



Masterarbeit am Institut für Informatik der Freien Universität Berlin,  
Arbeitsgruppe ID Management

# Attacking Differentially Private CNNs Trained with PATE

**Jannis Ihrig**

jannis.ihrig@fu-berlin.de

Matrikelnummer: 4384496

Betreuerin: Franziska Boenisch

1. Gutachter: Prof. Dr. Marian Margraf

2. Gutachter: Prof. Dr. Gerhard Wunder

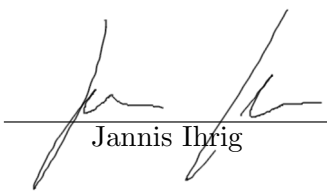
Berlin, den September 9, 2021



## Eidesstattliche Erklärung

Ich versichere hiermit an Eides statt, dass diese Arbeit von niemand anderem als meiner Person verfasst worden ist. Alle verwendeten Hilfsmittel wie Berichte, Bücher, Internetseiten oder ähnliches sind im Literaturverzeichnis angegeben, Zitate aus fremden Arbeiten sind als solche kenntlich gemacht. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungskommission vorgelegt und auch nicht veröffentlicht.

Berlin, den 09.09.2021



Jannis Ihrig



# Abstract

The increasing prevalence of machine learning (ML) models processing privacy-sensitive data makes it necessary to develop and employ privacy-preserving techniques for these algorithms. While Differential privacy (DP) was established as a robust and widely accepted framework allowing to give privacy guarantees for algorithms processing sensitive data, the relation between given guarantees and resulting concrete privacy is not fully understood.

This thesis evaluates the privacy of models trained with Private Aggregation of Teacher Ensembles (PATE), a recently proposed approach to implement DP for ML algorithms. To this end, various ML models for different datasets and settings of PATE hyperparameters are created, and subsequently targeted by attacks that aim at inferring sensitive information on the datasets used to create them.

The privacy granted by PATE to individuals whose sensitive data is contained in datasets used to train models with the framework is tested by attacking these with selected known methods for membership inference. Conducted experiments find that the framework overall reduces their accuracy and therefore increases privacy on an individual level. With regards to distribution-level privacy, the applicability of model inversion and property inference attacks is discussed. While a sketch for the modification of existing property inference attacks is given that allows their application to models created with the help of PATE, it is argued that the framework specifies a setting for training and deployment that makes a meaningful application of model inversion attacks questionable. Finally, a new distance-based property inference attack is presented and successfully applied to differentially private models trained with PATE, showing that the framework does not prevent inference of sensitive information on the distribution level.



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Background</b>	<b>5</b>
2.1	Machine Learning . . . . .	5
2.2	Attacks on Privacy in Machine Learning . . . . .	7
2.2.1	Adversarial Setting . . . . .	7
2.2.2	Membership Inference Attacks . . . . .	7
2.2.3	Property Inference Attacks . . . . .	9
2.3	Differential Privacy . . . . .	10
2.3.1	$(\epsilon, \delta)$ -Differential Privacy . . . . .	11
2.3.2	$(\alpha, \epsilon)$ -Rényi Differential Privacy . . . . .	12
2.3.3	Gaussian Mechanism . . . . .	13
2.4	Private Aggregation of Teacher Ensembles . . . . .	14
2.4.1	Overview . . . . .	14
2.4.2	Aggregation Mechanism . . . . .	15
2.4.3	Privacy Guarantees . . . . .	17
<b>3</b>	<b>Related Work</b>	<b>19</b>
3.1	Notions of Privacy . . . . .	19
3.2	Attacks on Privacy . . . . .	20
3.2.1	Membership Inference Attacks . . . . .	20
3.2.2	Model Inversion Attacks . . . . .	22
3.2.3	Property Inference Attacks . . . . .	23
3.3	Differential Privacy . . . . .	24
3.3.1	Differential Privacy for Machine Learning . . . . .	25
<b>4</b>	<b>Attacking Differentially Private CNNs Trained with PATE</b>	<b>27</b>
4.1	Adversarial Setting . . . . .	28
4.2	Evaluating PATE Privacy Guarantees through Membership Inference Attacks . . . . .	28
4.3	Model Inversion Attacks . . . . .	29
4.4	Property Inference Attacks . . . . .	30
4.4.1	Distance-based Property Inference Attack . . . . .	31
4.4.2	Deep Meta-Classifer . . . . .	36

<b>5</b>	<b>Experiments</b>	<b>39</b>
5.1	Creating Baseline and PATE models . . . . .	39
5.1.1	Datasets . . . . .	39
5.1.2	Model Architecture . . . . .	40
5.1.3	Model Training . . . . .	41
5.1.4	PATE Hyperparameters . . . . .	42
5.2	Membership Inference Attacks . . . . .	43
5.3	Property Inference Attacks . . . . .	44
5.3.1	Distance-based Property Inference Attack . . . . .	44
5.3.2	Deep Meta Classifier . . . . .	46
<b>6</b>	<b>Results</b>	<b>49</b>
6.1	PATE Training . . . . .	49
6.2	Membership Inference Attacks . . . . .	53
6.3	Property Inference Attacks . . . . .	57
6.3.1	Distance-based Property Inference Attack . . . . .	59
6.3.2	Deep Meta Classifier . . . . .	65
<b>7</b>	<b>Discussion</b>	<b>67</b>
7.1	Membership Inference and Model Performance . . . . .	67
7.2	Attacks on Distribution-level Privacy . . . . .	68
7.3	Future Work . . . . .	69
<b>8</b>	<b>Conclusion</b>	<b>71</b>



# 1 Introduction

Modern machine learning (ML) solves a variety of increasingly complex analysis problems, leading to a situation where ML models are embedded in – or even build the basis for – an increasing number of software systems. Their prevalence and data-driven nature thereby give rise to growing concerns regarding their application to privacy-sensitive data and the potential for privacy leaks. These worries are especially valid in domains processing privacy-sensitive data by nature, such as the medical field [45]. Here, the privacy of patient data stands in direct conflict with the utility of ML that is employed in research and diagnosis.

The relevance of these concerns is underlined by the existence of a growing corpus of literature demonstrating successful attacks on the privacy of ML models. Multiple classes of attacks, such as *membership* and *property inference* as well as *model inversion attacks* all give adversaries the option to extract private data. To do so, the attackers do not necessarily need access to the model’s training data itself. Rather, they achieve their goals by either querying the model or accessing its internals, demonstrated among others by Shokri et al. [53] and Ateniese et al. [4] for membership inference and property inference attacks respectively.

As ML techniques, especially deep learning, require sufficient and relevant data to create models that offer a reasonable utility, privacy-preserving techniques are employed to alleviate this conflict. Differential privacy (DP) [16] is a widely used and accepted framework to ensure the privacy of sensitive data. Originally developed for regular software access to databases, several techniques were proposed to make it applicable to ML algorithms, more recently also to deep learning [55][1]. Central to this work is one of these approaches to DP for ML called *Private Aggregation of Teacher Ensembles (PATE)*[43][44].

While DP allows giving privacy guarantees against the inference of private data, on the one hand, these privacy guarantees come with a loss of utility on the other hand, as the flow of information from the training data into a differentially private model has to be controlled carefully. The strictness of the privacy guarantees and with this, the trade-off between privacy and utility can be tuned via parameters, though the direct privacy implications of selecting a concrete set of them are not fully understood. Evaluation of an ML model with regards to utility is a known problem that can be solved with well-understood metrics such as accuracy, precision,

## 1 Introduction

and recall. Tying the selection of parameters for any formulation of DP to concrete privacy guarantees is more complex and an area of active research. One common approach is to test the privacy of a concrete model trained with a specific formulation of DP with known attacks on privacy.

Though existing publications use this method to evaluate the privacy granted by a variety of DP formulations and different ML techniques[47][28], an evaluation for the intuitively understandable PATE framework is still outstanding. This thesis thus follows the same path and evaluates PATE via multiple concrete CNN trained for varying sets of parameters for the DP framework<sup>1</sup>. The applicability of known attacks against differentially private models trained with PATE will be discussed in the context of the specific adversarial setting assumed by the framework. Further, the concrete privacy granted to individuals whose sensitive data is contained in the dataset used to create PATE models is tested with the help of membership inference attacks. On the distribution level, privacy is evaluated with the help of property inference attacks. To this end, a new distance-based attack is proposed, and its applicability to models trained with PATE is shown.

The main contributions of this work can be summarized as follows:

- The threat that different known attacks such as membership and property inference as well as model inversion attacks pose to the privacy of ML models is discussed in the context of PATE. While the meaningfulness of using model inversion attacks against models trained with PATE is questioned, a sketch for modifying existing property inference attacks to the given adversarial setting is given.
- A new distance-based property inference attack is proposed that targets models trained with PATE by comparing their behavior to multiple ensembles based on model outputs.
- Differentially private models are trained on multiple image datasets. While their utility is evaluated according to standard performance metrics, multiple attacks are executed against them to test their privacy, including membership inference and the newly proposed approach to property inference.
- An evaluation of the privacy of models trained with PATE is given based on the attacks' results.

The remainder of this thesis is organized as follows: Chapter 2 provides the theoretical background for this thesis. This includes a brief discussion of used attacks,

---

<sup>1</sup>Source code for these experiments can be found at [https://git.imp.fu-berlin.de/private\\_secure\\_ml/ihrig-msc-attacking-cnns-pate](https://git.imp.fu-berlin.de/private_secure_ml/ihrig-msc-attacking-cnns-pate). Raw result data is provided upon request.

DP, and the PATE framework, while further related work is reviewed in Chapter 3. The discussion of possible attacks in the context of PATE takes place in Chapter 4. The experiments proposed here are then detailed in Chapter 5, followed by results that are given in Chapter 6. Chapter 7 further discusses the meaning of these results for PATE as an ML framework for DP and gives an outlook on future work. Lastly, Chapter 8 concludes this thesis.



## 2 Background

### 2.1 Machine Learning

This section discusses the basics of ML as far as they are relevant for this thesis. The terms and definitions are thereby adapted from [5] and [21] if not noted otherwise. This section by no means aims at discussing the topic of ML exhaustively but at giving an overview that suffices in the context of this thesis.

It is possible to describe ML as the capability of an algorithm to extract knowledge from a *dataset* to solve a given *task*. The topic can be divided into several different categories with regards to the dataset that is input into a learning algorithm, from which *supervised* and *semi-supervised* learning will be relevant for this work. Therefore, it is first necessary to specify the notion of a dataset with the definition below:

**Definition 2.1.1** (Dataset, adapted from [16] p. 17). Given a universe of possible data  $\mathcal{U}$ , a dataset  $D$  is defined as

$$D := \{d_i\}_{i=1}^n, \quad (2.1)$$

where  $n \in \mathbb{N}$  is the dataset size and its elements  $d_i \in \mathcal{U}$  are called the *data points*. The terms database and record will be used equivalently for dataset and data point.  $\mathcal{D}$  denotes the set of all possible datasets with elements from the universe  $\mathcal{U}$ .

In the case of supervised learning, each data point in a dataset consists of a pair of feature  $\mathbf{x} \in \mathcal{X}$  and label  $\mathbf{y} \in \mathcal{Y}$ , where  $\mathcal{X}$  is the *feature space* and  $\mathcal{Y}$  the *label space*. The definition for a labeled dataset is given below. For this thesis, most datasets will contain both data points and labels but will simply be called datasets for brevity, as their structure becomes clear through their context.

**Definition 2.1.2** (Labeled Dataset). For a given feature space  $\mathcal{X}$  and label space  $\mathcal{Y}$ , a labeled dataset  $D$  is defined as

$$D := \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n, \quad (2.2)$$

for the dataset size  $n \in \mathbb{N}$  and features  $\mathbf{x}_i \in \mathcal{X}$  and labels  $\mathbf{y}_i \in \mathcal{Y}$ .

## 2 Background

In Semi-supervised learning, unlabeled data is added to the labeled dataset with the intention to improve the performance of the ML algorithm. The PATE framework evaluated in this thesis and discussed in Section 2.4 make use of such techniques.

For the rest of the thesis, it will be sufficient to concentrate on supervised learning. In this field, an ML algorithm aims at learning the association between feature and label space given through the examples in the dataset. In most cases, this is done by learning a *model*  $F$ . This process is commonly called the *training* of  $F$  with the model being its final output. The parameters of the ML algorithm that influence the training process are named *hyperparameters*, in contrast to the parameters on which the model itself might depend. More formally, a model can be defined as follows:

**Definition 2.1.3** (Model, Parametrized Model). For a given feature space  $\mathcal{X}$  and label space  $\mathcal{Y}$ , the function  $F: \mathcal{X} \rightarrow \mathcal{Y}$  is called a model. If  $F$  depends on a vector of adjustable parameters  $\theta$ , it is called a parameterized model. For clarity, they can be written as  $F_\theta$  to express this dependency.

As the above definition is abstract, models will always instantiate a concrete type of model such as a *neural network* (NN) which is a type of parameterized model. The models created for this work will all be *convolutional neural networks* (CNNs), a form of NN that is tailored to process features representing images. For brevity, these parameterized models will simply be called models in the following, keeping in mind that each of them depends on a vector of parameters.

One of the most common tasks in ML is *classification*, where the label space consists of a finite number of concrete categories or *classes*. To solve it, the ML algorithm has to produce a model that associates each given data point with one label or outputs a probability distribution over the possible labels. In the following, models solving the classification task will be called *classifiers* and it is assumed that their output have the latter form, called the *confidences* for a given data point. From these, a concrete label can be derived, which is called the models *prediction*.

**Definition 2.1.4** (Classifier). A model  $F: \mathcal{X} \rightarrow \mathcal{Y}$  for a given feature space  $\mathcal{X}$  and label space  $\mathcal{Y}$  is called a classifier if the label space consists of a finite set of discrete values  $Y := \{y_i\}_{i=1}^m$  for  $m \in \mathbb{N}$ .

**Definition 2.1.5** (Confidences). Let  $F: \mathcal{X} \rightarrow \mathcal{Y}$  be a classifier with feature space  $\mathcal{X}$  and labels space  $\mathcal{Y} := \{y_i\}_{i=1}^m, m \in \mathbb{N}$ . For a given input  $\mathbf{x} \in \mathcal{X}$ , the output of classifier  $F$

$$F(\mathbf{x}) := \{p_{y_i}\}_{i=1}^m, \quad (2.3)$$

with  $p_{y_i} \in [0, 1]$  and  $\sum_{i=1}^m p_{y_i} = 1$ , is called the confidences for  $\mathbf{x}$ . Each  $p_{y_i}$  is regarded as the probability that the input  $\mathbf{x}$  has the label  $y_i$ .

**Definition 2.1.6** (Prediction). Let  $F: \mathcal{X} \rightarrow \mathcal{Y}$  be a classifier with feature space  $\mathcal{X}$  and labels space  $\mathcal{Y} := \{y_i\}_{i=1}^m, m \in \mathbb{N}$ . The function  $g(p_{y_i}) := y_i$  returns the label  $y_i \in \mathcal{Y}$  for a given associated probability  $p_{y_i} \in [0, 1], i \in \{0 \dots m\}$ . For a given input  $\mathbf{x} \in \mathcal{X}$ , the prediction  $\hat{y}$  of the classifier  $F$  is defined as

$$\hat{y} := g(\max(F(\mathbf{x}))) . \quad (2.4)$$

## 2.2 Attacks on Privacy in Machine Learning

From the examples given in Chapter 1, the conflict between the potential utility given through the analysis of sensitive data via ML and the need to protect privacy becomes clear. Thereby, different notions of how privacy can be defined exist as outlined in Section 3.1. With DP, one of these specific understandings of privacy will be discussed in more detail in Section 2.3 as it builds the basis of the PATE framework. For this thesis, an attack on privacy will be regarded as a way to extract sensitive information from an ML model or the data used to create it [42].

In this section, possible *adversarial settings* in which an attack can take place will first be discussed, followed by the explanation for two different types of attacks that build the basis for experiments conducted below. A more comprehensive discussion of related work on attacks on privacy follows in Section 3.2.

### 2.2.1 Adversarial Setting

Attacks differ in the assumptions that are made regarding the *adversarial setting* [42] (also called *adversarial knowledge* as in [59]). Throughout this thesis, the common differentiation between *black-box* and *white-box* as presented in [42] will be used. In the former, an adversary may query the target model and observe the output for a given input but has no further specialized knowledge about the target or the data used to create it. Attacks of the latter category assume the adversary has full access to the target model, including its parameters and the information about the ML algorithm that was used to train it.

### 2.2.2 Membership Inference Attacks

A data point contained within a dataset is called a member of said set. Membership inference attacks against ML models such as first presented in [53] try to determine whether a given data point was part of the model's training dataset. In the following,

## 2 Background

two approaches from [53] and [50] will be discussed, where the first trains an attack model to attack the target model in contrast to the second that uses a threshold-based method. Both approaches thereby function in a black-box setting, meaning with only the possibility to query the target model and without direct access to its parameters.

### Shadow Training

The attack presented in [53] is called *shadow training*. It consists of four steps that need to be executed for an attack: First, a so-called *shadow dataset* is created that needs to be similarly distributed as the target models training dataset. [53] proposes multiple approaches to this step, from querying the target model with possible inputs to making the assumption that the attacker knows a noisy version of the training dataset.

In a second step, the shadow dataset is partitioned and used to train multiple *shadow models*. For this, each of the partitions is split into a training and test set, from which the training set is then used to create a shadow model. The goal for the created ensemble of shadow models is to use it as a stand-in for the target model.

The third step is to train the attack model. For the generation of the training dataset for this new model, each shadow model is fed its training and test set. The resulting confidences are labeled with “in” and “out” respectively, as well as with the data points original label. The collection of all the datasets of labeled confidences is then used to train the attack model in a supervised fashion.

Finally, the attack is executed by querying the target model with data points for which the adversary tries to infer membership information. The returned confidences are given to the attack model, which labels them as either “in” or “out”, and through this step classifies the targeted data points as members or non-members. Figure 2.1 depicts an overview of the steps necessary for the creation of the attack model.

### Threshold Attack

The attack from [50] is based on the assumption that the target model is expected to be more confident on data points that were used for its training. As a first step, the adversary queries the target model with a given target data point. They then select the highest value from the returned confidence vector and compare it to a given threshold. The target data point is considered a member of the target models training dataset if the confidence value surpasses the threshold.



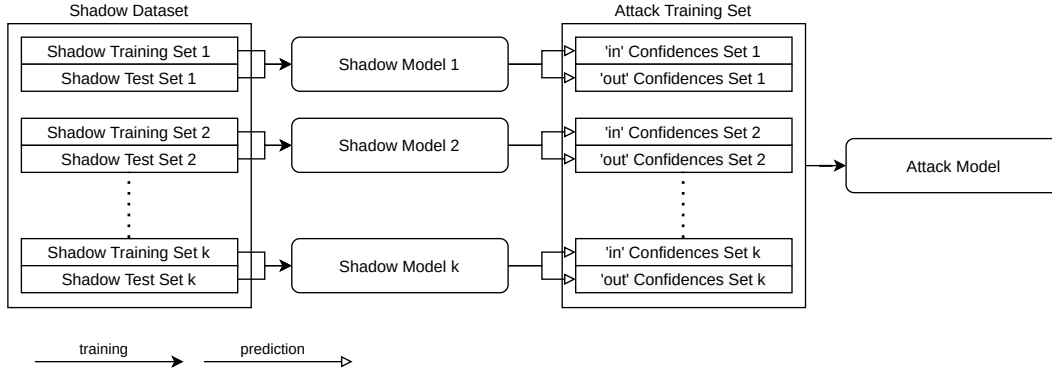


Figure 2.1: The above figure shows a schematic overview of the shadow training technique. The figure was adapted from [53].

The authors further discuss a general method on how to select a sensible threshold: First, random data points are generated that are considered non-members of the target models training dataset. Next, they are used to query the model, and the maximal values for the output confidences are recorded. These are then used to derive a threshold for the attack.

### 2.2.3 Property Inference Attacks

The attack was first proposed in [4] for support vector machines (SVM) and hidden markov models (HMM). It was named property inference attack in [20] where it is developed further to allow to target simple NN. The attack does not aim at directly extracting information on a single record of the training set of a model. Instead, an attack model called *meta-classifier* is trained to decide a binary property over the unknown training dataset of a target classifier. The attack as proposed in [4] takes place in a white-box scenario and consists of four steps:

First, the adversary selects a binary property  $\mathcal{P}$  that he wants to decide for the target models training dataset. An example is given in [4] in which the target property is to determine whether a model was trained on a dataset of voice samples for which all the contained data points come from speakers with Indian accents. They then generate  $k$  datasets, whereby  $\mathcal{P}$  is true for  $k/2$  and false for  $k/2$  of them. Here, it is assumed that the adversary has knowledge about the structure of the target models training dataset.

For the second step, the adversary trains one classifier for each of the  $k$  generated datasets and saves them as a set of feature vectors, labeled with the value of  $\mathbb{P}$  for the dataset that was used. The result is a training dataset for the attack model

## 2 Background

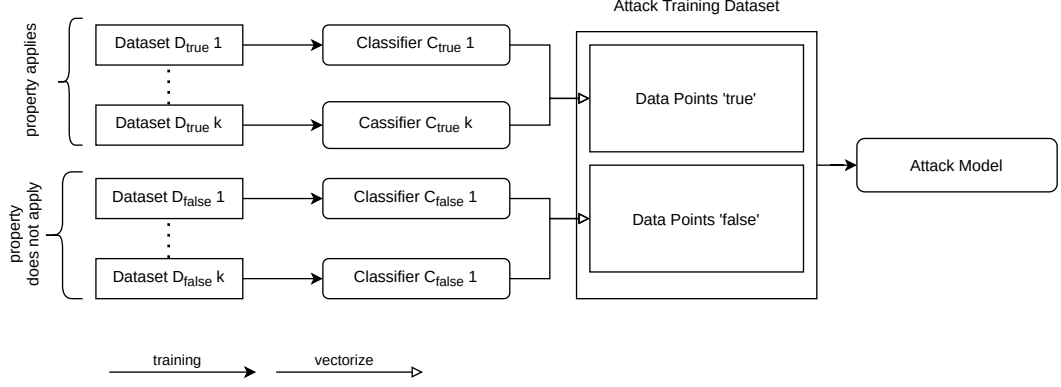


Figure 2.2: The above figure shows the schematic overview of the training of a meta-classifier attack model. The figure was adapted from [4].

of size  $k$ . It consists of  $k/2$  data points labeled 'true' and  $k/2$  data points labeled 'false'.

