AARHUS UNIVERSITET

AARHUS UNIVERSITY

DEPARTMENT OF ENGINEERING

MASTER'S THESIS
in
COMPUTER ENGINEERING

PRIVACY PRESERVING MACHINE LEARNING IN HEALTHCARE

AUTHOR	SUPERVISOR
Lasse Lildholdt	**Christian F. Pedersen**
201507170	Aarhus University

JANUARY 5, 2021

PREFACE

This thesis is the culmination of a two-year master's degree program at Aarhus University. It is written in collaboration with the University and the AIR research group in the period from August 31, 2020 to January 5, 2021.

I would like to thank my supervisor at Aarhus University, Christian Fischer Pedersen, as well as my co-supervisor, Christian Lillelund for supporting me throughout my thesis work and for invaluable feedback of the report.

I would also like to thank Fritz Dörmann, Osvald Lorenz Nygaard Frisk and Michael Quach for participating in a study course at Aarhus University covering privacy preserving machine learning. This study course has provided much appreciated discussions on key subjects throughout the thesis period.

Lastly I would like to thank Michael Harbo, head of DigiRehab, for data delivery as well as important domain knowledge.

Lasse Lildholdt

Date

ABSTRACT

In the Danish society the number of elderly people in need of caretaking are rising. This tendency calls for new and more effective ways of operating within the healthcare sector. One of the areas that needs focus is fall prevention. In order to prevent falls it is necessary to asses the elderly citizens risk of falling. If the risk is high, new assistive devices or training programs can be initiated. Today, the fall risk assesment is a manual and ineffective procedure. The streamlining of this procedure, with use of machine learning techniques, defines the scope of this thesis.

Machine learning uses large amounts of data in the training process. In normal cases this is not a challenge but in the scope of this thesis, the trainig data is private and therefore of sensitive character. This challenges the streamlining of machine learning and yields the core challenge covered in this thesis. How can machine learning algorithm be trained on sensitive datasets?

This thesis evaluates how federated learning can be used to train machine learning algorithms on decentralized datasets. Furthermore, it explores how encryption techniques and differential privacy can be used to secure machine learning models from leaking sensitive information. Lastly, it explores how complex machine learning models, especially deep learning models, can be made explainable such that the findings from the models, easier can be communicated to the elderly citiens.

By numerous experiments it is found that federated learning does offer a method for decentralized learning, but at the cost of model performance. It is also found how both encryption techniques and differential privacy can help securing machine learning models from leakage, however also at the cost of both model performance and complexity. Lastly it is found how a method called SHAP can help calculate what is refered to as SHAP feature values, presenting the machine learning models in a much more understandable way.

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

This Master thesis arises as a part of a project named AIR or AI-Rehabilitation. The AIR project aims at, as the name states, using artificial intelligence in the rehabilitation process for elderly people in the Danish municipalities.

In the Danish society it is seen how the amount of elderly people are increasing in recent years. This means that there are more people who needs caretaking, and therefore an increasing need for funding in the caretaking area exist [33]. The increased need, is something that is being discussed at a political level in the society [31] but with the AIR project it is investigated how we can use the current funding in the most efficient way.

By exploring the use of artificial intelligence, it can be evaluated which citizens are most likely to benefit from the resources within the caretaking area. Systems are fed with information about training sessions with elderly people, which assistive devices are being used etc. The system is then calculating how the resources are best used based on this input.

Up until now, that has been the main focus with the AIR project [10]. However, since these large amounts of data are already being collected, it is now explored how the data can be used for other cases within the domain. Fall-risk assessment is one of the new areas of investigation and is the direction set for this thesis. By using artificial intelligence and the large data sets which already are being collected, how is it possible to assist the fall-risk assessment made by the caretaking staff and how can results be communicated in the best way. That is the baseline for this thesis.

Another big challenge in the AIR project, is the privacy of the majority of the data collected from the Danish citizens within the caretaking programs. The data are protected by law [38], and it is getting more and more difficult to collect and store it. These data would be available from all the different municipalities in the Danish society but due to GDPR regulations, sharing data across the country is difficult. The other main perspective of this thesis is therefore to explore how we can use artificial intelligence without compromising the privacy of the data and therefore allow sharing of sensitive data.

1.1 Background

It is important to understand the motivation behind this thesis aswell as understand important concepts within AI and data protection, in order to understand the choices made throughout the thesis.

1.1.1 Elderly people in the danish society

As the average age of the citizens in the Danish society is increasing [16], the number of citizens who needs assistive caretaking is increasing accordingly. When a person is entering the group of people who are in need of assistive caretaking, the risk of them falling in their home is also increasing.

According to the Danish website "sundhed.dk" we see that 1/3 of all people above 65 years of age have at least one incident of falling within their home each year. Of those who fall in their home at least once a year, 50 percent of them have multiple incidents each year. Furthermore, we know that only about 10 percent of the fall incidents leads to serious injuries, but these still represent up to 30 percent of injury related hospitalizations [37].

It is clear that falling among elderly people, can have severe consequences both for the citizens future health and the society in terms of economy. This conclusion provides the foundation for using artificial intelligence to help streamline fall-risk assessment in order to decrease the amount of fall related incidents.

1.1.2 Artificial intelligence and machine learning

According to Jake Frakenfield [20], the definition of artificial intelligence is "the simulation of human intelligence in machines that are programmed to think like humans and mimic their actions". One of the things we as the human species are so good at, is recognizing patterns. A simple maneuver for us, but for computers it can be a challenging task. When we are using artificial intelligence, one of the actions we want to mimic from humans, is the ability to recognize patterns in data. That is exactly what machine learning is. "Machine learning is a field of study that gives computers the ability to learn without being explicitly programmed. The aim of machine learning algorithms is to learn how to perform certain tasks by generalizing from data. Such tasks might include giving accurate predictions or finding structures in data" [12].

The input to a machine learning algorithm is usually a set of samples, each containing some set of feature values. These feature values describes the sample and therefore distinguishes it from the other samples. Samples can either be assisted by a label connecting them to a certain class or be unlabeled. If labels are present, we are talking about supervised learning and if not, we are talking about unsupervised learning.

The task for the machine learning algorithm is to locate and find patterns. Within the healthcare domain an example could be with input samples of medical data with feature values being height, weight, medication track, assistive device track etc. The goal with this type of data could be determining the risk of falling. That is, each input sample is assigned a label, stating their risk of falling. The algorithm would now look for patterns. Patterns which determines how a certain combination of features often results in certain fall risks. After the algorithm has been trained it would be able to predict the risk of falling, given a new sample with new feature values.

When evaluating the performance of an algorithm, the training error can be investigated. The training error is a measure for how well the algorithm predicts the risk of falling. That means the training samples that were used to train the model are now used for evaluation. This error is often small since the model already knows a lot about these samples. The other, and more important measure used, is the test error. The test error is the measure for how well the algorithm predicts the fall risk based on new samples, unseen to the model. That means the test error is a measure for how well the model finds general patterns in the training samples that is not biased towards already seen data.

These definitions are important, because they will be used later in the thesis when discussing accuracy. Accuracy of a model is a meassure on how low the test error is in relation to the number of samples. A certain type of accuracy metric, will be used to compare performance of different algorithms used for machine learning.

Often these models can be adjusted in many ways (hyperparameter selection using cross validation, boosting etc.) such that they perform better with a lower test error. A more in depth explanation of how machine learning algorithms work can be found in chapter 3.

1.1.3 Privacy preserving machine learning

Among the various use cases for machine learning we find many which include data which to some degree is private to the citizens. Browsing history, purchase history, marketing preferences or as in this case medical healthcare information are examples of data that is used by companies for machine learning.

However, these types of data are getting more and more protected. As of May 25th 2016, a new data protection law was entered into force for the European citizens [38]. These General Data Protection Regulations (GDPR) limits the access companies have to their data and defines a stricter way of handling personal data.

The consequence for companies using personal data is that a strategy for handling existing and future data should be defined. While more and more data are collected, less and less data are available for machine learning. That arises a huge problem for companies that rely on their machine learning algorithms.

Furthermore, as stated by Roberta Kwok in [23] companies these days consider collected knowledge as very valuable assets. Therefore, it is common sense that most companies don't want to share their information with other instances. That is once again a case stating that without privacy introduced into the world of machine learning, data might be more and more difficult to achieve.

Given these circumstances, a new type of machine learning has been investigated in recent years named privacy preserving machine learning. With privacy preserving machine learning, algorithms are using machine learning while ensuring that the information leakage of the training samples are kept to a minimum. This means it is possible to reveal patterns within datasets, while keeping data private. By using privacy preserving machine learning, the increasing amount of data collected in the societies can still be used for machine learning, even though they are of private character. There are many different approaches to privacy preserving machine learning which will be investigated further in 3.

1.1.4 Deep learning

As stated in 1.1.2, machine learning is a part of the AI concept. Deep learning is a subset of machine learning and will be used in regards to privacy preserving machine learning algorithms, which reasons this brief introduction.

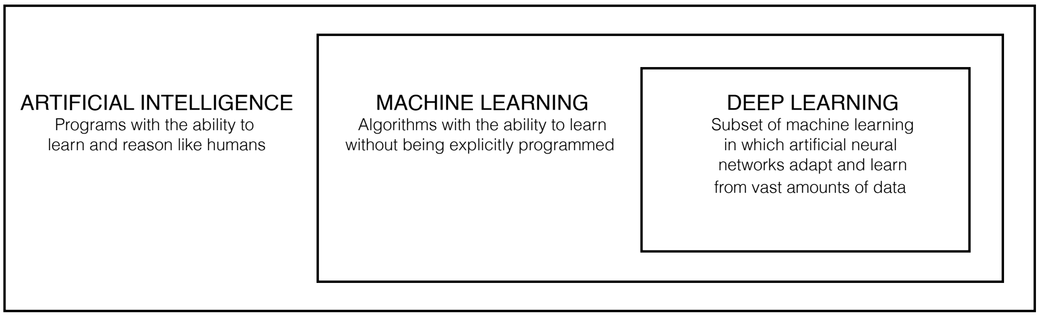
ARTIFICIAL INTELLIGENCE
Programs with the ability to learn and reason like humans
MACHINE LEARNING
Algorithms with the ability to learn without being explicitly programmed
DEEP LEARNING
Subset of machine learning in which artificial neural networks adapt and learn from vast amounts of data

Figure 1.1: Hierarchical illustration of the concepts of artificial intelligence

Deep learning models are based on neural networks which are a construction of several neurons connected in a large network. Each neuron is a computational unit which based on an input and a given weight computes a weighted sum which is then passed through an activation function. The activation function limits the range for the output of the network's layers. A simple neural network with 5 layers can be seen from 1.2.

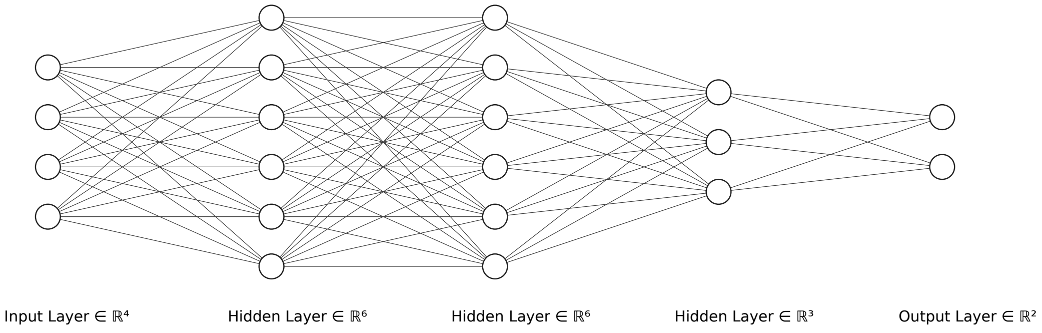

Figure 1.2: Simple illustration of an example deep learning network

Deep learning refers to more complicated neural networks with more layers, hence the name ”deep”. With deep learning, more complicated patterns in more complicated datasets can be found. However, this increased efficiency comes at the cost of computation. Large deep learning networks can often be very heavy to compute. The concept of deep learning with privacy and the theory behind it, is investigated further in 3.

