Rheinisch-Westfälische Technische Hochschule Aachen
Informatik 5, Information Systems
Prof. Dr. Stefan Decker

Master Thesis Proposal

Privacy Protection within Machine Learning Models trained on Distributed Data

Carsten Stoffels

November 14, 2018

Advisor	Benjamin Heitmann, Ph.D.
Supervisor	Prof. Dr. Stefan Decker

Contents

1 Introduction

1.1 Motivation

Throughout the growth of modern communication and information systems more and more data is produced. So much (roughly about 1 million exabytes[1]) that the incurred „Big Data“ can not be tracked or even analyzed by only humans. The need for automated analysis and usage gets more and more important in almost any data producing area. The former breakthrough and widely usage of machine learning approaches deal with some problems of data analysis and predictions for new unseen data. Nevertheless the good results produced by machine learning algorithms sometimes analyze and represent information they shouldn't include or reveal. Mostly this is the case when private data, one doesn't want to share with the rest of the world, is involved. For instance medical data, user information or business secrets are typical areas of private data. But just because the data should be kept secret, an analysis and usage of it shouldn't be denied, and can have a huge impact taking diagnose prediction on medical data as an example.

The main issue between privacy and machine learning is their contradictary approach to data. While privacy wants to protect data and wants to keep as much as possible hidden, machine learning tries the exact opposite: Analyze data, reveal its structure and use the gained knowledge in order to apply it in the future.

The motivation of this thesis takes this contradiction into account and tries to make a cooperation of machine learning and privacy happen. A solution where both parties are somehow satisfied would lead to a compromise and allow the usage of machine learning in fields where private data is involved.

1.2 Thesis Goal

The goal of the thesis is it to analyze the possibilites of machine learning in a private context. Focussing the structure of anonymized information, private machine learning models should be trained while information about individuals is protected. In the context of machine learning, the major challenge with regards to privacy, is prevention of information leakage about individuals on which publicly released machine learning model data is based. The training data itself is distributed along several participating parties which want to coorperate in order to gain a more powerful machine learning model. The storage and contribution of locally stored data of the parties should also protect against the leakage of private information. To guarantee a good balance between privacy protection and performance decrease an evaluation method taking both into account is needed to analyze the usefulness of the approach.

1.3 Outline

The following part of the proposal is structured as follows: Chapter 2 deals with the background and related work that has been done according to machine learning, privacy and the combination of both. After the overview Chapter 3 defines the setting and usecase of the thesis, mainly focusing a usefull context of the thesis goal. Building on top of that Chapter 4 describes how to engage the usecase of Chapter 3 and describes the realisation strategy, which is then concretized in Chapter 5. For a valuation of the thesis Chapter 6 describes the possible evaluation measurements and strategies. Chapter 7 concludes the proposal with an estimated project plan to manage and organize the work.

[1] Study of the journal Supercomputing Frontiers and Innovations

2 Background and Related Work

2.1 Privacy

Before tackling the „problems“ of privacy, one should define the term of privacy. Taking the definition of a dictionary privacy is „the state of beeing free from unwanted or undue intrusion or disturbance in one's private life of affairs“ or simply the „the freedom to be let alone“. So basically one should have the choice to interact with the outerworld and no not allowed interaction should be possible. Taking this „hard“ definition, it quickly becomes clear that guaranteeing privacy for everyone will be quite hard in the daily life. Taking tasks like real world shopping as an example, there would be a lot of overhead with respect to privacy. As a result, lots of work has to be put into accomplishing any persons privacy wish. Nevertheless, in the real world a situation is more or less completed when the shopping is done. The privacy can be restored to a certain degree after an interaction, if a person wants it that way.
Transferring the principale to data science and the storage of data, the definition can be applied in a similar manner. Instead of the interaction and disturbance itsself, the privacy can be defined on the produced data of a person. It should be choosable how data is treated by others and the control should stay by the producer. Nevertheless, using data analytics brings huge advantages on using the produced data. For example to improve research or user experience. Ignoring ethnic discussions about personalized information bubbles and other problems coming with personalization, data science on user generated data has a huge potential. But the problem of privacy remains. Since everybody has a choice on how his data is released, why would someone agree to be in a medical dataset revealing a potential illnesses. The cause could be disadvantes in real life situations like job applications or the social standing. A solution combining the usage of data without harming the producer should therefore be highly desirable goal.

2.1.1 Privacy Metrics

Privacy is a somehow intangible concept and everyone has a kind of his own definition of what privacy means to him or her. This makes it quite hard to measure privacy. Also privacy shares the same issues with security: How much protection is enough and what about possible unknown threats? Nevertheless some metrics and heuristics have been published which at least give a measurable hint on how private the data actually is.

k-Anonymity Published by L. Sweeny in 2002 k-anonymity [23] was introduced as privacy protecting model. The consideration that security isn't equal to privacy caused by the sometimes wanted access to data, leaded to the introduction of Quasi-Identifiern (QIDs). Basically QIDs are a set of Attributes $A_1, ..., A_n$ that can be used to describe a person in a unique way, making it identifiable within the dataset.

Definition 1. *(Quasi-identifier)*
Given a population of entitites U, an entity-specific table $T(A_1, ..., A_n), f_c : U \rightarrow T$ and $f_g : T \rightarrow U'$,where $U \subset U'$. A quasi-identifier of T, written Q_T, is a set of attributes $\{A_i, ..., A_j\} \subset \{A_1, ..., A_n\}$ where: $\exists p_i \in U$ such that $f_g(f_c(p_i)[Q_T]) = p_i$.

Based on the QIDs k-anonymity looks at combinations of them and is only given if each set of QIDs occurs at least k times. In other words: Each entry within the dataset can not be distinguished from at least $k-1$ other entries. Figure 1 gives an example for 2-anonymity.

Definition 2. *(k-anonymity)*
Let $RT(A_1, ..., A_n)$ be a table and QI_{RT} be the quasi-identifier associated with it. Then RT fullfills k-anonymity if and only if each sequence of values in $RT[QI_{RT}]$ appears with at least k occurrences $RT[QI_{RT}]$.

forename	age	gender
Lisa	20-25	female
Anna	20-25	female
Peter	30-35	male
Klaus	30-35	male
Gustav	18-19	male
Felix	18-19	male

(a) Not 2-Anonym

forename	age	gender
Lisa	20-25	female
Lisa	20-25	female
*	30-35	male
*	30-35	male
Felix	18-19	male
Felix	18-19	male

(b) 2-Anonym

Figure 1: Example for 2-Anonymity

l-Diversity Taking the definitions fron [18] l-diversity focuses more on the sensitive values stored in tabular data. K-anonymity measures the re-identification of persons in the datasets, l-diversity builds up on the existing equivalance classes exisiting in k-anonymity. The main purpuse is the consideration: If a person can not be distinguished from other persons, its information is still revealed when each entry of the persons store the same or similiar sensitive information. For instance, a dataset in which a group of persons has a heart disease an malicious party can not be identify a specific person, but still knows it has a heart disease. This leads to the following definition of the l-diversity-principle.