This dataset is then used in a third step to train the meta-classifier in a supervised fashion. The goal is for this attack model to learn to differentiate between vectors representing models for which  $\mathcal{P}$  is true and those for which it is false.

The fourth and last step consists of transforming the target model into the same representation as was used for the data points of the training dataset and feeding it to the meta-classifier. The attack model then decides  $\mathcal{P}$  for the training dataset of the target model on the basis of the presented vectorized representation.

### 2.3 Differential Privacy

DP gives a mathematically robust framework, fairly popular nowadays for protecting the privacy of individuals whose data is entered in a dataset. It was developed by Dwork [15], and follows the idea that the output of an algorithm that releases statistical information about a dataset  $D$  should not significantly change, whether or not an individual's data is contained in  $D$ <sup>1</sup>. Dwork et al. [16] argue that DP is an abstract concept, and a concrete algorithm called *mechanism* is needed to realize it. For it to give viable privacy guarantees, it is necessary to randomize the mechanism, which is achieved by adding controlled amounts of noise to its output.

In the following sections, two formulations of differential privacy will be presented.

<sup>1</sup>Note that it is implicitly assumed that the dataset contains at most one data point per individual, though DP extends to group privacy as well [15].

This will be  $\epsilon$ -DP and its relaxation  $(\epsilon, \delta)$ -DP [16], followed by another relaxation of DP called *Rényi Differential Privacy* (RDP) [38] and used for the PATE framework [44].

### 2.3.1 $(\epsilon, \delta)$ -Differential Privacy

This section is based on [16] if not noted otherwise.

When reasoning about  $(\epsilon, \delta)$ -DP it is necessary to first define the distance between two datasets:

**Definition 2.3.1** (Distance between Dataset, adapted from [29] Def. 2). The distance  $d(D_1, D_2)$  between the two datasets  $D_1, D_2 \in \mathcal{D}$  denotes the minimum number of data points that are required to be changed to transform  $D_1$  into  $D_2$ .

Two datasets with  $d(D_1, D_2) \leq 1$  are called *neighboring datasets*. Given this term, it is possible to formalize the intuitive understanding of DP that was given above:

**Definition 2.3.2** ( $(\epsilon, \delta)$ -Differential Privacy, adapted from [16] Def. 2.4). A randomized algorithm  $\mathcal{M}$  with domain  $\mathcal{D}$  is  $(\epsilon, \delta)$ -Differentially private if for all  $\mathcal{S} \subseteq \text{Range}(\mathcal{M})$  and all neighboring datasets  $D_1, D_2 \in \mathcal{D}$  the following applies:

$$\Pr[\mathcal{M}(D_1) \in \mathcal{S}] \leq \exp(\epsilon) \Pr[\mathcal{M}(D_2) \in \mathcal{S}] + \delta, \quad (2.5)$$

where the probability space is over the randomness of the algorithm  $\mathcal{M}$ . If  $\delta = 0$ , it is said that  $\mathcal{M}$  is  $\epsilon$ -differentially private.

For this formulation of DP, the first parameter  $\epsilon$  is often called the privacy budget, whereas the second parameter  $\delta$  allows for a relaxation of strict  $\epsilon$ -DP guarantees. Choosing  $\delta > 0$  gives a chance that the output of  $\mathcal{M}$  varies by more than  $\exp(\epsilon)$  – it is therefore important to choose a small value for  $\delta$ . It is recommended to choose  $\delta < 1/n$ , where  $n$  is the number of data points in a dataset, as each of them has the independent chance of  $\delta$  that information about it is leaked to the adversary [29].

From the definition above follows, the privacy guarantees given by this formulation of DP grow stronger the smaller parameters  $\epsilon$  and  $\delta$  are, as the guaranteed maximal difference in the output of  $\mathcal{M}$  for the neighboring databases gets smaller with them. The introduction of a  $\delta$ -parameter might weaken the privacy guarantees on the one hand but allow for different mechanisms such as the Gaussian Mechanism discussed in Section 2.3.3 that allow giving tighter bounds for them [38].

## 2 Background

One central feature of  $(\epsilon, \delta)$ -DP is its closeness under composition. This property allows the expression of privacy guarantees for more complex mechanisms that consist of multiple basic ones applied to a given dataset. Please refer to [16] for further information on the composition of  $(\epsilon, \delta)$ -DP. For this thesis, Rényi Differential Privacy and its composition is relevant, a topic that is discussed in Section 2.3.2.

### 2.3.2 $(\alpha, \epsilon)$ -Rényi Differential Privacy

This section is based on [38], in which Mironov presents Rényi Differential Privacy (RDP). It describes another relaxation of the strict  $\epsilon$ -DP which is similar to  $(\epsilon, \delta)$ -DP but uses the Rényi divergence. This divergence is a generalization of the Kullback-Leibler divergence (KLD) that can be defined as below:

**Definition 2.3.3** (Rényi Divergence, adapted from [38] Def. 3). For two probability distributions  $P$  and  $Q$  defined over an arbitrary result space  $\mathcal{R}$ , the Rényi divergence of order  $\alpha > 1$  is

$$\mathbb{D}_\alpha(P \parallel Q) := \frac{1}{\alpha - 1} \log \mathbb{E}_{x \sim Q} \left( \frac{P(x)}{Q(x)} \right)^\alpha, \quad (2.6)$$

where  $\log$  describes the natural logarithm and  $P(x)$  is the density of  $P$  at  $x$ .

With the definition of the Rényi divergence in place, it is now possible to formalize  $(\alpha, \epsilon)$ -RDP. Here, the divergence is used to determine the distance between the probability distributions output by a mechanism for neighboring datasets. Recalling the informal explanation of DP from above, a mechanism is differentially private if its output for two neighboring datasets does not differ in a significant way. This can be achieved by bounding the Rényi divergence for the output of a mechanism in the following way:

**Definition 2.3.4** ( $(\alpha, \epsilon)$ -Rényi Differential Privacy, adapted from [38] Def. 4). Let  $\mathcal{M}: \mathcal{D} \rightarrow \mathcal{R}$  be a randomized mechanism, where  $\mathcal{D}$  is the set of all possible datasets and  $\mathcal{R}$  the set of possible results for  $\mathcal{M}$ . The mechanism  $\mathcal{M}$  is said to satisfy  $\epsilon$ -Rényi differential privacy of order  $\alpha$ , or  $(\alpha, \epsilon)$ -RDP for short, if for all two neighboring datasets  $D_1, D_2 \in \mathcal{D}$  holds that

$$\mathbb{D}_\alpha(\mathcal{M}(D) \parallel \mathcal{M}(D')) \leq \epsilon. \quad (2.7)$$

With RDP, it is possible to determine the privacy guarantees for the composition of two mechanisms from their respective privacy guarantees. This applies for both the case in which the results for both mechanisms become known at the same time and for the situation in which the output of the second mechanism depends on the

output of the first. The privacy guarantees for the mechanism results from such a composition that can be constructed as follows:

**Lemma 2.3.1** (Composition of RDP mechanism, adapted from [38] Prop. 1). Let  $\mathcal{M}_1 : \mathcal{D} \rightarrow \mathcal{R}_1$  and  $\mathcal{M}_2 : \mathcal{D} \times \mathcal{R}_1 \rightarrow \mathcal{R}_2$  be two mechanism that satisfy  $(\alpha, \epsilon_1)$ -RDP and  $(\alpha, \epsilon_1)$ -RDP respectively. Then the composed mechanism

$$\mathcal{M}_3(D) := (\mathcal{M}_1(D), \mathcal{M}_2(\mathcal{M}_1(D), D)) \quad (2.8)$$

satisfies  $(\alpha, \epsilon_1 + \epsilon_2)$ -RDP.

Moreover, it is possible to express RDP privacy guarantees in the context of  $(\epsilon, \delta)$ -DP.

**Lemma 2.3.2** (From RDP to  $(\epsilon, \delta)$ -DP, adapted from [38] Prop. 3). If a mechanism satisfies  $(\alpha, \epsilon)$ -RDP, then it also satisfies  $(\epsilon + \frac{\log 1/\delta}{\alpha-1}, \delta)$ -DP for any  $0 < \delta < 1$ .

For a proof of the above Lemma 2.3.1 and Lemma 2.3.2, please refer to the proofs of Proposition 1 and 3 given in [38].

### 2.3.3 Gaussian Mechanism

The *Gaussian mechanism* is one of the basic mechanisms that can be used as a concrete implementation of DP. As other mechanisms are possible but not relevant for this work, they will only be briefly touched upon in the related work Chapter 3. The idea behind the Gaussian Mechanism is to randomize the output of a given function by adding noise drawn from the Gaussian distribution to its output [38]. For this work, the Gaussian mechanism is discussed in the context of RDP only, as it simplifies the task of expressing privacy guarantees for it, and therefore is used in PATE in this way [44].

**Definition 2.3.5** (Gaussian Mechanism, adapted from [38]). For a given real valued function  $f : \mathcal{D} \rightarrow \mathbb{R}$  and a dataset  $D \in \mathcal{D}$ , the Gaussian Mechanism takes the form

$$\mathcal{M}_{\sigma f}(D) := f(D) + N(0, \sigma^2) \quad , \quad (2.9)$$

where  $\mathcal{N}(0, \sigma^2)$  is the Gaussian distribution centered at 0 and a standard derivation of  $\sigma$ .

## 2 Background

One important factor influencing the guarantees that can be given for the Gaussian mechanism is the  $\ell_1$ -sensitivity of the function  $f$ . It describes the magnitude by which the output of function changes in case only one of the elements of the given dataset does.

**Definition 2.3.6** ( $\ell_1$ -sensitivity, adapted from [38]). For a given function  $f: \mathcal{D} \rightarrow \mathbb{R}$  and two datasets  $D_1, D_2 \in \mathcal{D}$ , the  $\ell_1$ -sensitivity  $\Delta_f$  is defined by

$$\Delta_f := \max_{D, D'} \|f(D) - f(D')\|_1 . \quad (2.10)$$

With the help of the  $\ell_1$ -sensitivity, it is now possible to finally specify the RDP guarantees the Gaussian mechanism gives. The following proposition can be verified by direct computation as shown in [38].

**Lemma 2.3.3** (RDP Guarantees for Gaussian mechanism, adapted from [38] Cor. 3). For a given function  $f: \mathcal{D} \rightarrow \mathbb{R}$  and its  $\ell_1$ -Sensitivity  $\Delta_f$ , the Gaussian mechanism  $\mathcal{M}_{\sigma, f}$  satisfies  $(\alpha, \alpha\Delta_f^2/(2\sigma^2))$ -RDP.

## 2.4 Private Aggregation of Teacher Ensembles

PATE is an approach for ML to satisfy DP. It was initially presented in [43], shortly afterwards an improved version was published with [44]. The first publication bases PATE on  $(\epsilon, \delta)$ -DP and the Laplacian mechanism, whereas the improved version builds upon RDP together with the Gaussian mechanism. This sections follows [44] if not otherwise noted, as it has been demonstrated that the improved PATE version allows achieving better utility for models with comparable privacy guarantees.

### 2.4.1 Overview

For training an ML model with PATE, the used dataset is expected to be split into sensitive, private training data  $D_{priv}$  with labels and a public split of data  $D_{pub}$  that is not sensitive and does not need to contain labels for the data points. For validating the results, a hold-out part called  $D_{test}$  is split from  $D_{pub}$ .

The central idea of PATE is to let the model finally output by the framework never come into direct contact with the sensitive part of the dataset – a setting that makes it harder for an adversary to extract private formation. Instead, an ensemble of so-called *teachers* is trained on disjoint partitions of the private data  $D_{priv}$  and their

knowledge is subsequently transferred into a *student* model through a process similar to knowledge distillation [25]. For this, unlabeled data from the public data  $D_{pub}$  is presented to the teacher ensemble, and from their output, labels are created. The public split that is thereby augmented with labels is then used to train the student model. After this step, the teachers are discarded while the student model and the public dataset  $D_{pub}$  can potentially be made public. Figure 2.3 gives a graphical overview of the PATE framework for further clarification.

The central element of the PATE framework is the creation of labels through the teachers as it is the only way for the student to gain information about  $D_{priv}$ . The teacher predictions on the input data points are interpreted as votes that are then aggregated to derive a label. Noise is added in this aggregation mechanism, and thereby PATE allows to train a differentially private student model.

After the teacher ensemble is trained in a supervised fashion, training the student is done with semi-supervision. This way, the complete public dataset  $D_{pub}$  is used, augmented with the labels created by the noisy aggregation mechanism of teacher votes. For the semi-supervised training, [43] proposes to employ *generative adversarial networks*<sup>2</sup> (GAN) such as from [51], and [44] uses Virtual Adversarial Training (VAT)[39].

### 2.4.2 Aggregation Mechanism

To create a label for a data point from the unlabeled dataset  $D_{pub}$ , PATE uses a so-called *aggregator* mechanism. When queried with a data point, the mechanism first collects the teachers' predictions for it. These predictions are also called the teacher votes and are counted by class. Subsequently, Gaussian noise is added to the teacher votes for each class, and the one with the highest result is returned. Hence, this aggregation mechanism is called the Gaussian NoisyMax aggregator (GNMax).

The counting of teacher votes can be formalized as shown below. Here,  $f_i$  denotes a single model from the teacher ensemble of  $k$  teachers, with  $i \in \{0, \dots, k\}$ .

**Definition 2.4.1** (Teacher Vote Count, adapted from [43]). For a given feature space  $\mathcal{X}$  and label space  $\mathcal{Y}$ , the vote count  $n_y$  for the label  $y \in \mathcal{Y}$  given a data point  $\mathbf{x} \in \mathcal{X}$  is defined as

$$n_y(\mathbf{x}) := |\{i | f_i(\mathbf{x}) = y\}| . \quad (2.11)$$

Given this, the GNMax aggregation mechanism can be defined:

---

<sup>2</sup>Also, see [22] and [21] for more information on GANs.

## 2 Background

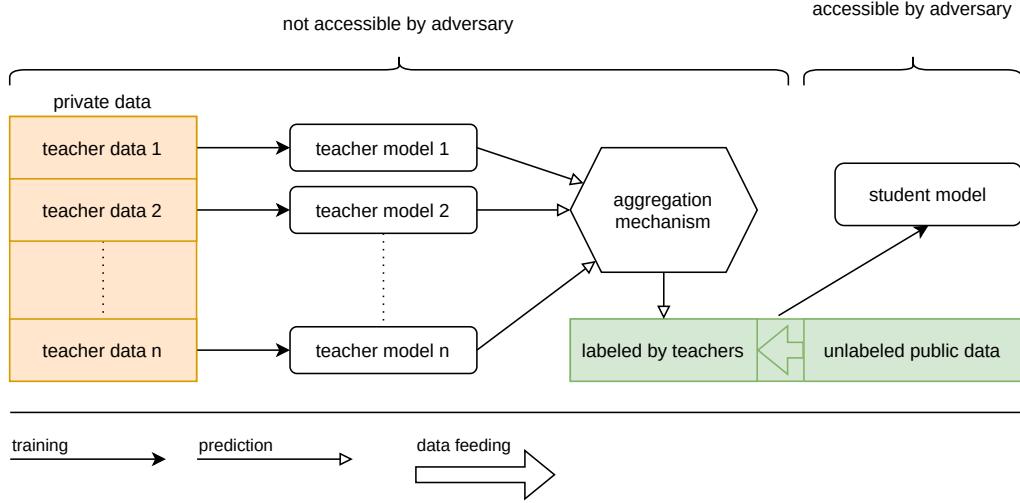


Figure 2.3: Overview over the PATE framework. The used dataset is split into sensitive, private data (orange) and potentially publicly available non-sensitive data (green). Teacher models trained on sensitive data are used to label public, non-sensitive data and discarded afterward. The partially labeled non-sensitive data is used to train a student model that can be made publicly available to be queried. Figure adapted from [43].

**Definition 2.4.2** (GNMax Aggregator, adapted from [44]). The GNMax aggregation mechanism is defined as

$$\mathcal{M}_\sigma(x) := \arg \max_{y \in \mathcal{Y}} \{n_y(x) + \mathcal{N}(0, \sigma^2)\} \quad , \quad (2.12)$$

where  $\mathcal{N}(0, \sigma^2)$  is the Gaussian distribution centered as 0 and a standard derivation of  $\sigma$ .

The number of partitions of the sensitive data split and teachers  $k \in \mathbb{N}$  as well as the standard deviation are hyperparameters of PATE and need to be adjusted to the size of a concrete dataset that is used.

The authors of PATE propose a variation of this mechanism, called *Confident-GNMax*, which will be used throughout this work. It solely returns labels for data points for which the teacher consensus is strong, meaning the votes for one label are higher than a given threshold. The aggregation of noisy teacher votes proceeds as above, with the difference that first Gaussian noise with a high standard deviation  $\sigma_1$  is applied to the voting results after which the result is compared to a threshold  $T$ . If this threshold is not surpassed by the maximum amount of votes given for one label, the complete query is rejected as the consensus between teachers was not strong enough. Otherwise, a second round of Gaussian noise with a smaller standard



deviation  $\sigma_2$  is applied. Then, the label with the highest amount of noisy votes is returned. This approach allows to reject labels with a high privacy cost but increases the number of hyperparameters for the Confident-GNMax by adding the threshold  $T$  and separating the standard deviation of Gaussian noise to  $\sigma_1$  and  $\sigma_2$ .

### 2.4.3 Privacy Guarantees

For the following, it is assumed that the partitioning of  $D_{priv}$  results in a situation where each data point only influences the training of one teacher. Moreover, the assumptions are made for a worst-case scenario, in which the presence or absence of one data point might change the teacher vote completely.

With the help of the GNMax aggregation mechanism, two different privacy bounds can be given for a query that is posed to it. The first is a loose bound that applies to all data points when they are labeled and results directly from the guarantees given by RDP. For the second, a data-dependent bound is given, which for many records produces tighter privacy guarantees. The privacy guarantees are thereby calculated in RDP and can be expressed in  $(\epsilon, \delta)$ -DP at the end of the labeling process.

**Lemma 2.4.1** (Loose Bound for GNMax, adapted from [44] Prop. 8). The GNMax aggregator  $\mathcal{M}_\sigma$  guarantees  $(\alpha, \alpha/\sigma^2)$ -RDP for all  $\alpha \geq 1$ .

*Proof.* From the requirement that teachers are trained on partitions of  $D_{priv}$  follows, that each data point can only influence the prediction of one teacher. For a neighboring dataset  $D'_{priv}$  in which one data point is changed, the teacher votes count can differ at most for two classes, each time by one. Following this, the GNMax aggregator is regarded as the composition of two Gaussian mechanisms with  $\ell_1$ -sensitivity of  $\Delta = 1$ . The loose bound therefore follows from Lemma 2.3.1 and Lemma 2.3.3.  $\square$

The data-dependent privacy bound can be applied in cases where the consensus between teachers is strong, meaning that the aggregation of teacher votes is especially high for one class.

**Lemma 2.4.2** (Tight Bound for GNMax, adapted from [44] Theo. 6 and Prop. 7). Let  $\mathcal{M}$  be a randomized algorithm that satisfies both  $(\alpha_1, \epsilon_1)$ -RDP and  $(\alpha_2, \epsilon_2)$ -RDP and suppose there exists a likely outcome  $y^*$  given a dataset  $D$  and a bound  $q \leq 1$  such that  $q \leq \Pr[\mathcal{M}(D) \neq y^*]$ . Additionally suppose that  $\alpha \leq \alpha_1$  and  $q \leq e^{(\mu_2-1)\epsilon_2} / \left( \frac{\mu_1}{\mu_1-1} \cdot \frac{\mu_2}{\mu_2-1} \right)^{\mu_2}$ . Then,  $\mathcal{M}$  satisfies  $(\alpha, \epsilon)$ -RDP with

$$\epsilon \leq \frac{1}{\alpha - 1} \log \left( (1 - q) \cdot \mathbf{A}(q, \alpha_2, \epsilon_2)^{\alpha-1} + q \cdot \mathbf{B}(q, \alpha_1, \epsilon_1)^{\alpha-1} \right), \quad (2.13)$$

## 2 Background

where  $\mathbf{A}(q, \alpha_2, \epsilon_2) = (1 - q) / \left(1 - (qe^{\epsilon_2})^{\frac{\epsilon_2 - 1}{\alpha_2}}\right)$  and  $\mathbf{B}(q, \alpha_1, \epsilon_1) = e^{\epsilon_1} / q^{\frac{1}{\alpha_1 - 1}}$ .

The probability  $\Pr[\mathcal{M}_\sigma(D) \neq y^*]$  thereby corresponds to the probability that the output of  $\mathcal{M}$  for a dataset  $D$  does not equal the likely outcome  $y^*$  and can be calculated by

$$\Pr[\mathcal{M}_\sigma(D) \neq y^*] \leq \frac{1}{2} \sum_{y \neq y^*} \operatorname{erfc}\left(\frac{n_{y^*} - n_y}{2\sigma}\right), \quad (2.14)$$

where  $\operatorname{erfc}$  is the complementary error function.