1.1.5 Summary

As stated in 1.1.1 we are seeing more and more elderly people in the Danish society. The elderly citizens are spread amongst different municipalities which are all keeping track of data collected within their own municipality. That means that data needs to be shared between the different municipalities in order to train machine learning models that benefits from all the findings across the country. However, this is a problem. As stated in 1.1.3 GDPR dictates strong regulations for which data can be shared and stored and how the process should be. Because of this problem, privacy preserving machine learning is introduced.

Privacy preserving machine learning, as stated in 1.1.3, defines a way of creating machine learning models without compromising the privacy of the samples used for training the model. That means that it will be possible to use data from all the different municipalities without ever compromising the private data in the machine learning training process. If perfect privacy is guaranteed (or close to it) the data protections laws will not be violated, and a shared machine learning approach could be embedded into the Danish healthcare system assisting in the process of fall-risk assessment and fall prevention.

1.2 Thesis goals

Machine learning have been proven to be very effective in many areas, one of them being healthcare. With machine learning it is possible to recognize patterns in datasets on previous patients and thereby state important facts on new ones. With those findind on new patients, the efficiency of the healthcare system could improve significantly. However, the challenge of using machine learning within the healthcare sector, is that information found from patient journals etc. are sensitive to the patients and are therefore protected by data regulations. The regulations prohibit data scientist from sharing data between municipalities and thereby prohibit the creation of a large dataset, spanning the entire society. This leads to first goal of this thesis:

> **Goal 1:** Provide a method for sharing data between municipalities without compromising the privacy of the individual data samples

With sharing of information being a possibility, machine learning models can be build based on observations from the entire Danish society. However, these models themselves can reveal private information about the citizens, whose information are being used for training. Therefore, it is important to create machine learning models which consider the privacy leakage of the model and ensures that sensitive data cannot be revealed from the model. This leads to the second goal for the thesis:

> **Goal 2:** By using privacy preserving machine learning techniques, provide a method for ensuring that sensitive information is not leaking from the trained models.

The reasoning for using machine learning in this thesis, is to assist caretakes in the fall-risk assessment process. If fall-risk assessment can be done more efficiently anti-fall initiatives can be applied earlier and thereby prevent many of these fall accidents. This can both increase life quality for the elderly citizens and save resources to be used in the ”more likely to fall” group of citizens. This defines the third and last goal of the thesis:

> **Goal 3:** By using privacy preserving machine learning, provide an assistive tool to be used for caretakers, which explains the conclusions made from the machine learning models.

All the goals presented for the thesis evaluates different privacy or explainability initiatives. The evaluation of these initiatives is the core goal of the thesis. That is, the domain in which the initiative is tested, is of less importance. The findings from the thesis work should easily be adaptable to other domains. Because of this direction, no special requirements for the developed system is presented, since this only is developed for benchmarking the privacy initiatives.

1.3 Thesis outline

The content of each chapter, apart from this one, is described in the following.

Chapter 2: A state-of-the-art discussion of how the healthcare system is using machine learning today, and how this can be improved. In this chapter the current approach to privacy preserving is also covered.

Chapter 3: An explanation of the fundamentals of how the different approaches to privacy preserving machine learning are working as well as an in-depth overview of the theory behind the different approaches. This theoretical analysis results in different solutions to the thesis goals. In this chapter, an explanation of the different threats which conventional data centralized machine learning models are vulnerable to, will also be covered.

Chapter 4: An explanation of the data which is used in the experimental phase of the thesis. In this chapter, the origin of the data and the needed manipulation of the data is covered.

Chapter 5: Experiments are presented along with their results. Each of the results will be evaluated, along with a discussion of how they should be interpreted. Furthermore, a description of the experiment setup is presented.

Chapter 6: A discussion of how the results from the experiments affects the solutions presented in chapter 3. The use cases for which the solutions are valuable are presented.

Chapter 7: An investigation of how the limitations of the presented results can be improved, by use of new strategies. This chapter presents the groundwork for future projects continuing this one.

Chapter 8: The concluding remarks on the thesis is presented. This involves the achieved results discussed against the thesis goals as well as a perspectivation of the personal outcome from the thesis work. Lastly a listing of the contributions made from this thesis is presented.

Chapter 2

STATE OF THE ART

This chapter outlines state-of-the-art within machine learning in the health care sector, fall-risk assessment and privacy preserving initiatives. The general idea behind this thesis is that that the Danish healthcare system can be improved, by using privacy preserving machine learning (with the focus being fall-risk assessment). In order to determine if this is possible it is important to understand state-of-the-art, in order to understand how to improve.

2.1 Machine learning in healthcare

In [15] they study the current use of machine learning in healthcare. They elaborate on the fact that the healthcare sector is an area where collected data amounts, are increasing drastically over the last few years. New and more digital approaches to the interaction with the patients leads to more data collection. As previously described in chapter 1, data is the core key to developing machine learning models. Large amounts of data means that it is easier to find statistical patterns in the data, which is the goal of machine learning.

In the article [15] they mention some important facts on the healthcare system. Of the cost related to the healthcare system, 50 percent is related to only 5 percent of the people embedded in the system. That means that some individuals are very expensive in terms of treatment. These individuals are often people with chronic diseases, but by the use of machine learning, these incidents might be revealed earlier. By starting treatment at an earlier stage, life quality can increase while and expenses can be kept lower.

Additionally, the article states: ”Close to 90 percent of emergency room visits are preventable. Machine learning can be used to help diagnose and direct patients to proper treatment all while keeping costs down by keeping patients out of expensive, time intensive emergency care centers”. This reveals the huge potential for machine learning used in healthcare.

In the article ”Machine learning in Healthcare” [17], they have a more detailed description of how machine learning models are applied in modern healthcare. All the examples mentioned, are examples where machine learning models are trained on, what is referred to as EHR’s or Electronic Health Records. These are data recorded from electronical programs deployed in the healthcare systems. That mimics the data that is used for the experiments for this thesis (see chapter 4), which means that the approaches are adaptable.

With the EHR's, [17] states examples of how machine learning models are used in two different cases. The first one is in predicting the onset of diseases. By using machine learning for this purpose, diseases might be found earlier because the models are able to find patterns in the usual patients with a given disease. The other case is in streamlining of hospital operations. Machine learning models are used to find patterns in different operation strategies, and thereby define the most effective approach.

Based on the findings from [17], it can be seen how these types of data, derived from the healthcare systems, are very valuable in regards to machine learning models. With the great results presented in the article, other areas as for example streamlining of fall risk assessment, are interesting to investigate.

2.2 Fall-Risk assesment in healthcare

Fall-risk assessment is the process of evaluating what the risk of falling is for individual citizens embedded in a healthcare program. With a well performing fall-risk assessment approach, it is possible to help the citizen lower their risk of falling. A fall can be very dangerous for elderly people. The risk of dying, after experiencing a fall, is three times higher for people over 70 years of age compared to people under 70 years of age [41]. This reveals the need for a well performing fall-risk assessment.

There are several different factors which define the risk of falling including individual and environmental parameters. In the article "Assessment and Management of Fall Risk in Primary Care Settings" [32] the authors define a model shown in 2.1

Individual	**Environmental**
Age-related changes	Medications
Cognitive deficits	Footwear
Strength or balance deficits	Assistive devices
Sensory deficits	Home features
Chronic conditions	Neighbothood features
Acite illnesses	Alchohol / Drugs
Behaviors / Choices	Support from caregivers

Table 2.1: Factors used in fall-risk assessment (table inspired from [32])

Given the amount of parameters in the fall-risk assessment the evaluation scheme must be highly complex. That is also the case in the article [32], where "The US Centers for Disease Control and Prevention (CDC)" have developed an algorithm that details each step of screening and assessment and guides interventions based on each individual's level of risk. This algorithm is the state-of-the-art in fall-risk assessment and is widely used. It reveals how a fall-risk assessment contains segments such as a brochure filling, exercise assessments and a physical examination. With other words, a complex setup.

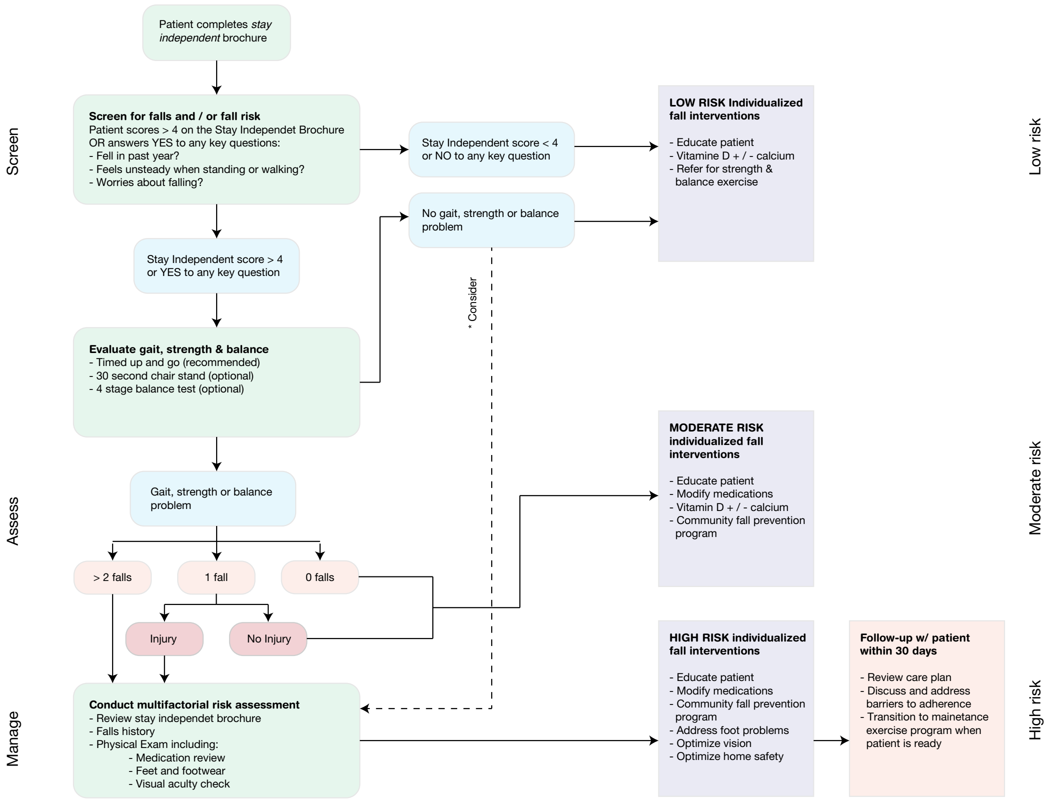
Screen
Assess
Manage
Low risk
Moderate risk
High risk
Patient completes stay independent brochure
Screen for falls and / or fall risk
Patient scores > 4 on the Stay Independet Brochure OR answers YES to any key questions:
- Fell in past year?
- Feels unsteady when standing or walking?
- Worries about falling?
Stay Independent score < 4 or NO to any key question
No gait, strength or balance problem
Stay Independent score > 4 or YES to any key question
Evaluate gait, strength & balance
- Timed up and go (recommended)
- 30 second chair stand (optional)
- 4 stage balance test (optional)
* Consider
Gait, strength or balance problem
> 2 falls
1 fall
0 falls
Injury
No Injury
Conduct multifactorial risk assessment
- Review stay independet brochure
- Falls history
- Physical Exam including:
- Medication review
- Feet and footwear
- Visual acuity check
LOW RISK Individualized fall interventions
- Educate patient
- Vitamine D + / - calcium
- Refer for strength & balance exercise
MODERATE RISK individualized fall interventions
- Educate patient
- Modify medications
- Vitamin D + / - calcium
- Community fall prevention program
HIGH RISK individualized fall interventions
- Educate patient
- Modify medications
- Community fall prevention program
- Address foot problems
- Optimize vision
- Optimize home safety
Follow-up w/ patient within 30 days
- Review care plan
- Discuss and address barriers to adherence
- Transition to mainetance exercise program when patient is ready

Figure 2.1: Illustration of fall-risk assessment algorithm (figure inspired from [32])

By using machine learning models, some of the data that is already collected amongst the different municipalities, can replace a lot of the individual work done in these assessments. Along the way, more data can be collected and help improve the model's performance. By exploring the use of machine learning, resources used in current fall-risk assessments can be lowered and used elsewhere in the healthcare programs.