Definition 3. *(l-diversity-principle)*
An equivalence class is said to have l-diversity if there are at least l "well-represented" values for the sensitive attribute. A table is said to have l-diversity if every equivalence class of the table has l-diversity

This well-representation can be measured in different ways, calling for different versions of l-diversity. Two of more famous definitions using distinct values and a entropy measurement.

Definition 4. *(Distinct l-Diversity) The simplest definition ensures that at least l distinct values for the sensitive field occure in each equivalence class*

Definition 5. *(Entropy l-Diversity) A table is entropy l-diverse if for every equivalence class eq and the set S of possible values:*

$$- \textstyle\sum_{s \in S} p_{(eq,s)} log(p_{(eq,s)}) \geq \log(l)$$

ϵ-Differential Privacy The main goal of differential privacy applied on data is it to not allow any "affects, adversely or otherwise, by allowing your data to be used in any study or analysis, no matter what other studies, data sets, or information sources, are available" [5]. According to definition 6 ϵ-differential privacy doesn't change the data itself, but adds obscurity or noise to the release mechanism. A smaller value for ϵ concludes a higher privacy guarantee according definition 6. In other words ϵ describes how likely an entry x of the dataset can be interpreted as another entry y, meaning if the output can be the same they can not be distinguished from another, leading to privacy. The parameter δ adds an addition fixed offset to this difference.

Definition 6 (Differential Privacy)**.** *A randomized algorithm M with domain $N^{|X|}$ is (ϵ, δ)-differntially private if for all $S \subseteq Range(M)$ and $\forall x, y \in N^{|X|}$ such that $\|x - y\|_1 \leq 1$:*

$$Pr[M(x) \in S] \leq exp(\epsilon) Pr[M(y) \in S] + \delta$$

Laplace Mechanism Differential privacy is just a measure metric and not an algorithm that can be used to describe the indistuingishability between outcomes of different entries. Therefore a procedure on how to achieve differential privacy is missing leading to the introduction of noise. Noise should be random in non-deterministic, because otherwise the same outcome for the same dataset would leak information. To sample such a random noise the laplace-distribution (shown in fig. 2) is often used [14]. Compared to other famous distributions like the gaussian distribution, laplace provides a strong peak around a specific value. This fact leads to a better remaining accuracy of the values, because the noisy outcome stays in a certain „small“ range around the original one. It remains to include laplacian noise within algorithms to obtain differential privacy, but this inclusion differs from problem to problem. This caused by the later elimination of the noise. Taking the task of determing the average over a huge dataset as an example, the noise sampling can be removed late on, because the expected value is known. The difficulty of including and removing of noise becomes more and more difficult with increasing complexity of the problem.

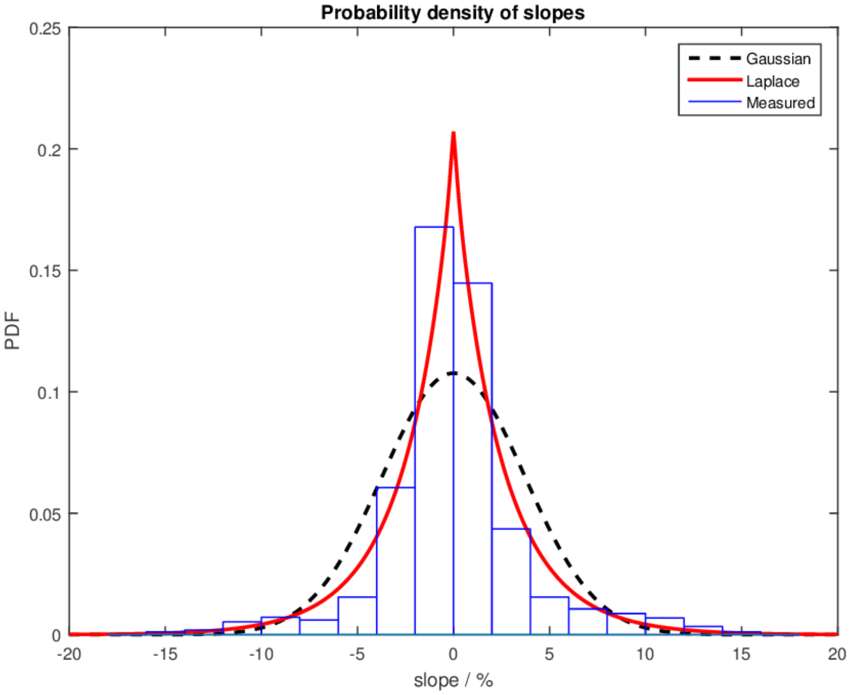


Figure 2: Laplace vs. Gaussian
Source: Laplace distribution models for road topography and roughness [16]

2.2 Machine Learning

Looking at numbers like 2.5 quintillion generated bytes everyday [2], it becomes quickly clear that a manual analysis of the so called „ Big Data “ seems to be impossible. A call for automated data analysis methods brings in machine learning.

Main goal of machine learning is it to recognize patterns within given data to predict future data or to perform other kinds of decision making under uncertainty [19]. Machine learning can be split into two different main types, called *supervised learning(predictive)* and *unsupervised learning(descriptive)*. A good general explanation is given by [12] and [3].

[2] https://www.domo.com/learn/data-never-sleeps-5

Supervised Learning The problem description for supervised learning is to learn a mapping from input data x to output data y given a labeled set of input-output pairs $D = (x_i, y_i)_{i=1}^{N}$ called *training set*, where N denotes the numbers of training examples. The input data x_i can be defined a D-dimensional feature vectors. Every entry of this vector represents a different kind of information, influencing the mapping. An optimal learned mapping would be able to map each input vector to a correct output label, no matter if the entry was in the training set or not. Supervised learning is often used for classification or regression tasks shown in figure 3.

Unsupervised Learning Unsupervised Learning is, different from supervised learning, performed on unlabeled data. The training procedure therefore doesn't have any insides, like correct classifications, into the data. Due to the abstinence of feedback, unsupervised learning can only analyze the given structure of the data. Typical applications are clustering, density estimations, or anomaly detection.