Informally, Lemma 2.4.2 expresses that the RDP for  $\mathcal{M}$  of order  $\alpha \leq \alpha_1, \alpha_2$  at  $D$  is bounded by a function of  $q, \alpha_1, \epsilon_1, \alpha_2, \epsilon_2$  that approaches 0 as  $q \rightarrow 0$ . For proof, please refer to the proofs for Theorem 6 and Proposition 7 given in [44].

The privacy guarantees are computed for each query to the GNMax and can be composed following the rules for RDP. Concretely, the privacy guarantees are tracked for a series of moments during the labeling process. Each step they are converted to  $(\epsilon, \delta)$ -DP guarantees and compared to the privacy budget. The labeling terminates before the privacy budget is crossed for all moments for which the privacy guarantees are calculated. Subsequently, a moment is selected according to the lowest tracked  $(\epsilon, \delta)$ -DP guarantees which are returned. They apply to any student model that is trained using the created labels. The authors of [44] note that these privacy guarantees are data-dependent, and publishing them would result in further loss of privacy. A smoothing that allows publishing this output is therefore proposed, which will not be applied here as the datasets used to train models with PATE are publicly known.

## 3 Related Work

The following section reviews previous work on different notions of privacy with a focus on DP and attacks against it.

### 3.1 Notions of Privacy

In the following, a brief overview of the literature regarding different notions of privacy will be given.

Trying to answer this question for what privacy is on a basic level, Li et al. suggests that it is easier to understand when privacy is breached, as it “is a social concept” and therefore hard to define directly [34]. The authors of [42] further differentiate between confidentiality and privacy, where breaching the former would take place when an adversary extracts information about an ML model, but the latter is breached if they successfully gain access to sensitive information on the data used to create it.

In the context of differential privacy which is discussed in more detail in Section 2.3, privacy is defined on an individual level. Here, a privacy breach occurs if an adversary can determine whether the data of an individual was part of a dataset [15]. Further, [53] argues that the case in which an adversary infers additional information about the members of a population given an ML model can not be considered a breach of privacy. This arises through the fact that the inferred information applies to the whole of the population and not to one individual, and the property of models to generalize on previously unknown data.

In contrast, the authors of [66] give arguments for the idea that besides attacks that breach privacy on this individual level, there exist those that breach it by extracting information on the training dataset or its distribution, referring among others to [20] discussed in Section 2.2.3. The authors subsequently call this form attribute privacy, for which they give formal privacy definitions.

## 3.2 Attacks on Privacy

This section gives an overview of related work regarding membership and property inference as well as model inversion attacks.

### 3.2.1 Membership Inference Attacks

Shokri et al. [53] demonstrate the class of membership inference attacks, that allows to predict from a given target record if it was part of the training set used to create the target ML model. It is shown that the degree to which a model is overfitted to its training dataset is related to the precision of its membership inference attack. Additionally, the authors state that overfitting is not the only cause for the shown vulnerability and the number of classes in the training dataset as an additional factor. The workings of this attack are described in greater detail in Section 2.2.2.

Following this initial attack, a variety of publications further examines the cause for ML models vulnerability to membership inference and proposes modifications of or alternative approaches to the shadow training technique:

Long et al. [35] describe the shadow training technique as untargeted as it is not specific to one target record and proposes two variations of a targeted attack that tries to infer the membership of one given record. The attack works by calculating the average KL divergence between the confidences output by the student model and two teacher ensembles - one for which the property holds and one for which it does not. Here, an ensemble of models is trained on datasets of which half contain the target record. The target record is fed to all shadow models, and the target model is classified as a member in case the target model's output is closer to the output from models whose training dataset contains the target record. For the first variation of the attack, the KL divergence is used as a distance measure. For the second variation, the model outputs are binned, and the frequency of models whose output falls into the same bin as the target model's output is calculated.

Salem et al. [50] aim to relax the assumptions made for the shadow training technique from [53]. They demonstrate that an adversary can execute a successful attack without knowing the ML algorithm of the model or the distribution of its training dataset as well as that the number of shadow models can be strongly reduced, up to only one employed model. The authors also propose a threshold-based attack that is independent of the target's training data distribution or any shadow models. Here the adversary inputs a target record into the target model and extracts the highest probability for a class. The target record is then assumed to be a member of the target model's training data if the extracted probability is higher than

a previously selected threshold. This threshold-based attack is further discussed in Section 2.2.2.

While the above attacks operate in a black-box setting where an adversary has no access to the model parameters, Nasr et al. [40] propose a white-box attack on neural networks that assumes the adversary has knowledge of the true label of a target record. In addition to computing the output for the record, a training step is executed on the neural network. Then, the training loss is recorded, in addition to the gradients extracted from the intermediate network layers. The authors state improved attack performance on selected target models in comparison to the black-box shadow training technique [53]. In contrast, the authors of [49] argue that this additional information would not improve the performance of a potential optimal black-box. They derive this from their search for an optimal attack technique and state that attacks from [53] and [64] are already coarse approximations of it.

Yeom et al. [64] further inspect the connection between vulnerability to membership inference attacks and overfitting, a relation which they confirm through a series of proposed attacks that depend on the output error of the target model. [36] extends this analysis by investigating the increased risk attacks pose to individual records that have a strong influence on the target models output. Especially outlier in the training dataset exhibit strong influence on the target model and an attack is demonstrated that exploits this. Records with strong influence are selected as targets, and the output distributions of the target model and an ensemble of reference models whose training data does not contain the target record are compared. The authors show that this can be done directly by querying the target record or even indirectly by querying other records for whom the output is influenced by the target record.

Truex et al. [59] analyze causes for membership inference vulnerability besides overfitting mainly based on the shadow training attack from [53]. They analyze multiple factors that influence the model’s vulnerability to membership inference. Most notably, the authors find the attack’s performance depending on the target’s model type and the structure of its training dataset. They further inspect the conditions under which attack models are transferrable. This property allows an adversary to not necessarily know the exact target models ML algorithm or the target dataset to mount an attack. Building on the above findings, [58] and [56] add the target model’s complexity and depths as potential factors for successful membership inference. The former publication thereby underlines the effect of skewed datasets and describes minority groups at especially at risk, a statement that is supported by [63].

While the above works already show that vulnerability to membership inference depends on task and target model complexity, they mostly focus on classification models that are trained from scratch. This is not the case in situations where *transfer learning* is employed to create a new model on the parameters of an existing one [21]. Zou et al. [68] explore this setting and presents results that indicate that

### 3 Related Work

the used attack is not able to restore membership information with regards to the training dataset of the base model. Other recent publications show that attacks are also feasible against complex models trained for different tasks, such as semantic segmentation [24][52] and image translation [52]. Moreover, attacks against generative methods have been shown, such as [23].

Another branch of research explores the feasibility of membership inference attacks against models that are trained in a federated fashion, such as described in [59]. Here, the effect of additional adversarial knowledge gained by a participant of a federated learning procedure is inspected with regards to the potentially increased performance of a black-box attack. [37] and [40] describe attacks where a malicious participant observes the parameters of the jointly trained model, calculates the parameter updates that were applied over time using these to execute the attack.

Song et al. [54] evaluate different approaches to membership inference attacks and mitigations. They propose to use a combination of attacks that train an ML model to infer membership information, as well as so-called metric-based attacks to benchmark the resilience of models against membership inference. The importance of metric-based attacks is underlined as the selection of suboptimal hyperparameters for those using models might lead to underestimating the risk for membership inference. Consequently, the authors present a new threshold attack in addition to a new *privacy risk score*. This metric estimates the probability for individual data points to be members of the target’s training dataset.

#### 3.2.2 Model Inversion Attacks

The initial *model inversion attack* presented by Fredrikson et al. [19] targets linear models in a black-box fashion to reconstruct a sensitive attribute for a partially given record. It is demonstrated for a model that outputs a patient-specific medicine dosage from which an adversary tries to extract a patient’s genotype with the help of extensive adversarial knowledge. This attack is further discussed in [64], where it is also called an *attribute inversion attack*. The advantage gained by the adversary through this attack for determining the unknown attribute is formally defined, and the connection between membership inference and attribute inference is explored.

The attack from [18] has the goal to reconstruct privacy-sensitive data from a given ML model and a label from the training dataset. The authors demonstrate this by inverting a facial recognition system consisting of a CNN trained on CIFAR-10. The goal here is to recover the image of a person given the name that is used as a label. The white-box attack calculates an image that minimizes the model’s training loss with respect to a given classification label. For this, a fixed number of iterations of gradient descent for this loss are calculated, and the gradients are applied to the

image.

Shokri et al. [53] state that the images recovered in the above attack do not correspond to actual images from the training data. Rather, the attack creates a representation of the average features by which the target model learned to characterize a given label. Later works try to improve in this area by using GANs. As one of these, the approach described in [3] trains a generator and discriminator by only querying the target model for labels. Through this black-box attack, the generator learns an approximated distribution of the images from the targets training dataset while the discriminator is able to imitate the target model’s behavior with respect to classification. Zhang et al. [67] and Chen et al. [10] show white-box attacks that improve upon the results from the approach above by being able to produce rather realistic images of faces. The former thereby allows the adversary to utilize different degrees of auxiliary knowledge, ranging from none over blurred to clear but partially masked images from the training dataset. Additionally, the authors show a correlation between “predictive power and its vulnerability to inversion attacks” [66].

While all other discussed attacks focus on locally trained models, there is also the possibility to invert models in federated learning: An adversary participant of such a collaborative learning procedure is shown to be able to gain information about the data of a label by updating the jointly trained model based on data generated with a GAN [26]. Wang et al. [60] on the other hand, focuses on the scenario in which a malicious server tries to gain information about private data while identifying participants it stems from.

### 3.2.3 Property Inference Attacks

Ateniese et al. [4] were the first to present a property inference attack. Its goal is not to directly extract training data but properties over the entire training dataset of the target model. The property under scrutiny is not directly related to the actual task of the target model but can be learned as a side-effect of the training procedure, such as whether a specific speaker accent is dominating in the samples of a speech recognition dataset. To this end, they train a so-called meta-classifier that is then input a representation of the target model to decide on a binary property of the training dataset. For a more detailed description on the training of the meta-classifier see Section 2.2.3.

The above approach was introduced to work on SVMs and HHMs, whose parameters can easily be described through a set of vectors [4]. [20] adapts the attack for neural networks. For these, the vectorized representation of a model is ambiguous, as the order in which units of one network layer are represented is irrelevant as long

### 3 Related Work

the connections between them are the same. To solve this problem, they introduce permutation invariant representations of neural networks, either achieved by sorting their units or by representing them through sets. These methods improve the attack performance on neural networks over [4].

Though not explicitly with the goal of performing a property inference attack, [17] introduce a *deep meta-classifier* that is trained in a similar fashion as the meta-classifier of [4] and operates on CNNs. The authors use this deep neural network to extract properties of the training setup of a target model, such as used datasets, hyperparameters, and whether the training data was augmented. To generate the training dataset for the deep meta-classifier, they write the parameter of trained CNNs as a vector, though it is noted that an improved representation of model parameters as data points would be beneficial. Additionally, it is stated that the proposed deep meta-classifier can potentially be used for property inference attacks due to its ability to reveal information about the training setup.

In contrast to the above works that operate on models that were trained locally, [37] demonstrates a property inference attack for federated learning. By observing the jointly learned model parameters, the adversary records model snapshots and tries to determine if these were based on training data containing a targeted property. With the help of auxiliary data for which the property is known, an attack model is then trained based on the gradients between model snapshots. Moreover, the described attack allows an adversary to introduce the classification task of determining the target property for data points to the jointly trained model.

### 3.3 Differential Privacy

DP was first introduced by Dwork in [15]. Informally speaking, it aims at giving a provable bound to the probability with which an adversary can distinguish between the output a DP-secure mechanism gives for two neighboring datasets, i.e. to datasets that only differ in one record. In its simplest form of  $\epsilon$ -DP, the parameter  $\epsilon$  is defined to describes the upper bound for this probability, and it is therefore called the *privacy budget*. A relaxed version was given with  $(\epsilon, \delta)$ -DP [16] to allow a private mechanism to fail the given guarantees with a small probability  $\delta$ .

Multiple varieties or relaxed versions of DP were proposed in the following, as they allow to find tighter bounds regarding the necessary privacy budget. Most of these relaxations aim at improving the privacy analysis for the Gaussian mechanism and the composition of mechanisms. As one of these approaches, [14] proposed Concentrated Differential Privacy, that tries to improve upon  $(\epsilon, \delta)$ -DP in these aspects. It was subsequently reformulated in [9], where the Rényi divergence [48] is used to



further improve upon the previous results. RDP [38], discussed in more detail in Section 2.3.2, is another approach that makes use of the Rényi divergence. In contrast to Concentrated Differential Privacy, it allows conversion from privacy guarantees given in RDP to  $(\epsilon, \delta)$ -DP. f-DP [13] aims at better interpretability compared to these divergence-based approaches, besides also providing improved tools to bound the privacy of mechanism and their compositions. It subsumes  $(\epsilon, \delta)$ -DP and therefore also allows conversion between f-DP and  $(\epsilon, \delta)$ -guarantees.

Another line of work deals with the problem of choosing appropriate parameters for DP, such as [33] which tries to give guidelines on how to choose the privacy budget for  $\epsilon$ -DP and argues that this is dependent on the setting in which a differential privacy mechanism is deployed. Similarly, [27] explores a method to find acceptable parameters for  $(\epsilon, \delta)$ -DP with respect to a trade-off between privacy and utility of a constructed system.

### 3.3.1 Differential Privacy for Machine Learning

Song et al. [55] apply DP to stochastic gradient descent (DP-SGD), theoretically making it available to a large number of number ML algorithms that process privacy-sensitive data, though the approach is limited to convex objectives and the need for non-meaningful privacy budget. Abadi et al. [1] build upon this by presenting an algorithmic technique they call the *moments accountant* to refine the analysis of the privacy cost for training an ML model and allows for non-convex objectives. With this, they enable the use of DP in further ML techniques as neural networks.

Other publications try to further improve upon these results in terms of model performance and privacy guarantees, such as [65] that in turn builds upon DP-SGD and the moments accountant. The authors identify several issues with the methods used by [1] and make use of Concentrated Differential Privacy from [14] among other modifications. In contrast to this, [8] demonstrates that it is possible to train differentially private neural networks with improved privacy guarantees without using the moments accountant, by applying f-DP from [13]. Moreover, works like [61] provide methods to train differentially private models with federated learning, or allow the creation of differentially private, synthetic data with GANs [30][57].

The PATE framework presented by Papernot et al. [43] also aims at providing a method to create models with competitive performance and privacy guarantees but also puts a heavy focus on explainability. Their approach is further refined in [44], among other things by using Rényi DP from [38] and an improved technique for transferring knowledge from teacher to student that is more economical regarding the privacy budget. The PATE framework is explained in more detail in Section 2.4.

### 3 Related Work

Another line of publications does not directly focus on creating models with increased performance or utility, but inspects the effects of DP has on the vulnerability of models to attacks on privacy described above. Here, Rahman et al. [47] apply membership inference attacks against differentially private models trained with the moment’s accountant from [1]. They find that models with low privacy budgets are more resilient to these attacks but at the same time show a drastic reduction of model performance. Similar findings are stated by Jayaraman et al. [28] who also apply membership as well as attribute inference attacks against models trained with different relaxations of DP. Zhang et al. [67], on the other hand, attack a differentially trained model with their presented model inversion attack, with the result does not help to protect against such an attack.

## 4 Attacking Differentially Private CNNs Trained with PATE

This section will discuss the PATE framework against the background of known attacks against ML models described in Section 2.2 and Section 3.2. The goal here is to inspect to which degree the structure of the approach itself protects sensitive data used to train an ML model, as well as to find a method to assess the concrete meaning of given privacy guarantees.

First of all, a look is taken at the setting in which PATE models are trained and in which potential attacks against them take place. From these considerations, it becomes already apparent that PATE most likely reduces the attack surface that can be used by an adversary to extract sensitive information. Next, this thesis follows previous work such as [47] in targeting differentially private models with known attacks to evaluate the concrete meaning of given privacy guarantees. Here, the dependency between the privacy budget used to train differentially private models and the resulting privacy will be inspected closely.

On the one hand, PATE is a mechanism to implement DP for ML models, and given privacy guarantees given by the framework relate to the privacy of the data of individuals (see Section 2.3). On the other hand, the attacks discussed in Section 2.2.3 and Section 3.2.3 target the extraction of sensitive information over the whole dataset or its underlying distribution. Hence, PATE will be examined under both aspects of privacy.

As seen in Section 2.4, the last step of training a model with PATE consists of semi-supervised learning, based on the unlabeled dataset  $D_{pub}$  augmented with labels created via the teacher ensemble. For this work, the training of the student models will be limited to a supervised training of the student model with the help of created labels and their associated data points. Semi-supervised learning is beyond the scope of this thesis but is expected to improve student and influence attack performance. Therefore, it is a promising avenue for future work.

The following section focuses on the discussion on how to utilize different classes of attacks for this evaluation while concrete experiments and their results will be described in Chapter 5 and Chapter 6

## 4.1 Adversarial Setting

One of the goals of PATE is that its privacy guarantees still hold if the student model and the dataset  $D_{pub}$  are made public [44]. On the one hand, this corresponds to a white-box setting with respect to the student model, in which an adversary has full access to the target model internals such as model architecture and parameters. Teacher ensemble and labels created to train the student, on the other hand, are discarded after the training process and are assumed not to be accessible by an adversary. The same applies to the sensitive dataset  $D_{priv}$ , which is kept private.

While the goal of an adversary is to extract information about  $D_{priv}$  which contains sensitive information, the structure of PATE described in Section 2.4 imposes multiple hurdles to this. One major point is that the adversary is assumed to have only access to the already public dataset  $D_{pub}$  and a student model that never was directly trained on sensitive data. The only channel for sensitive information to flow from  $D_{priv}$  to the student is through the noisy teacher voting mechanism, restricted by the set privacy budget. The Confident-GNMax mechanism that is relevant for this thesis randomizes the teacher votes with Gaussian noise, hiding the influence of single teachers and potentially disturbing the overall result. Moreover, results of the aggregator are merely labels, not confidences which further hides the behavior of the teacher ensemble. All this affects and limits the potential attack surface for an adversary.

With increasing privacy budget, the noisy teacher voting mechanism is expected to release more sensitive information in the form of labels for  $D_{pub}$ , which increases the potential for a successful attack on privacy. Nevertheless, the fact that data points used to train the student only stem from public data restrict the attack surface for the adversary independently of the privacy budget.

## 4.2 Evaluating PATE Privacy Guarantees through Membership Inference Attacks

The PATE framework allows training differentially private student models, and with this provides privacy guarantees regarding individuals. As discussed in Section 2.3, DP uses the parameter  $\epsilon$  to bound the influence of data points on a mechanism’s output. Membership inference attacks directly try to breach this understanding of privacy by detecting the difference that one data point of one individual creates in the behavior of a model [47], which is the reason why this type of attack lends itself to evaluate DP guarantees in practice. By doing so, this thesis follows previous work such as [47], [46], and [28].

The goal of an adversary executing a membership inference attack is to extract membership information with regard to sensitive data used to create the target model. In the case of PATE, this means an attack is successful if it correctly predicts whether a given data point is a member of  $D_{priv}$ . To do so, the adversary targets the student model that is produced by the PATE framework and afterward deployed or shared.

Moreover, the PATE framework assumes an adversary to have full access to the target model, which implies query access necessary for membership inference attacks. The full access also allows adversarial knowledge of the student’s model architecture which is required by attacks such as the shadow training technique discussed in Section 2.2.2. Depending on the concrete attack, additional adversarial knowledge such as knowledge of hyperparameters has to be assumed.

target model	PATE student (query access)
inferred information	membership with respect to $D_{priv}$
adversarial knowledge	(additional knowledge depending on concrete attack)

Table 4.1: Adversarial setting for membership inference attacks targeting PATE student models and  $D_{priv}$ .

The privacy guarantees given by PATE are dependent on the value that was chosen for the privacy budget hyperparameter. As this hyperparameter influences the number of labels created by the GNMax mechanism for  $D_{pub}$ , it also affects the performance of produced student models. The evaluation of PATE with the help of membership inference attacks will therefore be done with respect to the privacy budget while taking the reached student performance into account. Moreover, membership inference is affected by factors such as the degree to which models are overfitted [36, 53, 64] and the skewedness [56, 58, 63] and complexity [58] of datasets (see Section 3.2.1 for more detail). By partitioning the training data and using a majority vote for determining labels for  $D_{pub}$ , these effects might be increased by the PATE framework. The experiments discussed in Section 5.2 privacy will also take these factors into account.

### 4.3 Model Inversion Attacks

A direct model inversion attack, as discussed in Section 3.2.2, would require access to a model that is trained on the targeted data. For PATE, this would mean access to the models of the teacher ensemble as all sensitive data is contained  $D_{priv}$ . The PATE training procedure discussed in Section 2.4 explicitly states that teachers are to be discarded after the labeling process to prevent a breach of privacy in this way.

Moreover, attacks using model inversion discussed in Section 3.2.2, such as the ones presented in [18] and [67], focus on either reconstructing the average features learned by a model for a given class or try to derive a data point for the distribution underlying the training dataset. They target models that are used in settings such as facial recognition systems, where adversaries must be prevented to gain such knowledge. In the setting used to train models with PATE, it is assumed that an adversary can access the student, as well as the public dataset  $D_{pub}$ . Through this, the adversary has access to data points from the targeted distribution and can label these with the help of the student model. Model inversion is therefore not even necessary to reach the same results, and it is questionable if these attacks are a threat to privacy.

