Bachelorthesis at Fraunhofer AISEC

Department of Secure Systems Engineering (SSE)

Application and Evaluation of Differential Privacy in Health Data Classification Tasks

Maika Krueger

Matrikelnummer: 5199436

maika.krueger@fu-berlin.de

Betreuerin: Franziska Boenisch, Erstgutachter: Prof. Dr. Marian Margraf

Zweitgutachter: Prof. Dr. Jörn Eichler

Berlin, 29.11.2020

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 November 30, 2020

M. Krüger

Maika Krüger

Abstract

Data driven medical research enables promising novel methods of clinical decision support and personalised medicine. For example machine learning (ML) models are developed to detect unknown relationships between biomedical parameters to predict patients' diagnosis.

However, these benefits come with concerns about patients' privacy. Privacy protection is considered to be an important problem in ML, especially in the health care sector, where research is based on sensitive information such as medical histories, genomic data or clinical records. A model trained on sensitive data may store this sensitive information during the training process. Analysing the models parameters or output can reveal this sensitive information. Differential privacy (DP) is a strong mathematical framework that provides privacy guarantees in learning-based applications. Based on DP applications in several other areas its use has been proposed to protect an individual's privacy in ML contexts.

The Private Aggregation of Teacher Ensembles (PATE) is a state-of-the-art framework for private ML that is based on DP.

This thesis presents the evaluation of PATE when it is applied to medical classification task using a small medical data set. Therefore, the PATE model was compared to a non private baseline model. To evaluate the ML algorithms used in the PATE framework, two logistic regression (LR) classifiers, a support vector machine (SVM) and two neural networks (NN) were compared using cross validation, confusion matrix, F1 – Score, receiver operating characteristic curve (ROC), area under the ROC curve (AUC) and accuracy. Subsequently, the number of teachers, the noise injection and the accuracy of the PATE model trained on the medical data were evaluated.

Even with a small number of teachers the accuracy of the PATE model was 0.76. Compared to the baseline model (0.83), there is a low loss of accuracy, but still the guarantee of a strong privacy of $\epsilon = 2.16$. Moreover, with a small number of teachers, the agreement of the teachers about a predicted outcome was higher. Nevertheless, there were also limitations due to the small data set. Small training data sets resulted in overfitting. In addition, the model was less robust against noise injection due to the small number of teachers.

However, this thesis gives an introduction to the application of PATE on a health binary classification task using a small medical data set, thus providing an initial understanding of the application of PATE in in health care.

Contents

List of Figures

List of Tables

1 Introduction

In recent years, ML systems have become ubiquitous in society. They are emerging in many fields such as computer vision, nature language processing, and speech processing. On the basis of these techniques, infrastructures such as disease prediction systems based on healthcare data [20] achieved great success.[56] These infrastructures enable predictive, personalized medicine [34] and increase the need for additional data in health care [47]. For example, sharing medical data between hospitals to generate more data can benefit many aspects of medical research. However, these benefits come with concerns about patients' privacy. ML models are trained on medical data usually contains sensitive information about the patient such as clinical records, medical histories or genomic data.[47] In 2018, the General Data Protection Regulation (GDPR) of the European Union came into effect [2]. According to GDPR genomic and health related data are sensitive data [33].

It was discovered that by releasing ML models trained on sensitive data sets attackers can recover private information from model parameters [29], [51], [37], [36], [22], [55], [52], [42]. Since the model does implicitly memorize details about the distribution of the training data set [49], [62] the model can be highly impacted by a single record [18]. For example overfitting [59] or memorizing during the training process of NN [15] are potential weaknesses. Moreover, analysing the output of a model allows attackers to reconstruct the input data [29], [30].

Research into protecting privacy while releasing sensitive results of statistical analysis or ML has proposed many protections such as k–anonymity [37], l–diversity [37], and t–closeness [36]. However, these techniques cannot prevent attacks in which attackers already have information about the data set [56].

A promising approach to this problem is DP [22]. DP provides a strong privacy guarantee by injecting noise to the statistical results computed from the private data set [22].

DP has been widely adopted for privacy protection in ML [17], [19], [16], [58], [48], [63], [9], [43], [10], [35]. Some research focuses on developing a DP version of an existing ML algorithm such as DP–LR [17] or DP principal components analysis (PCA), [19] and some of them apply DP to release data in a privacy preserving manner [16], [58], [48]. Others developed general frameworks. Examples for these are DP stochastic gradient descent (DP–SGD) [9], which can be applied to algorithms optimized with stochastic gradient descent (SGD), and private aggregation teacher ensemble (PATE) [43]. PATE is based on a teacher–student framework [43] and outperformed existing approaches [56].

In the health care sector, research about DP–ML is still at the beginning. Researchers at the University hospital Berlin Charité developed an open source data anonymization tool called ARX which includes DP mechanisms and is used in a variety of contexts such as clinical trial data sharing [1]. Eicher et al. 2020 extended this tool with additional DP–ML features [28]. To predict drug sensitivity based on genomic data in clinical trials, Honkela et al. 2018 developed a DP Bayesian linear regression model [33]. The researcher of [30] provided a case study about the application of DP linear regression and DP histogramms in clinical trials. Not focusing on ML but on statistical tests, the application of DP to genome–wide association studies (GWAS) [3] is an other area of the application of DP in health care [54], [61], [50].

In this work, the original PATE framework is adopted to a medical classification problem using a small medical data set. The motivation of this thesis is to focus on the protection of the privacy of the individual whose data was used to train a ML model. The aim of this work is to give an introduction on PATE regarding its capability to be used in the medical context. Accordingly, this thesis gives a evaluation of the limitations and problems of PATE on the limited data problem in the health sector. To evaluate the performance of PATE, accuracy and privacy loss are the matrices.

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

- Application of PATE on a binary classification task using a small medical data set
- Evaluation of PATE by evaluation of the number of teachers, noise injection and comparison to a non private baseline model
- Discovery of limitations when applying PATE on a small data set
- Development of a PATE model with high accuracy and strong privacy

This thesis is organized as follows: The first section reviews the background of DP. The second section explains PATE in more detail and gives a short theoretical background on the privacy loss of PATE. The third section describes the methods used in the experiment followed by the experimental results reported in the next section. After that, the results are discussed. Lastly, the conclusion of this work and an outlook on future work is given.

2 Preliminary

The following chapter introduces the theoretical background on DP used in this thesis. The fist section deals with the mathematical background on DP starting with the original definition of DP, moving on to a more relaxed definition which is (ϵ, δ)–DP. Lastly, noise injection using the Laplace mechanism and l_1–sensitivity to achieve DP are explained.

2.1 Differential privacy

DP [22], [24] is a strong mathematical framework used for statistical and ML applications to measure privacy [22]. Given a ML model and a record, it should not be possible to determine whether this record was part of the training data set [64], [39], [49]. Intuitively, DP requires that the information and inference being released about a sensitive data set should be robust to any changes of one sample [56], [57]. The definition of DP formalizes this intuition [22], [24]. However, if personal information is general statistical information, DP does not guarantee that this information will remain private [57], [25].

2.1.1 ϵ – differential privacy

The original definition of DP is ϵ–DP [22]. Where ϵ represents the privacy guarantee of a random mechanism M (Section 2.2) [64]. A random mechanism M is applied on two neighboring data sets d and d'. Two data sets are neighboring if they differ at most in one record. The random mechanism M is ϵ–DP, if it will return the output o where $M(d) = o$ and $M(d') = o$ with similar probability. Because ϵ limits the loss of privacy (Section 3.5.3) a smaller ϵ results in stronger privacy guarantee and a higher ϵ in lower privacy guarantees.[24]

Definition 1 *ϵ–DP. Adapted from [22] A randomized mechanism M with domain D and range R gives ϵ–DP, if for any neighboring data sets (differing at most in one record) $d, d' \in D$ and for any subsets of output $S \subseteq R$, it holds that:*

$$Pr(\mathcal{M}(d) \in \mathcal{S}) \leq e^{\epsilon} Pr(\mathcal{M}(d') \in \mathcal{S}) \tag{2.1}$$

The equation also goes in the other direction.

2.1.2 (ϵ, δ) – differential privacy

In this work, a relaxed definition of the original definition of DP (Definition 1) is used which is (ϵ, δ)–DP. Where δ is the probability that ϵ can not be held. Accordingly, when $\delta = 0$, (ϵ, δ)–DP and ϵ–DP are equivalent.[26]

Definition 2 (ϵ, δ)*–DP. Adapted from [26] A randomized mechanism M with domain D and range R gives* (ϵ, δ)*–DP if for any neighboring data sets (differing at most in one record) $d, d' \in D$, and for any subsets of outputs $S \subseteq R$, it holds that:*

$$Pr(\mathcal{M}(d) \in \mathcal{S}) \leq e^{\epsilon} Pr(\mathcal{M}(d') \in \mathcal{S}) + \delta. \tag{2.2}$$

The equation also goes in the other direction.

2.2 Random mechanism, Laplace mechanism and sensitivity

The released information, the response or the true query answer, $resp \in \mathcal{R}$ from a sensitive data set $d \in \mathcal{D}$ is first computed by a deterministic query function $f : \mathcal{D} \to \mathcal{R}$ [56]. Afterwards, to the result $f(d)$ a random mechanism M adds noise [23]. Thus, the released information becomes a random variable (Section 3.5.3) [56]. The random mechanism M can be defined as follows:

$$\mathcal{M}(d) = f(d) + \eta(\epsilon, s(f)) \tag{2.3}$$

where $\eta(\epsilon, s(f))$ denotes the noise injection. Depending on the mechanism M, η can follow different distributions.[56] In this work, η follows the Laplace distribution.

The probability density function of the Laplace distribution is characterized by its location μ and scale $b > 0$ parameters. The probability density function of $Lap(\mu, b)$ is:

$$p(x|\mu, b) = \frac{1}{2b} exp\left(-\frac{|x - \mu|}{b}\right) \tag{2.4}$$

Figure 2.1 shows the Laplace distribution with $\mu = 0$.

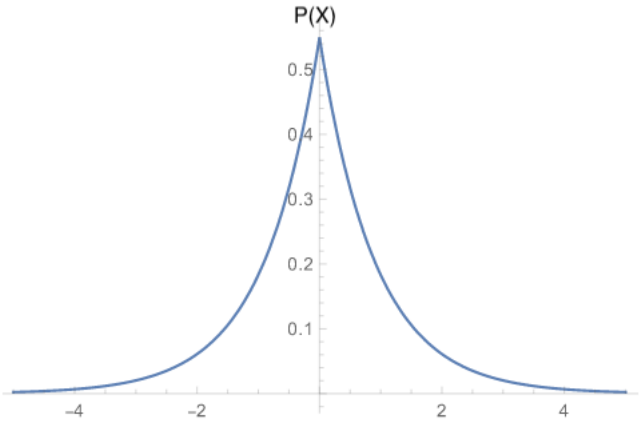


Figure 2.1: Laplace distribution. Adapted from [4]. Example of the probability density function $P(X)$ of the Laplace distribution with parameter $\mu = 0$.

Since, η follows the Laplace distribution, the random mechanism M is called Laplace mechanism [25]. The Laplace mechanism belongs to the global DP mechanisms since the data is perturbed at output time. In contrast, in local DP mechanisms data is perturbed at input time.[26] For the Laplace mechanism the noise injection is as follows:

$$\eta(\epsilon, s(f)) = \text{Lap}\left(\frac{s(f)}{\epsilon}\right) \tag{2.5}$$

where $\frac{s(f)}{\epsilon}$ is the scale parameter b of the Laplace distribution (Equation 2.4). The value $s(f)$ is the sensitivity (Definition 3) of the query function f (Equation 2.3).[25] The sensitivity will be explained in the following.