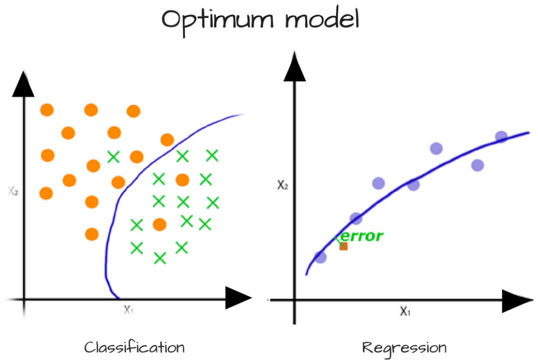
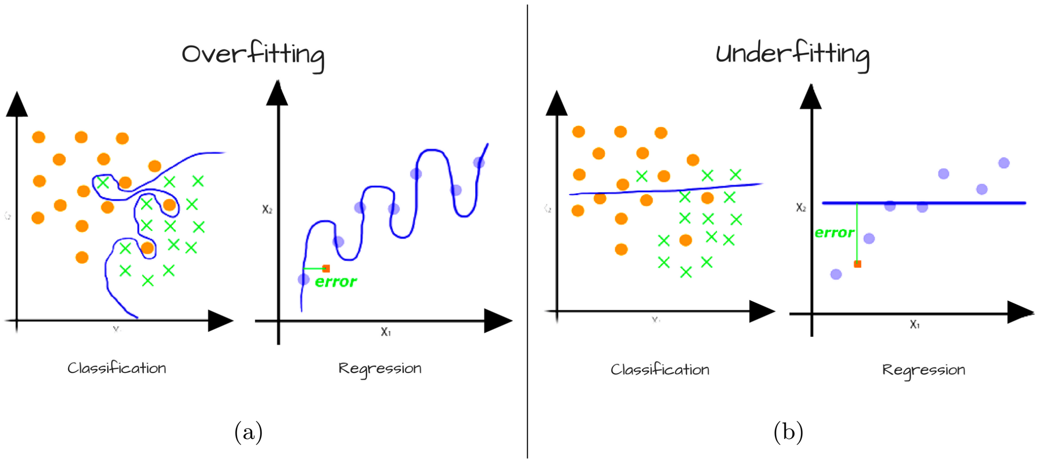


Figure 3: Classification vs. Regression
Source: https://medium.freecodecamp.org/using-machine-learning-to
-predict-the-quality-of-wines-9e2e13d7480d

Challenges within Supervised Learning Detached from the machine learning algorithm itself, the outcoming model has to deal with a huge tradeoff between overfitting and underfitting shown in figure 4. A model is underfitted if the performance on the given task is bad, meaning it produces a huge error rate. Learning characteristics of the given input data wasn't really successfull. In direct contrast an overfitted model has a „to good to be true“ performance. The data structure was learnt very well and can easily be reproduced by the model, so good that the model isn't able to make more generalized decisions anymore. This leads to model performing the given task with acceptable performance only on the training dataset. New unseen datasets with slightly different characteristics can be interpreted in a completly wrong way by the model.
Therefore supervised machine learning models always have to be balanced out between underfitting and overfitting to allow a good performance, but still let room for more generalized decisions.

2.2.1 Neural Networks

A neural network is a machine learning model often used for classification or regression problems. There exist several variations suitable for different tasks, differing in the internal structure. The foundation for modern state of the art neural network structures was made



(a)

(b)

Figure 4: Overfitting vs Underfitting
Source: https://medium.freecodecamp.org/using-machine-learning-to-predict-the-quality-of-wines-9e2e13d7480d

by Rosenblatt in 1960 [21], who invented the perceptron. It can be seen as binary classifier using a non-linear neuron taking a weighted input, an activation level and provides a weighted (classifying) output. The weights and the activation of the perceptron are learned by performing a task and gaining feedback over an error function, which is used to update the specific values. The standard perceptron is shown in figure 5, where the learning of $y(x)$ is understood as determing the weights of w. As a training process for a single training routine, the weight updates can be performed using training samples for the binary classification task according the simple rules: 1. If the output of the perceptron is correct, don't change anything. 2. If the output is incorrect and a one, substract the input vector from the weights. 3. If the output is incorrect and a zero, add the input vector to the weigts. This procedure is guaranteed to converge to a correct solution if such a solution exists.

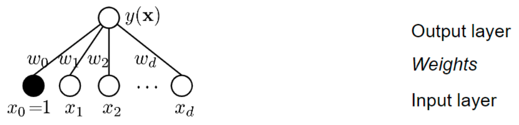


Figure 5: StandardPerceptron

Modern neural network structures are build up using many perceptrons for the modelling. So called, mutlilayer perceptrons or feedforward neural networks (shown in figure **??**) [5] consist of one input layer, one output layer and several hidden layers in between. As the standard perceptron a neural network tries to approximate a (classification) function $y = f^*(x)$, mapping an input x to a category y. In a supervised learning, training samples are fed into the network, while the weights are updated using error functions like cross-entropy [4] and squared loss [17] combined with the backpropagation algorithm [24]. The idea is to propagate the error through the network and update the weights along the gradients of the error function to minimize it.

2.3 Private Machine Learning

Machine learning in a private manner becomes more and more relevant caused by the increased usage of ML in real-life scenarios. Within the fast improvements made in general

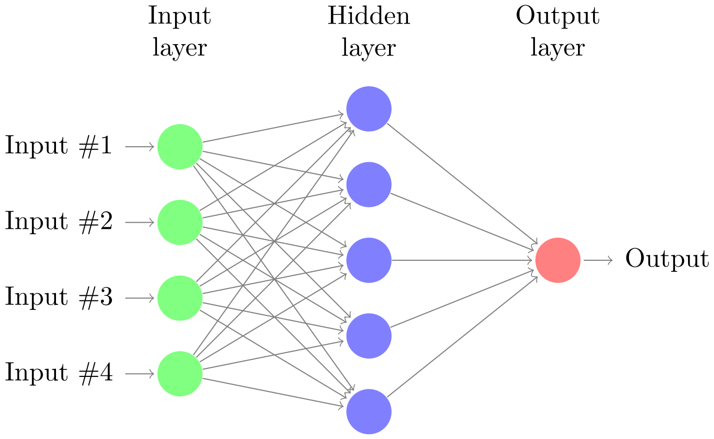


Figure 6: General Neural Network Structure

machine learning approaches, it is quite challenging to keep up with projects and publications considering privacy apsects. Nevertheless successfull developments have been made, taking [1], [6] and [15] as an example. This section should shortly consider some of the more relevant publications for this thesis.