Reconstructing the average features for a class as learned by the model could provide information about the dataset  $D_{priv}$  if its underlying distribution differs from that of  $D_{pub}$ . To identify this difference, the adversary would either need to know the true labels for  $D_{pub}$  or the distribution from which the dataset was sampled. Then again, a model inversion would not be necessary as they could compare it directly to the output created by the student model. Finally, targeting differences between the distributions underlying  $D_{priv}$  and  $D_{pub}$  is similar to extracting the value for a property from  $D_{priv}$ , and therefore might be simpler to achieve with the help of property inference attacks which will be discussed below.

For the above reasons, model inversion attacks will be regarded as a low threat to models trained with PATE and omitted from the experimental evaluation of the PATE framework.

## 4.4 Property Inference Attacks

As discussed in Section 2.2.3, the goal of property inference attacks is to extract information over the whole training dataset of a model with regards to a binary property  $\mathcal{P}$ . The model thereby only implicitly learns  $\mathcal{P}$ , as it is not explicitly contained in the dataset and not part of the model’s task. Property inference attacks were first demonstrated for models used in the field of voice recognition [4], with a property that can be adapted as  $\mathcal{P}_{accent}$ : *100% of the target model’s training dataset consists of data points created by people who speak an Indian English dialect*. Similar properties that are applicable to image datasets can be formulated. An example that is used below is  $\mathcal{P}_{contrast}$ : *Class 0 of  $D_{priv}$  used to create the target model contains 50% images with enhanced contrast*.

Note that the formulated properties for the inference attack refer to a target model. By inferring the property, the adversary gains information on sensitive data only indirectly. Targeting PATE students, the adversary has to bridge even more steps

of indirection. Here,  $D_{priv}$  is used to train the PATE teacher models, which in turn produce labels harnessed for the training of the targeted student model.

Two issues prevent the direct application of existing property inference attacks, such as described in Section 2.2.3, to student models trained with PATE. First, a method to vectorize trained CNNs for creating the training dataset of the meta-classifier needs to be found, as previous work targets either simpler models such as SVMs and HMMs or fully connected neural networks. This method needs to be able to retain the information on the parameters of the CNNs, to form a dataset with which a meta-classifier can be trained successfully. Second, creating the training dataset for the attack model out of fully trained PATE students would require massive amounts of computational power as each requires the creation of a complete teacher ensemble.

An alternative approach will therefore be focused first in Section 4.4.1, which utilizes the output of the student model rather than its parameters. Afterward, a modification to the attacks from [4] and [20] will be sketched, that could allow attacks on models trained with PATE through a meta-classifier.

##### 4.4.1 Distance-based Property Inference Attack

The goal of the method discussed in this section is to infer the value of a binary property for a PATE student model from its output. The attack is based on measuring the KLD between the confidences produced by the targeted student and the confidences output by models the adversary produced and for which they know the value of  $\mathcal{P}$ . The method is called a distance-based property inference attack, analogous to the distance-based membership inference attacks proposed in [35] by which it is motivated.

Two variants of the proposed attack are presented below. The goal for the first variant is to establish the basic approach and to determine whether it allows extracting information regarding  $\mathcal{P}$  from a PATE student. It is thereby an attack only in the theoretical sense as multiple strong assumptions regarding adversarial knowledge have to be made. These assumptions subsequently are loosened by a second variant and a method to restore the labeled subset  $D_{lpub}$  of  $D_{pub}$  with the help of membership inference. Lastly, an approach to extend the proposed distance-based property inference attack by using multiple properties is discussed.

**Attack using  $D_{priv}$ -based Ensembles**

For this theoretical distance-based property inference attack, the adversary is expected to know the model type and architecture, as well as hyperparameters for both models and the PATE framework. Additionally, they are assumed to know to the private dataset  $D_{priv}$ , and the part of  $D_{pub}$  that was labeled with the help of the GNMax mechanism, called  $D_{lpub}$ . The models targeted by this method are the student models resulting from executing the PATE framework. The attack is deemed successful if it correctly predicts the property  $\mathcal{P}$  from the target model's output.

target model	PATE student (query access)
inferred information	property $\mathcal{P}$ for target model
adversarial knowledge	$D_{priv}$ , $D_{pub}$ , $D_{lpub}$ , architecture and hyperparameters for PATE models

Table 4.2: Adversarial setting for the distance-based property inference attack using ensembles trained on  $D_{priv}$ .

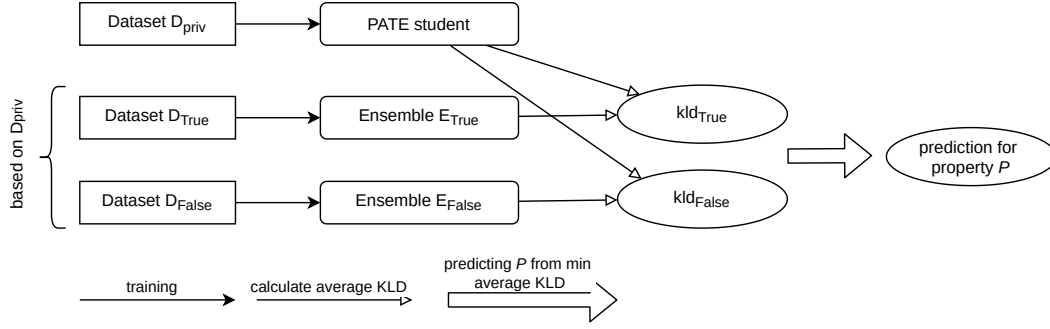
In the first step, the adversary creates two datasets  $D_{True}$  and  $D_{False}$  based on  $D_{priv}$ . Thereby,  $\mathcal{P}$  is *True* for models trained on  $D_{True}$  and *False* for those trained on  $D_{False}$ .

For the second step, the adversary trains two ensembles called  $E_{True}$  and  $E_{False}$  that are based on the datasets  $D_{True}$  and  $D_{False}$ . This is done in the same manner as training a PATE teacher ensemble for  $D_{priv}$ : The adversary partitions each of the datasets  $D_{True}$  and  $D_{False}$  according to the number of PATE teachers and trains one model per partition. They thereby use the model architecture and hyperparameters employed to train the PATE teachers.

For the third step, the adversary calculates the mean KLD between the student's confidences and the ensemble models' confidences over all data points in  $D_{lpub}$ . For simplicity, this measure will be called the average KLD between the student and a teacher ensemble. Note that  $D_{lpub}$  only contains data points, and no knowledge of the labels produced by the GNMax is necessary.

Finally, the adversary assumes that the target model has the same value for  $\mathcal{P}$  as the ensemble with the smaller average KLD to the student. Figure 4.1 gives an schematic overview of the attack while algorithm 1 shows pseudo-code for its last two steps.



Figure 4.1: Schematic overview of the distance-based property inference using  $D_{priv}$ .**Algorithm 1:** Distance-based attack using  $D_{priv}$ -based Ensembles

---

**input** : dataset  $D_{lpub}$ ,  
target student model  $S$ ,  
ensemble  $E_{True}$  based on  $D_{True}$ ,  
ensemble  $E_{False}$  based on  $D_{False}$   
**output**: prediction for property  $\mathcal{P}$

---

```

 $kld_{True} \leftarrow 0$ 
 $kld_{False} \leftarrow 0$ 
for  $d \in D_{lpub}$  do
     $sc \leftarrow S(d)$ 
    for  $T \in E_{True}$  do
         $tc \leftarrow T(d)$ 
         $kld_{True} \leftarrow kld_{True} + KLD(sc, tc)$ 
    for  $T \in E_{False}$  do
         $tc \leftarrow T(d)$ 
         $kld_{False} \leftarrow kld_{False} + KLD(sc, tc)$ 
if  $kld_{False}/|D_{lpub}| < kld_{True}/|D_{lpub}|$  then
    return  $True$ 
else
    return  $False$ 

```

---

The idea behind this approach is that the behavior of the teachers trained on  $D_{priv}$  is slightly modified depending on the property value of  $\mathcal{P}$ . The student reflects this difference in behavior, as the labels for  $D_{lpub}$  are based on the output of the teacher ensemble. By artificially creating two teacher ensembles with different values for  $\mathcal{P}$ , the adversary can determine which is more likely to be similar to the ensemble that produced the target student model. They can therefore infer the value of  $\mathcal{P}$  for the target model and gain information on  $D_{priv}$ .

**Attack using  $D_{pub}$ -based Ensembles**

For the second proposed variant of the distance-based property inference attack, the goal is to refine the approach from above by reducing the necessary assumptions made regarding the adversarial knowledge. Here, the adversary is not assumed to have access to dataset  $D_{priv}$  which results in a more realistic adversarial setting. The weaker assumption that the adversary knows  $D_{lpub}$ , which is the labeled portion of  $D_{pub}$ , remains.

target model	PATE student (query access)
inferred information	property $\mathcal{P}$ for target model
adversarial knowledge	$D_{pub}$ , $D_{lpub}$ , architecture and hyperparameters for PATE models

Table 4.3: Adversarial setting for the distance-based property inference attack using ensembles trained on  $D_{pub}$ .

Without access to  $D_{priv}$ , the adversary creates ensembles based on a dataset that is similar with regards to the underlying distribution. This has the goal to emulate the ensembles based on  $D_{priv}$  from above and their behavior. One candidate for such a dataset is  $D_{pub}$  as it is known to the attacker. Using only  $D_{upub} = D_{pub} \setminus D_{lpub}$  guarantees that the data used to create the attack models was not used for the training of PATE students. The rest of the attack is analogous to the first attack variant that uses ensembles based on  $D_{priv}$ .

**Restoring Labeled Data Points  $D_{lpub}$  from  $D_{pub}$** 

The distance-based property inference attacks described above require adversarial knowledge of  $D_{lpub}$ . This section discusses how an adversary can use membership inference to restore  $D_{lpub}$  from  $D_{pub}$ . The approach presented here does not reveal sensitive information but is solely used to reduce the adversarial knowledge necessary for subsequent attacks.

target model	PATE student (query access)
inferred information	membership with respect to $D_{lpub}$
adversarial knowledge	$D_{pub}$ , (additional knowledge depending on concrete attack)

Table 4.4: Adversarial setting for membership inference attacks extracting  $D_{lpub}$  PATE student models.

In the context of this work, the PATE student is trained in a supervised fashion, which takes place exclusively on  $D_{lpub}$  and the labels provided by the GNMax-

Aggregation mechanism with the help of teachers. The dataset  $D_{lpub}$  thereby is equal to the training dataset of a targeted PATE student model. An adversary trying to learn  $D_{lpub}$  can execute a membership inference attack against the PATE student model on the basis of  $D_{pub}$ . They then assume that data points for which the attack predicts membership for the training dataset of the module with high probability are part of  $D_{lpub}$ . All other data points are assumed to be part of  $D_{upub} = D_{pub} \setminus D_{lpub}$  which is the unlabeled part of the public data. Finally, predicting the labels created by the GNMax mechanism is then possible by calculating the targeted PATE student’s predictions for  $D_{lpub}$ .

### Extended Attack using Multiple Properties

The attack presented here is based on the previously discussed distance-based property inference attacks. Both of the proposed variants aim at determining whether a binary property applies to a PATE student model and therefore indirectly to  $D_{priv}$ . The properties formulated in [4] and Section 5.3.1 thereby state that a percentage of the sensitive data has a specific quality<sup>1</sup>. This is an Indian English accent or enhanced contrast for the given examples. The attack discussed here aims at inferring the percentage  $m$  by which a quality  $Q$  applies to a target model. The adversarial setting is thereby the same as for the chosen distance-based attack variant it is built on.

target model	PATE student (query access)
inferred information	percentage $m$ to which Quality $Q$ applies to $D_{priv}$
adversarial knowledge	$D_{pub}$ , $D_{lpub}$ , ( $D_{priv}$ depending on used basic attack), architecture and hyperparameters for PATE models

Table 4.5: Adversarial setting for the extended property inference attack using multiple properties.

To infer the percentage  $m$  to which a quality  $Q$  applies to the target model, the adversary formulates  $k$  properties of the type  $\mathcal{P}_Q^{m_i}$ : *The target model was trained on a dataset for which Quality  $Q$  applies to  $m_i\%$  of the data points, for  $i \in \{1, \dots, k\}$ .*

The subsequent attack is executed analogously to the distance-based property inference attack discussed above: For each of the properties  $\mathcal{P}_Q^{m_i}$ , the adversary produces a dataset  $D_Q^{m_i}$  based on either  $D_{priv}$  or  $D_{upub}$  and trains one ensemble of models  $E_{m_i}$  based on it. Thereby,  $D_Q^{m_i}$  is created so that  $\mathcal{P}_Q^{m_i}$  is *True* for models from  $E_{m_i}$ . The average KLD between each of these ensembles and the target model then is then calculated for  $D_{lpub}$ .

<sup>1</sup>The term property as used in [4] always applies to a complete dataset. The term quality is used here to denote a property applying only to one data point.

#### 4 Attacking Differentially Private CNNs Trained with PATE

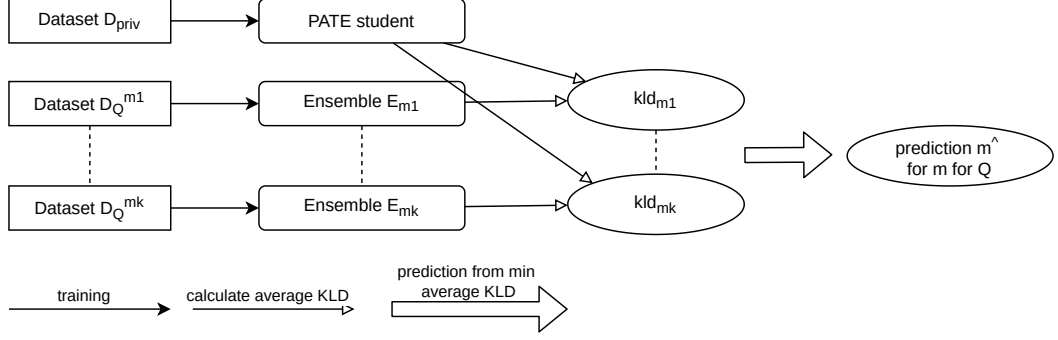


Figure 4.2: Schematic overview of the distance-based attack that predicts the percentage  $m$  by which a quality  $Q$  applies to  $D_{priv}$ .

The attack returns a prediction  $\hat{m}$  for percentage  $m$  according to the ensemble that shows the lowest average KLD to the student. It is deemed successful if  $\hat{m}$  is equal to the  $m_i$  for which  $|m - m_i|$  is minimal. A schematic overview of the attack is given in Figure 4.2

##### 4.4.2 Deep Meta-Classifier

Known property inference attacks as described in Section 2.2.3 use a meta-classifier to determine whether the property  $\mathcal{P}$  under scrutiny is *True* for the training dataset of the target model. At the beginning of this section, the complications were described that prevent the direct application of known attacks to CNNs trained with PATE. Here, a short sketch of how to solve them will be given that is later used to conduct a preliminary experiment as an initial test for the feasibility of the proposed method.

target model	PATE student (access to model parameters)
inferred information	property $\mathcal{P}$ for target model
adversarial knowledge	$D_{pub}$ , architecture and hyperparameters for PATE models

Table 4.6: Adversarial setting for the property inference attack using a deep meta-classifier.

The first problem arises from the need to vectorize models to create the attack dataset for the training of the meta-classifier. Different model types require different methods to transform their parameters to data points, as shown in [20], and those used in [4] and [20] are not necessarily applicable to CNNs. [17] uses a simple method to write the parameters of CNNs to a one-dimensional vector, which, as they argue,

retains some of the spacial structure of a CNN’s parameters<sup>2</sup>. Though the authors do not use this vectorization method for creating a dataset for property inference, their use case of recognizing training datasets and hyperparameters used to create a target model is similar. Therefore, not only their representation of model parameters is used, but also their proposed meta-classifier model. The latter is called a deep meta-classifier as it is a deep neural network.

The second complication when applying property inference attacks to models trained with PATE is the high number of models that need to be trained to create the attack dataset. The models used in [4], [20] and [17] are fairly simple, potentially as the creation of this dataset is very cost intensive. Tasks that require more complex models complicate the training of a meta-classifier in two ways: First, increased computational power is needed to train the more complex models that are used as data points. Second, the higher number of parameters in the models equals an increase in the input dimensions. This increases the complexity of the task for the meta-classifier, potentially making a more complex meta-classifier model necessary and again requiring more training data. For PATE, this problem is intensified by the fact that training a student model requires creating a complete ensemble of teachers, and executing the noisy teacher voting mechanism until the privacy budget is exhausted.

One potential way to circumvent executing the complete PATE framework for the training of each student is to create a dataset from models emulating PATE students. These models are trained similarly to the students but directly on a labeled dataset and not via teacher labels. With PATE, it is assumed that the adversary has access to the public dataset  $D_{pub}$  and the student model. In the following, it will also be assumed that they have knowledge of the hyperparameters used for the GNMax mechanism and the student training. By creating a teacher ensemble and executing the GNMax mechanism, the adversary estimates the number of labels used to train the student<sup>3</sup>. Then, the attacker creates one dataset of that size for each data point they aim to be contained in the attack dataset. Thereby, they manipulate the data so that  $\mathcal{P}$  is *True* for half of the created datasets and *False* for the other half. The rest of the attack then follows the one discussed in [4].

---

<sup>2</sup>The parameters of CNNs are ordered in layers of two-dimensional kernels, as discussed in more detail in [21].

<sup>3</sup>Here, the adversary might assume that the same model type and architecture were used for teachers and the student, as is the case in [44].



## 5 Experiments

The following sections discuss the concrete experiments that were conducted for the evaluation of PATE. First, Section 5.1 describes the setting in which multiple PATE student models for a variety of datasets and privacy budgets are trained. Subsequent sections outline the concrete setup for multiple experiments following the attacks discussed in Chapter 4.

### 5.1 Creating Baseline and PATE models

This section details the selected datasets, model architecture, and hyperparameters used to train all PATE models. Additionally, a series of non-private models were created as a baseline to which the experimental results are compared. All of them were written in and trained with the Keras API of the TensorFlow project [2].

#### 5.1.1 Datasets

The experiments of this section were conducted on the datasets SVHN [41] and FashionMNIST[62], as well as the Letters part of the EMNIST dataset [11]. The SVHN dataset contains  $32 \times 32$  color images of single digits cropped from photos taken of house numbers and labeled with 10 classes according to the digit they show. It was selected to ensure comparability of the results produced by the experiments conducted here with those shown in [44]. EMNIST is a superset of the MNIST dataset [32] from which different datasets can be derived. The Letters dataset is thereby a collection of  $28 \times 28$  grey-scale images of handwritten letters, which results in the 26 contained classes. This dataset was selected for its higher number of classes compared to the SVHN dataset. In addition, FashionMNIST was selected as a slightly more complex replacement for the standard MNIST dataset, depicting 10 different classes of pieces of clothing on  $28 \times 28$  grayscale images.

The choice for the above datasets was also influenced by the relatively high number of samples contained in them with regards to their complexity. Need for big datasets when training with PATE arises from the fact that  $D_{priv}$  is divided into partitions

## 5 Experiments

equal to the number of teachers. Initial experiments using the more complex CIFAR-10 dataset [31] indicated that dividing its train split used as  $D_{priv}$  into as few as 10 partitions resulted in a teacher test accuracy around 52% or below. For this, the same model architecture as described below was used, and changing it for a ResNet-20 did not improve the results. In contrast, a much higher teacher accuracy could be reached for the FashionMNIST dataset, though it does not contain significantly more data points. This is most likely due to the lower complexity of the classification task associated with it.

All three of the used datasets define a train and test split. SVHN additionally offers an extra split. The training of models for which PATE is not used was done directly on the train split, and the models are evaluated on the test split. For use with PATE, the datasets were restructured to match the structure described in Section 2.4. Therefore, the train split was regarded as sensitive data and used as  $D_{priv}$ . For SVHN, the extra split was also regarded as sensitive and added to  $D_{priv}$ , following [43]. The test split of the datasets was divided into 80% public data used in  $D_{pub}$  and 20% test data to form  $D_{test}$ . Thereby, It was made sure that the distribution of labels in all three data sets used for PATE roughly represents the distribution of the original datasets dataset. It should be noted that the SVHN dataset is naturally skewed, a fact that was considered in the experiments and their evaluation. The distribution of all datasets in the form as they were used for PATE is shown in Figure 5.1.

### 5.1.2 Model Architecture

The same model architecture of a simple CNN was used for all teacher and student models created with PATE for all three datasets. The same is true for all non-private baseline models trained without PATE. It was adapted from [43] where it was used for models trained on MNIST and SVHN. Additionally, the same architecture is reused in [44] for repeating previous experiments on the same datasets with the Confident-GNMax considered throughout this work. An overview of its layers is shown in Table 5.1.

The source code<sup>1</sup> published for [43] contains a model definition that allows enabling multiple dropout layers on creation, though the publication does not mention whether they were used for the presented experiments. As dropout is a powerful regularizer [21] and known mitigation against membership inference [50], it was not included in the experiments conducted here. This decision has been made to study the effects of PATE independently. Though also potentially affecting the performance of membership inference attacks, a weak L2-norm regularization was applied during

---

<sup>1</sup>The git repository for [43] can be found in the TensorFlow Privacy project at [https://github.com/tensorflow/privacy/tree/master/research/pate\\_2017](https://github.com/tensorflow/privacy/tree/master/research/pate_2017).



