, USA: IEEE Computer Society, 2019.

[Nas+21] Milad Nasr, Shuang Song, Abhradeep Thakurta, Nicolas Papernot, and Nicholas Carlini. *Adversary Instantiation: Lower Bounds for Differentially Private Machine Learning*. 2021. arXiv: `2101.04535 [cs.LG]`.

[New17] BBC News. *Google DeepMind NHS App Test Broke UK Privacy Law*. 2017. URL: `https://www.bbc.com/news/technology-40483202` (visited on 06/15/2018).

[NJ01] Andrew Y. Ng and Michael I. Jordan. “On Discriminative vs. Generative Classifiers: A Comparison of Logistic Regression and Naive Bayes”. In: *Proceedings of the Conference on Advances in Neural Information Processing Systems*. NIPS. Cambridge, MA, USA: MIT Press, 2001.

[NRK21] Thao Nguyen, Maithra Raghu, and Simon Kornblith. “Do Wide and Deep Networks Learn the Same Things? Uncovering How Neural Network Representations Vary with Width and Depth”. In: *Proceedings of the International Conference on Learning Representations*. ICLR. 2021.

[Nis16] Helen Nissenbaum. “Differential Privacy in Context: Conceptual and Ethical Considerations”. In: *Four Facets of Differential Privacy Symposium*. Princeton, NJ, USA: Institute for Advanced Study, 2016.

[Nis+18] Kobbi Nissim, Aaron Bembenek, Alexandra Wood, mark Bun, Marco Gaboardi, Urs Gasser, David R. O'Brien, Thomas Steinke, and Salil Vadhan. "Bridging the Gap between Computer Science and Legal Approaches to Privacy". In: *Harvard Journal of Law & Technology* 31 (2018), pp. 687–780.

[NRS07] Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. "Smooth Sensitivity and Sampling in Private Data Analysis". In: *Proceedings of the Symposium on Theory of Computing*. STOC. New York, NY, USA: ACM Press, 2007.

[NW18] Kobbi Nissim and Alexandra Wood. "Is Privacy Privacy?" In: *Philosophical Transactions of the Royal Society* 376.2128 (2018).

[Pap+18] Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Úlfar Erlingsson. "Scalable Private Learning with PATE". In: *Proceedings of the Conference on Learning Representations*. ICLR. Piscataway, NJ, USA: IEEE Computer Society, 2018.

[PVZ15] Omkar M. Parkhi, Andrea Vedaldi, and Andrew Zisserman. "Deep Face Recognition". In: *Proceedings of the British Machine Vision Conference*. BMVC. BMVA Press, 2015.

[PE16] European Parliament and Council of the European Union. "General Data Protection Regulation". In: *Official Journal of the European Union* 119.1 (Apr. 2016).

[Par14] Article 29 Data Protection Working Party. *Opinion 05/2014 on Anonymisation Techniques*. 2014. URL: `https://ec.europa.eu/justice/article-29/documentation/opinion-recommendation/files/2014/wp216_en.pdf` (visited on 04/21/2022).

[Rah+18] Md. Atiqur Rahman, Tanzila Rahman, Robert Laganière, and Noman Mohammed. "Membership Inference Attack against Differentially Private Deep Learning Model". In: *Transactions on Data Privacy* 11 (2018).

[Ran+07] Marc'Aurelio Ranzato, Fu Jie Huang, Y-Lan Boureau, and Yann LeCun. "Unsupervised Learning of Invariant Feature Hierarchies with Applications to Object Recognition". In: *Proceedings of the Conference on Computer Vision and Pattern Recognition*. WACV. Piscataway, NJ, USA: IEEE Computer Society, 2007.

[Rob21] Jonas Robl. *Balancing Privacy and Utility in Differentially Private Generative Models by Inference Attacks*. Technical Report. Heidelberg University, 2021.

[RHM19] Luc Rocher, Julien M. Hendrickx, and Yves-Alexandre Montjoye. "Estimating the Success of Re-identifications in Incomplete Datasets using Generative Models". In: *Nature Communications* 10.1 (2019).