2.3 Privacy preserving machine learning

With an increasing amount of data available in the healthcare sector and an increasing focus on data protection regulations, the need for machine learning methods which keeps data private is increasing with it. Privacy preserving machine learning are techniques which, to some degree, protect the data that is used as input for the training algorithms. It states a promise that the information used is not revealed to the outside world. A promise, which is important especially in the healthcare sector where most data is sensitive in some way.

Privacy preserving machine learning can be achieved in different ways. Studies reveal 3 main approaches to secure the privacy in machine learning. Differential privacy, federated learning and encrypted deep learning.

Differential privacy is the concept of adding a certain amount of noise to a query in order to imply plausible deniability for each of the input samples used. The noise adds the statistical insecurity of whether the input sample has true values or is scrambled by noise. Differential privacy is being used in numerous different applications like in [28] where it is being used to privately publish social network information. In chapter 3 differential privacy will be investigated more and applied on healthcare data to secure privacy.

Federated learning [27] addresses a well known problem in privacy protection. How can we share data between different data clusters without compromising privacy? GDPR dictates strong regulations for how it is allowed to share data between entities [38]. Federated learning twists the normal approach to machine learning. In conventional machine learning models, we centralize data, train a model and lastly deploy the model. Since this approach can be difficult with sensitive data and GDPR regulations, federated learning instead trains local models and centralize these instead. When models are trained, they are no longer leaking information on the input samples (if done correctly). Therefore, it is secure to share the local machine learning models. When shared, a datacenter then aggregates the models and deploy the aggregated model for future use in the different entities. In that approach local data can be used to improve performance in other entities without compromising privacy. In chapter 3, federated learning will be implemented in such a way that each municipality can train local models and distribute the information in the model in a secure way.

With encrypted deep learning another approach to privacy preserving is used. Encryption is a well-known way of protecting data, by converting the data into a secret code which can't be compromised. With encrypted deep learning, a neural network is constructed in way that allows the network to train using encrypted input samples. That means that the model also reveals encrypted information as output which can then be decrypted using the decryption scheme for the model. Encrypted deep learning has already been applied in industry as in [39] where medical data is encrypted and used in deep learning. In chapter 3 encrypted deep learning approaches are evaluated against healthcare data to investigate the degree of privacy which can be offered from encryption in deep learning.

Project tvaerspor is another project (apart from this thesis) carried out in Denmark regarding the use of machine learning in the Danish healthcare sector with private data [4]. The angle tvaerspor is using on healthcare data, is to predict both appropriate and inappropriate hospitalizations in elderly people. The data used in these predictions are, as the case is for this thesis, of private character which leads to the use of privacy preserving techniques.

With Tvaerspor the main goal is to collect and store data from various municipalities (Skanderborg, Horsens, Hedensted and Odder) at a central datacenter. This of course is normally not possible due to the same GDPR data restrictions that prohibits the use of conventional distributed machine learning techniques. The approach taken by tvaerspor is to collect the appropriate amount of legal documentation needed to collect data, and thereby create this central datacenter legally. The datacenter is then supposed to be publicly available under certain regulations.

Project Tvaerspors hard work in obtaining legal documentation for collecting data, could be omitted, if the right privacy preserving approaches were taken in the machine learning process. These privacy preserving techniques are made to avoid sharing data and thereby avoid the work in obtaining legal rights to data distribution. By the work carried out in this thesis, an alternative approach on how to use distributed sensitive data is proposed.

2.4 Summary

The first section in this chapter reveals that machine learning is already an asset used within the healthcare sector which enables the thought of using it in an even broader sense. The second section yields how fall-risk assessment today is a very slow and non-optimized process, in which machine learning has a potential. The third and last section covers how many different approaches already has been taken on how to preserve privacy when training machine learning algorithms. This leaves the question of how these methods perform with the data found within this domain, and what challenges and limitations the methods reveals. By exploring chapter 3, the theory behind the privacy preserving methods are covered, and multiple solutions for the domain is presented.

Chapter 3

FUNDAMENTALS OF PRIVACY PRESERVING MACHINE LEARNING

3.1 Introduction

Privacy preserving machine learning is the concept of developing and using machine learning algorithms without leaking information about the samples used in the training phase. In this chapter, different approaches to ensuring privacy in machine learning are investigated. As discussed in chapter 2, the different approaches to privacy preserving differ a lot. During this chapter the use cases for the different methods will be defined and related to the goals of this thesis. As a result, this chapter will present an approach on how to achieve the thesis goals, by use of the fundamentals of privacy preserving machine learning.

3.1.1 What is privacy?

As the name dictates, with privacy preserving machine learning, we want to use machine learning while preserving the privacy of the samples. In order to achieve this, it is first of all important to formally define what is meant by privacy.

Privacy, in a conventional context, would mean for people to be able to only reveal information about themselves according to their own choosing. That is, no undesired information about a person is revealed to the public.

With machine learning and data handling in general, we however introduce a problem that makes the conventional definition of privacy deficient. Assume some dataset is published only containing non sensitive information about the individuals. That is, no information in the dataset, can lead back to the individual behind the data. Later, another dataset is published. The new dataset shares some datatypes with the previous dataset but not all of them. Using statistical analysis on these datasets, it is possible to recover some sensitive information from the first set. This type of data attack is further elaborated in section 3.2.4. This example shows that the conventional definition of privacy may not be strong enough when challenged by statistical analysis. Instead another key aspect to the term is added. After the analysis the analyzer doesn't know anything about the people in the dataset, they remain "unobserved". To include this in the definition, another term is introduced, "differential privacy". One of the key pioneers in privacy in machine learning, Cynthia Dwork, defined the term differential privacy as:

"Differential privacy describes a promise, made by a data holder, or curator, to a data subject, and the promise is like this: You will not be affected, 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" [18]. This definition handles the previously mentioned situation by stating that information is kept private no matter what other information may be available in the future. A more formal definition can be seen from expression 3.1:

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

$$Pr[M(x) \in S] \leq \exp(\epsilon) Pr[M(y) \in S] + \delta \tag{3.1}$$

where M is a randomized algorithm, x and y are databases and S are all potential outputs of M. This definition does not promise differential privacy. Instead it measures how much privacy is afforded by a query M. In the definition there are two databases (or neural networks) x and y. These are paralel databases meaning only one entry differs between the two. The definition states that the distance between the results of a query on the two paralel databases are at a maximum of e^{ϵ}. Delta is a meassure on the probability of this definition not being valid. Often δ can be ignored, but this phenomena is elaborated further in section 3.3. In figure 3.1 it is illustrated how differential privacy ensures that the answers between two parallel networks can't be distinguished. As defined in definition 3.1, this uncertainty between answers are of course bounded by the privacy budget allowed (amount of leaked ε). See more details on privacy budget in section 3.3.6

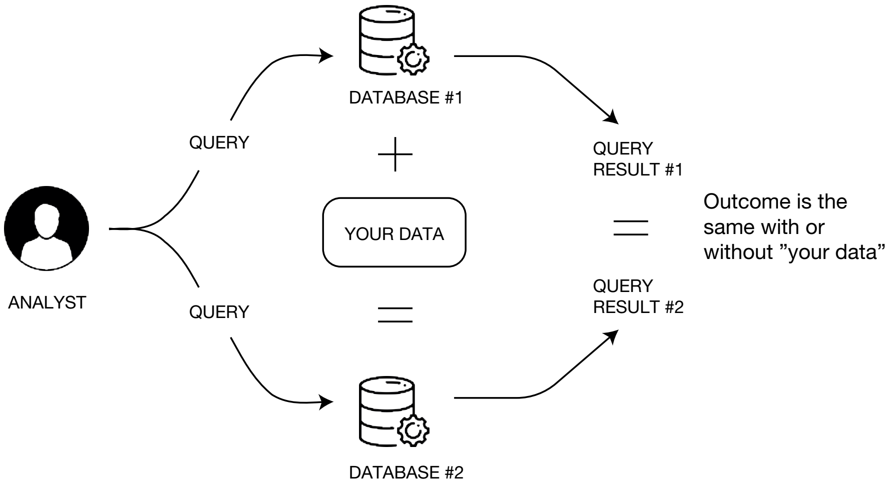

Figure 3.1: Privacy mechanism against adjacent databases (figure inspired from [34])

3.1.2 Privacy preserving needs in healthcare

As stated in section 1.2 the goals of this thesis are to provide a method for sharing data between different municipalities and making sure that the models provided, based on shared data, does not leak information about the citizens whose data has been used in the training process. This reveals two very different challenges which also needs to be solved with different approaches.

Privacy in the data sharing process means that all the data samples that are transferred between municipalities must be of private character. As will be investigated in section 3.2 normal anonymization of the data might not be strong enough to ensure privacy. Therefore, another technique to ensure data privacy when sharing data is needed. Federated learning is previously introduced in section 2.3 and provides a way of using data samples from different entities without ever sharing data. Instead of centralizing data in the training process, federated learning provides a method for centralizing models instead. This is the approach chosen, to solve this part of the listed goals.

Privacy in machine learning models means that the models cannot be used to track the information used to train them. As will be further elaborated in section 3.2, machine learning models can be compromised by adversarial attacks in a way that possibly will reveal sensitive information from the training process. To ensure this will not happen with sensitive healthcare data, the concept of differential privacy (introduced in section2.3) will be used. Differential privacy is a method for adding noise to the model training, in a way that ensures that no individual sample is too much exposed. Instead information from the entire population of samples are used in the training process.

By using both federated learning and differential privacy, it is possible to ensure both secure data sharing and secure model creation. The concepts and theory behind them will be investigated and applied in the following sections.

3.2 Threats

The problem with machine learning model creation is that the behavior of the model when exposed to new information might reveal sensitive information from the training samples. In this section, several different approaches on how these sensitive information can be extracted from the machine learning models are elaborated.

In figure 3.2, an overview of the different threats or attacks is illustrated. The different attacks require different amount of information regarding the model and the training process [12]. The reasoning for discussing several different attacks are that it is desired, that the solutions suggested by this thesis, covers as much ground as possible in terms of security levels.