Layer	Layer Type	Parameters
0	Input	$32 \times 32 \times 3$ (SVHN) / $28 \times 28 \times 1$ (Letters)
1	Conv2D	kernel: $5 \times 5$ , filters: 32, activation: 'ReLU'
2	BatchNormalization	
3	MaxPool2D	pool size: (3, 3), strides: 2
4	Conv2D	kernel: $5 \times 5$ , filters: 32, activation: 'ReLU'
5	BatchNormalization	
6	MaxPool2D	pool size: (3, 3), strides: 2
5	Flatten	
7	Dense	units: 128, activation: 'ReLU'
8	Dense	units: 10 (SVHN) / 26 (Letters)

Table 5.1: The model architecture used for all models on all datasets. The model has a total number of 633,546 parameters for SVHN, 509,533 parameters for Letters and 507,466 parameters for FashionMNIST.

the training process for the teachers based on SVHN as it allowed to reach a teacher accuracy that is comparable to the results from [43] and seems to be enabled by default for the teacher models used there.

### 5.1.3 Model Training

All models, the non-private baselines as well as the teacher and student models, were trained with the Adam optimizer and the exponential decay learning rate schedule from the TensorFlow project. The initial learning rate was set to be  $1e-3$  and decayed each epoch by 0.95. Table 5.2 shows the batch sizes and the number of training epochs for baseline, teacher, and student models for all datasets. Note that for training a teacher model, the associated partition of  $D_{priv}$  was repeated 20 times in different order each epoch<sup>2</sup>, increasing the effective number of epochs by 20 and results in a much more stable training procedure.

For SVHN, a simple data augmentation method was applied that scales the images to  $36 \times 36 \times 3$  and then produces a crop of the original image size at a random location. For Letters, the data augmentation was done by normalizing images followed by a small random zoom and rotation, whereas none was applied for the training of FashionMNIST. Augmenting the training data for teachers is possible, as each data point is assumed to completely influence the teacher outputs, as discussed in Section 2.4.3. Moreover, DP is immune to post-processing (see [16] Prop. 2.1). For this reason, no additional privacy loss incurs from the data augmentation used for

<sup>2</sup>This process of repeating and shuffling the training dataset was done for each partition of  $D_{priv}$  separately, making sure that each data point of  $D_{priv}$  only affects one teacher.

## 5 Experiments

Hyperparameter	SVHN	Letters	FashionMNIST
batch size	32	64	128
epochs (baseline)	20	20	20
epochs (teachers)	20	25	30
epochs (students)	50	70	50
patience (baseline)	3	3	-
patience (teachers)	10	25	-
patience (students)	10	30	-

Table 5.2: Hyperparameters for the training of models on all datasets. All models were trained with the Adam optimizer with a learning rate of  $1e - 3$  and exponential decay with a decay factor of 0.95 each epoch. The patience for the early stopping callback used for models trained on SVHN and Letters is given in the number of epochs.

the training of student models.

In addition to the above, early stopping was used for the SVHN and Letters datasets to reliably reach reasonable accuracy, with patience set to a number of epochs as shown in Table 5.2. Note that the early stopping callback of TensorFlow Keras restores the model that performed best according to a given performance metric, here set to the validation accuracy. Therefore, even patience values as high as the total number of epochs are sensible, as they guarantee that the training does not stop early, but the best model will be saved. For the FashionMNIST dataset, no early stopping was necessary as reliably accurate models could be trained without it.

### 5.1.4 PATE Hyperparameters

The size of the teacher ensembles, as well as the hyperparameters for the Confident-GNMax, have to be selected to train student models with PATE. For the latter, the most notable is the privacy budget, which ends the creation of labels upon its exhaustion. As the influence of the granted privacy budget on both student performance and concrete privacy granted through PATE is studied in the experiments presented further below, multiple students for privacy budgets  $\epsilon$  of 1, 2, 3, 5, and 10 were trained. All other parameters mainly depend on the dataset and are fixed.

For the SVHN dataset, the hyperparameters given in [44] were used to ensure comparability of results. The Letters and FashionMNIST datasets are similar to MNIST. Therefore, the hyperparameters for that dataset were also adapted from [44], with the exception of the  $\delta$  parameter for the Letters dataset. Here, with  $\delta = 1e - 6$ , a smaller value was chosen as the train split of Letters contains 128400 data points compared to the 60000 data points for the train split of MNIST. An overview of the

Hyperparameter	SVHN	Letters	FashionMNIST
number of teachers	250	250	250
T	300	200	200
$\sigma_1$	200	150	150
$\sigma_2$	40	40	40
$\delta$	$1e - 6$	$1e - 5$	$1e - 6$

Table 5.3: List of hyperparameters used for the training with PATE.

used PATE hyperparameters is given in Table 5.3.

## 5.2 Membership Inference Attacks

This experiment evaluates the concrete meaning of the privacy guarantees for PATE students regarding membership inference. For executing the membership inference attacks on both non-private baseline models and PATE students resulting from the previous experiment, TensorFlow Privacy<sup>3</sup> is used. The framework implements a range of threshold-based and model-based attacks that can be used to target TensorFlow models, from which the two attacks THRESHOLD\_ATTACK and MULTILAYERED\_PERCEPTRON are selected for the experiments conducted here.

The selected attacks correspond to those detailed in Section 2.2.2, though no new shadow models are trained for the MULTILAYERED\_PERCEPTRON in contrast to the original shadow training technique. Instead, the target model is used in their place, much like in a situation in which the adversary has access to a shadow model that emulates the target model perfectly [6]. Moreover, the implemented attacks expect that two sets are passed as arguments. These contain only members and non-members of the target model’s training dataset. The knowledge of the membership status of the contained data points is thereby only used to evaluate the attack.

As a first step, the selected attacks are executed against the baseline models trained without PATE to inspect their performance on ML models that are trained without DP. Here, the train and test splits are passed to the attacks for the set of data points that contain members, and non-members respectively.

Next, the selected attacks are executed against the PATE student models for the different privacy budgets  $\epsilon$  of 1, 2, 3, 5, and 10. For this, the dataset  $D_{priv}$  and  $D_{test}$  are passed as the two required sets with members and non-members. This way, the attack targets not the direct training dataset of the PATE student but tries to

<sup>3</sup>Source code available at <https://github.com/tensorflow/privacy>.

## 5 Experiments

determine the membership status for data points from the sensitive data contained in  $D_{priv}$ .

TensorFlow privacy also implements the privacy risk score from [54] which will also be calculated for all models created during the experiment above. In contrast to the two selected membership inference attacks that allow the evaluation of the vulnerability of models over the whole dataset, the privacy score is calculated for each data point. It is “defined as the posterior probability that it is from the training set [...] after observing the target model’s behavior over that sample” [54] and the shadow training technique from [53] is used to calculate it. Therefore, the privacy risk score allows estimating the vulnerability to membership inference for single data points, where scores around 0.5 indicate a low vulnerability.

All results will be interpreted in Section 6.2 in a way that clarifies the overall influence of PATE on the model’s resilience to membership inference on the one hand, and the meaning of selecting a concrete privacy budget on the other.

### 5.3 Property Inference Attacks

The following section details experiments following the property inference attacks discussed in section Section 4.4.

#### 5.3.1 Distance-based Property Inference Attack

This section first describes experiments conducted with regards to the two variants of the distance-based property inference attack proposed above, followed by the details for an experiment on the extended attack using multiple properties that builds onto them.

##### Attack using $D_{priv}$ -based Ensembles

The SVHN dataset is used to conduct the theoretical attack for determining the property value for a PATE student model with the help of two ensembles, as described in Section 5.3.1.

This experiment uses the attack to infer two different properties from target models. They have the form  $\mathcal{P}_x$ : *Class  $x$  of  $D_{priv}$  used to create the target model contains 50% images with enhanced contrast*, where  $x$  is a class from the target dataset. Figure 5.1

shows that SVHN is skewed, with class 0 being one of the classes with the least and class 1 the class with the most data points. Hence, the experiments here are conducted on both of them. For the properties  $\mathcal{P}_0$  and  $\mathcal{P}_1$ , the attack is first executed on student models trained with privacy budgets  $\epsilon$  of 1, 2, 3, 5, and 10 for which the property is *True*. Then, the results are validated by repeating the same attack on student models for which the property is *False*.

Here, the operation of enhancing the contrast of an image is understood as stretching its pixel values over the complete interval of possible pixel values. For an image  $\mathbf{x}$ , it can be written as:

$$\begin{aligned} \text{minval} &= \min(\mathbf{x}) \\ \text{maxval} &= \max(\mathbf{x}) \\ \mathbf{x} &= ((\mathbf{x} - \text{minval}) / (\text{maxval} - \text{minval})) \end{aligned}$$

The training of the two ensembles on  $D_{priv}$  used in this attack follows the standard-setting for training PATE teachers described in Section 5.1. It was also made sure that  $D_{priv}$  was shuffled differently for the creation of the PATE teachers that were used for creating the student and for the ensembles employed in the attack.

### Attack using $D_{pub}$ -based Ensembles

The goal of the method described in Section 5.3.1 is to reduce the necessary assumptions made regarding the knowledge of the adversary, by replacing the PATE teacher ensembles trained on  $D_{priv}$  with ensembles based on data that is not sensitive. Hence, the ensembles used to decide the targeted property here are trained based on  $D_{upub}$ , the unlabeled part of  $D_{pub}$ . The adversary could decide to use a different dataset with the same underlying distribution as  $D_{pub}$ , for example, if they do not know the labels that were used to train the student. Here,  $D_{upub}$  is selected for convenience and to make sure that the student and teacher models do not share data points in their training data.

Two ensembles are trained with the help of the SVHN dataset - one on an unmodified version of  $D_{upub}$  and the other on a version to which the property applies <sup>4</sup>. Thereby, the training dataset for the ensembles is partitioned to reflect the partition sizes for teachers that were trained in Section 5.1.  $D_{upub}$  was shuffled and partitioned multiple times to increase the teacher ensemble size. Then, the created ensembles are pooled to form one ensemble. The ensemble training is realized using the teacher model architecture and hyperparameters from Section 5.1. Finally, the experiment is executed analogous to Section 5.3.1 to determine the same properties  $\mathcal{P}_0$  and  $\mathcal{P}_1$ .

<sup>4</sup>As the experiment is executed on multiple students models that were trained with different labels,  $D_{upub}$  is calculated separately for each of them.

## 5 Experiments

### Restoring Labeled Data Points $D_{lpub}$ from $D_{pub}$

This experiment evaluates whether the attacker can use a membership inference attack to extract the data points from  $D_{pub}$  that were labeled by the GNMax mechanism. For this purpose, the attack THRESHOLD\_ATTACK is selected from TensorFlow Privacy and executed against the PATE student models.  $D_{lpub}$  and  $D_{upub} = D_{pub} \setminus D_{lpub}$  as datasets containing members and nonmembers were passed as input to the attack. As THRESHOLD\_ATTACK requires labels for both sets, the ones created by the GNMax are passed for  $D_{lpub}$  and the original labels from the dataset for  $D_{upub}$ . The selected membership inference attack is executed against the targeted PATE student models trained on all inspected datasets and for the privacy budgets  $\epsilon$  of 1, 2, 3, 5, and 10.

It is expected that especially low privacy budgets result in high overfitting of PATE students, allowing for successful membership inference attacks. Such results would indicate that an adversary can use membership inference attack to extract  $D_{lpub}$  for use in further attacks such as the distance-based property inference attacks detailed in Section 4.4.1.

### Extended Attack using Multiple Properties

In this section, the experiment executing the attack detailed in Section 4.4.1 is described. Thereby, the distance-based property inference attack variant making use of ensembles trained with the help of  $D_{priv}$  was used as a basis.

The quality  $Q$ : *The image has enhanced contrast* is used, where enhanced contrast is understood as in the experiments above. Three properties of the form  $\mathcal{P}_Q^{m_i}$ : *The target model was trained on a dataset for which Quality  $Q$  applies to  $m_i\%$  of the data points* are formulated, for  $m_i \in \{25\%, 50\%, 75\%\}$ . The ensembles  $E_{m_i}$  were trained the datasets  $D_{m_i}$  derived from  $D_{priv}$  for SVHN, which makes sure that the properties  $\mathcal{P}_Q^{m_i}$  apply to the models contained in the ensembles. The experiment is executed targeting a single PATE student model that was trained on an SVHN version with 50% high-contrast images in  $D_{priv}$ .

#### 5.3.2 Deep Meta Classifier

This preliminary experiment for a property inference attack with the help of a deep meta-classifier consists of two steps: First, it is necessary to train and vectorize a set of models to create the attack dataset with train, validation, and train splits.

Second, the deep meta-classifier is trained with the help of the train and validation split and finally evaluated on the test split.

The following property  $\mathcal{P}$  is used to create the training datasets for the models that are then vectorized: *The number of data points for class 0 in  $D_{priv}$  used to create the target model is reduced to 25%.* The dataset for each of these models is based on the  $D_{upub}$  set of Letters, which is the unlabeled part of the  $D_{pub}$  dataset. This way, it is guaranteed that the target model does not share data points with the models vectorized for the attack dataset. The reason for choosing the Letters dataset is the lower number of model parameters compared to that of models for SVHN, which reduces the computational power needed. The architecture of the single models is the same as detailed in Table 5.1, with hyperparameters that are described in Table 5.2<sup>5</sup>.

For each CNNs trained this way, a vectorization method built on the `get_weights()` method from TensorFlow Keras is called on each of its layers. The parameters are concatenated to a single numpy array, and non-trainable weights produced by the used batch normalization layers are removed. Each vector thus created is saved as one data point for the attack dataset.

See [17] for the architecture for the deep meta-classifier that encompasses 21 one-dimensional convolutional and fully-connected layers as well as additional one-dimensional max pooling and dropout layers. The architecture was reimplemented with the help of the TensorFlow Keras API in the context of this work. For its training, the batch size is set to 24<sup>6</sup> and the TensorFlow Adam optimizer is used with its default learning rate of 0.001 and an exponential decay learning rate schedule. The decay factor of 0.95 is thereby applied every two epochs of the training, over a total of 100 epochs. The early stopping callback of TensorFlow Keras is used to capture the best performing model with respect to the validation accuracy. The patience for this callback is set to 30 epochs.

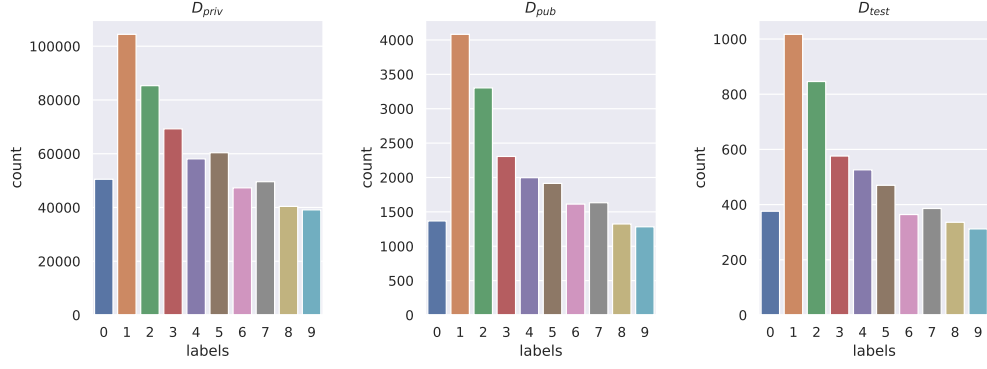
---

<sup>5</sup>Here, the hyperparameters from the PATE teacher training were used. Further experiments should switch to those used in the training of the PATE student models.

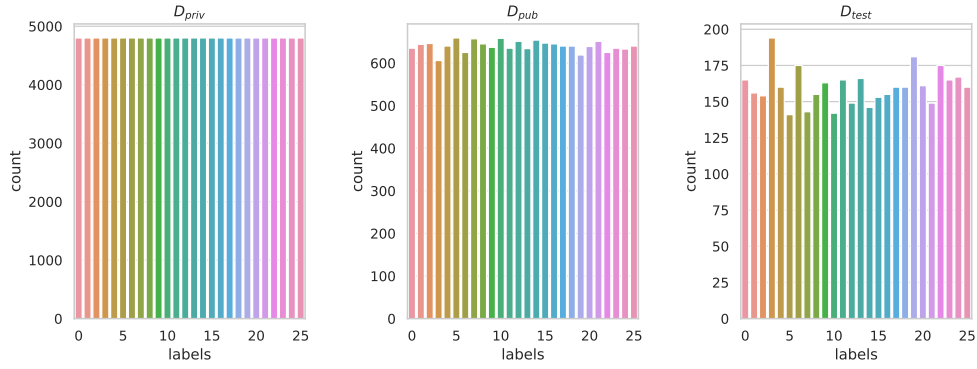
<sup>6</sup>The batch size of 24 was chosen due to the limitations of 12GB VRAM for the used NVIDIA Titan X graphics card. Training the model with a higher batch size would surely be beneficial.

## 5 Experiments

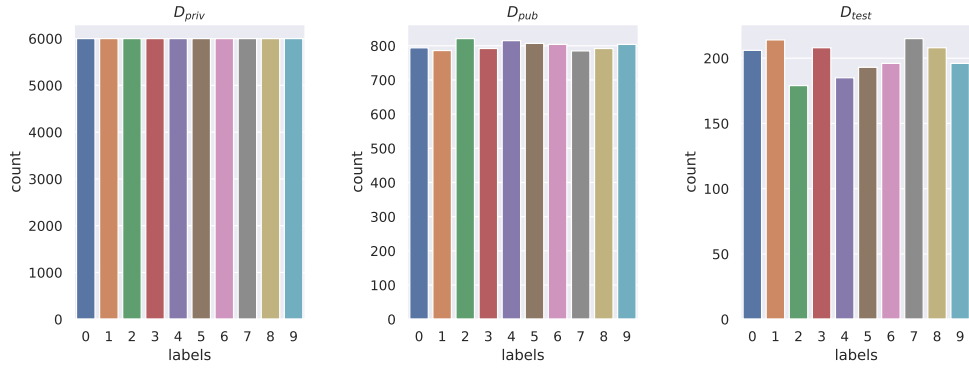
Label Distribution of Datasets



(a)



(b)



(c)

Figure 5.1: Distribution for the labels in SVHN (a), Letters (b) and FashionMNIST (c) after being separated into the PATE data splits.



## 6 Results

### 6.1 PATE Training

For the SVHN dataset, the PATE teacher ensemble reached an average test accuracy of 85.78%, a slightly better result than the reported average test accuracy of 83.18% from [43]. A high gap between this test accuracy and an average training accuracy of 99.43% indicates a high degree of overfitting for the teachers. Due to splitting the SVHN dataset into 250 partitions, this is not unexpected and does not give an advantage to attackers in the setting assumed by PATE, as teachers are discarded after training the student model.

The teachers for Letters and FashionMNIST reached similar train accuracy but lower test accuracies of 80.25% and 78.25%. It is possible to explain this through the dataset sizes and the number of labels created through the Confident-GNMax. For Letters, the  $D_{priv}$  dataset has 124800 data points compared to 604388 data points for SVHN. When taking into account the 26 classes for Letters, the partition used to train a teacher contains a mean of 19 samples per class, compared to 242 for SVHN. Similar is true for FashionMNIST, as the dataset has about the same number of classes as SVHN, but its associated  $D_{priv}$  dataset is more than ten times smaller.

The results from the of Confident-GNMax mechanism are shown in Table 6.1. When comparing the ones for SVHN with those presented in [44], the number of answered queries (i.e., generated labels) for the reached privacy bound  $\epsilon = 4.96$  closely match those for the privacy bound of 5 here. Therefore, the results shown here seem valid in this regard.

It is clearly visible that the number of created labels by the Confident-GNMax aggregator increase with the privacy budget. For the SVHN and Letters datasets and privacy budget of 10, the labeling process stopped before deleting it completely, as the maximum number of queries was reached, i.e. the size of  $D_{pub}$ . The label accuracy is noteworthy as well, ranging between 94% and 95% for SVHN, which is around 8-10% higher than the average teacher accuracy. The fact that the accuracy is stable over the budgets is not surprising as the same teacher ensemble is used for the noisy voting mechanism. The increase in accuracy compared to a single teacher

Dataset	Privacy Budget	$\epsilon$	Queries	Labels	Succ. Queries	Label Acc.	Student Acc.
SVHN	1	0.993	415	148	35.66%	94.18%	62.35%
	2	1.997	1611	573	35.57%	94.11%	81.26%
	3	2.998	3733	1309	35.07%	95.11%	85.06%
	5	5.000	9230	3206	34.73%	94.94%	87.58%
	10	8.072	20823	7248	34.81%	94.58%	89.42%
Letters	1	0.992	178	96	53.93%	88.90%	57.10%
	2	1.995	763	393	51.51%	91.35%	78.96%
	3	2.999	1553	796	51.26%	91.13%	81.54%
	5	4.999	3759	1959	52.11%	91.11%	84.00%
	10	9.998	11825	6175	52.22%	91.36%	85.17%
FashionMNIST	1	0.994	288	144	50.00%	92.22%	73.65%
	2	1.998	984	506	51.42%	90.04%	80.07%
	3	2.996	1949	1020	52.33%	89.45%	81.75%
	5	4.998	4873	2557	52.47%	88.89%	82.85%
	10	6.714	8000	4212	52.65%	88.70%	82.78%

Table 6.1: Results from Confident-GNMax label creation and Student Accuracies.

They show an average of 3 executions for each privacy budget. The teacher ensembles are based on varying partitions of  $D_{priv}$  and seeds for the initialization of model parameters. The baseline models trained without DP reach a mean test accuracy of 91.87% for SVHN, 92.19% for Letters, and 92.00% for FashionMNIST. The values from column  $\epsilon$  together with the  $\delta$ s from Table 5.2 form the average final  $(\epsilon, \delta)$ -DP guarantees depending on the privacy budget. The column Queries represents the median number of queries that were posed to the Confident-GNMax, whereas Labels gives the number of answered queries resulting in a label. The percentage to which the Confident-GNMax mechanism returned labels for posed queries is shown in the next column (Succ. Queries). The label accuracy (Label Acc.) was measured against the true labels of the  $D_{pub}$  dataset.