2.3.1 Private Aggregation of Teacher Ensembles

The *Private Aggregation of Teacher Ensembles* (PATE) [20] approach addresses the problem of machine learning applications, storing sensitive information within the model itself. Therefore it tries to add privacy guarntees using ϵ-differential privacy 2.1.1 during the training process in a general applicable way. The major advantage compared to e.g. differential private stochastic gradient descent described in 2.3.2 is the adaptivity to arbitrary machine learning algorithms, caused by a voting procedure independent of the model itself.
In 7 the overview of the PATE approach is described. Given a dataset containing sensitive information, the dataset is split into n smaller datasets. The machine learning procedure of the users choice is performed on these subsets to obtain a model for each. A model can then be referred as teacher, since they are used to create a new aggregated teacher. Until now, no privacy is included during the process, which means each teacher still may contain sensitive information. Like in 2.1.1 already mentioned somehow (laplacian) noise has to be added in order to blur the origin of a specific information. This is done in the next step by creating the aggregated teacher.
To aggregate a teacher containing information about the complete dataset the predcition is done using formula 1, which takes the label with the maximum votes of each teacher and adds the laplacian distribution to each class. The distribution is centered around 0 and has the scale $\frac{1}{\gamma}$, where γ denotes the privacy parameter provided by ϵ-differential privacy. Nevertheless taking the ensemble isn't enough, since an attacker may have access to model parameters. Also with increasing predictions made, the required noise level increases, too. To fix this problem, another model is trained called the student. To train this student model another dataset, different from the data used to train the teaches is used. This dataset should be unlabeled and publicly available. If no such datasets exist it should at least be easy to construct. The labelling of this new dataset is done by the private teacher aggregation, which is basically the previously described voting procedure including noise. The new labels already include privacy and can therefore be called private labels. Feeding this new training data in the general training procedure results in the rleased model, called student. The student only saw private labels and provides because of that privacy for the

unseen sensitve information used for training the teacher models.

$$f(x) = arg\,\underset{j}{max}\{n_j(\vec{x}) + Lap(\frac{1}{\gamma})\} \quad (1)$$

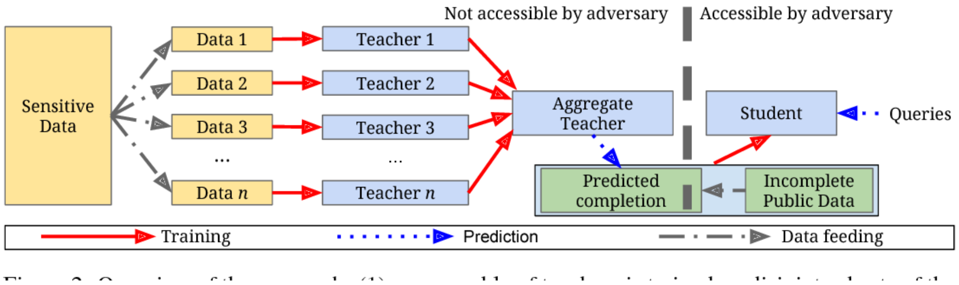


Figure 7: Private Aggregation Teacher Ensamble (PATE)
Source: Semi-Supervised Knowledge Transfer For Deep Learning From Private Training Data [20]

2.3.2 Differential Private Stochastic Gradient Descent

Different to PATE, Wu, Xi and Kumar used a different strategy in their differential private version of the stochastic gradient descent (DPSGD) [25]. [20] works detached from the learning procedure, meaning known non-private algorithms can be used for the training. In contrast DPSGD uses the output pertubation method [7] , a basic paradigm to achieve differential privacy, within the training procedure. This limits the approach to applications using SGD and can not be generalized. Nevertheless, the „simplicity“ of running normal SGD for a constant number of passes and then including random noise to the resulting output already enables a fast achievement on ϵ-differential privacy, even if small noise is included. Besides that they also provided some optimizations according convex and strongly convex models. Since they proofed the privacy guarantees provided by the noise factor they evaluated the accuracy on different datasets. The performance already came close to the baseline model at a choosen $\epsilon = 1$, but drops rapidly afterwards. In [25] a comparision is made between two other approaches for private stochastic gradient descent. One approach by Song, Chaudhuri and Sarwate [22] also uses differential privacy and adds noise dircetly into the update equation of SGD. The second one from Bassily, Smith and Thakurta[2] combined their approach on achieving differential privacy with optimizing non-smooth loss functions. [25] showed in experiments , that their approach achieved a better accuracy and an easier embedding into the training method.

2.3.3 Anonymized Representations with Adversarial Neural Networks

Another approach on anonymized representations using adversarial neural networks [13] and the domain adaption framework [11] was propossed by Feutry, Piantanida, Bengio, and Duhamel in [8]. An generative adversarial neural network (GAN) is used to create synthetic data, meaning its output is similiar to other known input data. For instance a GAN can be used to create pictures of handwritten digits given a input label of the specific number. To achieve this the GAN architecture, shown in figure8, consists of two main components: A generator and a discriminator.
The task of the generator is it to generate outputs, like pictures of handwritten digits, given

an arbitrary label as input. Feeding the generated output into the discriminator, the task is to determine how to good the generated output suits the label. The discriminator itself is trained on real input data (like images of handwritten digits) to be able to fullfill its purpose.

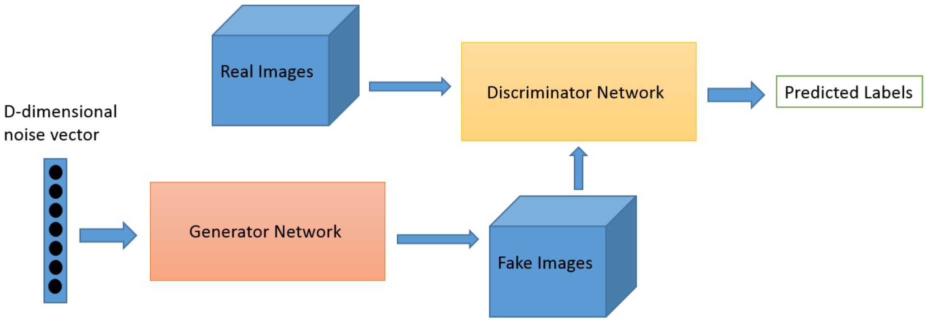


Figure 8: GAN Overall Architecture
Source: https://www.oreilly.com/learning/generative-adversarial-networks-for-beginners