The scale of the noise injected is controlled by the sensitivity $s(f)$ and the privacy parameter ϵ [25]. Over any two neighboring data sets, the sensitivity $s(f)$ bounds the possible change in computing the output of the query f [56], [22], [26]. In other words, the sensitivity $s(f)$ is the maximum absolute distance of $f(d)$ and $f(d')$ [26]. The sensitivity $s(f)$ of the Laplace mechanism is called the l_1 - sensitivity and is defined as:

Definition 3 *l_1 - sensitivity of Laplace mechanism. Adapted from [26] The sensitivity $s(f)$ of two neighboring data sets $d, d' \in \mathcal{D}$ differing in at most one record and the deterministic function $f : \mathcal{D} \to \mathcal{R}$ is defined as:*

$$s(f) = \max_{d,d' \in \mathcal{D}} \|f(d) - f(d')\|_1 \tag{2.6}$$

Figure 2.2 shows the distribution of the outcome of the Laplace mechanism M on two neighboring data sets d and $d\prime$. In Figure 2.2, the probability that M returns an output $M(d) = o$ and $M(d') = o$ with the same probability is shown at point o.

For a more detailed introduction and the mathematical background on DP see [12].

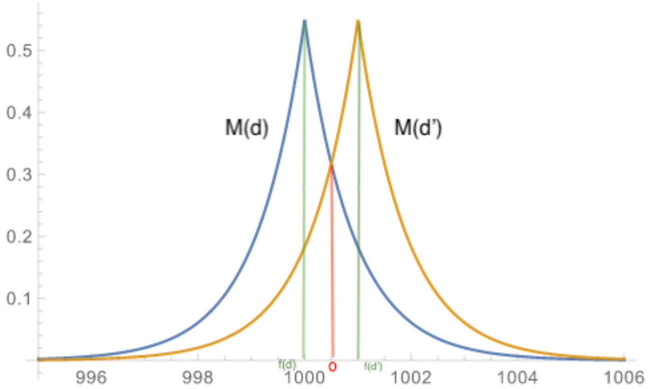


Figure 2.2: Probability density function of Laplace mechanism M on two neighboring data sets d (blue) and d′ (yellow). Adapted from [4]. The probability that $M(d)$ and $M(d\prime)$ return the same value is similar for both data sets at outcome o (red). The outcome of the query function f to data set d is $f(d)$ (green) and to data set d′ is $f(d\prime)$ (green).

3 PATE

This chapter gives an introduction to the original PATE framework [43]. Each section describes a key concept of PATE. This chapter starts with data portioning needed to train the PATE model, followed by the explanation of a teacher model and a student model. The next section explains the noisy maximum aggregation mechanism, which guarantees DP. Lastly, in the final section a privacy analysis of PATE is given.

Papernot et al. introduced PATE in [43] and expanded it in [44]. PATE is based upon a structured application of knowledge aggregation and transfer [13] which has been explored by Nissim et al. [41], Pathak et al. [46] and Hamm et al. [32]. Additionally, PATE is based on an ensemble of ML models [21] which is a common technique in ML [21]. In order to overcome the privacy concerns in ML, PATE is based on the following ideas. Firstly, the ensemble is trained on disjoint data subsets making the predictions made by most of the models independent of a single sensitive data point. Moreover, in order to prevent attackers from being able to inspect or access the internals of the learning model, the internal of the model is kept private. In addition, the student model, which will be published, never sees the sensitive data set. [43] Figure 3.1 shows the different aspects of PATE.

3.1 Data partitioning

In the sensitive training data set (X, Y) of the teacher ensemble, X denotes the set of inputs, Y the set of labels and r the number of teachers. Therefore, having r teacher, r sub data sets are generated. However, being trained independently, each teacher solves the same ML problem. The training set X and the labels Y is spit into r disjoint data sub sets (X_i, Y_i) where $i \in \{1, ..., r\}$. The subsets have the same size. On each of these subsets, one single teacher is trained separately.[43]

3.2 Teacher

The teachers, where a single teacher's prediction is f_i where $i \in \{1, ..., r\}$ (Section 2.2), are classifier trained on the sensitive sub data sets (Section 3.1). Therefore, the teachers are not published. Since PATE is agnostic to the underlying ML models of the teachers, there are no constraints on their training.[43]

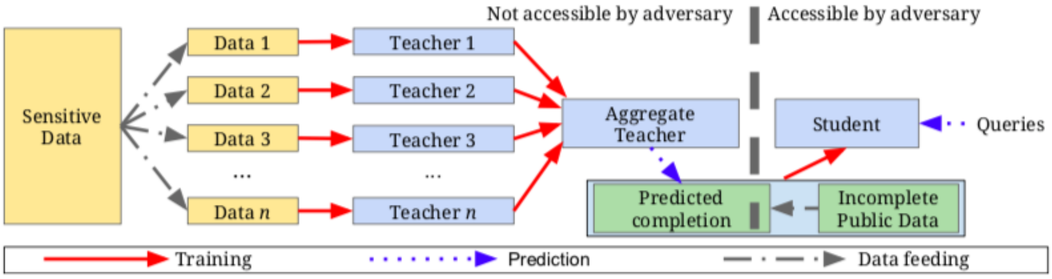


Figure 3.1: PATE framework. Taken from [43]. Step 1: Ensemble of teachers is trained on disjoint sensitive data sets (1). Step 2: Teacher ensemble makes predictions by using the maximum aggregation on unlabeled public training data set of the student (2). Step 3: Student is trained on these public data using resulting labels from noisy maximum aggregation of the teacher predictions (3). Step 4: Student model gets published to solve ML tasks (4).

3.3 Student

The student is a classifier trained on a public data set Z and the resulting labels from section 3.4. In this thesis, the student is trained in a supervised way. The student asks the teacher ensemble to label its unlabeled training data set which is done by the maximum aggregation mechanism (see section 3.4). In this thesis, these queries of the student to the teachers are called "label queries" or "student queries". The total amount of the label queries is T. As the student is trained in a supervised way, the total amount of label queries T is equal to the size of the training data Z of the student. After the student is trained, the student answers predictions queries from the end users. At this point, the privacy loss of PATE is immutable and the adversary is not able to infer individual information of the training data by analysing the student model or the students' output. Therefore, the student can be published and privacy can be maintained in terms of DP.[43]

3.4 Noisy maximum aggregation mechanism

The ensemble of r trained teachers predicts on unseen, non-sensitive data Z. Z is the training data set of the student with size a. Each teacher makes a prediction $f_i(z_k)$ where $i \in \{1, ..., r\}$ on input $z_k \in Z$ where $k \leq a$.

Given m classes. Then, the label count $n_j(z_k)$ for a given class $j \leq m$ and an input $z_k \in Z$ is the sum of the number of teachers that assigned class j to input z_k.[43] In this thesis, label count and vote counts are used interchangeably.

To enforce privacy, the Laplace mechanism is used (Section 2.2). This means, random noise from the Laplace distribution $Lap\left(\frac{1}{\gamma}\right)$ is added to the vote counts n_j of the teachers on class j and input z_k.

Regarding the Laplace mechanism, in this thesis, the parameter μ is 0 and the scale parameter b is $\left(\frac{1}{\gamma}\right)$ of the Laplace distribution (Section 2.2).

Next, the perturbed vote counts for each class j on an input z_k are compared. The class which has the highest perturbed vote counts is the final prediction of the PATE model on input z_k.[43]

$$g(z_k) = argmax_j \left\{ n_j(z_k) + Lap\left(\frac{1}{\gamma}\right) \right\} \tag{3.1}$$

where g is used as the final prediction of the PATE model on input z_k, γ (see section 3.5.1) is the privacy parameter for a single label query of the student to the teachers and $Lap(b)$ is the Laplace distribution with location parameter $\mu = 0$ and scale parameter $b = Lap\left(\frac{1}{\gamma}\right)$.

3.5 Privacy analysis

3.5.1 Influence of γ

The parameter γ is the privacy parameter for a single label query (Equation 2.5). Therefore, it influences the privacy guarantee of PATE. The scale parameter b of the Laplace distribution is the inverse of γ (Equation 2.5). Thus, the scale of the Laplace distribution increases with a small γ. This results in a strong privacy guarantee. Since the scale of the Laplace distribution decreases with a large γ, a large γ leads to a weak privacy guarantee. However, a small γ may result in worse accuracy because adding random samples from Laplace distribution to the label votes may cause the noisy maximum to differ from the true maximum of the label counts.[43]

3.5.2 Privacy cost of noisy maximum aggregation mechanism

For two neighboring databases d and d' and r teacher, r-1 teachers with predictions f_i where $i < r$, get the same same training data partition of d and d'. Teacher f_i where $i = r$ gets the sub sets of d and d' differing in one entry.

As a result, the label counts n_j for a given class $j \in m$ and an input $z_k \in Z$ of r-$S1$ teachers should be the same. The label counts n_j of the teachers with predictions f_i where $i = r$ for a given class $j \in m$ and an input z_k differ by at most 1 on both data sets d and $d_\prime$. As a result, the sensitivity in one class is 1. Therefore, $Lap(1/\gamma)$ is added to the label count n_j on class j of input z_k with $\epsilon = \gamma$ [43].

Since the label counts differ in at most two classes, the sensitivity of the maximum aggregation mechanism is 2 [56]. According to the Laplace mechanism, we have $\epsilon = 2\gamma$ (Equation 2.5) [56]. Thus, the noisy maximum aggregation mechanism (Section 3.4) is $(2\gamma, 0)$ – DP for one label querying of the student to the teacher ensembles (Section 3.3).[43]

3.5.3 Deriving the privacy of loss PATE

Privacy loss and privacy loss random variable

First of all, the privacy loss and the privacy loss random variable are defined. They show the difference of the probability distribution of running $M(d)$ and $M(d')$ [43].

The privacy loss ϵ is calculated at a specific value $o \in \mathcal{R}$. The actual value of the privacy loss depends on the random mechanism M (Equation 2.3). Therefore, the actual value of the output o depends on a random phenomenon. The function mapping the input d, d', some auxiliary input aux and M to the possible random output for all $M(d)$ is the privacy loss random variable $\mathcal{C}(\mathcal{M}, aux, d, d')$. The definition is as follows:

Definition 4 *Privacy loss at outcome o. Taken form [43] Let $\mathcal{M} : \mathcal{D} \rightarrow \mathcal{R}$ be a randomized mechanism and d, d' a pair of neighboring databases . Let aux denote an auxiliary input. For outcome $o \in \mathcal{R}$, the privacy loss at o is defined as:*

$$c(o; \mathcal{M}, aux, d, d') \equiv log\left(\frac{\mathcal{P}r(\mathcal{M}(aux, d) = o)}{\mathcal{P}r(\mathcal{M}(aux, d') = o)}\right) \tag{3.2}$$

The privacy loss random variable $\mathcal{C}(\mathcal{M}, aux, d, d')$ is defined as $c(\mathcal{M}(d); \mathcal{M}, aux, d, d')$ i.e. the random variable defined by evaluating the privacy loss at an outcome samples from $\mathcal{M}(d)$

Moments accountant

The student queries the teachers T–times to get enough labeled training data. The noisy maximum aggregation mechanism (Section 3.4) is applied once per query (Section 3.3). Therefore, PATE is a composition of all maximum aggregation mechanisms used to label each label query. To measure the total privacy loss of PATE, strong composition theorem [27] and moments accountant [9] have been proposed. The moments accountant method provides a more accurate calculation of the privacy loss [9]. Therefore, it is used in this work. The properties of the moments accountant method are proven and introduced by Abadi et al. [9], building on previous work [14], [40].