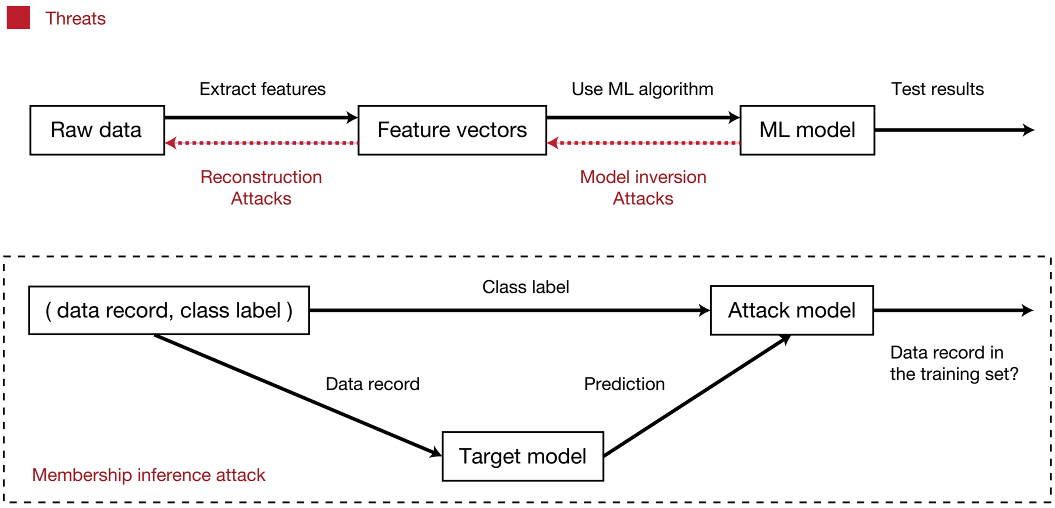

Figure 3.2: Machine learning privacy threats (figure inspired from [12])

3.2.1 Reconstruction attacks

As shown in figure 3.2, a reconstruction attack is an attack which aims at reconstructing raw input information, found from the feature vectors. Feature vectors are created as a preprocessing step for machine learning models by extracting the important information from the input data [14]. These feature vectors do often not reveal sensitive information by themselves but if white box access to the model is present, the raw sensitive input information samples can be reconstructed.

An example of this type attack is a case where adversarial access to these feature vectors were given. The feature vectors were minutiae templates compacting information of fingerprint images [19]. By using the information found in these templates the fingerprint images where reconstructed.

In order to avoid these types of reconstruction attacks, the computation server must not use machine learning techniques which require storage of these feature vectors (SVM or kNN [22]). Furthermore, the machine learning techniques that are used should reveal as little information as possible, in order to avoid leakage that could lead to the feature vectors.

The existence of reconstruction attacks prove that it is not enough to "hide" sensitive information behind feature vectors, because the sensitive information can be reconstructed if the right access is given to an adversarial.

3.2.2 Model inversion attacks

Model inversion attacks are, as showed in figure 3.2, attacks where the machine learning model are "reverse engineered" to expose some of the information that was used in the training phase. Model inversion attacks can be carried out with adversarial "white box access", meaning that the model itself is exposed, or with "black box access", meaning that only the output of the model with certain queries are exposed. These types of attacks do not require for the machine learning algorithm to store sensitive information, which were the case with reconstruction attack (storing of feature vectors). This means that also neural networks could be exposed to model inversion attacks.

The goal with model inversion attacks is to replicate the feature vectors, that were used in the training process of the machine learning model [21]. By using a strategic approach in applying different samples to the model to process, certain patterns can be found from the answers produced, which can reveal information. Model inversion attacks are therefore also mostly effective when a single feature vector is representing some label.

To avoid model inversion attacks, first the adversarial should be limited to only black box access. That however is not enough, because of the strategy explained above. Therefore, the data owner should limit the access to the output of the model or manipulate the output in some way that limits the information which an adversarial could gain from different queries.

3.2.3 Membership inference attacks

As shown in figure 3.2, membership inference attacks aim at producing attack models which can state whether a data record was used to train a machine learning model or not. The attack works by creating the attack model stated by Shokri et al. in [35] and then inputting both data labels and predictions from the actual machine learning model (black box access). The attack model would then state if the samples was used or not, based on statistical analysis (the math behind the technique is not restated here but can be found in [35]).

The result from this type of attack, is a definition on whether some information was used in the training process or not. This does violate the privacy of the people behind the training data because their information, if found to be used in training, is potentially leaked.

In order to avoid this type of attack, it is needed to separate the connection between a data record and the corresponding label. That is exactly was is done by using differential privacy. Differential privacy, adds a certain amount of noise to the outputting result of the machine learning network. This, by definition, rules out membership inference attacks, because even though the attack model states that a record is used in training, the added noise add plausible deniability to that ruling.

3.2.4 De-anonymization attacks

De-anonymization attack is another category of attacks against machine learning algorithms. With this type of attack, the adversarial is attacking against a countermeasure already done by the data owner. The data owner typically anonymizes data before using it, in order to stay within the data regulations (an example of these regulations is GDPR). The data owner then trust, that the anonymization process secures the privacy of the people kept within the dataset.

De-anonymization attacks aim at ”reversing” the anonymization process and therefore regenerating the sensitive information which were removed from the dataset. This type of attack works by using ”outside found” information that more or less correlates with the anonymized dataset. By finding patterns in the ”outside found” and anonymized dataset, it is possible to regenerate the sensitive data with a certain amount of accuracy.

A popular case being discussed in relation to this type of attack, was the Netflix AI competition [30]. This was a competition where Netflix published 500.000 customers movie ratings from their personal database. They wanted to get the public’s help in creating an even better recommendation system for their movie service. The published movie ratings were anonymized by removing both movie titles and usernames.

Researchers at Texas university was able to deanonymize some of the released data by comparing rankings and timestamps between the Netflix dataset and public available information found from the international movie database (IMDB). By removing the top 100 rated movies from each user, they found that the ratings were quite individual. This is very valuable, because the individuality in the dataset makes it possible to distinguish single samples in the datasets.

This type of de-anonymization attack reveals that the conventional anonymization process is not enough. It is not secure to publish anonymized data because even though no other correlating dataset exist currently, it is impossible to know what will be released in the future.

3.2.5 Summary

Threats against privacy when building machine learning models are a real factor. Different kinds of attack exposes different types of information from different areas in the machine learning building process. Therefore, countermeasures needs to be taken. By using differential privacy, the added noise to the queries of the machine learning models, makes the output of the model ”plausible deniable”, meaning that the adversarial can’t be sure that the output of the model is actually ”the right one” based on the input.

This makes both model inversion and membership inference attacks much harder if not impossible. By using differential privacy in collaboration with a neural network it is furthermore possible to avoid reconstruction attacks. With neural networks, feature vectors are not stored which is very much desirable.

Federated learning aim at centralizing ”small” machine learning models and aggregating them instead of centralizing data. Because centralized data is not an issue, anonymization is therefore not needed, since the data is always kept local . This countermeasure makes sure that the de-anonymization attacks can’t be carried out and expose private data. Encryption techniques can furthermore be used to ensure that models does not leak information after creation.

With differential privacy, federated learning and encryption techniques working in combination, it should be possible to create a robust mechanism for training and using a neural network-based machine learning model with sensitive private information.

3.3 Differential privacy

To recap from the state-of-the-art in chapter 2, differential privacy is the concept of adding noise to queries of some database or neural network, with the intention of hiding the "true" result of the query and therefore protecting against privacy attacks as described in section 3.2.

In figure 3.3 it is seen how the output of several deep learning networks is hidden behind a gaussian noise distribution. Given some result R it is not possible to determine with hundred percent ensurance which network produced the result. In theory, every single network could have produced the result, with the probability decreasing as moving away from the R in the figure.

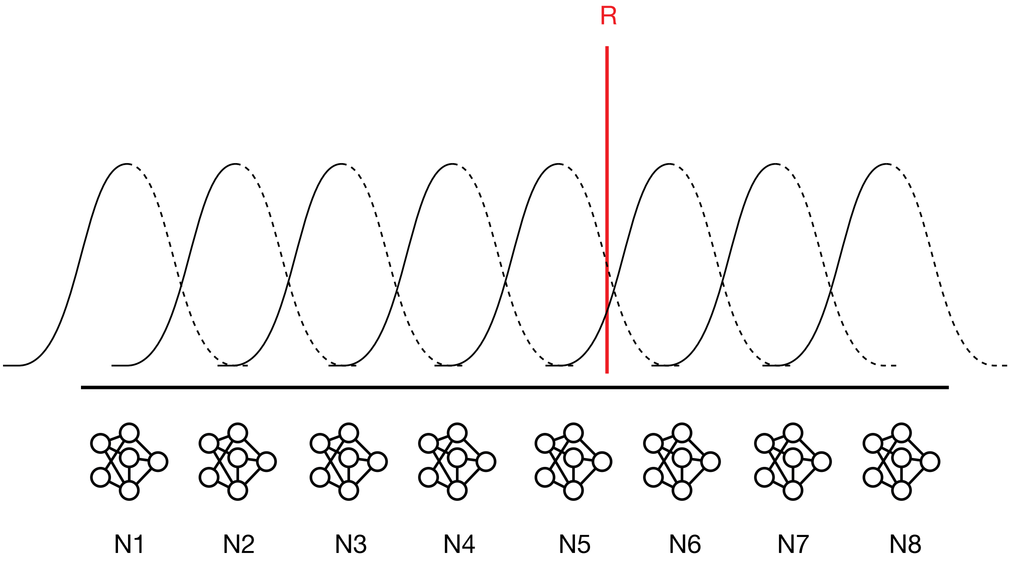

Figure 3.3: Machine learning models with gaussian noise

In this chapter the theory behind differential privacy will be investigated to prepare for the experiments carried out in chapter 5. Differential privacy can be achieved in various different ways and in this chapter a solution applicable for the domain of this thesis will be presented along with arguments as to why this approach is chosen.

To formalize the definition of differential privacy, the standard definition defined by Cynthia Dwork will be used [18].

A randomized mechanism: $M : D \rightarrow R$ with a domain M and a range R, satisfies (ε, δ)-differential privacy if for any two adjacent datasets $d, d' \in D$ and for any subsets of outputs $S \subseteq R, Pr[M(d) \in S] \leq \exp(\varepsilon)Pr[M(d') \in S] + \delta$.

This standard definition have previously been elaborated but basically states that if a function is ran on two adjacent databases, then the output of the two functions are bounded by ε.

The local sensitivity of a function $f : \mathbb{N}^{|\mathcal{X}|} \to \mathbb{R}^k$ with respect to databases x and y is defined by equation 3.2, and is used in the continued investigation of differential privacy.

$$\Delta f = \max_{y \text{ adjecent to } x} \|\mathrm{f}(x) - \mathrm{f}(y)\|_1 \tag{3.2}$$

By using the above definition for differential privacy proposed by Cynthia Dwork, the qualilative properties prososed along with it can be listed. The properties are useful in the general understanding of what differential privacy promises. The qualities are listed below and is found from [18].

- **Protection against arbitrary risks:** Moving beyond protection against re-identification

- **Automatic neutralization of linkage attacks:** Including all those attempted with all past, present, and future datasets and other forms and sources of auxiliary information

- **Quantification of privacy loss:** Differential privacy is not a binary concept, and has a measure of privacy loss. This permits comparisons among different techniques: for a fixed bound on privacy loss, which technique provides better accuracy? For a fixed accuracy, which technique provides better privacy?