[Sae16] Aaqib Saeed. *Implementing a CNN for Human Activity Recognition in Tensorflow*. Nov. 4, 2016. URL: `https://aqibsaeed.github.io/2016-11-04-human-activity-recognition-cnn/` (visited on 04/03/2022).

[Sal+16] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. "Improved Techniques for Training GANs". In: *Proceedings of the Conference on Neural Information Processing Systems*. NIPS. Red Hook, NY, USA: Curran Associates Inc., 2016.

[SS98] Pierangela Samarati and Latanya Sweeney. *Protecting Privacy when Disclosing Information: k-Anonymity and its Enforcement Through Generalization and Suppression*. Technical Report. 1998.

[SW10] "Accuracy". In: *Encyclopedia of Machine Learning*. Ed. by Claude Sammut and Geoffrey I. Webb. Boston, MA, USA: Springer US, 2010, pp. 9–10.

[Sam59] Arthur L. Samuel. "Some Studies in Machine Learning Using the Game of Checkers". In: *IBM Journal of Research and Development* 3.3 (1959), pp. 210–229.

[San+09] Sriram Sankararaman, Guillaume Obozinski, Michael I. Jordan, and Eran Halperin. "Genomic Privacy and Limits of Individual Detection in a Pool". In: *Nature Genetics* 41 (2009).

[sci] scikit-learn. *Precision-Recall*. URL: https://scikit-learn.org/stable/auto_examples/model_selection/plot_precision_recall.html (visited on 01/19/2022).

[Seg+17] Aaron Segal, Antonio Marcedone, Benjamin Kreuter, Daniel Ramage, H. Brendan McMahan, Karn Seth, K. A. Bonawitz, Sarvar Patel, and Vladimir Ivanov. "Practical Secure Aggregation for Privacy-Preserving Machine Learning". In: *Proceedings of the Conference on Computer and Communications Security*. CCS. New York, NY, USA: ACM Press, 2017.

[SS15] Reza Shokri and Vitaly Shmatikov. "Privacy-Preserving Deep Learning". In: *Proceedings of the Conference on Computer and Communication Security*. CCS. New York, NY, USA: ACM Press, 2015.

[Sho+17] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. "Membership Inference Attacks against Machine Learning Models". In: *Proceedings of the Symposium on Security and Privacy*. S&P. Piscataway, NJ, USA: IEEE Computer Society, 2017.

[Sil+16] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. "Mastering the Game of Go with Deep Neural Networks and Tree Search". In: *Nature* 529.7587 (2016), pp. 484–489.

[SZ15] Karen Simonyan and Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *Proceedings of the Conference on Learning Representations*. ICLR. Piscataway, NJ, USA: IEEE Computer Society, 2015.

[SCS13] Shuang Song, Kamalika Chaudhuri, and Anand D. Sarwate. "Stochastic Gradient Descent with Differentially Private Updates". In: *Proceedings of the Global Conference on Signal and Information Processing*. GlobalSIP. Piscataway, NJ, USA: IEEE Computer Society, 2013.

[Sri+14] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. “Dropout: A Simple Way to Prevent Neural Networks from Overfitting”. In: *Journal of Machine Learning Research* 15.1 (Jan. 2014), pp. 1929–1958.

[Tak+20] Tsubasa Takahashi, Shun Takagi, Hajime Ono, and Tatsuya Komatsu. *Differentially Private Variational Autoencoders with Term-Wise Gradient Aggregation*. 2020. arXiv: `2006.11204 [cs.LG]`.

[Tan+17] Jun Tang, Aleksandra Korolova, Xiaolong Bai, Xueqiang Wang, and Xiaofeng Wang. *Privacy Loss in Apple’s Implementation of Differential Privacy on MacOS 10.12*. 2017. arXiv: `1709.02753 [cs.CR]`.

[TAM19] Om Thakkar, Galen Andrew, and H. Brendan McMahan. *Differentially Private Learning with Adaptive Clipping*. 2019. arXiv: `1905.03871 [cs.LG]`.