The moments accountant method (Definition 5) is based on the moment generating function of the privacy loss random variable (Definition 4). Before defining the moments accountant method, the moments of a random variable and the moments generating function of a random variable are shortly explained.

The moments of a random variable describe the characteristics of the distribution of a random variable. For example, the first moment of a random variable is the expectation value and the second moment is the variance of the random variable. All moments together describe the distribution function of the random variable.[45]

The moment generating function is used to calculate the moments of a distribution. Moreover, the distribution of summations of two random variables is the product of the two moment generating functions. If the moment generating function exists of a random variable exists, then its' distribution function exists.[45]

The cumulant generating function, which is the natural logarithm of the moment-generating function, is used to calculate the moments of a distribution. The λth derivative of the cumulant generating function is the λth moment of the random variable.[45]

The moments accountant $\alpha_{\mathcal{M}}(\lambda)$ is the maximum of the cumulant generating functions of the privacy loss random variables $\mathcal{C}(\mathcal{M}, aux, d, d')$ (Section 3.5.3). The moments accountant method calculates the cumulant generating function $\alpha_{\mathcal{M}}(\lambda; aux, d, d')$ for each random variable of the privacy loss. By comparing the resulting cumulant generating function, the maximum of them is chosen as the total privacy loss. Thus, the worst case scenario of the privacy loss is taken. As an example: calculating the first moment, which is the expectation variable [45], and taking the maximum of all, is the largest value for the privacy loss and the worst case scenario.

The moments accountant are defended as follows:

Definition 5 *Moments accountant. Taken from [9] Let $\mathcal{M} : \mathcal{D} \rightarrow \mathcal{R}$ be a randomized mechanism and d, d' a pair of neighboring databases. Let aux denote an auxiliary input. The moments accountant is defined as:*

$$\alpha_{\mathcal{M}}(\lambda) \equiv max_{aux,d,d'} \alpha_{\mathcal{M}}(\lambda; aux, d, d') \tag{3.3}$$

where $\alpha_{\mathcal{M}}(\lambda; aux, d, d') \equiv \ln E(exp(\lambda \mathcal{C}(\mathcal{M}, aux, d, d')))$ is the moment generating function of the privacy loss random variable.

The composition theorem of the moments accountant allows to bound $\alpha(\mathcal{M}(\lambda))$ at each step of PATE and sum them up to bound the moments of PATE [9].

Theorem 1 *Composability. Taken from [43] Suppose that a mechanism $\mathcal{M}$ consists of a sequence of adaptive mechanisms $\mathcal{M}_1 ... \mathcal{M}_2$ where $\mathcal{M}_i$ is*

$$\prod_{j=1}^{i-1} \mathcal{R}_j \times \mathcal{D} \rightarrow \mathcal{R}_i \tag{3.4}$$

Then for any output sequence $o_1 ... o_{i-k}$ and any λ

$$\alpha_{\mathcal{M}}(\lambda; d, d') = \prod_{i=1}^{k-1} \alpha_{\mathcal{M}_i}(\lambda; o_1 ... o_{k-1}, d, d') \tag{3.5}$$

where $\alpha_{\mathcal{M}}(\lambda; d, d')$ is conditioned on $\mathcal{M}_i$'s output being o_i for $i < k$.

To convert the moments to the DP–guarantee the tail–bound is used [43]. The tail–bound maps δ to a given ϵ. The theorem is as follows:

Theorem 2 *Tail bound. Taken from [43] For any $\epsilon > 0$, the mechanism M is (ϵ,δ)–DP for*

$$\delta = min_{\lambda} exp(\alpha_{\mathcal{M}}(\lambda) - \lambda\epsilon) \tag{3.6}$$

Over T labels queries from the student to the teachers the following privacy loss for PATE results based on the moments accountant:

Theorem 3 *Privacy loss of PATE. Taken from [43] For any γ and δ, over T steps using the aggregation mechanism with noise Lap $\left(\frac{1}{\gamma}\right)$, which is $(2\gamma, 0)$ - DP, satisfies (ϵ, δ) - DP, where*

$$\epsilon = 4T\gamma^2 + 2\gamma\sqrt{2T \ln \frac{1}{\delta}} \tag{3.7}$$

Determined by the number of queries T made to the teachers during training the student, the privacy loss is independent from the end-user queries made to the student [43].

4 Methods and Data

This chapter gives an overview of the data set, methods and libraries used to implement and evaluate the PATE classifier. First, the most common Python DP-libraries will be introduced. After that, the Pima Indian diabetic data set (PID) will be described. Third, the method to split training and test data will be shown. Then, the different classifiers, which are teachers, student and baseline model will be introduced. Therefore, 2 NN, 2 LR and 1 SVM were implemented. Moreover, hyper parameter tuning of the classifier using $GridSearch()$ and Hparams were explained. Hparams is a Tensorflow (TF) dashboard for hyper parameter tuning of NN. The next section deals with the evaluation of the classifier using ROC, confusion matrix and cross validation. Then, the explanation which classifier is chosen as the baseline model is given. The last section explains the steps to implement the PATE model, which includes the implementation of the noisy maximum mechanism (Section 3.4), the calculation function of epsilon (Theorem 3), the implementation of different numbers of teachers and the implementation to evaluate the number teachers, accuracy, noise injection and privacy loss.

4.1 Differential privacy library

There are three famous open-source Python libraries for DP–ML: Opacus [7], TF Privacy [8] and PySyft [6]. Recently, the open-source library Opacus was released by Facebook. Opacus trains DP–PyTorch models. Since this library was not released during this work, it was not used in this work.[7] Furthermore, PySyft is a library for secure and private deep learning developted by OpenMined community extending the libraries TF, Kreas and Pytorch. It gives a framework for the PATE $syft.frameworks.torch.dp.pate$. Currently, according to the developer of the PySyft library, the library is not secure and should not be used to protect data. [6] Therefore, this library was not used in this work. Lastly, TF Privacy was developed by Google. One of the engineers is Nicolas Papernot who is the main researcher of PATE. Moreover, he released his source code for PATE in the repository of the TF Privacy library which is adapted in this work.

4.2 Pima Indian diabetic data set

The PID was collected form the National Institute of Diabetes and Digestive and Kidney Diseases [4]. as part of the Pima Indians Diabetes data base. The data set includes 768 patients. All patients are at least 21 years old and are

Table 4.1: Python libraries for DP–ML.

library	producer	models
Opacus [7]	Facebook	PyTorch models
TF Privacy [8]	Google	TF models
PySyft [6]	OpenMined	TF, Kreas and Pytorch models

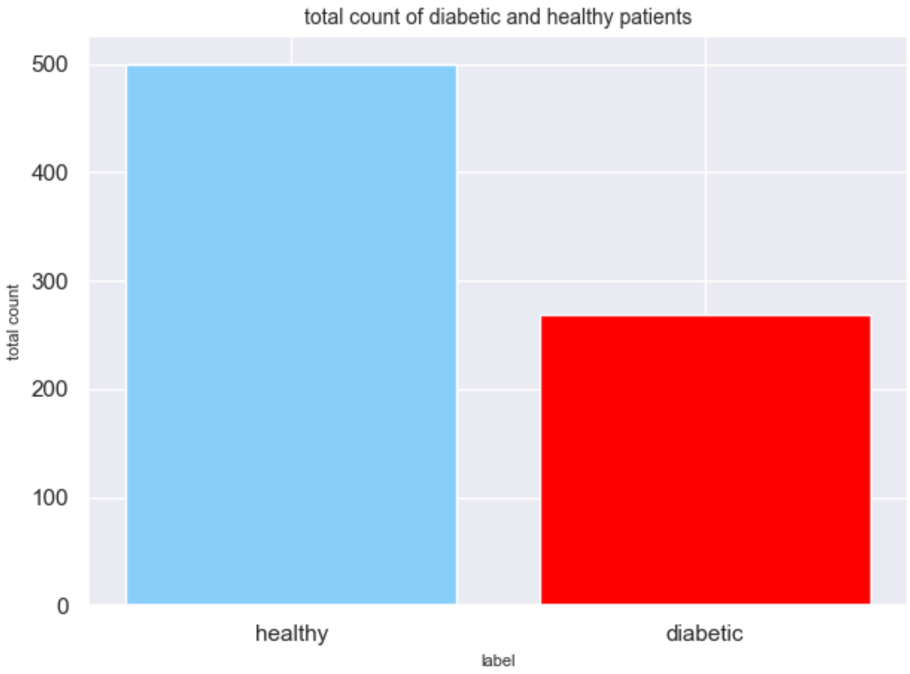


Figure 4.1: Imbalanced classes of PID. Absolute count of diabetic (red) and non-diabetic (blue) patients included in the PID.

female Pima Indian heritage. Based on diagnostic measurements, ML models can predict whether or not a patient has diabetes as a binary classification problem.[5]

4.2.1 Classes

The classes are 0 if non-diabetic (negative) and 1 if diabetic (positive). The target distribution between positives and negatives is imbalanced. This means, that the two classes are not represented equally. Figure 4.1 shows that there are 500 non-diabetic and 268 diabetic patients.

4.2.2 Missing values

The PID has missing values. In order to keep the already small data set at 768 data points, the missing values are replaced by the median of the class on the corresponding label.

4.2.3 Features

The data consists of eight medical predictor variables as features. Figure 4.2 gives an overview of the probability distribution and a description of each feature from the data set. To get the correlation between the features, the correlation matrix was plotted. Figure 4.3 shows the correlation matrix between features. If the correlation factor is closer to one the higher the correlation between the features. The figure shows that pregnancies, Glucose and Body mass index (BMI) have significant correlation with the outcome. Moreover, the features of BMI and skin thickness, Insulin and Glucose, age and pregnancies have a significant correlation with each other.

4.2.4 Split test and training data

Using the function $traintestsplit()$ from the Python ML library Scikit Learn, the data set was split into test and training data. 30% of the data belonged to the test or validation set and 70% to the training set. For reproducibility, the parameter $randomstate$ was set to 0.

4.3 Classifier

LR, SVM and NN classifier were compared as potential base line model (Section 4.6), teacher model and student model (Section 4.7.4).

4.3.1 Support vector machine

The SVM was created using Scikit Learns' classifier $svm.SVC()$. For binary classification a linear kernel is needed. Therefore the parameter, $kernel$ was set to $linear$.

4.3.2 Logistic regression

The LR model was created using Scikit Learns' classifier
$linearmodel.LogisticRegression()$.