- **Composition:** Perhaps most crucially, the quantification of loss also permits the analysis and control of cumulative privacy loss over multiple computations. Understanding the behavior of differentially private mechanisms under composition enables the design and analysis of complex differentially private algorithms from simpler differentially private building blocks

- **Group privacy:** Differential privacy permits the analysis and control of privacy loss incurred by groups, such as families.

- **Closure under post-processing:** Differential privacy is immune to post-processing: A data analyst, without additional knowledge about the private database, cannot compute a function of the output of a differentially private algorithm M and make it less differentially private. That is, a data analyst cannot increase privacy loss, either under the formal definition or even in any intuitive sense, simply by sitting in a corner and thinking about the output of the algorithm, no matter what auxiliary information is available.

3.3.1 Local vs. global differential privacy

Differential privacy has two aspects, local differential privacy and global differential privacy [6]. With global differential privacy a central aggregator exists which receives raw non-private data. The central aggregator will then perform the aggregation and add the noise to the result. An illustration of the concept can be seen in the left part of figure 3.4. The central aggregator is typically located at a server separate from the entities in the system.

With global differential privacy the noise is only added once, in the end of the process. This reveals one big advantage. Since the noise is only added once, the amount of noise is significantly less than in local differential privacy. With a lower amount of noise added and keeping a fair amount of privacy, the accuracy of the model can be kept higher. This can be a significant advantage for many use cases.

The catch of using global differential privacy is that, since the noise is first added at the end of the process, the aggregator must be trustworthy. The raw data from the individuals are send to the aggregator and are therefore not private. Because of that, the aggregator can't leak information and must not be compromised if the privacy should be kept for the individuals.

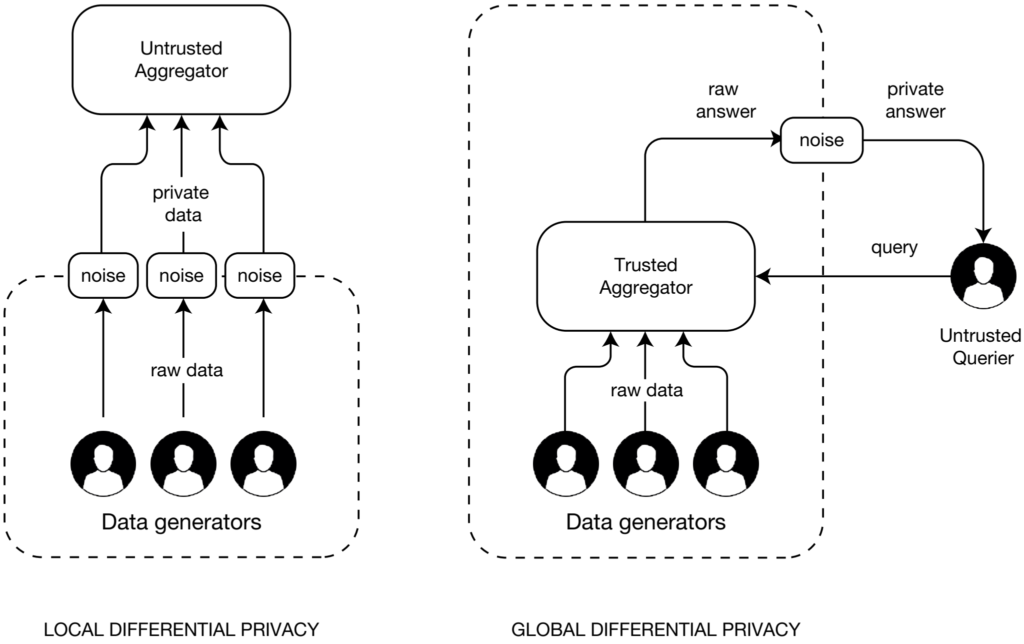

Figure 3.4: Local vs global differential privacy

With local differential privacy, noise is added to each individual and their data is therefore at a private level, even when placed at the aggregator. This ensures the highest amount of privacy since the aggregator does not necessarily have to be trusted. The data is protected. However, when using this method the amount of noise added increases with the sample population size and therefore the reachable accuracy decreases.

For the case presented in this thesis, local differential privacy is chosen. If the global version was chosen, the municipalities would have to find a secure server which they trusted with their data, which would be able to work as the central aggregator. This challenge is outside the scope for the thesis, and therefore the local version is chosen, since a secure server is not required. This choice reveals challenge regarding accuracy, which will be further elaborated in the following section.

3.3.2 Privacy-accuracy tradeoff

The choosing of local differential privacy has the challenge of reaching a high enough accuracy due to the amount of noise used in the algorithm. There are two main factors that affect the accuracy. That is, sample size and the noise level. The sample size affects the accuracy due to the intuitive reason that if more samples are used, the distribution from the samples more resembles the true distribution. As an example, the gaussian distribution is used. To illustrate the intuitive reason, see figure 3.5. Here different number of samples are drawn from a normal distribution. What is seen is that when the sample size increases the amount of sample "outliers" decreases and the shape of the distribution gets more symmetrical. This means that noise is more likely to "average out" when the distribution is closer to the real gaussian distribution. When the noise tends to "average out" the accuracy increases.

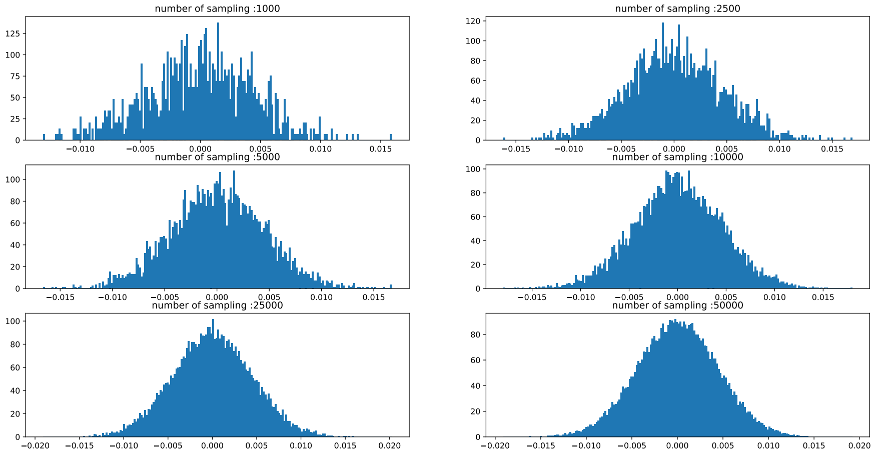

Figure 3.5: Effect of sample size in accuracy

The other concept which affects the accuracy is the privacy-accuracy tradeoff. When the amount of noise that is added to each individual is sample increases, naturally the accuracy decreases [13]. To illustrate this concept, an example from [1] is shown.

"This simulation demonstrates the Laplacian Noisy Counting mechanism (a differentially private algorithm). In a nutshell, this mechanism preserves the privacy of the ground truth (which is 3). With each new query from an adversary, our data curator returns the ground truth with additional noise. This noise is drawn from a zero-centered Laplace distribution with a parameter equal to (1 / epsilon). The quantity epsilon is a positive, tunable variable, and is the center of entire field of differential privacy. To be brief, the smaller the epsilon, the more private the results. On the x-axis, we have the number of consecutive queries that an adversary sends to the curator. Notably, it is logarithmically scaled. On the y-axis, we've plotted the average of the query replies that the curator sends out, along with the 95 percent confidence interval of those queries" [1].

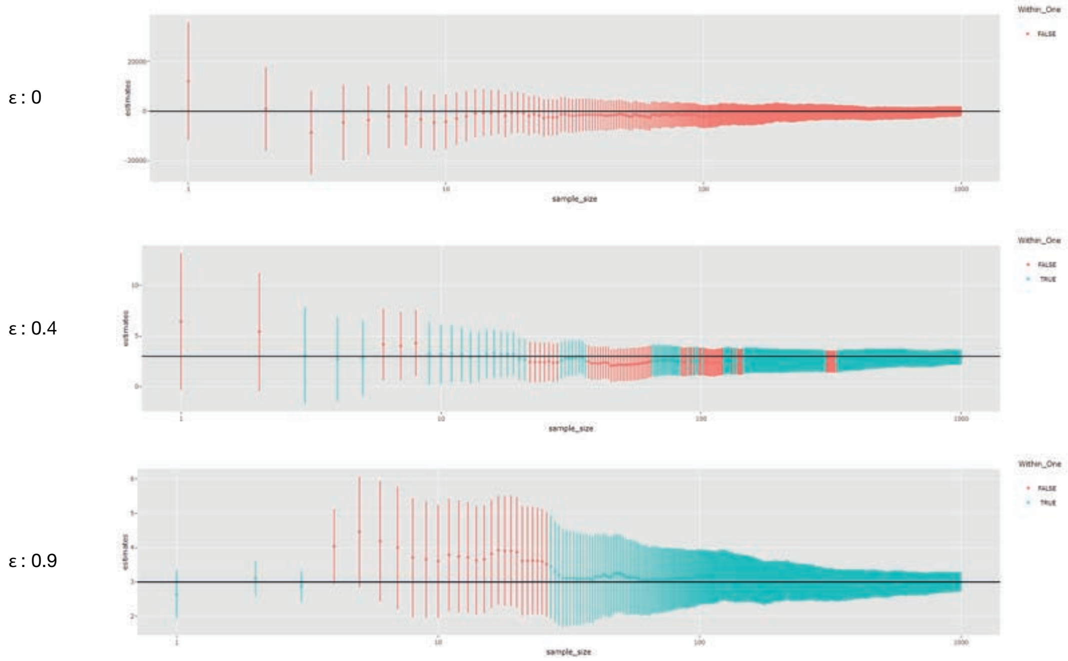

Figure 3.6: Effect of ε on accuracy

Figure 3.6 illustartes the effect ε has on the correctness of queries. It is seen how an increase in ε allows for more true answers and therefore less privacy. With ε equal to zero, the privacy is total, and no true answers are exposed. This however also destroys the model completely because of the effect on the accuracy.

In order to use local differential privacy, both the sample size and the noise level must be chosen carefully. These parameters can tweak the model performance and the privacy provided by the model.

3.3.3 Sample size increase privacy

Another beneficial factor with larger samples sizes is that the privacy which can be promised by differential privacy increases accordingly.

This property is kind of counter intuitive. If more datapoints are used in the training process, then more points need to be kept private and the intuitive understanding is that the privacy leakage will increase. However, the way differential privacy works is that it looks for patterns in datasets rather than looking for individual data structures. That is, if more data samples are available, the probability of finding patterns within the dataset increases. If patterns are found these can be used to explain the dataset better without leaking private information about the individual sample.

To investigate this further, an experiment is carried out in a Udacity course on private and secure AI [7]. Here the concept of increased sample sizes is tested against varying amounts of noise, and it is revealed how larger datasets can accept more noise while keeping the same accuracy.

This concept is beneficial for the domain of this thesis. Medical data collections are getting larger and larger, which mean that the sample sizes used within the individual municipalities will increase over time and thereby improve on the reachable accuracy and privacy.

3.3.4 Differential privacy with deep learning

To recap from earlier explanations, deep learning is the concept of effectively fitting a complex function. The deep learning models are often represented as a set of weights θ, that given some input x will produce some prediction $\theta(x)$. The prediction $\theta(x)$ is then evaluated and the loss $\mathcal{L}(\theta(x))$, the difference between the prediction and the true answer, is computed. Training the model will involve finding the parameters θ that minimizes the loss function over the different inputs x.