[Tin10a] Kai Ming Ting. “Precision and Recall”. In: *Encyclopedia of Machine Learning*. Ed. by Claude Sammut and Geoffrey I. Webb. Boston, MA, USA: Springer US, 2010, pp. 781–781.

[Tin10b] Kai Ming Ting. “Sensitivity and Specificity”. In: *Encyclopedia of Machine Learning*. Ed. by Claude Sammut and Geoffrey I. Webb. Boston, MA, USA: Springer US, 2010, pp. 901–902.

[TKP19] Reihaneh Torkzadehmahani, Peter Kairouz, and Benedict Paten. “DP-CGAN: Differentially Private Synthetic Data and Label Generation”. In: *Proceedings of the Conference on Computer Vision and Pattern Recognition Workshops*. CVPRW. Piscataway, NJ, USA: IEEE Computer Society, 2019.

[Tra+16] Florian Tramèr, Fan Zhang, Ari Juels, Michael K. Reiter, and Thomas Ristenpart. “Stealing Machine Learning Models via Prediction APIs”. In: *Proceedings of the USENIX Security Symposium*. Berkeley, CA, USA: USENIX Association, 2016.

[TG96] Kagan Tumer and Joydeep Ghosh. “Estimating the Bayes Error Rate through Classifier Combining”. In: *Proceedings of the Conference on Pattern Recognition*. ICPR. Piscataway, NJ, USA: IEEE Computer Society, 1996.

[Vin+10] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. “Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion”. In: *Journal of Machine Learning Research* 11 (2010).

[Wan21] Hansi Lo Wang. “Long-Awaited Redistricting Data is Expected in August after a Legal Fight Cools”. In: (2021). URL: `https://www.npr.org/2021/06/29/992912397/a-supreme-court-fight-over-census-data-privacy-and-redistricting-is-likely-comin?t=1642667965931` (visited on 01/19/2022).

[Wan+20] Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris Papailiopoulos, and Yasaman Khazaeni. *Federated Learning with Matched Averaging*. Piscataway, NJ, USA, 2020.

[Wan+17] Tianhao Wang, Jeremiah Blocki, Ninghui Li, and Somesh Jha. “Locally Differentially Private Protocols for Frequency Estimation”. In: *Proceedings of the USENIX Security Symposium*. Berkeley, CA, USA: USENIX Association, 2017.

[War65] Stanley L. Warner. “Randomized Response: A Survey Technique for Eliminating Evasive Answer Bias”. In: *Journal of the American Statistical Association* 60.309 (1965).

[Web18] MLPerf Website. *MLPerf – Fair and Useful Benchmarks for Measuring Training and Inference Performance of ML Hardware, Software, and Services*. 2018. URL: `https://mlperf.org/` (visited on 05/20/2020).

[Weg+22] Benjamin Weggenmann, Valentin Rublack, Michael Andrejczuk, Justus Mattern, and Florian Kerschbaum. “DP-VAE: Human-Readable Text Anonymization for Online Reviews with Differentially Private Variational Autoencoders”. In: *Proceedings of the Web Conference*. WWW. New York, NY, USA: ACM Press, 2022.

[WH60] Bernard Widrow and Marcian E. Hoff. “Adaptive Switching Circuits”. In: *1960 IRE WESCON Convention Record, Part 4*. New York, NY, USA: Institute of Radio Engineers, 1960.

[WH00] Rüdiger Wirth and Jochen Hipp. "CRISP-DM: Towards a Standard Process Model for Data Mining". In: *Proceedings of the Conference on Practical Applications of Knowledge Discovery and Data Mining*. Practical Application Company, 2000.

[Wun+21] Dominik Wunderlich, Daniel Bernau, Francesco Aldà, Javier Parra-Arnau, and Thorsten Strufe. *On the Privacy-Utility Trade-off in Differentially Private Hierarchical Text Classification*. 2021. arXiv: `2103.02895 [cs.CR]`.

[WWD16] Apple WWDC. *Engineering Privacy for Your Users*. 2016. URL: `https://developer.apple.com/videos/play/wwdc2016/709/` (visited on 04/24/2022).