4.3.3 Neural network

The NN was implemented using TF 2.0. The NN consisted of 1 input layer, 3 hidden layers and 1 output layer. The hidden layers had the activation function rectified linear activation function ($relu$) as activation function. The output layer had the activation function *sigmoid*. To get a binary classifier, the output layer consisted of 1 neuron, binary cross entropy was used as a loss function and a threshold of 0.5 was chosen. Values greater or equal to the

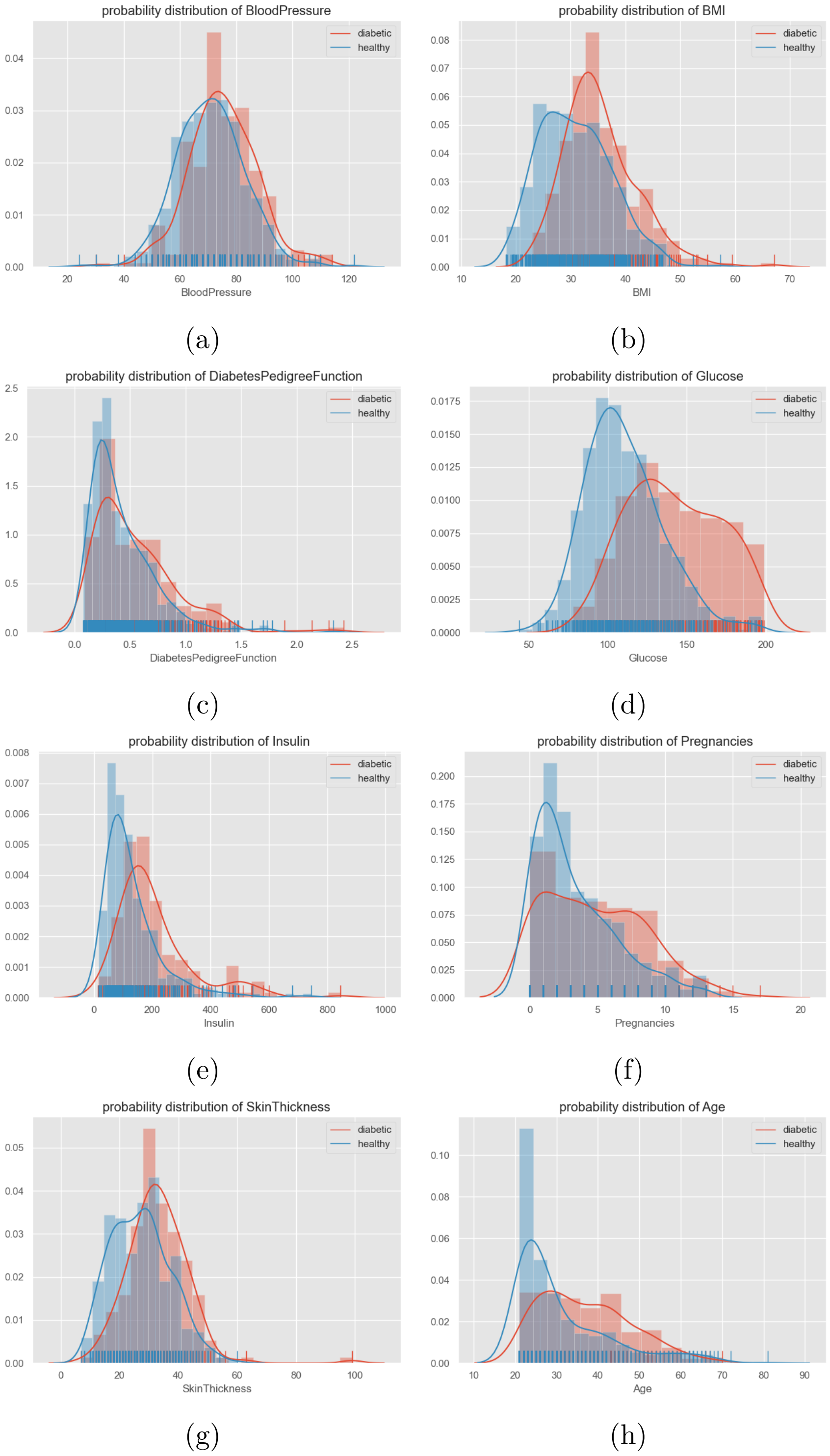


Figure 4.2: Probability distribution of features of the PID set. (a) Blood Pressure: Diastolic blood pressure (mm Hg), (b) BMI: Body mass index $((kg_{weight})/(m_{height})^2)$, (c) Diabetic Pedigree Function: scoring the likelihood of diabetes based on family disease history, (d) Glucose: Plasma glucose concentration, (e) Insulin: Insulin concentration (mu U/ml), (F) Pregnancies: Number of pregnancies per person, (g) Skin Thickness: Triceps skin fold thickness (mm), (h) Age: Age (years)

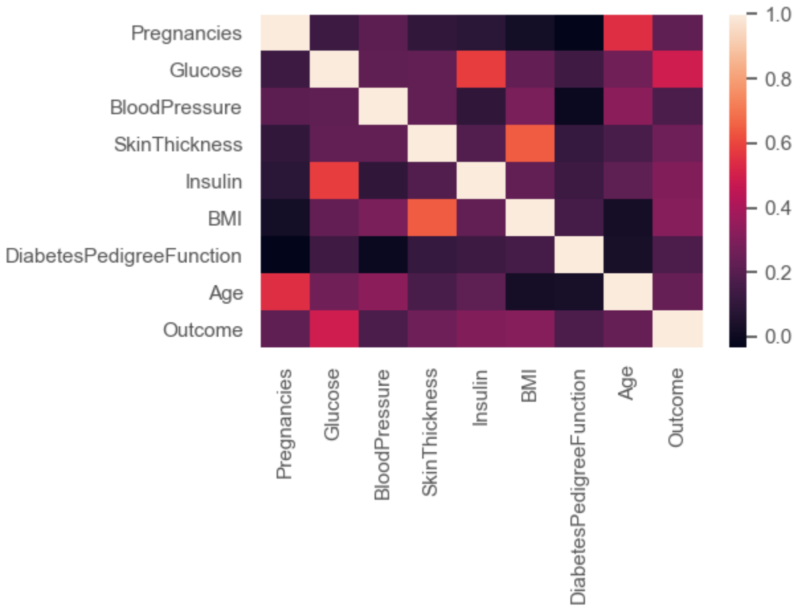


Figure 4.3: Correlation between features of PID where a higher correlation factor belongs to a higher correlation between two features (brighter colours). The closer the correlation factor is to zero, the lower the correlation between two features (darker colours).

threshold were predicted as positive, otherwise as negative. This was realised in an extra function.

4.4 Hyper parameter tuning

4.4.1 Grid search

The function $GridSearchCV()$ from Scikit Learn was used to optimize SVM and LR accuracies. $GridSearch()$ evaluates the metrics, in this work accuracy, for each paramter partition.

$Gridsearch()$ for SVM was done with the following parameter: the parammeter C was set to: 0.001, 0.002, 0.003, 0.004, 0.005, 0.006, 0.007, 0.008, 0.009, 0.1, 0.2, 0.3, 0.4, 0.5, 1, 10, 100, 1000. And the $kernel$ was set to $linear$.

$Gridsearch()$ for LR was done with the following values for the parameter for C : 0.001, 0.002, 0.003, 0.004, 0.005, 0.006, 0.007, 0.008, 0.009, 0.1, 0.2, 0.3, 0.4, 0.5, 1, 10, 100, 1000, the paramter $penalty$ was set to $l1$, $l2$, the $solver$ was set to $liblinear$. The second $GridSearch()$ for LR was done with: C set to: 0.001, 0.002, 0.003, 0.004, 0.005, 0.006, 0.007, 0.008, 0.009, 0.1, 0.2, 0.3, 0.4, 0.5, 1, 10, 20, 30 ,40, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 70, 80, 90, 99, 100, 101, 102, 105, 110, 120, 1000, $penalty$ set to $l2$ and $solver$ set to $lbfgs$.

4.4.2 Hparams and Tensorboard

The accuracy of the NN was optimized by using the TF plugin Hparams and Tensorboard. Hparams was differing optimizer, number of neurons in hidden layers and batchsizes. The following optimizers were used: *adam*, *sgd*, *RMSprop*. In the first layer were 4, 8, in the second 4, 8, 16 and in the third layer 4, 8, 16 neurons plagued in. The experiment was done for batchsize 25 and 32 and 100 epochs each. For each partition the accuracy was evaluated using the test data set (Section 4.2.4).

4.5 Model evaluation

4.5.1 Cross validation

The optimized models (Section 4.4) were evaluated by cross validation using $KFold()$ and $StratifiedKFold()$ from Scikit Learn starting from 2 folds up to 20 folds. Additionally, for each fold size, the average accuracy and the standard deviation was measured. Moreover, the PATE model itself was evaluated using $KFold()$ with 5 folds.

In order to be able to use Sikit Learns functions on the NN, a wrapper function was implemented using the $KerasClassifier()$ from $keras.wrappers.scikitlearn$. The resulting network was used for further analyzing and evaluation steps.

4.5.2 Confusion matrix

In order to choose the baseline model, and the teachers and the student model, the confusion matrices for the optimized models were calculated. Therefore the function $confusionmatrix()$ from Scikit Learn was used. By combinations of predicted and actual values, the confusion matrix is an other measurement to evaluate the performance of a classification model. Its outcome is the number of true positive (TP), true negative (TN), false positive (FP), false negative (FN).

4.5.3 Receiver operating characteristic curve

The ROC was plotted. The ROC curve visualizes the trade off between TP rate and FP rate using different probability thresholds. However, in this work, the probability threshold is fixed at 0.5, and the area under the curve (AUC) of the different models is compared. The AUC value should be close to one.

4.6 Baseline model

The aim of the baseline model is to get a justified comparison with the PATE model. To evaluate the performance of linear and non–linear classifier, LR, NN and SVM (Section 4.3) were compared.

4.7 PATE model

4.7.1 Data set

The data sets resulting from section 4.2.2 were used to train and validate the PATE model. The split of the test and training was 70% to 30%. The split results from section 4.2.4.

4.7.2 Disjoint data splits for teachers and student

The training data from section 4.7.1 was split into disjoint sub training data sets. The number of sub sets results from the sum of the number of teachers and plus the student. Therefore, each teacher and student gets its own training data set. Afterwards, each teacher was trained on its training data sub set (Section 3.1, Section 3.2).

4.7.3 Noisy max aggregation votes of the teachers

Each trained teacher predicted labels for the student training data (Section 3.3). To calculate the noisy maximum aggregation votes from 3.4 of the teachers, the $noisymax()$ function was adapted from TF Privacy [8] and was changed to a binary classification problem. This function takes as input a list including the predictions of the teachers and the scale value b of the Laplace distribution (Section 2.5). Then, the label counts (Section 3.4) and the noisy maximum aggregation of the votes is computed (Equation 2.3). Therefore, it randomly samples from the Laplace distribution using the function $random.laplace()$ from Python library Numpy. Afterwards, the most frequent label for each data point was returned.

4.7.4 PATE classifier

First of all, to evaluate the models of the teachers (Section 3.2) and students (Section 3.3), the accuracy of the optimized models resulting from section 4.4 were compared. The model with the highest accuracy was chosen. After that, PATE (Chapter 3) was implemented with 2, 3, 4, 5, 10 and 20 teachers. Then, the student was trained on the data resulting from section 4.7.1 and the output labels of the noisy maximum aggregation mechanism of the teachers (Section 4.7.3). After that, the trained student is used to make predictions on unseen data sets.

4.7.5 Evaluation of privacy loss

To calculate the privacy loss of the PATE model, the equation of Theorem 3 was implemented as function $epsilon()$. The function gets T, γ and δ and returns the privacy loss for the PATE model as described in section 3.5.3. For this purpose, δ was fixed at 0.00001. T is the total amount of the students queries to the teacher. As in this work the student learns from the teacher in a supervised fashion, T is fixed and is equal to the size of the training data of the student (Section 3.3).

4.8 Evaluation of accuracy

In the following experiment, the PATE model predicted on unseen data. The aim was to get as high accuracy as possible and to not overfit the data while epsilon was not considered. For this, accuracy and γ per label query were plotted for every number of teachers from section 4.7.4. Then, epsilon was calculated based on section 4.7.5.