With supervised learning models, the model takes an input pair $(x_i, y_i$ and produce some prediction $\theta(x_i)$. The loss is then computed as $\mathcal{L}(\theta(x_i), y_i)$. During training some batch of insputs x_b is chosen to the model and the gradient on the loss for the batch is computed as $\nabla\mathcal{L}(\theta(x_i), y_i)$. When using stochastic gradient decent (SGD) we can then update the model parameters as $\theta_{t+1} = \theta_t - \eta\nabla\mathcal{L}(\theta(x_i), y_i)$.

When using differential privacy with deep learning there are multiple approaches. As previously discussed in section 3.3.1, differential privacy can be achieved both locally and globally. The local version with deep learning would mean adding noise to the individual datapoints before inputting them to the deep learning network. The global version would mean adding noise to the final weights of the network which would then produce differential private predictions. However, it might be hard to determine which weights have what dependence on the training data which makes this approach difficult.

An alternative approach, also in the scope of local differential privacy, is called differential private stochastic gradient decent (DPSGD). With this approach the algorithm for SGD is extended to involve steps that ensures differential privacy. When computing the gradients, DPSGD adds two steps involving clipping the gradients to a certain threshold which ensures that no input has significantly more power over the output and adding noise to the gradients which ensure the differential privacy. This procedure is proposed in [11] and the algorithm is presented in algorithm 1:

```
Algorithm 1: Differentially private SGD (Outline)
Input: Examples $\{x_1, ..., x_N\}$, loss function $\mathcal{L}(\theta) = \frac{1}{N}\sum_i \mathcal{L}(\theta, x_i)$. Parameters:
  learning rate $\eta_t$, noise scale $\sigma$, group size $L$, gradient norm bound $C$.
  Initialize: $\theta_0$ randomly
  for $t \in [T]$ do
    Take a random sample $L_t$ with sampling probability $L/N$
    Compute gradient
    For each $i \in L_t$, compute $\mathbf{g}_t(x_i) \leftarrow \nabla_{\theta_t}\mathcal{L}(\theta_t, x_i))$
    Clip gradient
    $\bar{\mathbf{g}}_t(x_i) \leftarrow \mathbf{g}_t(x_i)/max(1, \frac{\|\mathbf{g}_t(x_i)\|_2}{C})$
    Add noise
    $\tilde{\mathbf{g}}_t \leftarrow \frac{1}{L}(\sum_i \bar{\mathbf{g}}_t(x_i) + \mathcal{N}(0, \sigma^2 C^2 \mathbf{I})$
    Decent
    $\theta_{t+1} \leftarrow \theta_t - \eta_t \tilde{\mathbf{g}}_t$
Output $\theta_T$ and compute the overall privacy cost $(\varepsilon, \delta)$ using a privacy accounting
  method.
```

3.3.5 Noise distributions for differential privacy

In the algorithm 1 it is seen how gaussian noise is used to augment the gradients as is proposed in the original algorithm in [11]. This, along with laplacian noise, are the two most used distributions when adding noise with differential privacy [18]. There are both pros and cons to both noise distributions. With gaussian noise, the added noise is of the same type as conventional noise, as these tend toward a gaussian distribution. Another advantage with gaussian distributions is that the sum of two gaussian variables reveal another gaussian which can help in the statistical analysis of the privacy mechanism, as proposed in [18].

With laplacian noise the main advantage over gaussian, is the ability to set δ at zero. With conventional differential privacy, two parameters are considered (ε, δ). ε is the bound on the privacy between two queries of the same type against two adjacent networks while δ is a meassure of the probability of ε not holding. With laplacian noise distribution it can be proved that δ can be ignored as it will always equal to zero and yield $(\varepsilon, 0)$-differential privacy. This proof is presented here:

The general probability that a certain result is achieved by querying some database (or neural network) given that the result is added with Laplacian noise, can be expressed by the following definition.

$$Prob(R = x|D) = \frac{\varepsilon}{2\Delta f} \cdot \exp\left(-\frac{|x - f(D)|\varepsilon}{\Delta f}\right) \tag{3.3}$$

where R is the result drawn from the database, D is the database and Δf is the sensitivity. It is desired to define the reason why we can ignore δ (δ equals zero) when using the Laplacian mechanism to add noise to a query. To do so an expression can be build stating that the probability of achieving some result on one database given some query, in relation to the same query on an adjacent database, should only equal $\exp(\varepsilon)$. That expression can be seen below.

$$\frac{Prob(R|Q(D_I))}{Prob(R|Q(D_{I\pm1}))} \leq \exp(\varepsilon) \tag{3.4}$$

where Q defines a query on a database D and R is the obtained result. It was already defined in equation 3.3 how a probability of some result given the Laplacian mechanism is defined. That definition is now substituted into the expression from equation 3.4 yielding the expression seen below.

$$\frac{\frac{\varepsilon}{2\Delta f} \cdot \exp\left(-\frac{|R-f(D_I)|\varepsilon}{\Delta f}\right)}{\frac{\varepsilon}{2\Delta f} \cdot \exp\left(-\frac{|R-f(D_{I\pm1})|\varepsilon}{\Delta f}\right)} \leq \exp(\varepsilon) \tag{3.5}$$

In this expression, $\frac{\varepsilon}{2\Delta f}$ is present both in the numerator and denominator and therefore cancel out.

$$\frac{\exp\left(-\frac{|R-f(D_I)|\varepsilon}{\Delta f}\right)}{\exp\left(-\frac{|R-f(D_{I\pm1})|\varepsilon}{\Delta f}\right)} \leq \exp(\varepsilon) \tag{3.6}$$

Since there now are exponents in both the numerator and denominator, the expression can be presented in a simpler way using normal fraction calculus.

$$\exp\left(-\frac{|R - f(D_{I\pm1})|_{\varepsilon}}{\Delta f} + \frac{|R - f(D_{I\pm1})|_{\varepsilon}}{\Delta f}\right) \leq \exp(\varepsilon) \tag{3.7}$$

In the exponent we now have common variables in both expressions which mean we can collect them to simplify the expression. ε and Δ f are collected, yielding the expression below.

$$\exp\left(\frac{\varepsilon}{\Delta f} \cdot |f(D_I - f(D_{I\pm1})|\right) \leq \exp(\varepsilon) \tag{3.8}$$

In this final step, much of the expression cancel out. Δf is defined as the sensitivity. The sensitivity is the maximum distance between the same query applied to two parallel databases. That is exactly what is seen in the second part of the left side in the equation. Therefore, since there are both divided by Δf and multiplied by Δf ($|F(D_I) - F(D_{I\pm1})|$) those cancel out and reveal how the probability of the achieving some result with an equal query on parallel databases are only defined by ε and therefore setting δ equal to zero when using the Laplacian mechanism.

$$\exp(\varepsilon) \leq \exp(\varepsilon) \tag{3.9}$$

When the Laplacian mechanism reveals $(\varepsilon, 0)$-differential privacy it is intuitively seen as an advantage, since there is no chance for the privacy bound not holding. However, with the addition of a small δ it is possible to tighten the bound under composition by a great amount, as is seen with moments accountant (see section 3.3.7). Because this tighter bound under composition is possible, it is often seen how the gaussian mechanism is chosen with the addition of a small δ.

3.3.6 Composition and privacy budget

As it was seen from equation 3.1, that ε defined the bound between two queries against two adjacent models. That is, ε describes the potential leakage between the queries. For conventional differential privacy, it is said the mechanism is bound under composition. Cynthia Dwork described the theorem as [18]:

Theorem 3.1 *For any $\varepsilon > 0$ and $\delta \in [0, 1]$ the class of (ε, δ)-differentially private mechanisms satisfy $(k\varepsilon, k\delta)$-differential privacy under k-fold adaptive composition.*

This theorem describes how multiple queries bounded by a leakage of maximum ε per query, will add up. This reveals a noticeable challenge with differential privacy. For a use case like the one seen from the domain of this thesis, the municipalities might agree on a certain level of privacy, that is, a certain level of allowed ε. However, if an adversarial were to have unlimited access to the model or models, it will be possible, by the use of statistics, to reveal the true answer of a certain query, by asking the same query multiple times. Because of this ”weakness” with differential privacy, it is common to use what is referred to as a privacy budget, like seen in [25]. A privacy budget is a upper limit for how much privacy is allowed to be leaked from some model or database. When this limit is reached, the model or database will stop answering queries. This is done to prevent reaching the budget and allowing adversarial to reconstruct true answers and compromise privacy.

By using privacy budgets, differential privacy models remain completely secure against adversarial attacks. However, this yields another challenge. If the model stops answering queries it can no longer be used. That means that an adversarial could ”use up the budget” and make the models unusable for the caretakers in the municipalities.

3.3.7 Moments accountant

When adding noise in differential privacy, the bound between adjacent queries are, as mentioned, defined by ε. Much research has been devoted to understanding how privacy loss and noise distributions are related. In algorithm 1 used for DPSGD it can be seen how noise from the gaussian distribution is added with the statement:

$$\mathcal{N}(0, \sigma^2 C^2 \mathbf{I}) \tag{3.10}$$

This gaussian distribution needs to be ”mapped” to a certain ε in order for the developer to secure a machine learning network to a certain privacy level. By the work of Cynthia Dwork in [18], it has been argued that σ should be set at:

$$\sqrt{2 \cdot \log(\frac{1.25}{\delta}})/\varepsilon \tag{3.11}$$

If this is done, each step is (ε, δ)-differential private. This however, considers the normal composition theorem described in the previous section. That is, at multiple queries the privacy leakage adds up linearly. What is found from [11] is that the normal composition theorem has a loose bound on the actual privacy leakage. From the perspective of the normal composition theorem, the noise distribution used for differential privacy is not taking into consideration when defining the privacy leakage under composition. By the work proposed in [11], a tighter bound on the privacy leakage is revealed, known as moments accountant. By using conventional composition, the differential privacy described in equation 3.12 is achieved.

$$O((q\varepsilon\sqrt{T \cdot \log(\frac{1}{\delta}})), qT\delta)\text{-differential privacy} \tag{3.12}$$

By using the moments accountant instead, a tighter bound can be achieved as shown with equation 3.13. It is seen how a factor of $\sqrt{log(1/\delta)}$ is saved for ε and a factor of qT is saved for δ where T is the number of runs and q is the sampling probability.

$$O((q\varepsilon\sqrt{T}), \delta)\text{-differential privacy} \tag{3.13}$$

The effect on the tighter bound from the moments accountant can be further seen by figure 3.7. The leaked amount of ε is plotted against number of epochs for both the conventional composition and for the moments accountant. A significant savage of leaked ε can be seen. For further details on moments accountant see [11].