The contribution of [8] besides a series of mathematical results, is a training procedure, which is shown in 9, differing from [11] in two main points. First they use an adversarial network to classify an given input vector to a specific person identity. These identities can be seen as a mapping that should be protected since it enables the linking of private information to persons. The second contribution is a more robust training by introducing a lower bound for the cross-entropy-loss. The reason for this cross-entropy of the private-labels-predictor, which can be abitrarly bad (leading to very large gradients). Actually, there is no need for it to be higher than a random guessing predictor.
The overall structure and training procedure uses three components. The first part deals with the encoding of a private input into an anonymized representation, which is then split upt into one regular branch and one private branch. The regular branch is used to keep utility within the representation or in other words: It tries to perform a specified task as good as possible. The private branch on the other hand tries to retrieve private data from the anonymized representation in order to gain a somewhat good trade-off between utility and privacy. The training procedure, also described in [8], is performed as follow:

1. Pre-train encoder and regular label predictor using the regular cross-entropy as loss function.
2. Freeze encoder and pre-train private label predictor, also using cross-entropy as loss function
3. Adversial Training alternating between label predictors training and encoder training
   (a) Sample N training examples and update both branch predictors
   (b) Sample N training examples and update encoder to minimize adversarial objective

2.3.4 Model Inversion Attack

As already mentioned in section 2.1.1 measuring privacy is a difficult task. In order to show the information leakage Fredrikson, Jha and Ristenpart developed two attacks to extract information of a given machine learning model trained on sensitive data. Their first approach published in 2014 [10], dealt with a regression on personalized warfarin dosing. The

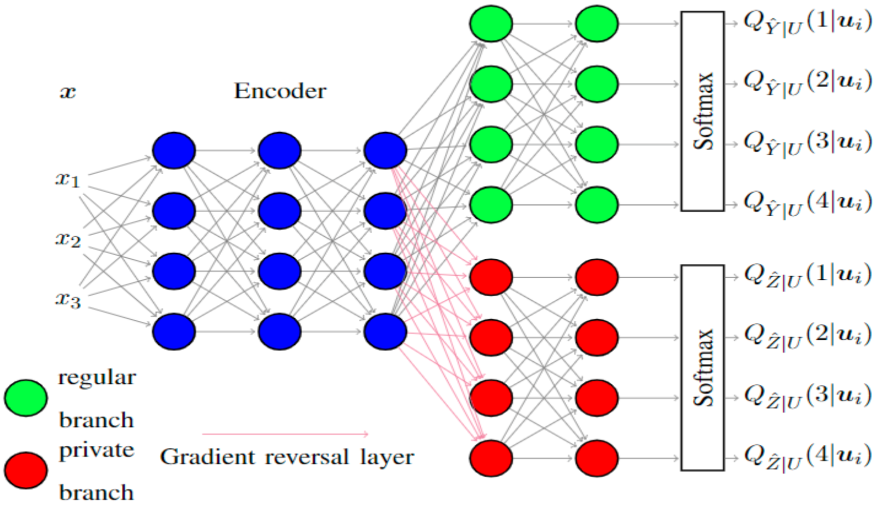

$\boldsymbol{x}$
Encoder
x_1
x_2
x_3
Softmax
$Q_{\hat{Y}|U}(1|\boldsymbol{u}_i)$
$Q_{\hat{Y}|U}(2|\boldsymbol{u}_i)$
$Q_{\hat{Y}|U}(3|\boldsymbol{u}_i)$
$Q_{\hat{Y}|U}(4|\boldsymbol{u}_i)$
Softmax
$Q_{\hat{Z}|U}(1|\boldsymbol{u}_i)$
$Q_{\hat{Z}|U}(2|\boldsymbol{u}_i)$
$Q_{\hat{Z}|U}(3|\boldsymbol{u}_i)$
$Q_{\hat{Z}|U}(4|\boldsymbol{u}_i)$
regular branch
private branch
Gradient reversal layer

Figure 9: Deep neural network architecture of [8]

assumption for the attack is having background knowledge about the target, excluding one sensitive attribute. For this attribute different possible values are considered and analyzed. The prediction for the missed value is based on the minimal error produced by the model combined with calculated marginal priors. Using a gaussian error model, attributes far off the prediction label y are penalized. The algorithm is shown in figure 1. Their simulated attack achieved an accuracy of 70-80% on guessing the correct sensitive attribute. They also applied ϵ-differential privacy within their regression algorithms to improve privacy guarantees, resulting in a performance decrease of the attack, based on ϵ.

Algorithm 1 Generic inversion attack for nominal target features

```
procedure ADVERSARY $A^f(err, p_i, x_2, ..., x_t, y)$:
    for each possible value $v$ of $x_1$ do
        $x' \leftarrow (v, x_1, ..., x_t)$
        $r_v \leftarrow err(y, f(x')) * \prod_i p_i(x_i)$
    return $\arg\max_v \boldsymbol{r_v}$
```

In 2015 they built up on their first approach, using confidence information and basic countermeasurements [9]. The attack uses revealed information on previous predictions. Knowing the complete structure of the model, meaning the attack is no longer a black-box attack, a label of a person and the corresponding activation of neurons, Fredrikson et. al. were able to reconstruct training images by propagating through the network back to the inputs. Figure 10 shows one reconstructed target person and the original training picture.

3 Use Case and Requirements

The usecase of this thesis is a bank marketing scenario collecting distributed data in different locations of the organisation. The main goal of the company is the optimization of their negotiations about term deposit contracts with customers. Therefore an analysis regarding successfull contracts in the past should be done to improve the chance of successfull contracts in the future and therefore the profit of the company. Also targeted advertisement and

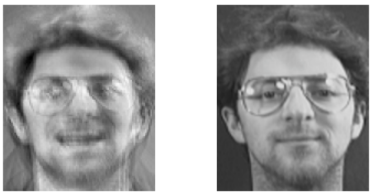


Figure 10: An image recovered using a new model in-version attack (left) and a training set image of thevictim (right). The attacker is given only the per-son's name and access to a facial recognition systemthat returns a class confidence score
Source: [9]

customer contact can enhance the customer experience in order to just get the information and consultation he needs in his situation.
The bank has several institutes to collect and analyze data locally. To gain a more powerfull prediction model the information of all institues should be combined.
Taking privacy regulations and customer protection into account, storing, sharing, and training prediction models on private customer information can't be done without including privacy enhancing technologies. On the otherhand the management don't want to forfeit to much of the accuracy and advantages of the analysis, because of the affection on profit and customer satisfaction. This leads to the following requirements to the system.