4.9 Evaluation of noise injection

In this experiment, epsilon was fixed at 2, 5, and 8. To calculate epsilon, different values for γ were inserted in the implementation of section 4.7.5. Then, the PATE models with different numbers of teachers and differing ϵ predicted on unseen data. Lastly, their accuracy was evaluated.

4.10 Evaluation of the number of teachers

The aim of this experiment was to evaluate the confidence and the agreement of the teachers about the output of the PATE model. For example, consider a PATE model with one teacher. The output of the PATE model only depends on the prediction of this teacher. Therefore, the teacher always agree 100% on the outcome of the PATE model and is confident about that. In contrast, considering a PATE model with 100 teachers, depending on the agreement of the teachers about the output of the PATE model, a single teachers prediction may not change the outcome of the PATE model. This may be because the difference between the highest label count n_i on an input z_k and the second highest label count n_j on the same input z_k differ in more than one vote. The mean of the difference is called absolute gap gap_a. A larger absolute gap relative to the number of teachers indicates a stronger confidence of the teachers.

To evaluate this hypothesis, the percentage gap was calculated. If the percentage gap is higher, then the teachers agree on their predictions and the model is more confident about the output. To calculate the percentage gap, the mean of the absolute gap was calculated for different numbers of teachers. Then, the difference was normalized by the number of teachers r and the multiplied by 100. This was done for 1, 2, 3, 4, 5, 6, 7, 8, 9 and 10 teachers.

$$gap_p = \frac{gap}{r} * 100 \tag{4.1}$$

To give a short example, the following gives a calculation of different percentage gaps resulting from 6 teachers. Given 6 teachers where 5 teachers predict outcome positive and one predicts outcome negative, the absolute gap is 4. With the calculation from above, this results in a percentage gap of 66.66%. Additionally, if 4 teachers predict outcome positive and 2 predicts outcome negative then the percentage gap is 33.33%. Finally, if all predict outcome positive, the percentage gap is 100.00%. 100% means that the teachers totally agree on the output. For an even number of teachers a special case occurs where the gap is 0. For example: given a PATE model with 6 teachers. 3 teachers predict outcome positive and 3 predicts outcome negative it is 00.00%

5 Results

The following chapter presents the results of this thesis. First of all, the base line model is chosen. A description of the results of the different classifiers for the baseline model can be found in Table 5.1. The second section shows the results for the different classifiers, which were evaluated with accuracy as the main parameter. Firstly, two LR models with parameters (Table 4.4.1) were compared. Afterwards, two NN, differing in batchsizes, were evaluated (Table 4.4.2). Lastly, by comparing the accuracy, confusion matrices, ROC curve and K–fold cross validation up to 20 folds the LR model, the NN and the SVM model were compared to each other. The next section shows the results of the evaluation of the accuracy of the PATE model. Therefore, different PATE models with 2, 3, 4, 5, 10 and 20 teachers were compared. Their accuracy was plotted against the noise injection per label query γ. Moreover, for 2 to 5 teachers, the resulting privacy loss was calculated. The next section deals with the evaluation of the noise injection. Therefore, the value for the privacy loss was fixed at 2, 5 and 8. Delta was fixed at $1e-5$. The accuracy of the PATE model with 2, 3, 4 and 5 teachers with these privacy losses were compared. Afterwards, the number of teachers were evaluated. Therefore, the gap, which is the average difference of the number of votes between the top most and the second most frequented labels, were plotted. Lastly, the accuracy of the PATE classifiers with the best performance was compared to the accuracy of the baseline model.

5.1 Baseline model

As the baseline model, the model with the best performance of 5.2.1 was chosen. The resulting model was NN with accuracy of 0.87013, with batchsize 25, 8 neurons in the first, 8 neurons in the second and 8 neurons in the third layer, the $RMSprop$ optimizer and with $l2$ regularizer set to 0.01.

5.2 PATE

5.2.1 Evaluation of classifier as teachers and student model

Comparing LR

The two different models for LR models, LR 2 and LR 1 , had the same accuracy on the test data set. Their accuracy was 0.77. Table 5.1 shows the corresponding parameter. Since both models have the same accuracy, the F1–Score of both were compared. The F1 score for positive outcome was 0.61

(LR 1) and 0.60 (LR 2). Therefore, LR 1 has been chosen for the following evaluations. In the following the optimized LR is LR 1.

Table 5.1: Comparison of two LR models. Model LR 1 has a higher F1–Sore than model LR 2. Both models have the same accuracy.

model	accuracy	C	penalty	solver	F1 score (positives)
LR 1	0.77	0.005	l2	liblinear	0.61
LR 2	0.77	0.2	l2	lbfgs	0.60

Comparing NN

Figure 5.1 shows the evaluation of the different NN. Two NN for batchsize 25 and batchsize 32 were implemented. Table 5.2 shows the parameters of the NN with the highest accuracy. Since NN 1 had higher accuracy than NN 2, NN 1 was chosen for the next evaluations. In the following the optimized NN is NN 1.

Table 5.2: Parameter of the NN with highest accuracy with batchsize 25 (NN 1) and batchsize 32 (NN2).

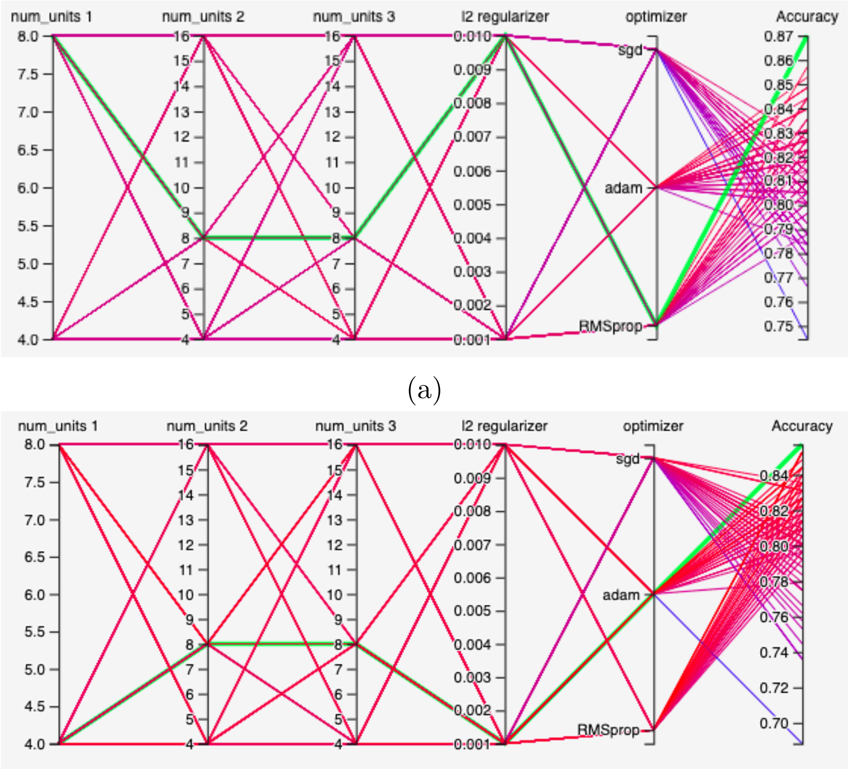
model	accuracy	neurons	optimizer	l2 regularizer	batchsize
NN1	0.87013	8, 8, 8	RMSprop	0.01	25
NN2	0.85714	4, 8, 8	adam	0.001	32

Accuracy of NN, SVM, LR

Table 5.3 shows the test accuracy of the optimized models. The NN (using a Kreas Wrapper (Section 4.5.1)) had the highest accuracy (0.83) followed by SVM (0.78) followed by LR (0.77).

Cross validation

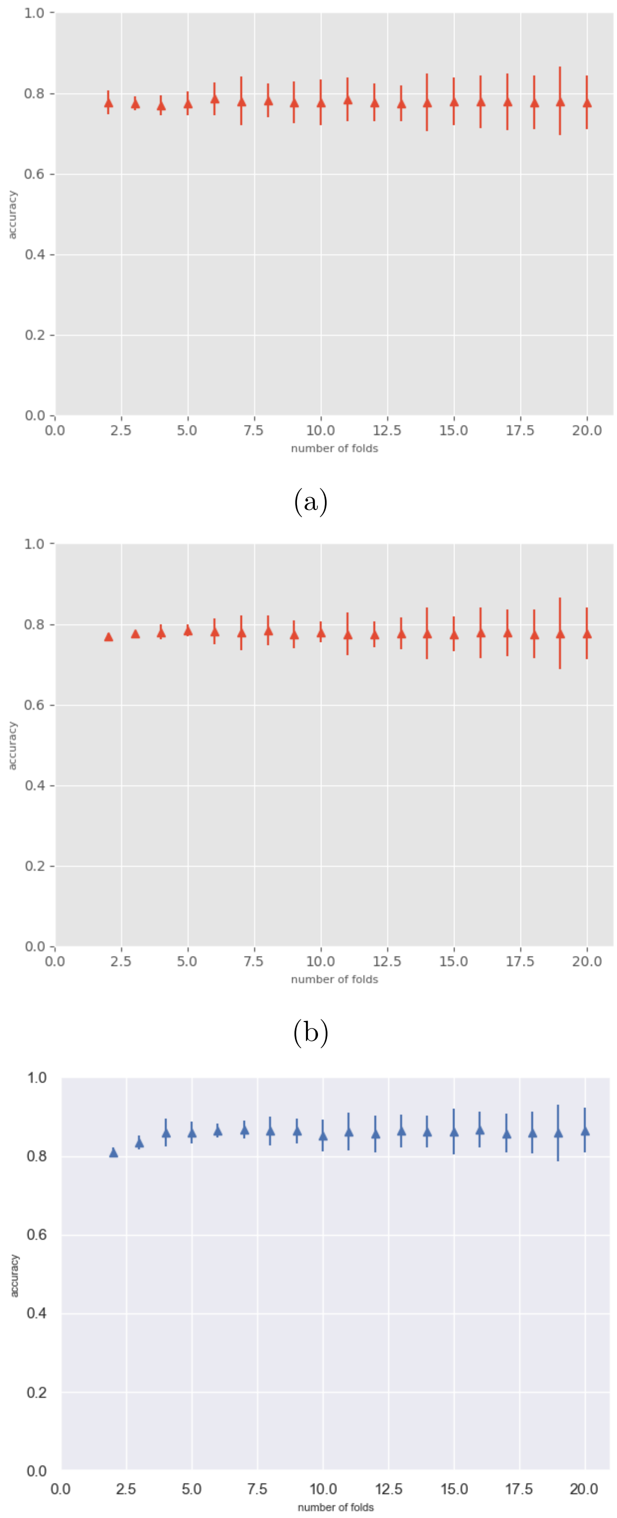
Figure 5.2 and Figure 5.3 show the results of cross validation of the optimized models. The folds ranged from 2 to 20. Comparing the fold sizes, the greater the fold size, the smaller the data sub set and the higher the variance. The accuracy fluctuated around the test accuracy of the models in 5.2.1. This was similar for stratified K–fold and K–fold. Comparing the models in each fold, the NN had the highest accuracy for all fold sizes. Followed by SVM and LR which had similar accuracy for fold sizes.



(a)

(b)

Figure 5.1: Comparison of different batchsizes using Hparams and Tensorboard. NN 1 had batchsize = 25 (a), NN 2 had batchsize = 32 (b), The models with the highest accuracy are shown in green.