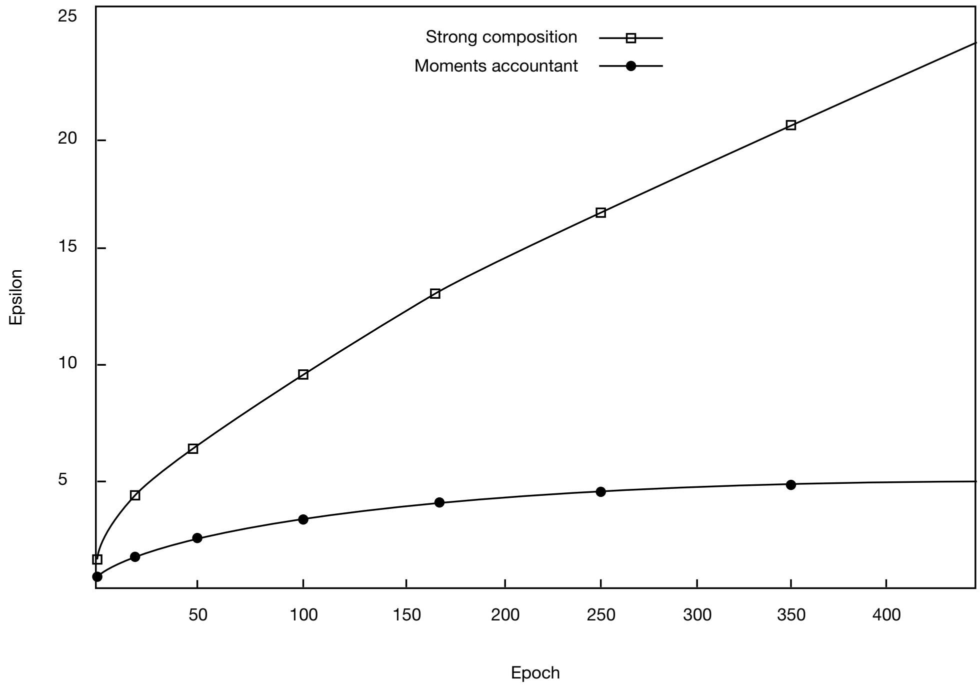

Figure 3.7: Moments accountant vs normal composition

3.3.8 Determining the right value of ε

As discussed in the previous section, the value of ε is important. By exhausting the privacy budget, the amount of leaked ε increases. That is both with normal composition and moments accountant. When differential privacy should be used, the value for ε therefore needs to be set, in order to investigate if the training process exceeds the acceptable amount of ε.

The calculation of the acceptable amount of ε is challenging because the promise made by differential privacy is offered against all the available data in the world. No matter what other potential dataset is available, the promise of differential privacy by some amount of ε is still present. This "feature" of differential privacy makes the estimation of needed ε difficult.

The basic calculation of ε can be found from equation 3.14 that is presented in [24]. Here Δf refers to the sensitivity of the feature of protection in the dataset, Δv refers to the sensitivity against any possible dataset in the world, n refers to the number of entries in the dataset and ρ refers to the probability of being identified in the database.

$$\varepsilon \leq \frac{\Delta f}{\Delta v} \cdot \ln \frac{(n-1) \cdot \rho}{1-\rho} \tag{3.14}$$

The challenge with this calculation is determining both the local sensitivity for some feature Δf but more importantly determining Δv. In order to determine Δv, all possible datasets in the world must be known, in order to choose the max distance from the local one. This is "impossible" which means some subset of datasets must be chosen when determining Δv.

Note that when both n and ρ increases the minimum ε increases accordingly. This means that with larger datasets and larger probability of being identified, less noise needs to be added to achieve some amount of ε.

3.4 Federated learning

Federated learning is another technique used in the area of privacy preserving machine learning. While differential privacy is a technique used to secure a machine learning model against adveserial attacks, federated learning is a technique for sharing knowledge between entities, without sharing of actual data. The first goal of this thesis is to define a method for sharing data between municipalities. As is known from section 1.1.3, conventional sharing of data between municipalities is not an option.

The idea behind federated learning, is to share models instead of data. That is, federated learning is a decentralized data approach. In figure 3.8 the concept is illustrated with the domain of the thesis. Each municipality would create their own machine learning model based on their local data and then centralize the trained model, instead of the data. At a datacenter, the models would be aggregated to gain equal information from each municipality. The averaged model would be redistributed to the municipalities, for further use. Whenever new data is available, the sequence will start over.

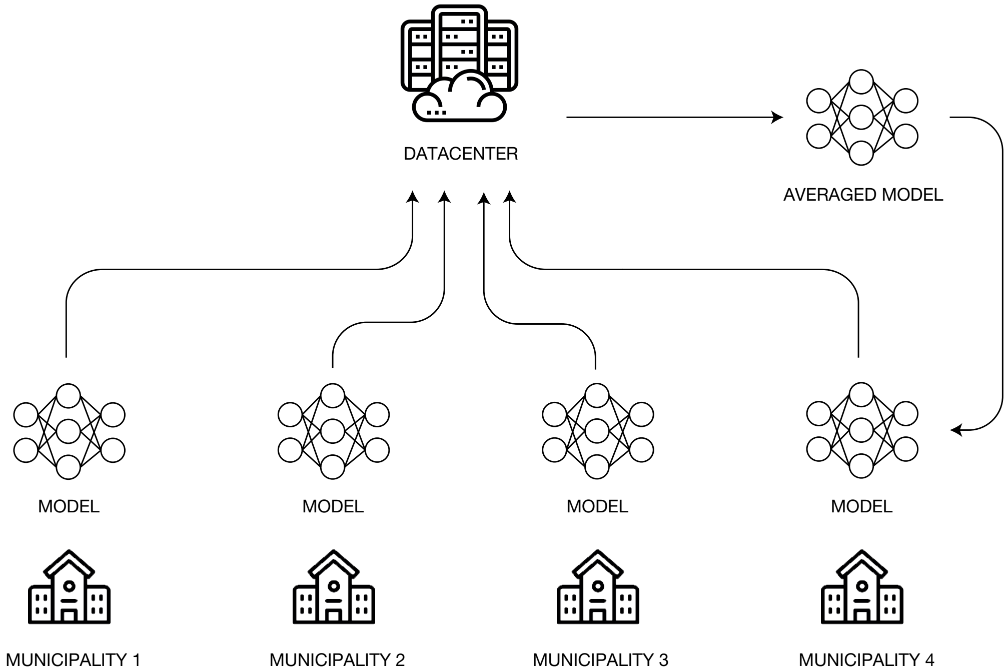

Figure 3.8: Conventional federated learning

Google are one of the pioneers behind federated learning. They introduced the concept in 2017 with the blogpost ”Federated Learning: Collaborative Machine Learning without Centralized Training Data” [2]. Google describes the concept as (for the google case, federated learning is used to train a machine learning model, which predicts word usage in their keyboards on mobile devices):

It works like this: your device downloads the current model, improves it by learning from data on your phone, and then summarizes the changes as a small focused update. Only this update to the model is sent to the cloud, using encrypted communication, where it is immediately averaged with other user updates to improve the shared model. All the training data remains on your device, and no individual updates are stored in the cloud.

Some primary benefits of using federated learning is presented in table 3.1.

Benefits	Description
Decentralized data	With federated learning it allows multiple entities to learn a shared prediction model without sharing protected data
Decentralized learning	Federated learning allows for moving the learning of the model to the ”edge”. That is, to devices like smartphones or tablet, or in this case even organazations like hospitals or municipalities.
Real-time prediction	Federated learning works faster than conventional machine learning, since the predictions is happening within the devices or organazations. Federated learning removes the time lag that occurs when transmitting data to a server for predictions.
Robustness	Since the models are stored on the devices and not on a server, it makes predictions possible even when there is no connectivity in the network.

Table 3.1: Benefits from federated learning

3.4.1 Challenges with federated learning

While federated learning proposes a method for training a global model without centralizing data, it still reveals some core challenges, which needs to be coped with when implementing a privacy solution.

With federated learning the data is, as mentioned, kept local. That is, each node in the network will never share data that could be of private character, with other nodes or the central server. However, each node does share the trained model, by sharing the gradients of the local trained models. As described in section 3.2, there exists methods for reversing these gradients back to the original training data. These attacks are known as model inversion attacks. When creating a federated learning network, the potential leakage of private information from the centralized models, therefore needs to be covered by another security measure if perfect privacy preserving is needed.

Another challenge with federated learning is communication. In a conventional setup with federated learning, as the one created by google [2], millions of devices are participating in a collaborative learning process. This can lead to bottlenecks in the communication link, meaning that the communication cost can far exceed the computational cost of local training [26].

In the domain of this thesis, the number of "devices" (in this case municipalities) is limited to under 10 (potentially in the future this will increase by a small amount), which means that communication bottlenecks will not be a factor.

3.5 Secure Multiparty Computation

Another field within privacy preserving machine learning is called Secure MultiParty Computation (referred to as SMPC). SMPC is a technique for multiple parties to compute a function jointly without sharing their private inputs to the function [8]. In terms of machine learning, this jointly computed function could be the loss function which needs to be evaluated at each epoch to perform the gradient decent for SGD.

SMPC works within the field of encryption. That is, the way SMPC keeps the promise of not leaking sensitive information, is to use encryption. There are many ways in which this encryption scheme can be implemented. This means that SMPC is a general idea for privacy preserving and many different implementations of the concept can be made.

SMPC has the most significant use case in what is referred to as, machine learning as a service. With machine learning as a service, companies offer machine learning capabilities for customers, whose data can be of private character. In the domain of this thesis, machine learning as a service could provide a secure computation of the machine learning predictions, without sharing sensitive information about citizens.

3.5.1 Additive Secret Sharing

One of the implementation of SMPC uses a protocol known as Additive Secret Sharing (referred to as ASS). ASS use the idea that information can be split into segments and all segments must be known in order to reveal the information. In figure 3.9 an illustration of the concept of ASS is shown.

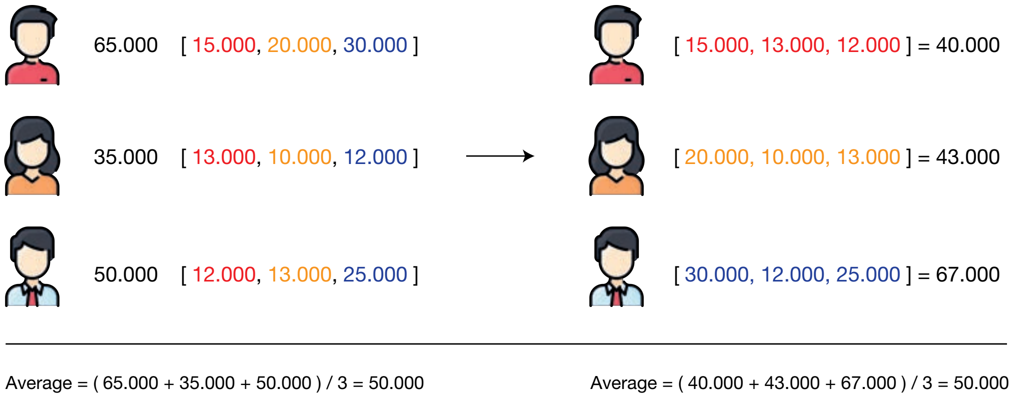

Figure 3.9: Additive Secret Sharing protocol

In the example from figure 3.9, three people wants to know their average salary. However, none of the people wants to share their own personal salary. In order to compute the average, ASS is used. Each person split their own salary in three random segments. The random segments are shared between the other parties. Since every segment is necessary to compute the sensitive salary information and an assumption is made, that none of the parties are working together, the information is kept private. Each person will now sum up their shares and reveal the result to the other parties. This summed result still does not leak information, since every share must be known. The summed results can now be averaged to reveal the true average salary of the three persons.

This example shows the capabilities of ASS. It works because operations like addition and subtraction can be performed on the individual shares without changing the result. For a more in-depth explanation of the capabilities of ASS see [42].

3.5.2 Additive Secret Sharing encryption

With the example carried out in figure 3.9, the shares of each party is distributed to the other parties. While it is still the case that each party must know the other shares in order to decrypt the information, some knowledge is still gained from a single received share. Since the example case is based on salary, the other parties might have some preliminary knowledge of the field or scope in which the salaries exist. That is, there are a limited number of possible sharing combination within the field. As a minimum, the share receiver can determine that the true salary of the sender of the share must be greater than the share received. It is not a lot of leaked information, but this "deficiency" with ASS can be fixed with encryption.