Performance of the prediction model is crucial since it becomes useless if the accuracy drops to low. It is hard find a good balance between privacy and performance, because privacy enhancement mostly involves performance decrease. The main reason for this is the information loss of identifiable information removed from the data, which surely protect indiviudals but also remove important training information.

Training Speed To determine if the developed system can be used in the real world, the training speed is important. If the training procedure takes to long the overhead would lead to a unusable model in practise, even if the performance is outstanding.

Amount of data One of the most expensive resources in training machine learning models is the training data itself. This includes raw data as well as adding labels to it. A powerfull model needing less training data is always desired since it reduces costs and time creating it. Also the creation of new data isn't

Privacy Protection Privacy protection should be included in every step of the procedure to minimize the risk of information leakage. The complete eliminiation of privacy threats isn't possible caused by the competing concepts between privacy and utility. Maximal privacy can easily be achieved by removing private information. Obviously this would also lead to zero utility, since nothing is left for the training. The main task is it to add just enough privacy for a good protection that still allows performing machine learning.

Data Distribution The process should be able to handle data distribution. This means data storage can't be seen as one partition including the whole information, but as many pieces that can not be combined without taking privacy protection into account.

4 Conceptual approach

The conceptual approach illustrated in figure 11, begins with different datasets containing sensitive information. The task performed on the datasets is a classification task. Taking standard datasets from e.g.: the uci machine learning repository, like the heart disease[3] or the bank marketing dataset[4] allow a comparison to other approaches. Also the structure of the dataset should contain different types of attributes like categorical, numerical, binary and string values, allowing a better analysis on privacy metrics and encoding of the values. Distribution of the data is simulated by splitting up the given dataset in different chunks. Manipulations like anonymization on them are made in a splitted manner such that the privacy regulartions can already be met at different storage locations of the company. As comparison for the normal usage without added privacy, a non-private machine learning model is trained. To analyze the possiblilites of adding privacy to this model, several other models are trained using different techniques to achieve privacy. As a standard machine learning model a somehow „simple“ neural network is taken without spending to much time in its optimization since the influence of the performance is measured in the first place. Optimization steps in the non-private and private versions can be considered later as a bonus. First the described PATE (section 2.3.1) approach is used to add noise to achieve ϵ-differential privacy (6). A combination of previous data manipulation and anonymization is considered and applied to both the PATE and the non-private model, to measure influence on performance and privacy. In the next step different optimization possibilities should be tested in order to increase the accuracy of the mode, focussing the embedding of privacy techniques. These could be the representation of the data, the form of the data embedded into the model, the privacy parameter ϵ used in differential privacy described in 2.1.1 or the neural network input structure for sensitve attributes.

Through blackbox and/or whitebox access to the models a simulated attack is implemented and used in order to retrieve sensitive information of the released models. The first version of the blackbox attack is similiar to the model inversion attack described in 2.3.4, taking known background knowledge as input.It should aim for sensitive attributes of persons by exploiting given background knowledge. Improvements to the attack according to performance, used background knowledge and the structure of attacked information can be made. Improvements to less needed background knowledge or better predicitions using the known structure of the datasets are planned in the next step. A possible additional whitebox attack can implemented using neural network understanding techniques. Such techniques are used to derive rules from models. The exploration and application of those techniques can lead to retrieve rules of the network revealing information about general sensitive information within the model. This whitebox attack focusses more on general information gain instead of attack individuals.

5 Realisation/ Implementation

The genereal implementation strategy is shown in figure 12. Most of the work is done in python using tensorflow as machine learning plattform combined with data science python packages like pandas.

Baseline Model As baseline machine learning model a neural network designed and implemented in tensorflow. The structure itself should have different implementations since they make affect the influence of added privacy within the model. An adaption to the input and

[3] https://archive.ics.uci.edu/ml/datasets/heart+Disease

[4] https://archive.ics.uci.edu/ml/datasets/bank+marketing

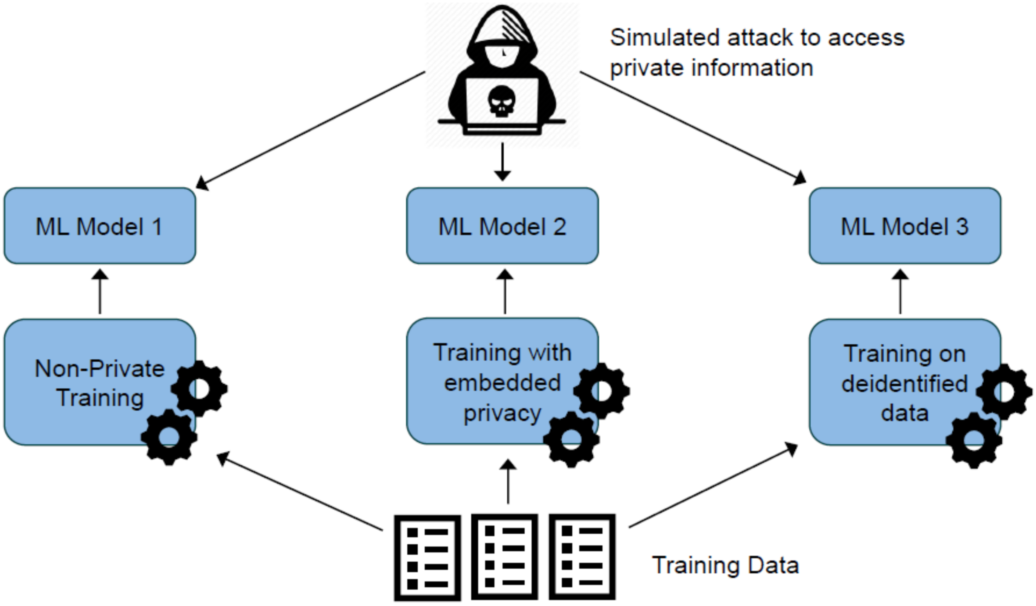

Simulated attack to access private information
ML Model 1
ML Model 2
ML Model 3
Non-Private Training
Training with embedded privacy
Training on deidentified data
Training Data

Figure 11: Conceptual Approach

output for each of the used datasets is needed to be able to compare the results later on. To allow the evaluation of the model and the built-up private models, statistics like accuracy, training-speed and amount of data should already be considered within the implementation. Tensorflows tensorboard already allows a good monitoring of the training procedure and therefore should be included.

PATE The PATE approach described in section 2.3.1 is already implemented in a github project[5] that needs to be adapted. The implementation is also done with tensorflow leading to a good embedding of the baseline model. Additionaly the data handling and writing needs to be adapted in the same way.