[YFJ17] Samuel Yeom, Matt Fredrikson, and Somesh Jha. *The Unintended Consequences of Overfitting: Training Data Inference Attacks*. 2017. arXiv: `1709.01604 [cs.CR]`.

[Yeo+18] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. "Privacy Risk in Machine Learning: Analyzing the Connection to Overfitting". In: *Proceedings of the Computer Security Foundations Symposium*. CSF. Piscataway, NJ, USA: IEEE Computer Society, 2018.

[ZLH19] Ligeng Zhu, Zhijian Liu, and Song Han. "Deep Leakage from Gradients". In: *Proceeding of the Conference on Neural Information Processing Systems*. NeurIPS. Red Hook, NY, USA: Curran Associates Inc., 2019.

Academic Curriculum and Publications

Academic Curriculum

Oct. 2019 –	**University of Stuttgart, Germany.** External Ph.D. candidate at the Institute of Information Security. Supervisor: Prof. Dr. Ralf Küsters
Jan. 2015 – Mar. 2021	**SAP SE, Karlsruhe, Germany.** Industrial Ph.D. candidate at SAP Security Research. Supervisor: Prof. Dr. Florian Kerschbaum
Apr. 2013 – Aug. 2015	**Technical University of Berlin, Germany.** **University of Rennes 1, France.** Master of Science: Innovation in Information and Communication Technology. Master Sciences, Technologies, Santé: Recherche en Informatique. Thesis title: *Evaluating the Impact of Big Data on Intrusion Detection: Anomaly Detection and Alert Correlation in SAP landscapes* Supervisor: Prof. Guillaume Pierre
Apr. 2008 – Aug. 2011	**University of Applied Sciences Ludwigshafen am Rhein, Germany.** Bachelor of Science: International Business Administration and Information Technology. Thesis title: *Evaluating structured code search performance on an in-memory, columnar database* Supervisor: Dipl. Inf. Susan Hickl

Publications

- Jonas Böhler, Daniel Bernau, and Florian Kerschbaum. *Privacy-Preserving Outlier Detection for Data Streams*. In: *Proceedings of the 31st Annual IFIP WG*

11.3 Conference on Data and Applications Security and Privacy (DBSec 2017). Springer, 2017.

- Günther Eibl, Kaibin Bao, Philip W. Grassal, and Daniel Bernau, Hartmut Schmeck. *The Influence of Differential Privacy on Short Term Electric Load Forecasting*. In: *Proceedings of the 7th Annual DACH+ Conference on Energy Informatics*. Springer, 2018.

- Benjamin Hilprecht, Martin Härterich, and Daniel Bernau. *Monte Carlo and Reconstruction Membership Inference Attacks Against Generative Models*. In: *Proceedings of the 19th Privacy Enhancing Technologies Symposium (PETS 2019)*. Sciendo, 2019.

- Daniel Bernau, Jonas Robl, Philip W. Grassal, Steffen Schneider, and Florian Kerschbaum. *Comparing Local and Central Differential Privacy Using Membership Inference Attacks*. In: *Proceedings of the 35th Annual IFIP WG 11.3 Conference on Data and Applications Security and Privacy (DBSec 2021)*. Springer, 2021.

- Daniel Bernau, Günther Eibl, Philip W. Grassal, Hannah Keller, and Florian Kerschbaum. *Quantifying Identifiability to Choose and Audit ϵ in Differentially Private Deep Learning*. In: *Proceedings of the 47th Annual Conference on Very Large Databases (VLDB 2021)*. IEEE, 2021.

- Dominik Wunderlich, Daniel Bernau, Francesco Aldà, Javier Parra-Arnau, and Thorsten Strufe. *On the Privacy-Utility Trade-Off in Differentially Private Hierarchical Text Classification*. Technical report arXiv:2103.02895. arXiv, Mar. 4, 2021. URL: `http://arxiv.org/abs/2103.02895`.

- Daniel Bernau, Jonas Robl, Florian Kerschbaum. *Assessing Differentially Private Variational Autoencoders under Membership Inference*. In: *Proceedings of the 36th Annual IFIP WG 11.3 Conference on Data and Applications Security and Privacy (DBSec 2022)*. Springer, 2022.