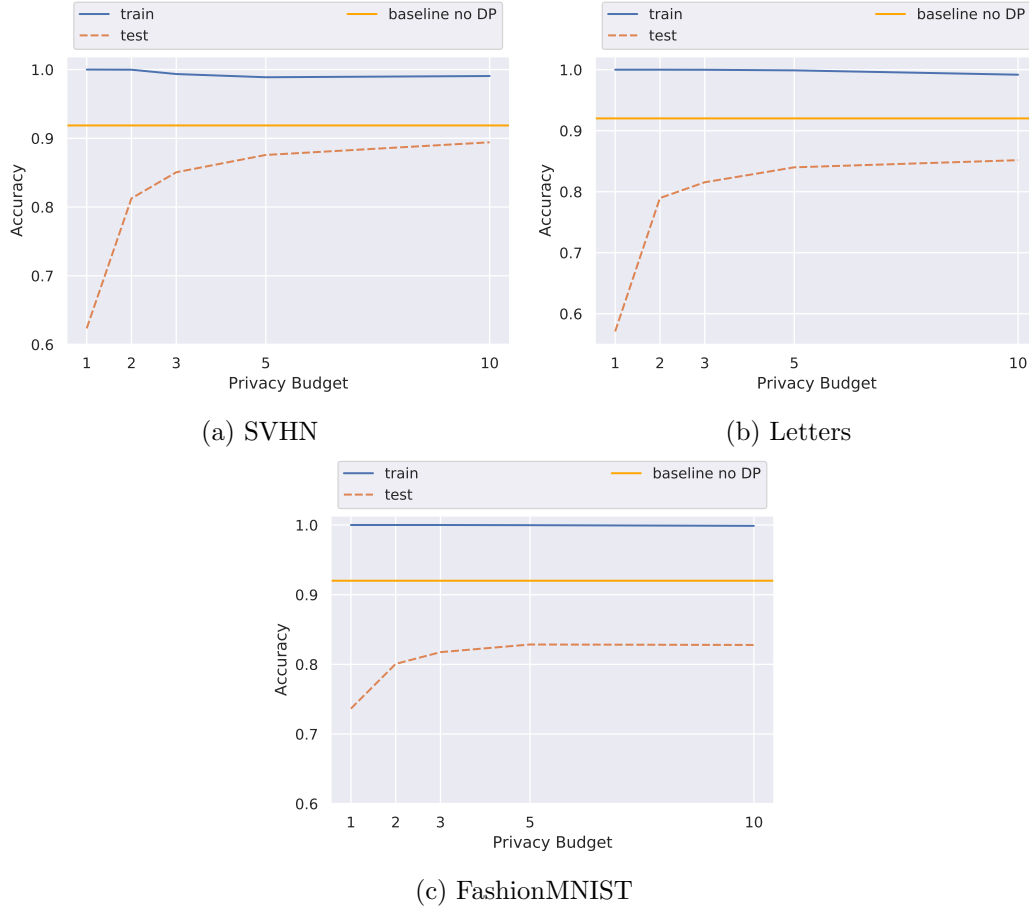


Figure 6.1: Accuracy for PATE students trained with different privacy budgets. The orange line represents the accuracy of the baseline models trained without PATE on the respective datasets. For PATE students, lower values for the privacy budget  $\epsilon$  result in a higher gap between train and test accuracy, meaning a higher degree of overfitting. This can be explained by the low number of labeled data points available to student training at low privacy budgets.

## 6 Results

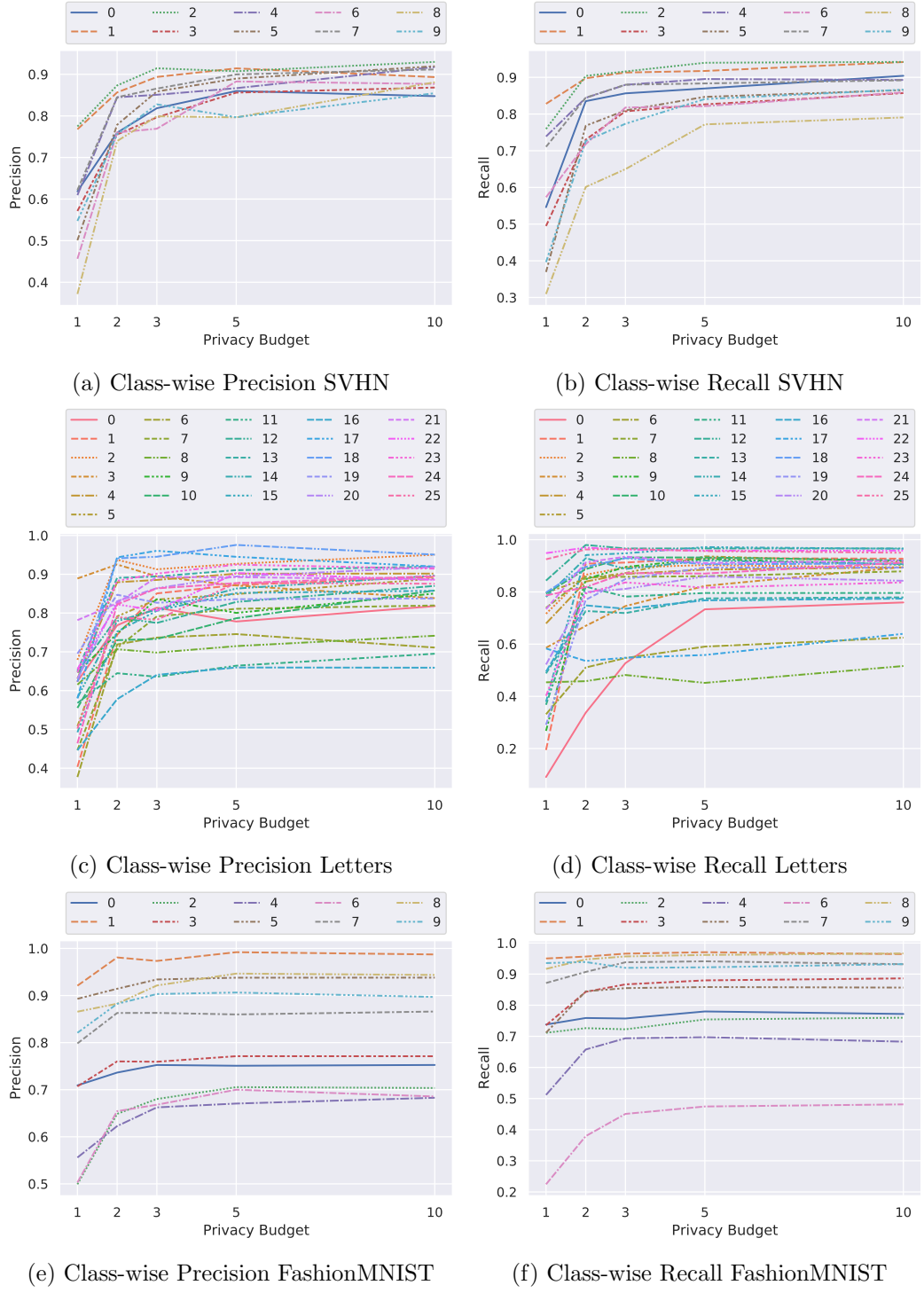


Figure 6.2: Class-Wise Performance of PATE Students

can be expected as the labels are the result of an ensemble [7][21].

For the Letters and FashionMNIST dataset, the label accuracy is significantly lower, with values ranging from 88% to slightly more than 91%. This most likely results from the lower teacher test accuracy and significantly limits the possibility of training students with high accuracies.

From Table 6.1 and Figure 6.1, it is clear that the low number of labeled data points reduces the student accuracy. Additionally, the 5-6% of mislabeled data potentially also limits the final accuracy of the student. Increasing the privacy budget from  $\epsilon = 1$  to  $\epsilon = 2$  increases the student accuracy drastically. This effect is not as strong for further increases of the privacy budget, although the students trained with budgets of  $\epsilon = 5$  and  $\epsilon = 10$  reach test accuracies comparably close to the baseline model. The accuracy for the student trained for SVHN with  $\epsilon = 5$  is only 4% lower than that created for [44] with semi-supervision and a comparable budget.

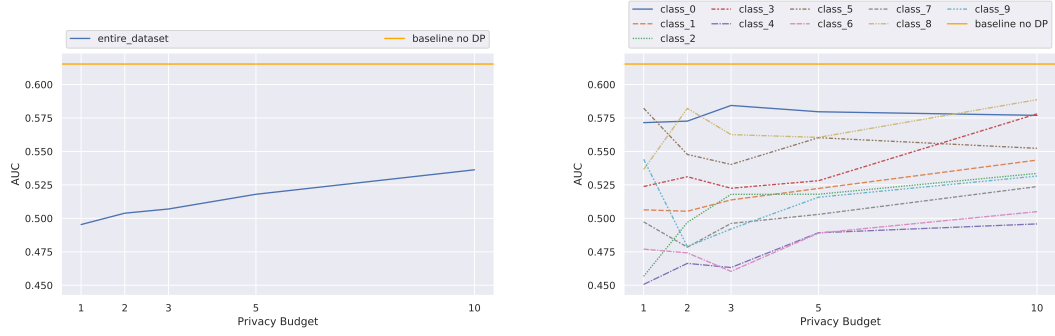
## 6.2 Membership Inference Attacks

Figure 6.3, Figure 6.4 and Figure 6.5 show the results of the attacks selected from TensorFlow Privacy when applied to the models trained with PATE on SVHN, Letters and FashionMNIST. They are displayed in relation to the privacy budgets chosen for the PATE training. The results for the non-private baseline models are given as a horizontal orange line and form a basis for comparison. The displayed Area Under Curve (AUC) aggregates the true-positive to the false-positive ratio of the attack results, where 0.5 is expected for a random guess. A higher AUC signifies a higher performance of the attacks.

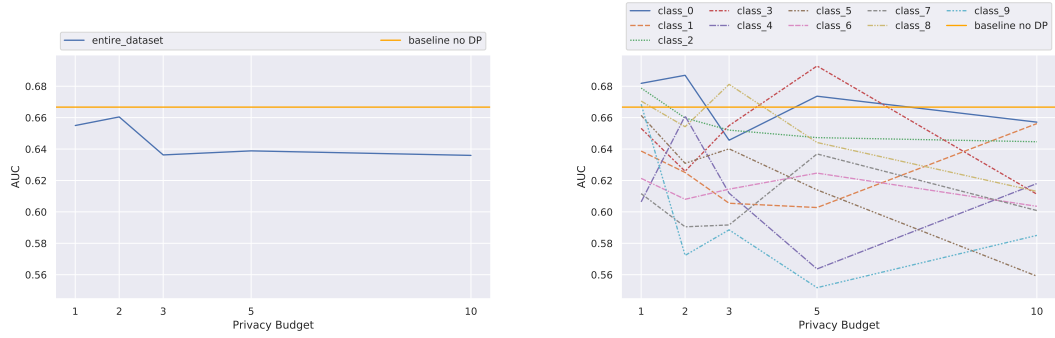
The results for SVHN clearly show a reduced vulnerability of PATE students for all privacy budgets to the selected membership inference attacks in comparison to the non-private baseline. They also indicate a slight increase in model vulnerability for the threshold-based attack for higher privacy budgets. In contrast to that, the MLP-based attack exhibits a higher performance on models created for privacy budgets of  $\epsilon = 1$  and  $\epsilon = 2$ . The latter attack type might benefit from the high degrees of overfitting that the PATE models exhibit, especially for the two lowest privacy budgets.

When considering the class-wise results for the models trained with PATE, it becomes apparent that the performance of the attacks is related to the class-wise performance of the PATE students displayed in Figure 6.2. For SVHN, classes 0 and 8 have low precision and recall for all tested budgets. When inspecting the class-wise results for the membership inference, it shows that these classes are among those

## 6 Results



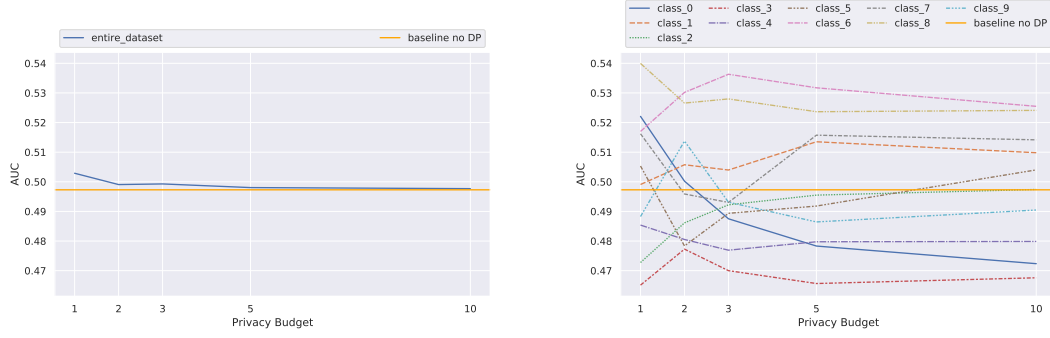
(a) Threshold-based Attack over the entire dataset (left) and class-wise(right).



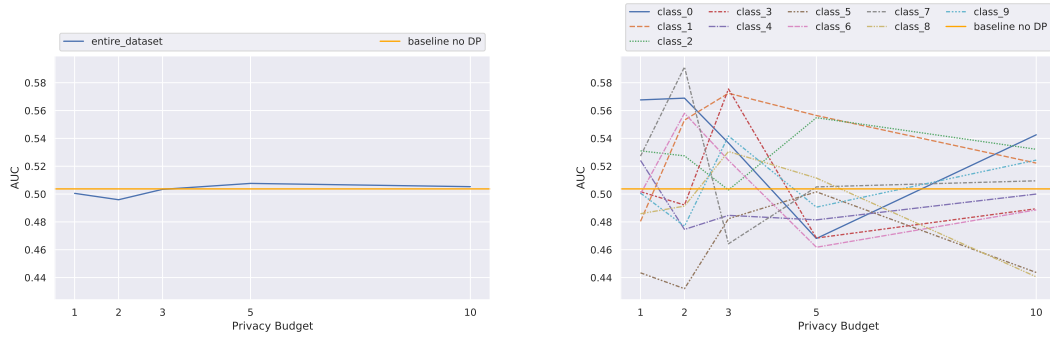
(b) MLP-based Attack over the entire dataset (left) and class-wise(right).

Figure 6.3: Membership Inference against PATE Students trained on SVHN. The baseline model was trained on the same dataset without the use of PATE.

## 6.2 Membership Inference Attacks



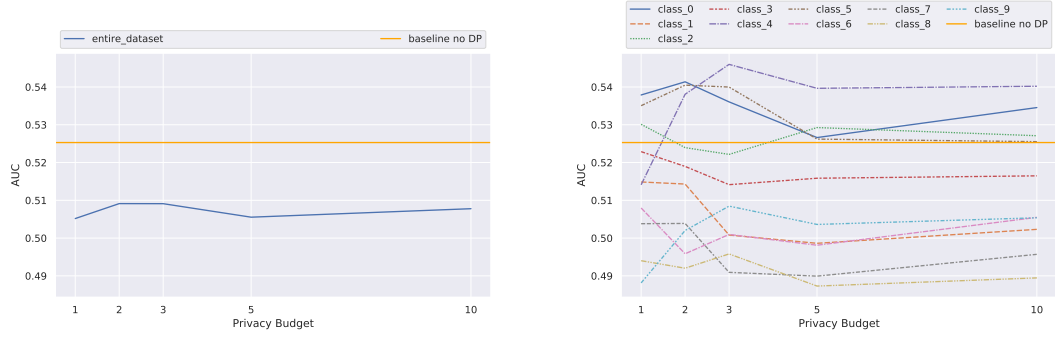
(a) Threshold-based Attack over the entire dataset (left) and class-wise(right).



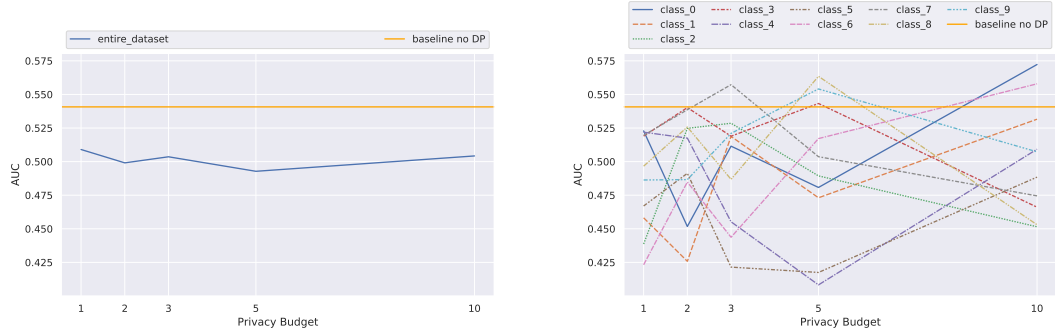
(b) MLP-based Attack over the entire dataset (left) and class-wise(right).

Figure 6.4: Membership Inference against PATE Students trained on Letters. The baseline model was trained on the same dataset without the use of PATE.

## 6 Results



(a) Threshold-based Attack over the entire dataset (left) and class-wise(right).



(b) MLP-based Attack over the entire dataset (left) and class-wise(right).

Figure 6.5: Membership Inference against PATE Students trained on FashionM-NIST. The baseline model was trained on the same dataset without the use of PATE.



on which both threshold- and MLP-based attacks are most successful. A possible explanation for this is a low generalization of the model for the affected classes, which leads to clearly distinct model behavior for datasets containing members and non-members. This low generalization might be affected by the low number of data points that describe these classes. A low number of data points per class does not seem the single factor, though, as class 9 has a similar number of data points and below-average class-wise precision and recall but does not seem affected.

Similar to models trained on SVHN, the results for FashionMNIST in Figure 6.5 show a reduced vulnerability to membership inference for PATE models trained for all different privacy budgets compared to the baseline trained without DP. The measured AUC for the attacks on models trained with PATE indicates a success rate close to a random guess, though also the performance of both threshold- and MLP-based attacks against the baseline models is much less accurate than in the case of SVHN. The accuracy of the attacks thereby does not increase noticeably for models trained with higher privacy budgets.

The results for the Letters dataset in Figure 6.4 show that the selected membership inference attacks are not successful, neither when performed on the baseline models nor when targeting the PATE models. For both threshold- and MLP-based attacks, AUC values around 0.5 were measured for all models, indicating that the attacks perform about as good as a random guess. One reason for this might be the simplicity of the Letters dataset that reduces the difference in the behavior of the model for members and non-members.

In Figure 6.6, the privacy risk score is displayed for the baseline and PATE models in the form of a Cumulative Distributive Function (CDF). For the SVHN and FashionMNIST datasets, the CDFs for all models trained with PATE show that the score for most data points is closer to 0.5 than for the compared baseline models. This effect becomes weaker for models trained with PATE and higher privacy budgets. While for SVHN, bigger portions of the dataset are affected, the results for FashionMNIST only show vulnerabilities for some outliers. The privacy risk score calculated for the Letters dataset and the PATE models trained on them does not significantly differ from those calculated for the baseline model trained without DP. This is consistent with the results for the two selected membership inference attacks.

## 6.3 Property Inference Attacks

The following sections give results for the experiments detailed in Section 5.3.

## 6 Results

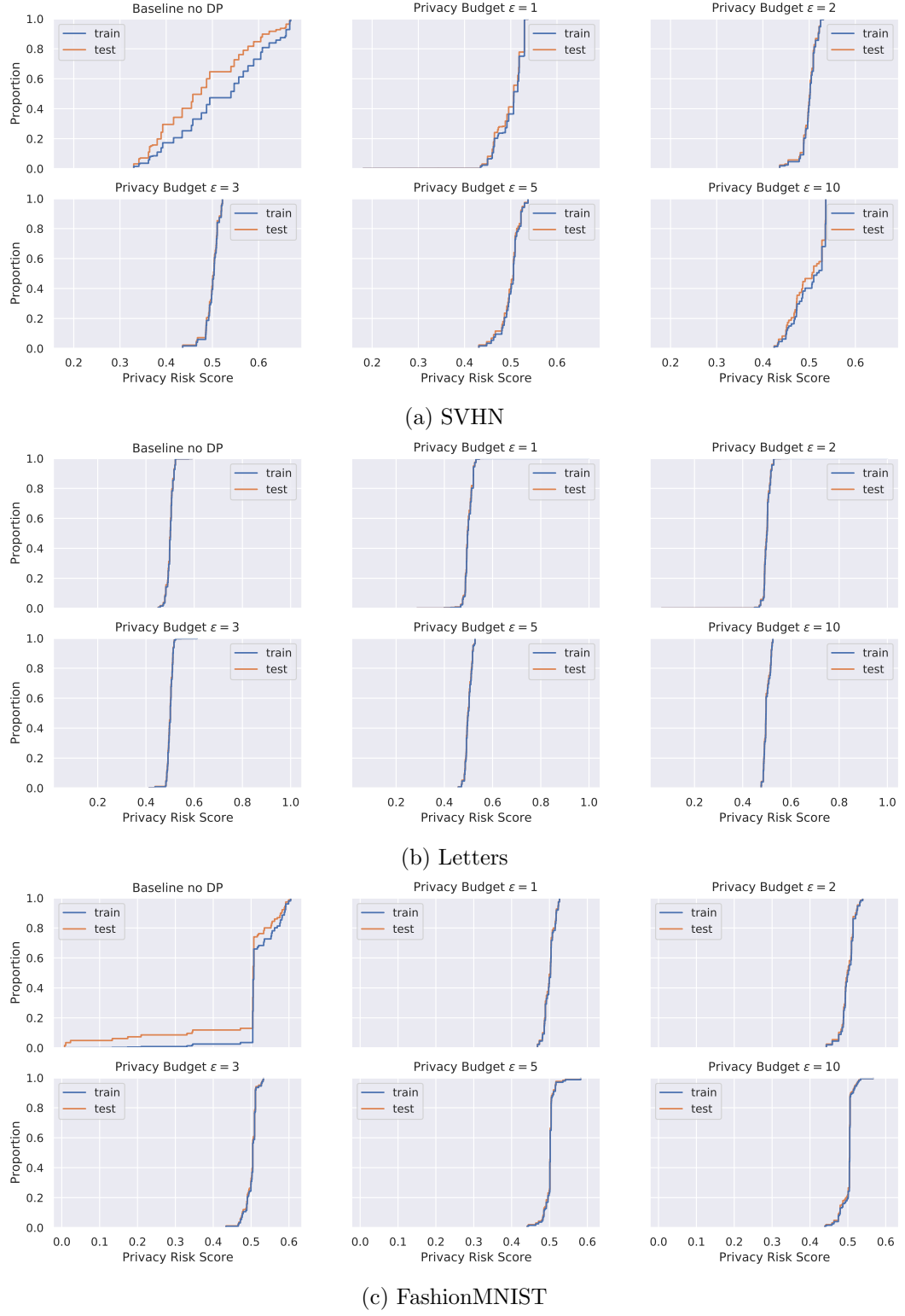


Figure 6.6: Privacy Risk Score from [54] calculated for PATE students.