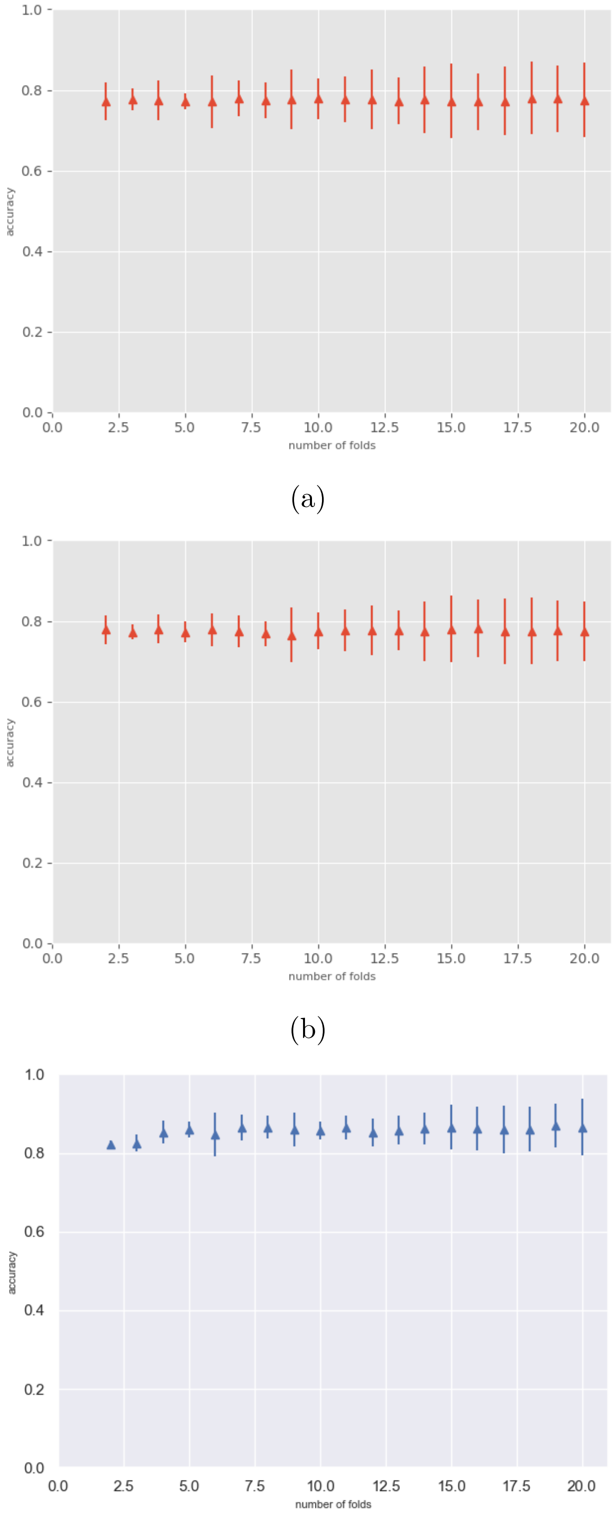


(a)

(b)

(c)

Figure 5.2: K–fold cross validation of LR (a), SVM (b), NN (c). Folds values ranged from 2 up to 20 folds. The variance increased with the fold size.



(a)

(b)

(c)

Figure 5.3: Stratified K–fold cross validation of LR (a), SVM (b), NN (c). Folds values ranged from 2 up to 20 folds. The variance increased with the fold size.

Table 5.3: Test accuracy of LR, NN and SVM. LR had parameter C is 0.005, *penalty* is $l2$, *solver* was *liblinear*, NN had 8 units in the first, 8 units in the second and 8 units in the third layer. The optimizer was $RMSprop$, the $l2$ regularizer had value 0.01 and the number of epochs is 25. For comparison, the accuracy of the NN was evaluated using the Kreas Wrapper (Section 4.5.1). SVM had parameter C with 0.4 and *kernel* was set to *linear*.

model	accuracy
LR	0.77
SVM	0.78
NN	0.83

Confusion matrix

Table 5.4 shows the confusion matrix of LR, SVM and NN. The test set consisted of 72 actual positive (P) and 159 actual negative (N) entries. The outcome of the models included more negative predictions than there were actual negative entries in the training data set which is discussed in chapter 6. The recall, which is TP/P, of NN is 0.71, LR is 0.60 and SVM is 0.58. The precision, which is $TP/(TP+FP)$, of NN is 0.72, SVM is 0.68 and LR is 0.63.

Table 5.4: Confusion matrix of LR (a), SVM (b), NN (c).The test set consisted of 72 actual positive and 159 actual negative entries.

	predicted negative	predicted positive
actual negative	134	25
actual positive	29	43

(a)

	predicted negative	predicted positive
actual negative	139	20
actual positive	30	42

(b)

	predicted negative	predicted positive
actual negative	140	19
actual positive	21	51

(c)

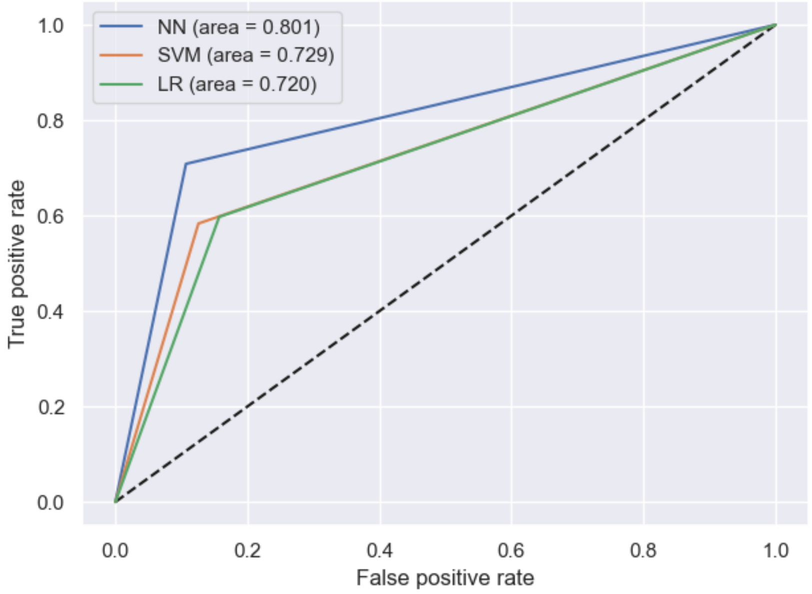


Figure 5.4: ROC curve of NN, SVM, LR. NN had the largest area (0.801), followed by SVM (0.729), followed by LR (0.720).

ROC curve

Figure 5.4 shows the ROC curves for a fixed threshold of 0.5. The closer the area under the ROC curve to 1, the better the performance of the model. NN had the largest area (0.801), followed by SVM (0.729), followed by LR (0.720).

According to the results in Sections 5.2.1, 5.2.1, 5.2.1 the NN outperformed the SVM and the LR. Therefore, NN was used in PATE as student and teacher model.

5.2.2 Evaluation of accuracy

Accuracy

The results from sections 5.2.1, 5.2.1, 5.2.1 showed that the NN outperformed the SVM and the LR. Therefore, the NN was chosen as the model for the teachers and the student.

Figure 5.5 shows the accuracy of different numbers of teachers with respect to γ. Small values of γ corresponded to a large noise amplitude. Large values of γ corresponded to a small noise amplitude (Equation 2.5).

The training accuracy of the models increased while the value of γ increased. Moreover, the higher the number of teachers was, the lower the test accuracy and the higher the training accuracy for increasing γ. For 2 (a), 3 (b), 4 (c) and 5 (d) teachers, the validation and training accuracy increased while γ increased. For 2 or 3 teachers, the validation accuracy mostly stayed higher than the training accuracy. For 4 and 5 teachers, the training accuracy mostly increased more than the validation accuracy and stayed higher than the validation accuracy. For a γ around 0.6 the training accuracy tended to be larger than the test accuracy. For 10 and 20 teachers, their validation accuracy stayed flat while γ increased. Moreover, their training accuracy increased.

For validation accuracy higher than training accuracy, 2 teachers had a peak at $\gamma = 0.2$ and validation accuracy 0.79, 3 teachers had a peak at $\gamma = 0.48$ and validation accuracy 0.8, 4 teachers had a peak at $\gamma = 0.38$ and validation accuracy 0.81, and 5 teachers had a peak at $\gamma = 0.65$ and validation accuracy 0.83. For the next evaluation steps, 10 and 20 teachers were dropped out due to the limited training data size which is further discussed in chapter 6.

Privacy loss

In the following, the resulting privacy loss of PATE was calculated for a γ greater or equal 0 and a validation accuracy higher than the training accuracy. Table 5.6 shows the results of the estimation. The accuracy increased with the number of teachers. 5 teachers had the highest accuracy (0.83) with an ϵ of 209.25 and 89 label queries. The lowest accuracy was identified for 2 teachers (0.79), resulting in ϵ of 54.32 with 179 label queries.

Table 5.6: Evaluation of accuracy. Delta is fixed at $1e - 5$. The highest accuracy of varying numbers of teachers where the test accuracy is higher than the training accuracy.

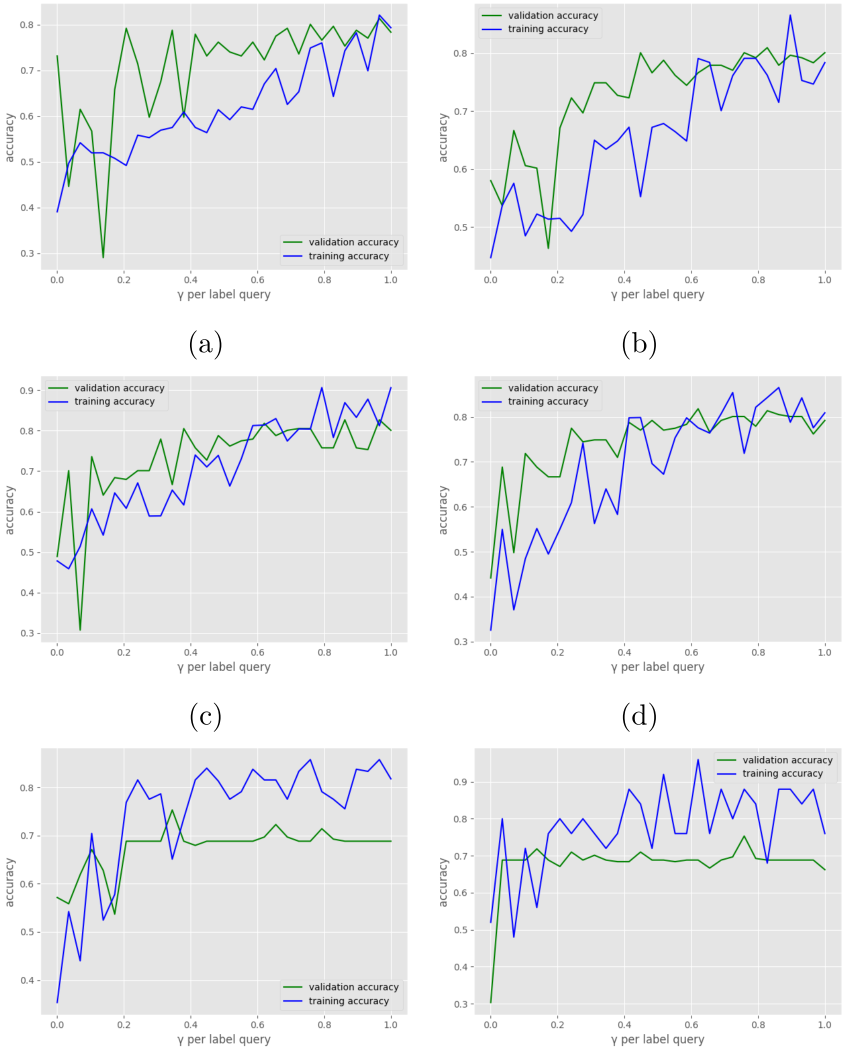
teachers	T	gamma	ϵ	accuracy
2	179	0.2	54.32	0.79
3	134	0.48	176.81	0.80
4	107	0.38	99.52	0.81
5	89	0.65	209.25	0.83

5.2.3 Evaluation of noise injection

Table 5.7 shows the accuracy reached by the models when having a privacy loss around 2, 5 and 8. Only 2 and 5 teachers reached an accuracy above 0.70. 4 teachers did not reach an accuracy above 0.60. In an accuracy comparison for each teacher, 2 teachers reached the highest accuracy of 0.74 by having a privacy loss of 8.01. The weakest performance occurred for 4 teachers with an accuracy of 0.31 and a privacy loss of 5.06.