Data Modification The data modification should mostly transform the training and test data such that they fullfill specific privacy metrics. It would be desirbale if the implementation could also be done in python. Existing libraries like the ARX anonymization tool[6] are implemented in java, which could lead to some problems according automation or compatibility. A first hard-coded version based on the given dataset should be implemented which then can be extended to a more generic approach for arbitrary datasets.

Model Inversion Attack The model inversion attack described in section 2.3.4 is a whitebox attack on the model. Therefore it needs somehow access to the model. Since the communication structure is already given by tensorflow it makes sense to built up the reimplementation on that. On top of that extensions to whitebox attacks need to access to the model structure making it indispensable adapting them directly to the model.

[5] $https://github.com/tensorflow/models/tree/master/research/differential_{p}rivacy/pate$
[6] $https://arx.deidentifier.org/anonymization-tool/$

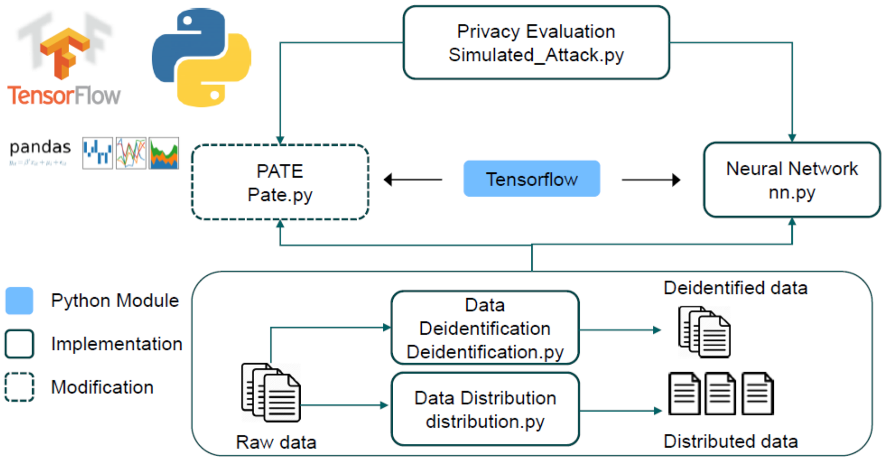


Figure 12: Implementation Structure

6 Evaluation

The evaluation of the thesis focusses on four main requirements, the runtime or training time of the implementations, characteristics of the training data, accuracy of the models and the enabeling of privacy guarantees. Also a combined view on priavcy and accuracy is given to analyze the utility vs. privacy tradeoff.

6.1 Runtime/Training Time

The training time of the models is measured related to the used hardware to allow somehow a comparison. Next to the computational power the used time is relevant for real-world application, since one could not wait several month for the training to converge.
The prediction time for the models is relevant within the simulated attack caused by the many prediction requests. The number of requests combined with the spent time is recorded to evaluate the expenditure of the attack.

6.2 Training Data

The data amount needed to achieve good performance within a machine learning model is crucial for real-life scenarios, because it is very expensive obtain. Labelling existing or collecting datasets often requires expert knowledge and time. Also some datasets consist of very rare data, even if time and money is invested. To evaluate the performance regarding the dataset size, different parts of different size should be used for the training. Also different datasets in general are taken to achieve a more generalized view of the performance and the behaviour in different domains. This is mainly done due to the fact a single dataset can contain specific characteristics, which make it hard or easy to learn.

6.3 Accuracy

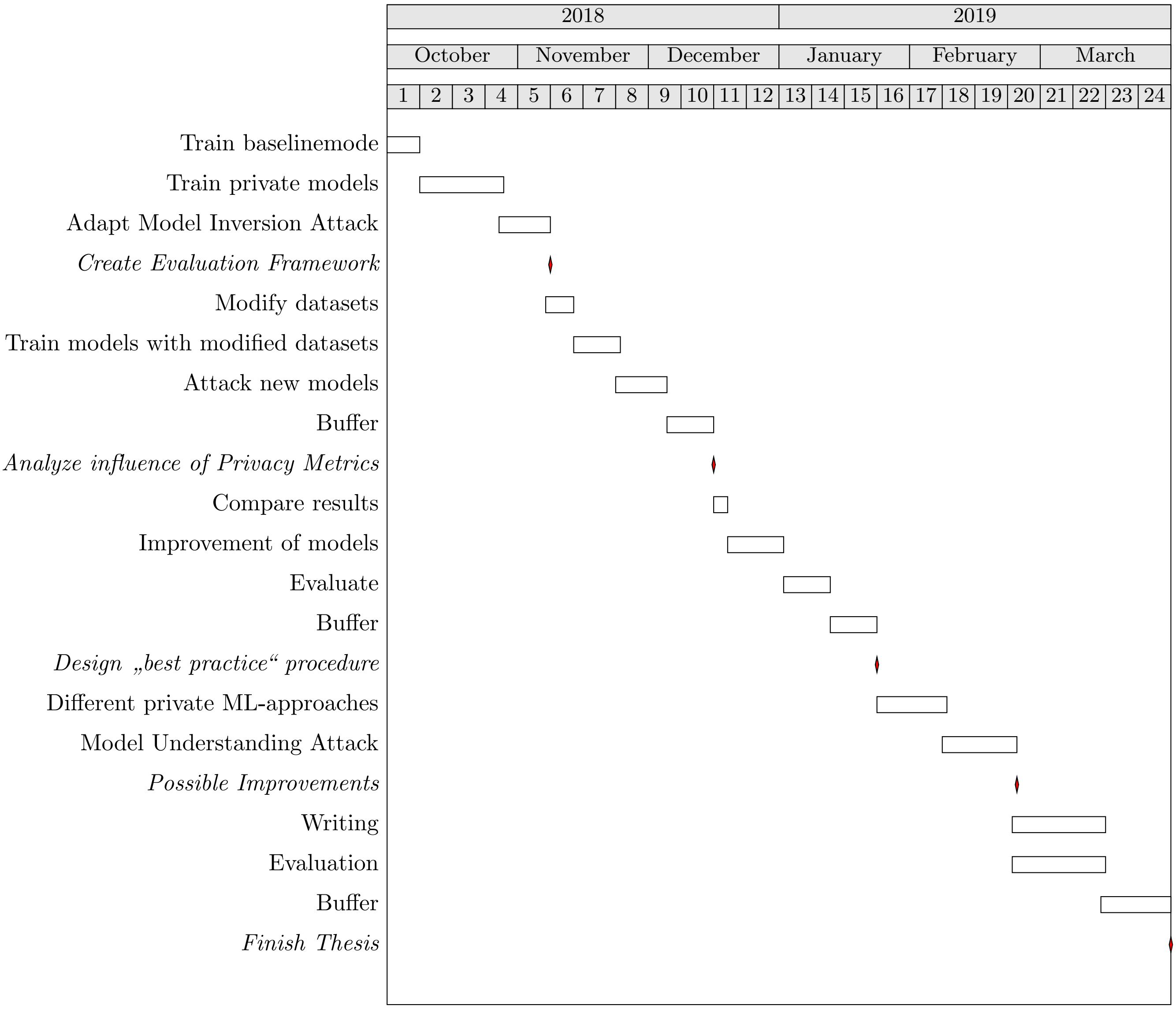
Accuracy is measured of all trained models using test sets of data. The models have to predict the known label of the test data to determine how much percent they predcit correctly. Directly from this follows the test accuracy. Next to the test accuracy direct metrics like the loss function are calculated and recorded during the training. These heuristics also contain hints about the performance and are therefore considered.