### 6.3.1 Distance-based Property Inference Attack

This section discusses the results for the experiments described in Section 5.3.1 regarding the distance-based property inference attacks proposed in Section 4.4.1.

#### Attack using $D_{priv}$ -based Ensembles

In the experiment detailed in Section 5.3.1, PATE students are targeted with the distance-based property inference attack discussed in Section 4.4.1 for the properties of the form  $\mathcal{P}_x$ : *Class  $x$  of  $D_{priv}$  used to create the target model contains 50% images with enhanced contrast.*, specified for class 0 as  $\mathcal{P}_0$  and for class 1 as  $\mathcal{P}_1$ . Figure 6.7 and Figure 6.8 display the results of the conducted experiments. The attack is successful if it correctly infers the inspected property’s value for the target model, meaning the average KLD to the ensemble with the correct property value is lower than to the other. This was achieved for 19 out of the 20 targeted student models.

Note that the attack is successful for properties over both classes 0 and 1 of SVHN though class 1 contains double the number of data points compared to class 0. Teachers for which  $\mathcal{P}_1$  is true therefore have more modified and unmodified data points in their training dataset than those for which  $\mathcal{P}_0$  is true. The result might be that these teachers generalize well on both modified and unmodified data points. In such a situation, their behavior could become indistinguishable from the behavior of teachers that are trained on either only modified or unmodified data. That, in turn, would be reflected in the labels created by the Confident-GNMax mechanism and, finally, the student. Such a situation does not seem to occur, though, as the proposed method to property inference works both for properties  $\mathcal{P}_0$  and  $\mathcal{P}_1$ .

The results presented here have to be regarded with care, as the number of PATE executions was low and were taken over only one comparison between a student model and two teacher ensembles per privacy budget for the student model. Further experiments are necessary, for which a higher number of both student models and teacher ensembles is created, testing if the attack consistently produces similar results to the ones shown here.

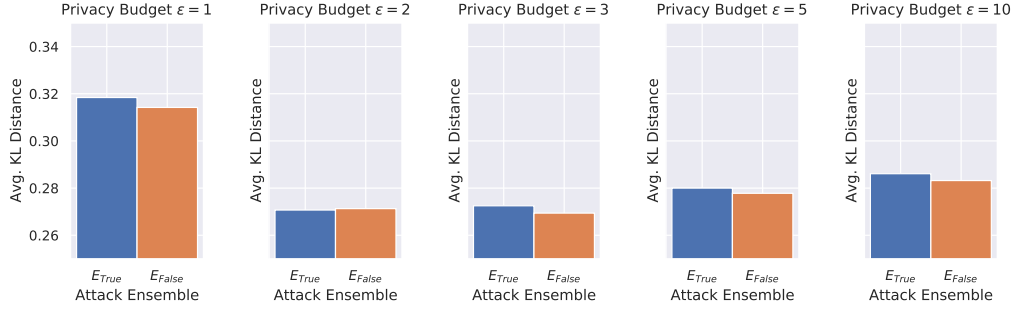
#### Attack using $D_{pub}$ -based Ensembles

Figure 6.9 and Figure 6.9 show the results for the experiment detailed in Section 5.3.1. They are based on the attack discussed in Section 4.4.1 that infers a property for a target model with the help of ensembles created on the basis of  $D_{pub}$ .

## 6 Results

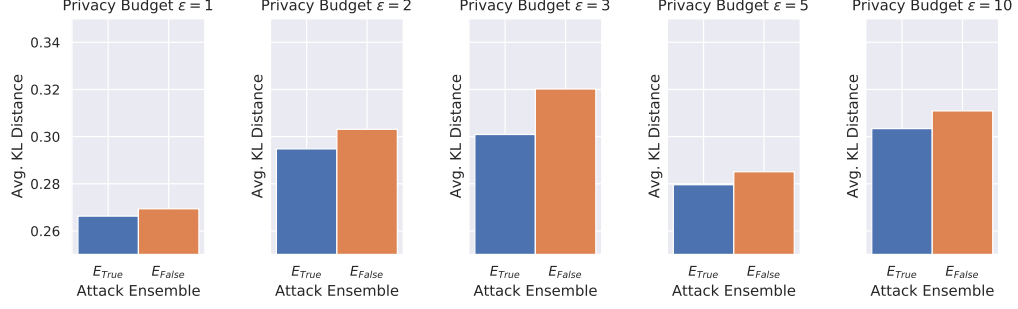


(a)  $\mathcal{P}_0$  is *True* for the targeted PATE student. The attack is successful if the average KLD from student to  $E_{True}$  is lower than to  $E_{False}$ .

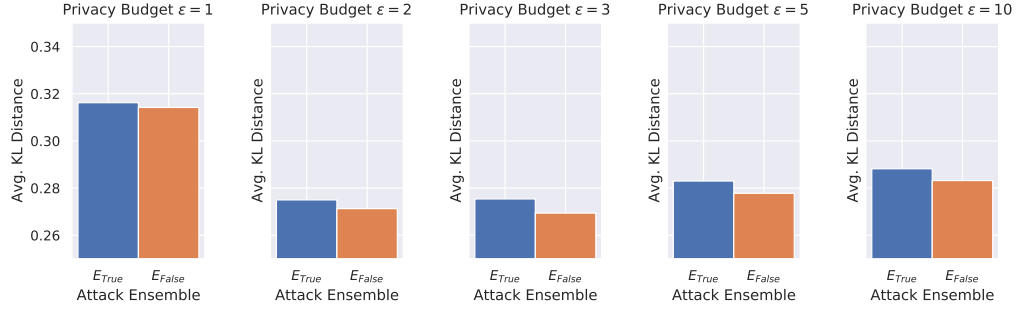


(b)  $\mathcal{P}_0$  is *False* for the targeted PATE student. The attack is successful if the average KLD from student to  $E_{False}$  is lower than to  $E_{True}$ .

Figure 6.7: Results for distance-based attacks with ensembles based on  $D_{priv}$  for the property  $\mathcal{P}_0$ : *Class 0 of  $D_{priv}$  used to create the target model contains 50% images with enhanced contrast.* This was done by calculating the average KL divergence between the confidences output by the student model and two teacher ensembles - one for which the property holds and one for which it does not.



- (a)  $\mathcal{P}_1$  is *True* for the targeted PATE student. The attack is successful if the average KLD from student to  $E_{True}$  is lower than to  $E_{False}$ .



- (b)  $\mathcal{P}_1$  is *False* for the targeted PATE student. The attack is successful if the average KLD from student to  $E_{False}$  is lower than to  $E_{True}$ .

Figure 6.8: Results for distance-based attacks with ensembles based on  $D_{priv}$  for the property  $\mathcal{P}_1$ : *Class 1 of  $D_{priv}$  used to create the target model contains 50% images with enhanced contrast*. This was done by calculating the average KL divergence between the confidences output by the student model and two teacher ensembles - one for which the property holds and one for which it does not.

## 6 Results

As for results shown in the previous section, the underlying attack is successful if it correctly predicts the inspected property’s value for the target model, meaning the average KLD to the ensemble with the correct property value is lower than to the other. This was achieved for 14 out of the 20 targeted student models.

Note that though switching the training dataset for the ensembles used for this attack from  $D_{priv}$  to  $D_{pub}$ , the attack still achieves an accuracy of 70%, which is better than a random guess (50%). Moreover, the attack accuracy might be increased further by creating ensembles on the basis of  $D_{pub}$  that emulate the PATE teachers used to train the targeted PATE student even better. When inspecting the metrics for the ensembles, it becomes apparent that the teachers used to create the PATE student reach an average test accuracy of 85.80% whereas the ensemble created on  $D_{pub}$  reaches a test accuracy of 90.49%. This difference stems from slightly larger partitions for the models trained on partitions of  $D_{pub}$ . Additionally, the ensembles used to conduct the attack consist of 16 models in contrast to 250 PATE teachers used to train the PATE student. The small ensemble size was chosen solely to reduce the computational cost, and increasing it might lead to better accuracy.

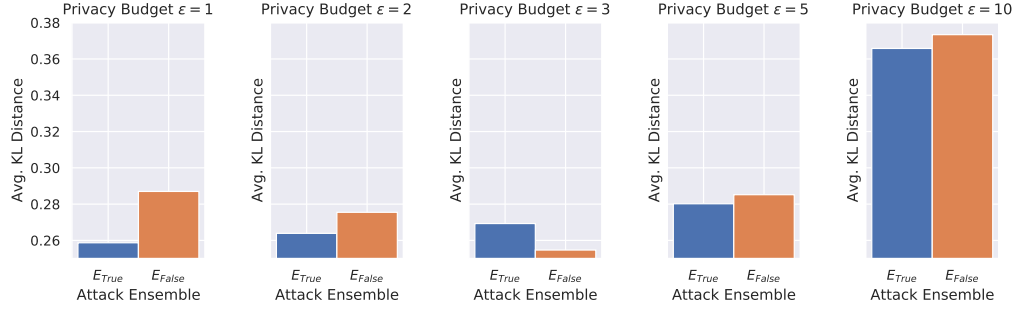
### Restoring Labeled Data Points $D_{lpub}$ from $D_{pub}$

Figure 6.11 shows the results for the selected membership inference attack when executed on the PATE student models and the datasets  $D_{lpubl}$  and  $D_{upub}$ . As expected, models with a low privacy budget are more vulnerable to the attack, which most likely stems from the high degree by which they are overfitted to the training data. For PATE students that are solely trained in a supervised fashion, these results mean that a lower privacy budget results in stronger privacy guarantees but potentially allows the attacker to restore the data points and labels a target is trained on with better accuracy.

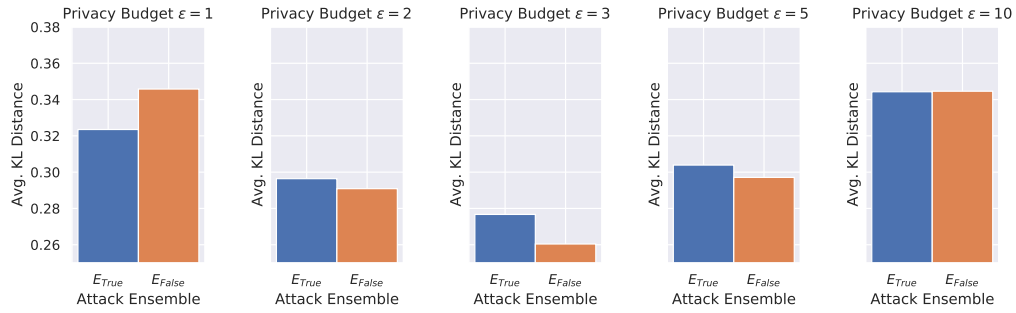
### Extended Attack using Multiple Properties

The results discussed in this section were produced by the experiment described in Section 5.3.1. The attack tries to infer to which percentage  $m$  the quality  $Q$ : *The image has enhanced contrast* applies to the data points from the dataset  $D_{priv}$  which was used in creating the targeted PATE student.

The sensitive data  $D_{priv}$  used for creating the target model contained 50% images with enhanced contrast. The results displayed in Figure 6.12 show, that the attack correctly infers  $\hat{m} = 50\%$ . For a student based on a training dataset with 40% high-contrast images, the attack should also show a similarly small distance to the teacher



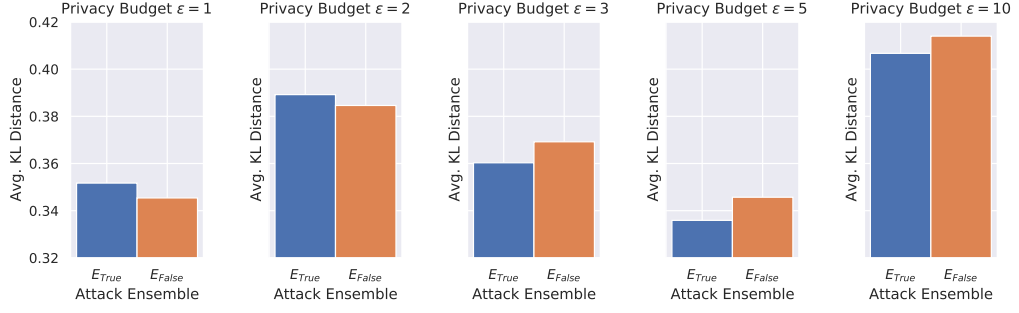
- (a)  $\mathcal{P}_0$  is *True* for the targeted PATE student. The attack is successful if the average KLD from student to  $E_{True}$  is lower than to  $E_{False}$ .



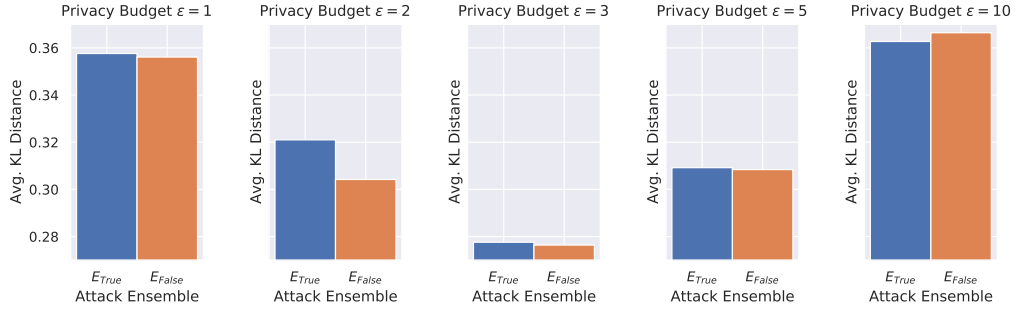
- (b)  $\mathcal{P}_0$  is *False* for the targeted PATE student. The attack is successful if the average KLD from student to  $E_{False}$  is lower than to  $E_{True}$ .

Figure 6.9: Results for distance-based attacks with ensembles based on  $D_{pub}$  for the property  $\mathcal{P}_0$ : *Class 0 of  $D_{priv}$  used to create the target model contains 50% images with enhanced contrast*. This was done by calculating the average KL divergence between the confidences output by the student model and two teacher ensembles - one for which the property holds and one for which it does not.

## 6 Results



(a)  $\mathcal{P}_1$  is *True* for the targeted PATE student. The attack is successful if the average KLD from student to  $E_{True}$  is lower than to  $E_{False}$ .



(b)  $\mathcal{P}_1$  is *False* for the targeted PATE student. The attack is successful if the average KLD from student to  $E_{False}$  is lower than to  $E_{True}$ .

Figure 6.10: Results for distance-based attacks with ensembles based on  $D_{pub}$  for the property  $\mathcal{P}_1$ : *Class 1 of  $D_{priv}$  used to create the target model contains 50% images with enhanced contrast*. This was done by calculating the average KL divergence between the confidences output by the student model and two teacher ensembles - one for which the property holds and one for which it does not.



Figure 6.11: Membership inference attack on the PATE students to restore  $D_{lpub}$ . The attack was executed over the whole of  $D_{pub}$  while data points from  $D_{lpub}$  were regarded as members.





Figure 6.12: Determining the percentage to which a property  $Q$ : *The image has enhanced contrast* applies to the data points of the sensitive data  $D_{priv}$  used to create a PATE student model. In the given results, the student was created with the help of a teacher ensemble that is based on a training dataset that contains 50% images with enhanced contrast. The attack was executed measuring the average KLD between the PATE student model and the three ensembles  $E_{25\%}$ ,  $E_{50\%}$  and  $E_{75\%}$ . It is successful if the average KL from the targeted PATE student to  $E_{50\%}$  is lower than to the other two ensembles.

ensemble with 50% high-contrast images in its training dataset and therefore assign that property magnitude to the student model. This would have to be shown in further experiments.

### 6.3.2 Deep Meta Classifier

The creation of the attack dataset takes most of the computational power needed for the property inference attack with a deep meta-classifier. Due to the huge number of features in each data point - 509,533 for the Letters dataset and the architecture from Table 5.1 - as well as the depths of the attack model, it is necessary to train a high number of models for the training split of the attack dataset.

First, a dataset with 2014 data points in its train split as well as 206 data points for the validation and test split was created. The property  $\mathcal{P}$  was thereby true for half of the data points of each split and false for the other half. Training the attack model on this dataset showed that it overfitted to the training data but was not able to generalize well with only reaching 52.17% test accuracy. Therefore, the number of data points in the training dataset was doubled to 4028. With this, it was possible

## 6 Results

to train a deep meta-classifier that exhibits a test accuracy of 60.87% on the test set. Even with the bigger size of the dataset, the model still strongly overfits its training data, and the model’s test accuracy fluctuates strongly between epochs. The latter most likely is caused by the extremely high-dimensional feature space of its input. Due to this, the optimizer modifies the model parameters to reach a local minimum in the loss function according to the training data, which contains too few samples to catch the complexity of the actual underlying distribution. A much bigger training dataset and batch size would be required to tackle the problem and increase the performance of the attack model.

## 7 Discussion

The following section discusses the proposed methods, conducted experiments and the gained results presented above.

### 7.1 Membership Inference and Model Performance

This section discusses, the results from training differentially private models with PATE and their evaluation through membership inference attacks.

The training of PATE students showed specific requirements with regards to the setting in which it takes place. Next to the need for non-sensitive data that can be made publicly available, a sufficiently large training dataset of sensitive data is required. The experiments presented in this work showed that PATE teacher performance is dependent on the latter, and small sizes of the private dataset  $D_{priv}$ , therefore, might result in low student utility. The privacy budget  $\epsilon$  strongly influences the utility of PATE students. Models trained with a very low privacy budget display a strong performance loss while increasing the budget above  $\epsilon = 3$  only grants a moderate increase in student utility.

The conducted experiments show that the privacy gained by using PATE to train models is less dependent on the privacy budget  $\epsilon$ . All membership inference attacks executed against the PATE students trained on SVHN and FashionMNIST showed decreased accuracy compared to those applied to the non-private baseline models<sup>1</sup>. Only for the threshold-based attack executed against PATE students trained on SVHN, increasing the privacy budget leads to a slight increase of accuracy of the membership inference attack (see Figure 6.3a).

Therefore, it can be generally assumed that the vulnerability of models to membership inference is reduced by training them with PATE, meaning a concrete increase of privacy for the sensitive data they are trained on. Note that, though the presented results show that increasing the privacy budget  $\epsilon$  has a positive influence

---

<sup>1</sup>For the Letters dataset, membership inference attacks on both differentially private models trained with PATE as well the non-private baseline models showed accuracies around 50%. The results for this dataset are therefore not relevant here.

on the utility for PATE students while not introducing relevant additional risk for membership inference, higher values for  $\epsilon$  exponentially weaken the given DP privacy guarantees.

## 7.2 Attacks on Distribution-level Privacy

Besides evaluating PATE with membership inference attacks, this thesis inspected its resilience to attacks that aim at extracting information with regards to properties over the whole sensitive dataset. PATE gives privacy guarantees in the context of DP, which protects the privacy of individuals but does not give any assurances regarding privacy for properties over a complete dataset. Nevertheless, this thesis examined the effect PATE has on the latter understanding of privacy, as leaking such properties potentially leads to a leak of sensitive information.

Though the privacy guarantees given by the framework make no statement regarding attacks against this understanding of privacy, it reduces the thread from many known attacks. This is achieved for one through the structure of the PATE framework and the setting in which models are trained and deployed. The entire class of model inversion attacks potentially becomes irrelevant as the public dataset  $D_{pub}$  is known to the adversary, and restoring data points or features does not result in an information gain. Moreover, attacks such as from [4] are not completely impossible, but the computational costs for creating a dataset for the training of an attack model becomes prohibitive.

The latter might be circumvented with the method to train a deep meta-classifier sketched here, which has the potential to be extended to a property inference attack against PATE. Both the experiments in [17] and this work show that it is possible to create the training dataset for such an attack model and train it to classify targets by a property of their underlying training dataset.

The distance-based property inference attack proposed here that uses the target’s output circumvents the problem of creating an expensive attack dataset. Moreover, experiments here show that the attack can be applied to PATE students to successfully infer properties over the sensitive data used to create the target model. Additionally, the proposed extended attack using multiple properties allows the extraction of more fine-grained information. These presented attack variants demonstrate that the information on properties is present in student models trained with PATE and can be extracted. Therefore, differentially private models trained with PATE bear the risk of revealing information on properties over the complete sensitive dataset.