5.2.4 Evaluation of the number of teachers

Figure 5.6 shows the percentage gap. With a smaller number of teachers, the percentage gap increased. With a higher number of teachers, the percentage gap decreased. 1 teacher showed a percentage gap of 100%, while 8 and 9 teachers showed the smallest percentage gap which was less than 60%.



(a) (b)

(c) (d)

(e) (f)

Figure 5.5: Accuracy of the noisy aggregation and varying γ per label counts for 2 teachers (a), 3 teachers (b), 4 teachers (c), 5 teachers (d), 10 teachers (e), 20 teachers (f). Small values of γ correspond to a large noise amplitude. Large values of γ corresponded to a small noise amplitude. The training accuracy was the mean of the K–fold cross validation using the training data set. The validation accuracy was the accuracy resulting from applying the model on the test data set.

Table 5.7: Comparison of PATE classifier with 2, 3, 4 and 5 teachers. The privacy loss is fixed around 2, 5 and 8. Delta is fixed at $1e - 5$.

teachers	T	gamma	epsilon	accuracy
2	179	0.015	2.08	0.39
2	179	0.033	5.02	0.57
2	179	0.049	8.01	0.74
3	134	0.018	2.17	0.57
3	134	0.038	5.00	0.61
3	134	0.057	8.07	0.62
4	107	0.020	2.16	0.58
4	107	0.043	5.06	0.31
4	107	0.063	7.95	0.54
5	89	0.022	2.16	0.76
5	89	0.047	5.04	0.55
5	89	0.069	7.94	0.69

5.3 Comparison PATE vs baseline model

Comparing the accuracy of PATE with the accuracy of the baseline model, the baseline model outperformed the PATE model. The base line model reached an accuracy of 0.83 5.2. The PATE model reached an accuracy of 0.76 and a privacy loss of 2.16. The PATE model consisted of 5 teachers and the number of student queries was 89 (Table 5.7).

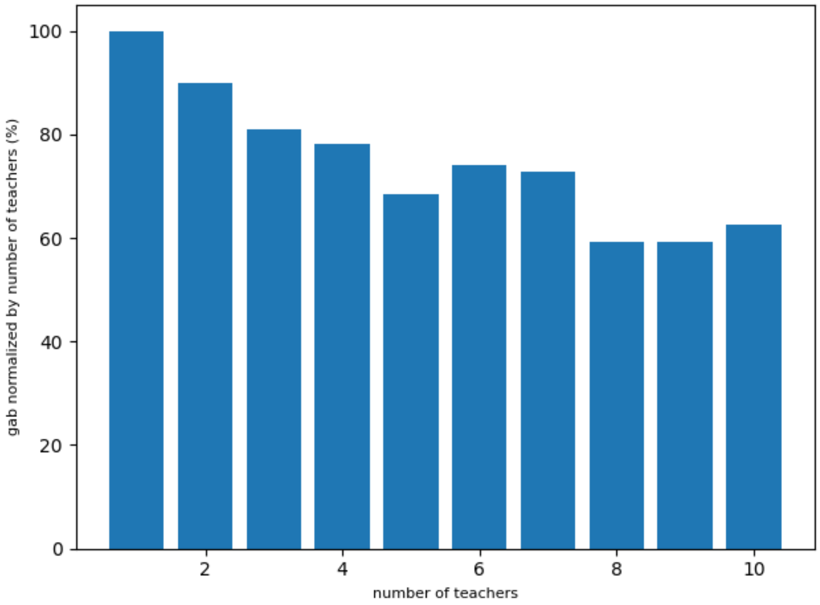


Figure 5.6: Confidence of the teachers. Percentage gap which is normalized by the number of teachers. A higher value indicates that the teachers agree on their predictions. Therefore, the teachers are more confident about the outcome of the PATE model.

6 Discussion

In this chapter, the results of this work will be discussed. The focus here will be on aspects and possibilities for increasing the accuracy and performance further, while comparing the results to recent literature. Therefore, the partition training data set, the number of teachers, the agreement of the teachers and the comparisons to the baseline model will be included.

6.1 Discussion of accuracy, number of teachers, noise injection partition of data set

According to Papernot et al. 2017, the performance of the PATE model greatly depends on the number of teachers [43]. The researchers trained 250 teachers in their work. This conflicts with the results of this thesis. In this thesis, the PATE model with best performance had 5 teachers. Even with a small number of teachers, compared to the number of teachers used in the work of Papernot et al. 2017, an accuracy of 0.76 and a strong privacy guarantee of 2.16 were reached. Comparing the accuracy of the baseline model (0.83) and the accuracy of PATE showed just a difference of 0.07 while introducing a strong privacy guarantee.

Moreover, having r teachers, each teacher trains on $(1/r)$ fraction of the data. If r is small, the teachers are trained on larger data sets resulting in a higher performance. This may be the reason why in this thesis a smaller number of teachers results in a strong confidence of the teacher ensemble about the outcome prediction.

Nevertheless, there were also some limitations due to the small data set. First of all, a large number of teachers r ended up in training sets that were too small. As a result, when trained with 10 and 20 teachers, the PATE classifier ended up overfitting the test data (Figure 5.5).

Secondly, noise injection may have a greater impact on the outcome, when training a number of teachers smaller than 10. This might also explain the accuracy of 0.31 and 0.39 for 2 and 4 teachers (Table 5.7). An additional explanation, which does not exclude the previous, may be that an even number of teachers can have the same amount of label counts in both classes. Therefore, the outcome is random.

To get a better performance of PATE on small data sets, Wang et al. 2019 used transfer learning to have a large number of teachers on a small data set. Their model extends PATE and is called "TrPATE". In transfer learning the performance of a model is improved by training on a target domain and transferring information from a related domain. The model is the teacher ensemble, the target domain is the sensitive training data set for the teachers and the source data set, with information from a related domain, is a related public data set. The knowledge from a related non-private data set is shared with every teacher of the ensemble. Thus, each teacher is trained on both the private data and the transferred knowledge.[56] Moreover, if the target data set is significantly smaller than the source data set, learning transfer enables training a large target network without overfitting [60].

In addition, the data set plays a crucial role for developing a high performing model [38], [31]. The Pimia Indian diabetic data set includes more negative than positive entries which may have introduced a bias towards negative predictions in this thesis. Evidence confirming this may be that all models predicted more entries as non-diabetic (negative) than were actually in the training data set in order to maximize its accuracy. Additionally, the models had a recall between 0.71 and 0.58 (Table 5.4). This shows a shift to negative predictions. Thus, the accuracy fails to report true positive or true negative outcomes of the models [53]. In order to meet the requirements of a good performance measurement, the ROC curve, the confusion matrix and the F1 score were evaluated as well.

The sub data sets for the teachers were randomly sampled. Arora argues that if the disjoint data sets are not representative and diverse, the individual teachers are not trained efficiently. Due to this, no consensus of the teachers is reached. To overcome this issue, Arora developed the guided PATE. The researcher clustered data points using k-Medoids clustering, where each cluster contains data points. The data points are added evenly and sequentially to each sub data set of the teachers.[11]

Another method which may lead to an even a higher accuracy was developed by Papernot et al. 2017. The researcher trained a collection of binary experts. An expert for class j was trained to predict 1 if the sample was in class j and 0 otherwise. Using binary experts improved the student's accuracy compared to the student trained on arbitrary data with the same number of label queries. However, the absolute increases in accuracy were limited between 1.5% and 2.5%.[43]

6.2 Summary of discussion

In summary, this thesis showed that the accuracy loss of the PATE model was small on a small medical data set compared to the baseline model. Moreover, a small number of teachers was used compared to the original implementation of PATE [43]. This thesis is an early stage work. Several challenges remain. However, the implementation of this thesis guarantees a strong privacy. Thus, PATE has the capability to be used in the medical context. Therefore, an introduction towards using PATE for private machine learning in health care was given.

7 Conclusion and Outlook

In this work, the original PATE was applied to a binary classification problem on a small medical data set.

The performance of 2 LR, a SVM and two NN was compared using cross validation, confusion matrix, F1 Score, AUC and ROC curve and accuracy. NN had the best performance. Therefore, PATE was implemented using NN. Furthermore, the number of teachers, the noise injection, and the accuracy of PATE were evaluated. Therefore a NN was used as a benchmark. The final PATE classifier had an accuracy of 0.76 with an privacy loss of 2.16, 5 teachers and 89 student queries.

This work demonstrated that PATE can be applied even on a small medical data set with reasonable accuracy (0.76) compared to the baseline model. Compared to the original work of Papernot et al. 2017, in this thesis a smaller number of teachers reached a strong privacy guarantee of 2.16. Additionally, the potential bottlenecks for increasing the accuracy further were highlighted. Nevertheless, PATE has the efficiency to be applied on sensitive medical learning tasks.

The results of this project may help to develop other applications of PATE on medical learning tasks. Further work should evaluate PATE using synthetic data. This may overcome the limited data problem in some medical tasks. In Addition, the number of label queries of the student to the teacher ensemble should be implemented independent from training data size of the student. Hereby, the number of label queries will be an additional hyper parameter which can be manipulated to get a stronger privacy. Moreover, an efficient implementation for use in hospitals should be developed. Lastly, the semantics of DP and PATE are complex. Their concepts should be explained such that a non–technical audience for example ethic committees and political decision makers are able to to understand DP. This may also help to get patients' trust and data.

Bibliography

[1] ARX - Data Anonymization Tool | A comprehensive software for privacy-preserving microdata publishing. https://arx.deidentifier.org/ (visited on 11/29/2020).

[2] General Data Protection Regulation (GDPR) Compliance Guidelines. https://gdpr.eu/ (visited on 11/29/2020).

[3] Genome-Wide Association Studies (GWAS). https://www.genome.gov/genetics-glossary/Genome-Wide-Association-Studies (visited on 11/29/2020).

[4] The privacy loss random variable - Ted is writing things. https://desfontain.es/privacy/privacy-loss-random-variable.html (visited on 11/29/2020).

[5] UCI Machine Learning Repository. http://archive.ics.uci.edu/ml/index.php (visited on 11/29/2020).

[6] OpenMined/PySyft. https://github.com/OpenMined/PySyft (visited on 11/29/2020), Oct. 2020.

[7] Pytorch/opacus. https://github.com/pytorch/opacus (visited on 11/29/2020), Oct. 2020.

[8] Tensorflow/privacy. https://github.com/tensorflow/privacy (visited on 11/29/2020), Oct. 2020.

[9] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang. Deep Learning with Differential Privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 308–318, Vienna Austria, Oct. 2016. ACM.