6.4 Privacy

Measurment of the privacy guarantees provided, the simulated attack is used. An accuracy calculation on how many sensitive attributes are extracted correctly is taken as comparison. On top of that, the level of data transformation and achieved privacy metrics is also compared.

7 Project plan

The project plan envisages to start with the development of the evaluation framework as first milestone. A basic routine to allow the measurement is the goal. The next three milestones focus on performance and privacy improvement and try to consider different methods and approaches for this. The evaluation is somehow included in almost every step of the project plan, since performance and attacking the model are the significant measurements. In the last part of the project the results are written down in order to finish the thesis.


2018
2019
October
November
December
January
February
March
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
Train baselinemode
Train private models
Adapt Model Inversion Attack
Create Evaluation Framework
Modify datasets
Train models with modified datasets
Attack new models
Buffer
Analyze influence of Privacy Metrics
Compare results
Improvement of models
Evaluate
Buffer
Design „best practice“ procedure
Different private ML-approaches
Model Understanding Attack
Possible Improvements
Writing
Evaluation
Buffer
Finish Thesis

References

[1] Martin Abadi et al. “Deep Learning with Differential Privacy”. In: *23rd ACM Conference on Computer and Communications Security (ACM CCS)*. 2016, pp. 308–318. URL: https://arxiv.org/abs/1607.00133.

[2] Raef Bassily, Adam Smith, and Abhradeep Thakurta. “Private empirical risk minimization: Efficient algorithms and tight error bounds”. In: *Foundations of Computer Science (FOCS), 2014 IEEE 55th Annual Symposium on*. IEEE. 2014, pp. 464–473.

[3] Christopher M Bishop. “Pattern recognition and machine learning, 2006”. In: (2012).

[4] Pieter-Tjerk De Boer et al. “A tutorial on the cross-entropy method”. In: *Annals of operations research* 134.1 (2005), pp. 19–67.

[5] Cynthia Dwork and Aaron Roth. “The Algorithmic Foundations of Differential Privacy”. In: *Foundations and Trends® in Theoretical Computer Science* 9.3–4 (2014), pp. 211–407. ISSN: 1551-305X. DOI: 10.1561/0400000042. URL: http://dx.doi.org/10.1561/0400000042.

[6] Cynthia Dwork, Guy N Rothblum, and Salil Vadhan. “Boosting and differential privacy”. In: *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*. IEEE. 2010, pp. 51–60.

[7] Cynthia Dwork et al. “Calibrating noise to sensitivity in private data analysis”. In: *Theory of cryptography conference*. Springer. 2006, pp. 265–284.

[8] Clément Feutry et al. “Learning Anonymized Representations with Adversarial Neural Networks”. In: *arXiv preprint arXiv:1802.09386* (2018).

[9] 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*. ACM. 2015, pp. 1322–1333.

[10] Matthew Fredrikson et al. “Privacy in Pharmacogenetics: An End-to-End Case Study of Personalized Warfarin Dosing.” In: *USENIX Security Symposium*. 2014, pp. 17–32.

[11] Yaroslav Ganin and Victor Lempitsky. “Unsupervised domain adaptation by backpropagation”. In: *arXiv preprint arXiv:1409.7495* (2014).

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

[13] Ian Goodfellow et al. “Generative adversarial nets”. In: *Advances in neural information processing systems*. 2014, pp. 2672–2680.

[14] Moritz Hardt, Katrina Ligett, and Frank McSherry. “A simple and practical algorithm for differentially private data release”. In: *Advances in Neural Information Processing Systems*. 2012, pp. 2339–2347.

[15] Zhanglong Ji, Zachary C Lipton, and Charles Elkan. “Differential privacy and machine learning: a survey and review”. In: *arXiv preprint arXiv:1412.7584* (2014).

[16] Pär Johannesson, Krzysztof Podgórski, and Igor Rychlik. “Laplace distribution models for road topography and roughness”. In: *International Journal of Vehicle Performance* 3.3 (2017), pp. 224–258.

[17] Wee Sun Lee, Peter L Bartlett, and Robert C Williamson. “The importance of convexity in learning with squared loss”. In: *IEEE Transactions on Information Theory* 44.5 (1998), pp. 1974–1980.

[18] Ashwin Machanavajjhala et al. “L-diversity: Privacy Beyond K-anonymity”. In: *ACM Trans. Knowl. Discov. Data* 1.1 (Mar. 2007). ISSN: 1556-4681. DOI: 10.1145/1217299.1217302. URL: http://doi.acm.org/10.1145/1217299.1217302.

[19] Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective.* The MIT Press, 2012. ISBN: 0262018020, 9780262018029.

[20] Nicolas Papernot et al. “Semi-supervised knowledge transfer for deep learning from private training data”. In: *arXiv preprint arXiv:1610.05755* (2016).

[21] Frank Rosenblatt. “Perceptron simulation experiments”. In: *Proceedings of the IRE* 48.3 (1960), pp. 301–309.

[22] Shuang Song, Kamalika Chaudhuri, and Anand D Sarwate. “Stochastic gradient descent with differentially private updates”. In: *Global Conference on Signal and Information Processing (GlobalSIP), 2013 IEEE.* IEEE. 2013, pp. 245–248.

[23] Latanya Sweeney. “k-anonymity: A model for protecting privacy”. In: *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 10.05 (2002), pp. 557–570.

[24] Paul J Werbos. “Backpropagation through time: what it does and how to do it”. In: *Proceedings of the IEEE* 78.10 (1990), pp. 1550–1560.

[25] Xi Wu et al. “Differentially private stochastic gradient descent for in-RDBMS analytics”. In: *CoRR, abs/1606.04722* (2016).