## 7.3 Future Work

One of the limitations of the experiments conducted for this work is the supervised training of the PATE student models. The presented results need to be validated with the full PATE framework that uses semi-supervised techniques in its last step. Such validation would bring the additional benefit of allowing to inspect the contribution of different components of PATE to the overall privacy the framework grants.

Moreover, the variety of membership inference attacks used to evaluate the PATE framework here is limited due to the constrained scope of this work. It is necessary to expand this study to more attack types in the future as a higher number of membership inference attacks is known. Moreover, datasets used in the industry such as ImageNet [12] have higher complexity and number of classes than the datasets on which PATE models were trained here. Extending the experiments presented here to these datasets would lead to a more comprehensive understanding of the effects of PATE on membership inference.

Here, the modification of known property inference attacks to allow them to function against PATE students was only sketched and tested with a preliminary experiment. Future work is necessary to genuinely show that such a modified attack can be executed successfully, which first and foremost requires the creation of a substantially larger dataset to train the deep meta-classifier.

Finally, though the newly proposed distance-based property attack was shown to be effective against PATE students trained in the given setting, it is necessary to validate the results for other datasets and a higher number of models. Additionally, subsequent experiments need to reveal whether the attack allows successful property inference attacks against PATE students trained in a semi-supervised fashion. Such future work could finally demonstrate if differentially private models trained with the full PATE framework are susceptible to property inference.



## 8 Conclusion

This thesis evaluated the privacy guarantees granted to models trained with PATE. Though the framework allows giving guarantees regarding the privacy of models by implementing a mechanism for DP, their concrete meaning is not well understood. This work thus evaluated the privacy of differentially private models trained with PATE for a variety of privacy budgets and datasets through executing concrete attacks against them. The selected attacks thereby target privacy at the level of individuals as well as at the level of the complete dataset and its distribution.

Training models on multiple datasets showed that using PATE comes with several challenges. Among these are intense computational costs and the need for an extensive private dataset. Further, it was demonstrated that the performance of produced models strongly depends on the chosen privacy budget, with low values reducing model utility severely. Nevertheless, the use of higher privacy budgets allows training models that show only a moderately reduced performance in comparison to the non-private baseline models.

Further, selected known membership inference attacks executed against PATE students displayed a reduced accuracy in comparison to the attack results for non-private baseline models. This concrete gain in protection against membership inference thereby showed to be mostly independent of the privacy budget. A careful selection of the privacy budget when training differentially private models with PATE is nevertheless necessary, as higher privacy budgets result in models with increased performance but quickly lead to meaningless privacy guarantees.

Next, the privacy granted by PATE was inspected at the distribution level. A discussion of model inversion in the context of the specific setting required by PATE revealed that it is questionable whether this class of attacks poses a relevant threat to models trained with the framework. It was shown that known property inference attacks are not directly applicable to models trained with PATE due to the structure of the framework. Hence, this thesis proposed a new distance-based property inference attack that was successfully applied to PATE student models. The attack infers properties from the output of a targeted PATE student model, making it feasible in the specific setting for which PATE models are trained. Multiple variations of the method were proposed to reduce the necessary adversarial knowledge and increase the granularity to which information on sensitive data can be extracted.

## 8 Conclusion

Additionally, a sketch for a modified known property inference attack was shown that potentially allows attacking PATE students by using a deep meta-classifier.

To summarize, by attacking differentially private models trained by PATE with both membership and property inference attacks, the concrete privacy that the framework grants was evaluated in its different facets. This thesis showed that the structure of the PATE framework overall exposes a small surface to potential attacks. Regarding distribution-level privacy, a new approach to property inference was demonstrated in addition to a sketch to modify an existing attack to work against PATE models. As a mechanism for DP, PATE was not build to prevent such information leaks, and this work shows that it is possible to extract distribution-level privacy from models trained with a version of the PATE framework that uses supervised training for student models. Nevertheless, it was demonstrated that the PATE framework grants efficient protection against membership inference attacks which protects privacy on an individual level, while produced models show only a moderately reduced utility.



# Bibliography

- [1] Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. “Deep Learning with Differential Privacy”. In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. CCS ’16. New York, NY, USA: Association for Computing Machinery, Oct. 2016, pp. 308–318. ISBN: 978-1-4503-4139-4. DOI: 10.1145/2976749.2978318.
- [2] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mane, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viegas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. “TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems”. en. In: *arXiv:1603.04467 [cs]* (Mar. 2016). arXiv: 1603.04467 [cs].
- [3] Ulrich Aïvodji, Sébastien Gambs, and Timon Ther. “GAMIN: An Adversarial Approach to Black-Box Model Inversion”. en. In: *arXiv:1909.11835 [cs, stat]* (Sept. 2019). arXiv: 1909.11835 [cs, stat].
- [4] Giuseppe Ateniese, Giovanni Felici, Luigi V. Mancini, Angelo Spognardi, Antonio Villani, and Domenico Vitali. “Hacking Smart Machines with Smarter Ones: How to Extract Meaningful Data from Machine Learning Classifiers”. In: *arXiv:1306.4447 [cs, stat]* (June 2013). arXiv: 1306.4447 [cs, stat].
- [5] Christopher Bishop. *Pattern Recognition and Machine Learning*. en. Information Science and Statistics. New York: Springer-Verlag, 2006. ISBN: 978-0-387-31073-2.
- [6] Oussama Bouanani. “The Influence of Training Parameters and Architectural Choices on the Vulnerability of Neural Networks to Membership Inference Attacks”. en. In: (), p. 101.
- [7] L. Breiman. “Bagging Predictors”. In: *Machine Learning* (1996). DOI: 10.1007/BF00058655.

- [8] Zhiqi Bu, Jinshuo Dong, Qi Long, and Weijie J. Su. “Deep Learning with Gaussian Differential Privacy”. eng. In: *Harvard Data Science Review* 2020.23 (2020). ISSN: 2644-2353. DOI: 10.1162/99608f92.cfc5dd25.
- [9] Mark Bun and T. Steinke. “Concentrated Differential Privacy: Simplifications, Extensions, and Lower Bounds”. In: *TCC* (2016). DOI: 10.1007/978-3-662-53641-4\_24.
- [10] Si Chen, Ruoxi Jia, and Guo-Jun Qi. “Improved Techniques for Model Inversion Attacks”. en. In: *arXiv:2010.04092 [cs]* (Oct. 2020). arXiv: 2010.04092 [cs].
- [11] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and André van Schaik. “EM-NIST: An Extension of MNIST to Handwritten Letters”. In: *arXiv:1702.05373 [cs]* (Mar. 2017). arXiv: 1702.05373 [cs].
- [12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. “ImageNet: A Large-Scale Hierarchical Image Database”. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. June 2009, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848.
- [13] Jinshuo Dong, Aaron Roth, and Weijie J. Su. “Gaussian Differential Privacy”. In: *arXiv:1905.02383 [cs, stat]* (May 2019). arXiv: 1905.02383 [cs, stat].
- [14] C. Dwork and G. Rothblum. “Concentrated Differential Privacy”. en. In: *undefined* (2016).
- [15] Cynthia Dwork. “Differential Privacy”. en. In: *Automata, Languages and Programming*. Ed. by Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2006, pp. 1–12. ISBN: 978-3-540-35908-1. DOI: 10.1007/11787006\_1.
- [16] Cynthia Dwork and Aaron Roth. “The Algorithmic Foundations of Differential Privacy”. en. In: *Foundations and Trends® in Theoretical Computer Science* 9.3-4 (2013), pp. 211–407. ISSN: 1551-305X, 1551-3068. DOI: 10.1561/04000000042.
- [17] Gabriel Eilertsen, Daniel Jönsson, Timo Ropinski, Jonas Unger, and Anders Ynnerman. “Classifying the Classifier: Dissecting the Weight Space of Neural Networks”. en. In: *arXiv:2002.05688 [cs]* (Feb. 2020). arXiv: 2002.05688 [cs].
- [18] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. “Model Inversion Attacks That Exploit Confidence Information and Basic Countermeasures”. In: *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. CCS ’15. New York, NY, USA: Association for Computing Machinery, Oct. 2015, pp. 1322–1333. ISBN: 978-1-4503-3832-5. DOI: 10.1145/2810103.2813677.

- [19] Matthew 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. UNIX Security Symposium 2014* (Aug. 2014), pp. 17–32.
- [20] Karan Ganju, Qi Wang, Wei Yang, Carl A. Gunter, and Nikita Borisov. “Property Inference Attacks on Fully Connected Neural Networks Using Permutation Invariant Representations”. In: *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security. CCS ’18*. New York, NY, USA: Association for Computing Machinery, Oct. 2018, pp. 619–633. ISBN: 978-1-4503-5693-0. DOI: 10.1145/3243734.3243834.
- [21] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. Adaptive Computation and Machine Learning. Cambridge, Massachusetts: The MIT Press, 2016. ISBN: 978-0-262-03561-3.
- [22] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. “Generative Adversarial Nets”. en. In: (), p. 9.
- [23] J. Hayes, Luca Melis, G. Danezis, and Emiliano De Cristofaro. “LOGAN: Evaluating Privacy Leakage of Generative Models Using Generative Adversarial Networks”. In: *ArXiv* (2017).
- [24] Yang He, Shadi Rahimian, Bernt Schiele, and Mario Fritz. “Segmentations-Leak: Membership Inference Attacks and Defenses in Semantic Image Segmentation”. en. In: *Computer Vision – ECCV 2020*. Ed. by Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm. Vol. 12368. Cham: Springer International Publishing, 2020, pp. 519–535. ISBN: 978-3-030-58591-4 978-3-030-58592-1. DOI: 10.1007/978-3-030-58592-1\_31.
- [25] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. “Distilling the Knowledge in a Neural Network”. en. In: *arXiv:1503.02531 [cs, stat]* (Mar. 2015). arXiv: 1503.02531 [cs, stat].
- [26] B. Hitaj, G. Ateniese, and F. Pérez-Cruz. “Deep Models Under the GAN: Information Leakage from Collaborative Deep Learning”. In: *CCS* (2017). DOI: 10.1145/3133956.3134012.
- [27] J. Hsu, M. Gaboardi, A. Haeberlen, S. Khanna, A. Narayan, B. C. Pierce, and A. Roth. “Differential Privacy: An Economic Method for Choosing Epsilon”. In: *2014 IEEE 27th Computer Security Foundations Symposium*. July 2014, pp. 398–410. DOI: 10.1109/CSF.2014.35.
- [28] Bargav Jayaraman and David Evans. “Evaluating Differentially Private Machine Learning in Practice”. en. In: *28th {USENIX Security Symposium ({USENIX Security 19)}*. 2019, pp. 1895–1912. ISBN: 978-1-939133-06-9.
- [29] Zhanglong Ji, Zachary C. Lipton, and Charles Elkan. “Differential Privacy and Machine Learning: A Survey and Review”. In: *arXiv:1412.7584 [cs]* (Dec. 2014). arXiv: 1412.7584 [cs].

- [30] James Jordon, Jinsung Yoon, and Mihaela van der Schaar. “PATE-GAN: Generating Synthetic Data with Differential Privacy Guarantees”. en. In: *International Conference on Learning Representations*. Sept. 2018.
- [31] Alex Krizhevsky. “Learning Multiple Layers of Features from Tiny Images”. en. In: (), p. 60.
- [32] Yann Lecun. “Gradient-Based Learning Applied to Document Recognition”. en. In: *PROCEEDINGS OF THE IEEE* 86.11 (1998), p. 47.
- [33] Jaewoo Lee and Chris Clifton. “How Much Is Enough? Choosing  $\epsilon$  for Differential Privacy”. en. In: *Information Security*. Ed. by Xuejia Lai, Jianying Zhou, and Hui Li. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2011, pp. 325–340. ISBN: 978-3-642-24861-0. DOI: 10.1007/978-3-642-24861-0\_22.
- [34] Ninghui Li, Wahbeh Qardaji, Dong Su, Yi Wu, and Weining Yang. “Membership Privacy: A Unifying Framework for Privacy Definitions”. en. In: *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security - CCS '13*. Berlin, Germany: ACM Press, 2013, pp. 889–900. ISBN: 978-1-4503-2477-9. DOI: 10.1145/2508859.2516686.
- [35] Yunhui Long, Vincent Bindschaedler, and Carl A. Gunter. “Towards Measuring Membership Privacy”. In: *ArXiv* (2017).
- [36] Yunhui Long, Vincent Bindschaedler, Lei Wang, Diyue Bu, Xiaofeng Wang, Haixu Tang, Carl A. Gunter, and Kai Chen. “Understanding Membership Inferences on Well-Generalized Learning Models”. en. In: *arXiv:1802.04889 [cs, stat]* (Feb. 2018). arXiv: 1802.04889 [cs, stat].
- [37] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov. “Exploiting Unintended Feature Leakage in Collaborative Learning”. In: *2019 IEEE Symposium on Security and Privacy (SP)*. May 2019, pp. 691–706. DOI: 10.1109/SP.2019.00029.
- [38] Ilya Mironov. “Renyi Differential Privacy”. In: *2017 IEEE 30th Computer Security Foundations Symposium (CSF)* (Aug. 2017), pp. 263–275. DOI: 10.1109/CSF.2017.11. arXiv: 1702.07476.
- [39] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. “Virtual Adversarial Training: A Regularization Method for Supervised and Semi-Supervised Learning”. In: *arXiv:1704.03976 [cs, stat]* (June 2018). arXiv: 1704.03976 [cs, stat].
- [40] M. Nasr, R. Shokri, and A. Houmansadr. “Comprehensive Privacy Analysis of Deep Learning: Passive and Active White-Box Inference Attacks against Centralized and Federated Learning”. In: *2019 IEEE Symposium on Security and Privacy (SP)*. May 2019, pp. 739–753. DOI: 10.1109/SP.2019.00065.
- [41] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. “Reading Digits in Natural Images with Unsupervised Feature Learning”. en. In: (), p. 9.

- [42] N. Papernot, P. McDaniel, A. Sinha, and M. P. Wellman. “SoK: Security and Privacy in Machine Learning”. In: *2018 IEEE European Symposium on Security and Privacy (EuroS P)*. Apr. 2018, pp. 399–414. DOI: 10.1109/EuroSP.2018.00035.
- [43] Nicolas Papernot, Martín Abadi, Úlfar Erlingsson, Ian Goodfellow, and Kunal Talwar. “Semi-Supervised Knowledge Transfer for Deep Learning from Private Training Data”. In: *arXiv:1610.05755 [cs, stat]* (Mar. 2017). arXiv: 1610.05755 [cs, stat].
- [44] Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Úlfar Erlingsson. “Scalable Private Learning with PATE”. In: *arXiv:1802.08908 [cs, stat]* (Feb. 2018). arXiv: 1802.08908 [cs, stat].
- [45] W. Nicholson Price and I. Glenn Cohen. “Privacy in the Age of Medical Big Data”. en. In: *Nature Medicine* 25.1 (Jan. 2019), pp. 37–43. ISSN: 1078-8956, 1546-170X. DOI: 10.1038/s41591-018-0272-7.
- [46] Apostolos Pyrgelis, C. Troncoso, and Emiliano De Cristofaro. “Knock Knock, Who’s There? Membership Inference on Aggregate Location Data”. In: *NDSS* (2018). DOI: 10.14722/NDSS.2018.23183.
- [47] Atiqur Rahman, Tanzila Rahman, Robert Laganieri, Noman Mohammed, and Yang Wang. “Membership Inference Attack against Differentially Private Deep Learning Model”. en. In: *Transactions on Data Privacy* 11 (2018), p. 19.
- [48] Alfréd Rényi. “On Measures of Entropy and Information”. EN. In: *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*. The Regents of the University of California, 1961.
- [49] Alexandre Sablayrolles, Matthijs Douze, Yann Ollivier, Cordelia Schmid, and Hervé Jégou. “White-Box vs Black-Box: Bayes Optimal Strategies for Membership Inference”. en. In: *arXiv:1908.11229 [cs, stat]* (Aug. 2019). arXiv: 1908.11229 [cs, stat].
- [50] A. Salem, Y. Zhang, M. Humbert, M. Fritz, and M. Backes. “ML-Leaks: Model and Data Independent Membership Inference Attacks and Defenses on Machine Learning Models”. In: *NDSS* (2019). DOI: 10.14722/NDSS.2019.23119.
- [51] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. “Improved Techniques for Training GANs”. en. In: *arXiv:1606.03498 [cs]* (June 2016). arXiv: 1606.03498 [cs].
- [52] Avital Shafran, Shmuel Peleg, and Yedid Hoshen. “Reconstruction-Based Membership Inference Attacks Are Easier on Difficult Problems”. en. In: *arXiv:2102.07762 [cs]* (Feb. 2021). arXiv: 2102.07762 [cs].
- [53] R. Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. “Membership Inference Attacks Against Machine Learning Models”. In: *2017 IEEE Symposium on Security and Privacy (SP)* (2017). DOI: 10.1109/SP.2017.41.

- [54] Liwei Song and Prateek Mittal. “Systematic Evaluation of Privacy Risks of Machine Learning Models”. en. In: *arXiv:2003.10595 [cs, stat]* (Dec. 2020). arXiv: 2003.10595 [cs, stat].
- [55] Shuang Song, K. Chaudhuri, and A. Sarwate. “Stochastic Gradient Descent with Differentially Private Updates”. In: *2013 IEEE Global Conference on Signal and Information Processing* (2013). DOI: 10.1109/GlobalSIP.2013.6736861.
- [56] Shakila Mahjabin Tonni, Dinusha Vatsalan, Farhad Farokhi, Dali Kaafar, Zhi-gang Lu, and Gioacchino Tangari. “Data and Model Dependencies of Membership Inference Attack”. en. In: *arXiv:2002.06856 [cs, stat]* (July 2020). arXiv: 2002.06856 [cs, stat].
- [57] Reihaneh Torkzadehmahani, Peter Kairouz, and Benedict Paten. “DP-CGAN: Differentially Private Synthetic Data and Label Generation”. en. In: *arXiv:2001.09700 [cs, stat]* (Jan. 2020). arXiv: 2001.09700 [cs, stat].
- [58] Stacey Truex, Ling Liu, Mehmet Emre Gursoy, Wenqi Wei, and Lei Yu. “Effects of Differential Privacy and Data Skewness on Membership Inference Vulnerability”. en. In: *arXiv:1911.09777 [cs, stat]* (Nov. 2019). arXiv: 1911.09777 [cs, stat].
- [59] Stacey Truex, Ling Liu, Mehmet Emre Gursoy, Lei Yu, and Wenqi Wei. “Towards Demystifying Membership Inference Attacks”. en. In: *arXiv:1807.09173 [cs]* (Feb. 2019). arXiv: 1807.09173 [cs].
- [60] Zhibo Wang, Mengkai Song, Zhifei Zhang, Yang Song, Qian Wang, and Hairong Qi. “Beyond Inferring Class Representatives: User-Level Privacy Leakage From Federated Learning”. en. In: *arXiv:1812.00535 [cs]* (Dec. 2018). arXiv: 1812.00535 [cs].
- [61] Kang Wei, Jun Li, Ming Ding, Chuan Ma, Howard H. Yang, Farokhi Farhad, Shi Jin, Tony Q. S. Quek, and H. Poor. “Federated Learning With Differential Privacy: Algorithms and Performance Analysis”. In: *IEEE Transactions on Information Forensics and Security* (2020). DOI: 10.1109/TIFS.2020.2988575.
- [62] Han Xiao, Kashif Rasul, and Roland Vollgraf. “Fashion-MNIST: A Novel Image Dataset for Benchmarking Machine Learning Algorithms”. In: *arXiv:1708.07747 [cs, stat]* (Sept. 2017). arXiv: 1708.07747 [cs, stat].
- [63] Mohammad Yaghini, Bogdan Kulynych, Giovanni Cherubin, and Carmela Troncoso. “Disparate Vulnerability: On the Unfairness of Privacy Attacks Against Machine Learning”. en. In: *arXiv:1906.00389 [cs, stat]* (July 2020). arXiv: 1906.00389 [cs, stat].
- [64] S. Yeom, I. Giacomelli, M. Fredrikson, and S. Jha. “Privacy Risk in Machine Learning: Analyzing the Connection to Overfitting”. In: *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*. July 2018, pp. 268–282. DOI: 10.1109/CSF.2018.00027.

- [65] Lei Yu, Ling Liu, Calton Pu, Mehmet Emre Gursoy, and Stacey Truex. “Differentially Private Model Publishing for Deep Learning”. en. In: *2019 IEEE Symposium on Security and Privacy (SP)* (May 2019), pp. 332–349. DOI: 10.1109/SP.2019.00019. arXiv: 1904.02200.
- [66] Wanrong Zhang, Olga Ohrimenko, and Rachel Cummings. “Attribute Privacy: Framework and Mechanisms”. en. In: *arXiv:2009.04013 [cs, stat]* (Sept. 2020). arXiv: 2009.04013 [cs, stat].
- [67] Y. Zhang, R. Jia, Hengzhi Pei, Wenxiao Wang, B. Li, and D. Song. “The Secret Revealer: Generative Model-Inversion Attacks Against Deep Neural Networks”. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020). DOI: 10.1109/cvpr42600.2020.00033.
- [68] Yang Zou, Zhikun Zhang, Michael Backes, and Yang Zhang. “Privacy Analysis of Deep Learning in the Wild: Membership Inference Attacks against Transfer Learning”. en. In: *arXiv:2009.04872 [cs, stat]* (Sept. 2020). arXiv: 2009.04872 [cs, stat].