[10] M. Abadi, Ú. Erlingsson, I. Goodfellow, H. B. McMahan, I. Mironov, N. Papernot, K. Talwar, and L. Zhang. On the Protection of Private Information in Machine Learning Systems: Two Recent Approches. In *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*, pages 1–6, Aug. 2017.

[11] H. Arora. Guided PATE for Scalable Learning. page 3.

[12] F. Boenisch. Differential Privacy: General Survey and Analysis of Practicability in the Context of Machine Learning. page 107.

[13] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, Aug. 1996.

[14] M. Bun and T. Steinke. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In M. Hirt and A. Smith, editors, *Theory of Cryptography*, pages 635–658, Berlin, Heidelberg, 2016. Springer Berlin Heidelberg.

[15] N. Carlini, C. Liu, Ú. Erlingsson, J. Kos, and D. Song. The Secret Sharer: Evaluating and Testing Unintended Memorization in Neural Networks. In *28th {USENIX} Security Symposium ({USENIX} Security 19)*, pages 267–284, 2019.

[16] T. Chanyaswad, C. Liu, and P. Mittal. RON-Gauss: Enhancing Utility in Non-Interactive Private Data Release. *Proceedings on Privacy Enhancing Technologies*, 2019(1):26–46, Jan. 2019.

[17] K. Chaudhuri and C. Monteleoni. Privacy-preserving logistic regression. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 289–296. Curran Associates, Inc., 2009.

[18] K. Chaudhuri, C. Monteleoni, and A. D. Sarwate. Differentially Private Empirical Risk Minimization. *Journal of Machine Learning Research*, pages 1069–1109, Feb. 2011. Comment: 40 pages, 7 figures, accepted to the Journal of Machine Learning Research.

[19] K. Chaudhuri, A. Sarwate, and K. Sinha. Near-optimal Differentially Private Principal Components. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 989–997. Curran Associates, Inc., 2012.

[20] M. Chen, Y. Hao, K. Hwang, L. Wang, and L. Wang. Disease Prediction by Machine Learning Over Big Data From Healthcare Communities. *IEEE Access*, 5:8869–8879, 2017.

[21] T. G. Dietterich. Ensemble Methods in Machine Learning. In *Multiple Classifier Systems*, Lecture Notes in Computer Science, pages 1–15, Berlin, Heidelberg, 2000. Springer.

[22] C. Dwork. Differential Privacy. In M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, editors, *Automata, Languages and Programming*, Lecture Notes in Computer Science, pages 1–12, Berlin, Heidelberg, 2006. Springer.

[23] C. Dwork. Differential Privacy: A Survey of Results. In M. Agrawal, D. Du, Z. Duan, and A. Li, editors, *Theory and Applications of Models*

of Computation, Lecture Notes in Computer Science, pages 1–19, Berlin, Heidelberg, 2008. Springer.

[24] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor. Our Data, Ourselves: Privacy Via Distributed Noise Generation. In S. Vaudenay, editor, *Advances in Cryptology - EUROCRYPT 2006*, volume 4004, pages 486–503. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[25] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In S. Halevi and T. Rabin, editors, *Theory of Cryptography*, pages 265–284, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[26] C. Dwork and A. Roth. The Algorithmic Foundations of Differential Privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3-4):211–407, 2013.

[27] C. Dwork, G. N. Rothblum, and S. Vadhan. Boosting and Differential Privacy. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 51–60, Oct. 2010.

[28] J. Eicher, R. Bild, H. Spengler, K. A. Kuhn, and F. Prasser. A comprehensive tool for creating and evaluating privacy-preserving biomedical prediction models. *BMC medical informatics and decision making*, 20(1):29, Nov. 2020.

[29] M. Fredrikson, S. Jha, and T. 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*, pages 1322–1333, Denver, Colorado, USA, 2015. ACM Press.

[30] M. Fredrikson, E. Lantz, S. Jha, S. Lin, D. Page, and T. Ristenpart. Privacy in Pharmacogenetics: An End-to-End Case Study of Personalized Warfarin Dosing. In *23rd {USENIX} Security Symposium ({USENIX} Security 14)*, pages 17–32, 2014.

[31] V. Ganganwar. An overview of classification algorithms for imbalanced datasets. *International Journal of Emerging Technology and Advanced Engineering*, 2:42–47, Jan. 2012.

[32] J. Hamm, P. Cao, and M. Belkin. Learning Privately from Multiparty Data. *International Conference on Machine Learning, pp.555-563, 2016*, page 9, 2016.

[33] A. Honkela, M. Das, A. Nieminen, O. Dikmen, and S. Kaski. Efficient differentially private learning improves drug sensitivity prediction. *Biology Direct*, 13(1):1, June 2018.

[34] L. Hood and S. H. Friend. Predictive, personalized, preventive, participatory (P4) cancer medicine. *Nature Reviews Clinical Oncology*, 8(3):184–187, Mar. 2011.

[35] J. Li, J.-J. Yang, Y. Zhao, B. Liu, M. Zhou, J. Bi, and Q. Wang. Enforcing Differential Privacy for Shared Collaborative Filtering. *IEEE Access*, 5:35–49, 2017.

[36] N. Li, T. Li, and S. Venkatasubramanian. T-Closeness: Privacy Beyond k-Anonymity and l-Diversity. In *2007 IEEE 23rd International Conference on Data Engineering*, pages 106–115, Apr. 2007.

[37] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam. L-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data*, 1(1):3–es, Mar. 2007.

[38] M. A. Mazurowski, P. A. Habas, J. M. Zurada, J. Y. Lo, J. A. Baker, and G. D. Tourassi. Training neural network classifiers for medical decision making: The effects of imbalanced datasets on classification performance. *Neural Networks*, 21(2):427–436, Mar. 2008.

[39] H. B. McMahan, G. Andrew, U. Erlingsson, S. Chien, I. Mironov, N. Papernot, and P. Kairouz. A General Approach to Adding Differential Privacy to Iterative Training Procedures. *arXiv:1812.06210*, Mar. 2019. Comment: Presented at NeurIPS 2018 workshop on Privacy Preserving Machine Learning; Companion paper to TensorFlow Privacy OSS Library.

[40] I. Mironov. Rényi Differential Privacy. In *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*, pages 263–275, Aug. 2017.

[41] K. Nissim, S. Raskhodnikova, and A. Smith. Smooth sensitivity and sampling in private data analysis. In *Proceedings of the Thirty-Ninth Annual ACM Symposium on Theory of Computing*, STOC '07, pages 75–84, San Diego, California, USA, June 2007. Association for Computing Machinery.

[42] S. J. Oh, B. Schiele, and M. Fritz. Towards Reverse-Engineering Black-Box Neural Networks. In W. Samek, G. Montavon, A. Vedaldi, L. K. Hansen, and K.-R. Müller, editors, *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, Lecture Notes in Computer Science, pages 121–144. Springer International Publishing, Cham, 2019.

[43] N. Papernot, M. Abadi, Ú. Erlingsson, I. Goodfellow, and K. Talwar. Semi-supervised Knowledge Transfer for Deep Learning from Private Training Data. *arXiv:1610.05755 [cs, stat]*, Mar. 2017. Comment: Accepted to ICLR 17 as an oral.

[44] N. Papernot, S. Song, I. Mironov, A. Raghunathan, K. Talwar, and Ú. Erlingsson. Scalable Private Learning with PATE. *arXiv:1802.08908 [cs, stat]*, Feb. 2018. Comment: Published as a conference paper at ICLR 2018.

[45] A. Papoulis and S. U. Pillai. *Probability, Random Variables, and Stochastic Processes.* McGraw-Hill, Boston, 4th ed edition, 2002.

[46] M. Pathak, S. Rane, and B. Raj. Multiparty Differential Privacy via Aggregation of Locally Trained Classifiers. In J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 1876–1884. Curran Associates, Inc., 2010.

[47] J. L. Raisaro, n. Gwangbae Choi, S. Pradervand, R. Colsenet, N. Jacquemont, N. Rosat, V. Mooser, and J.-P. Hubaux. Protecting Privacy and Security of Genomic Data in i2b2 with Homomorphic Encryption and Differential Privacy. *IEEE/ACM transactions on computational biology and bioinformatics*, 15(5):1413–1426, 2018 Sep-Oct.

[48] X. Ren, C.-M. Yu, W. Yu, S. Yang, X. Yang, J. A. McCann, and P. S. Yu. LoPub : High-Dimensional Crowdsourced Data Publication With Local Differential Privacy. *IEEE Transactions on Information Forensics and Security*, 13(9):2151–2166, Sept. 2018.

[49] R. Shokri, M. Stronati, C. Song, and V. Shmatikov. Membership Inference Attacks Against Machine Learning Models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18, San Jose, CA, USA, May 2017.

[50] S. Simmons, B. Berger, and C. Sahinalp. Protecting Genomic Data Privacy with Probabilistic Modeling. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, 24:403–414, 2019.

[51] L. Sweeney. K-ANONYMITY: A MODEL FOR PROTECTING PRIVACY. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, Oct. 2002.

[52] F. Tramèr, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart. Stealing Machine Learning Models via Prediction APIs. In *25th {USENIX} Security Symposium ({USENIX} Security 16)*, pages 601–618, 2016.

[53] M. F. Uddin. Addressing Accuracy Paradox Using Enhanced Weighted Performance Metric in Machine Learning. In *2019 Sixth HCT Information Technology Trends (ITT)*, pages 319–324, Nov. 2019.

[54] C. Uhlerop, A. Slavković, and S. E. Fienberg. Privacy-Preserving Data Sharing for Genome-Wide Association Studies. *The Journal of privacy and confidentiality*, 5(1):137–166, 2013.

[55] B. Wang and N. Z. Gong. Stealing Hyperparameters in Machine Learning. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 36–52, May 2018.

[56] L. Wang, J. Zheng, Y. Cao, and H. Wang. Enhance PATE on Complex Tasks With Knowledge Transferred From Non-Private Data. *IEEE Access*, 7:50081–50094, 2019.

[57] A. Wood, M. Altman, A. Bembenek, M. Bun, M. Gaboardi, J. Honaker, K. Nissim, D. O'Brien, T. Steinke, and S. Vadhan. Differential Privacy: A Primer for a Non-Technical Audience. *SSRN Electronic Journal*, pages 1–69, 2018.

[58] C. Xu, J. Ren, Y. Zhang, Z. Qin, and K. Ren. DPPro: Differentially Private High-Dimensional Data Release via Random Projection. *IEEE Transactions on Information Forensics and Security*, 12(12):3081–3093, Dec. 2017.

[59] 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)*, pages 268–282, July 2018.

[60] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? *arXiv:1411.1792*, pages 3320–3328, Nov. 2014. Comment: To appear in Advances in Neural Information Processing Systems 27 (NIPS 2014).

[61] F. Yu, S. E. Fienberg, A. B. Slavković, and C. Uhler. Scalable privacy-preserving data sharing methodology for genome-wide association studies. *Journal of Biomedical Informatics*, 50:133–141, Aug. 2014.

[62] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. Understanding deep learning requires rethinking generalization. *arXiv:1611.03530*, Feb. 2017. Comment: Published in ICLR 2017.

[63] J. Zhang, X. Xiao, Y. Yang, Z. Zhang, and M. Winslett. PrivGene: Differentially private model fitting using genetic algorithms. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, SIGMOD '13, pages 665–676, New York, New York, USA, June 2013. Association for Computing Machinery.

[64] Z. Zhao, N. Papernot, S. Singh, N. Polyzotis, and A. Odena. Improving Differentially Private Models with Active Learning. *arXiv:1910.01177*, Oct. 2019.