Encryption is added to ASS by defining a large field in which the shares can be defined. This is usually a very large prime number [7]. In conventional ASS the shares must sum to the sensitive number. With encryption added, the shares must sum to the summation modulus the field. The field is referred to as Q. By adding this encryption, the field in which the shares are created is now infinite. Also, there will be an infinite number of possible shares that all map to the same share value, since the modulus operator ”wraps around” the field. For details, an implementation of the ASS encryption process is shown in the code snippet below [7].

```
import random

def encrypt(x, n_shares=3):

      Q = 23740629843760239486723

      shares = list()

      for i in range(n_shares - 1):
            shares.append(random.randint(0, Q))

      final_share = Q - (sum(shares) % Q) + x

      shares.append(final_share)

      return tuple(shares)
```

Code Listing 3.1: ASS encryption

The reason for choosing a large field (a large number for Q), is that we cannot represent numbers that are bigger than the field. In the decryption process, the decrypted number would be the encrypted number modulus the field. Since the modulus operation ”wraps around”, information would be lost if larger numbers that the field were used.

3.5.3 Fixed presicion encoding

ASS, as shown in the example carried out in figure 3.9, works with integer numbers. In that example the domain was salary and the jointly computed function was as averaging operation. In order to use ASS within the domain of this thesis, it is necessary to be able to work with decimal numbers since deep learning model gradients tends to be decimal numbers.

In order to accomplish this, the encoding of the numbers used, can be changed. The gradients would normally be in floating point encoding, which must be changed to what is referred to as fixed precision encoding. Fixed precision encoding enables decimal numbers to be represented as integers defined to some precession. It does so by raising the decimal number to some predefined precession using a base number. Once again, a large prime number Q is used to define the field in which the encoded numbers can be represented. This is again secured with the modulus operator. An example of both an encoding and a decoding scheme for fixed precession encoding can be seen in the code snippet below:

```
BASE = 10
PRECISION = 4
Q = 23740629843760239486723

def encode(x_dec):
      return int(x_dec * (BASE ** PRECISION)) % Q

def decode(x_fp):
      return (x_fp if x_fp <= Q / 2 else x_fp - Q) / (BASE ** PRECISION)
```

Code Listing 3.2: Fixed precision encoding

3.6 Shapley Additive Explanations

With machine learning in general, there is a trade-off between complexity and interpretability. This means that if a very simple machine learning algorithm is made, it will be easy to explain the prediction. Similarly, if an algorithm is very complex it will be hard to explain which features are moving the model in which directions. An example of this could be normal linear regression. Linear regression typically has the form seen from equation 3.15.

$$f(x_1, x_2, ..., x_n = \phi_1 x_1 + \phi_2 x_2 + ... + \phi_n x_n) \tag{3.15}$$

Each feature is assigned a weight coefficient and summed up to the result. With this simple model the coefficient is directly showing the importance of the feature on the outcome. This interpretability however comes at the cost of complexity. This simple model is only able to predict linear patterns in the data. For real use cases the patterns in datasets are often non-linear which is also part of the definition of deep learning networks (looking to fit a complex non-linear function to a set of features). This reveals a problem. How can we get interpretability of a machine learning model and still have it be complex enough to predict on non-linear data patterns?

One solution is Shapley Additive Explanations or SHAP values [5]. The expression for calculating SHAP values can be seen from equation 3.16

$$\phi_i(p) = \sum_{S \subseteq N/i} \frac{|S|!(n - |s| - 1)!}{n!} \cdot (p(S \cup i) - p(S)) \tag{3.16}$$

The last part of the expression, $(p(S \cup i) - p(S))$ is intuitive. This yields how the importance of i is the difference between the model output to the input set containing i and to the input set not containing i. However, this is not the entire explanation of

SHAP values. Because of the complexity of the models, the patterns are not linear. That is, the output of the model might change accordingly to the order, in which it sees certain features. This means that if the feature evaluated before i changes, then the way i affects the model output could change accordingly. To compensate for this, the importance of i is calculated on all possible sets of the features and is added by the term $\sum_{S \subseteq N/i}$. The weight coefficient for each feature is calculated as how many permutations of the sets exist normalized by the features in total. That is done by $\frac{|S|!(n-|s|-1)!}{n!}$.

The outcome are the SHAP values which explains the importance of a certain feature with respect to all possible combinations of the remaining features. This is of course very computational heavy, but certain libraries exist for optimizing the computations. An example plot of SHAP values from a soccer statistic dataset can be seen from figure 3.10. Here the model is predicting if a soccer team had the player who was named "man of the match". In the dataset a "1" refers to having the "man of the match" player and "0" refers to not having the "man of the match" player. From the top left datapoint it is seen how a low value in the "Goal Scored" feature moves the model outcome towards "not having man of the match player" with approximately 0.28.

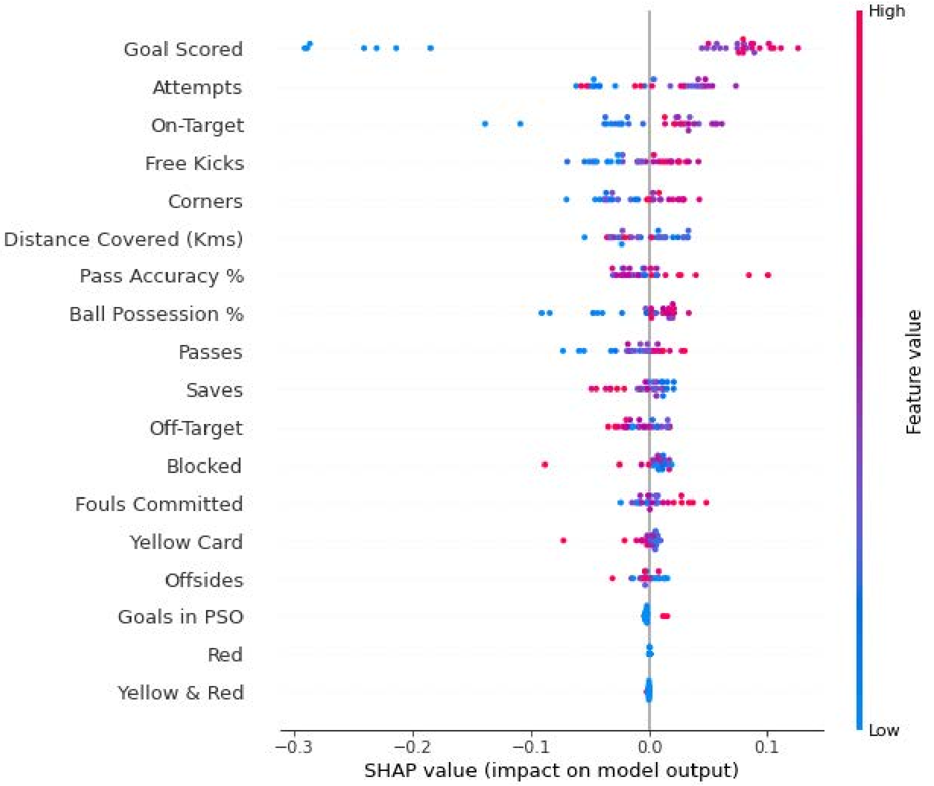

Figure 3.10: SHAP value plot for complex model explanation

SHAP values offer a great explanatory tool for understanding how machine learning models work. Within the healthcare domain, this can be important. Under the assumption that a privacy preserving machine learning model works as intended in terms of both privacy and accuracy, it would be used to evaluate elderly citizens risk of falling. Upon a new prediction of high fall-risk for a citizen, SHAP values could be helpful in determining the right direction for the ”treatment”. The SHAP values will possibly be able to explain which of the features, are most responsible for the classification as ”in risk of falling”, and thereby explain what should be changed in order to potentially affect the outcome of the model.

SHAP values helps adapt machine learning models from being a scientific algorithm to being an explanatory tool which is understandable for the citizens. It might help citizens to trust the algorithm more because it reduces the complexity of the message that the caregivers are trying to communicate when a treatment is proposed.

3.7 Solutions

Using state-of-the-art theory presented in this chapter, a list of different possible solutions have been created. Each of the solutions has advantages drawn from the theory and will have capabilities for adding value to the overall goals of the thesis. However, much of the theory also dictates limitations, which means that some of the approaches needs to work in cooperation in order to gain usable value.

In the following, five different approaches to privacy preserving machine learning will be presented along with its advantages and limitations. These solutions will be the basis of the experimental phase in chapter 5, where the value of the solutions will be evaluated against the goals of the thesis.

3.7.1 Solution 1: Data central deep learning

In figure 3.11, the first solution is presented. This figure illustrates the conventional method of performing machine learning in a network of multiple entities. Each of the entities, in this case municipalities, share their data with a central server or datacenter, where all the calculations for a machine learning network will be carried out. This approach naturally has the advantage of having all the raw data from all the municipalities available, which make the creation of a machine learning network very feasible and the performance very strong.

The natural limitations of this approach is already presented in the introductory work of the thesis. Because of GDPR restrictions, the collection of datasets of private and sensitive information is not possible. This means that even though the solution is the best in terms of performance, it is not usable for this domain. However, this solution will be refered to as the reference for how well other solutions are performing.

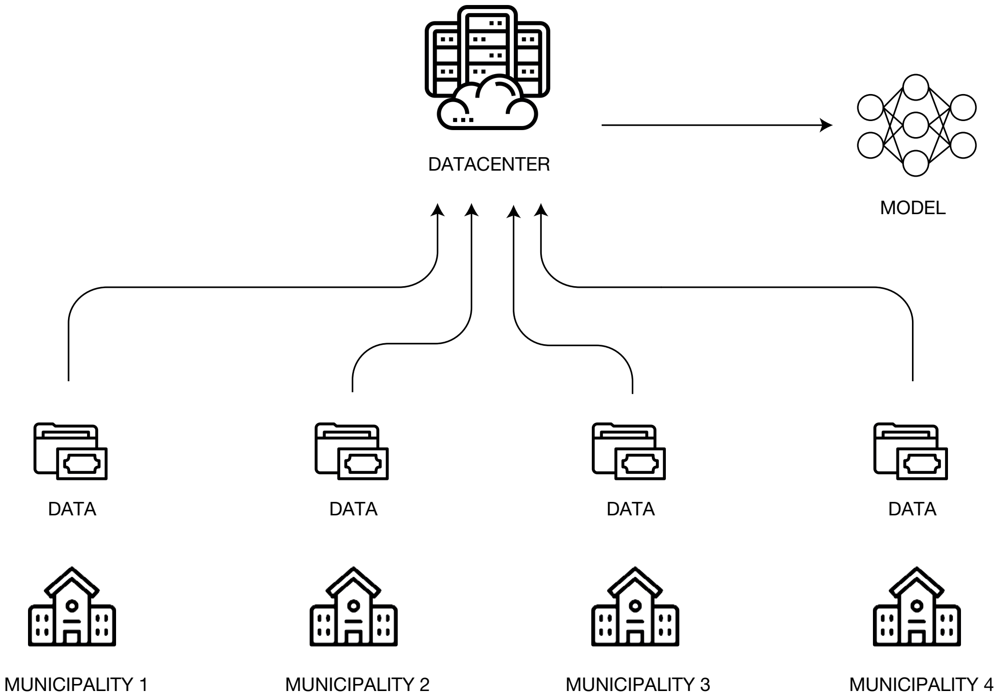

Figure 3.11: Data central machine learning approach

3.7.2 Solution 2: Conventional federated learning

In figure 3.12 the second solution is illustrated. This approach is based on the conventional idea of federated learning. The fundamentals of federated learning have already been presented in section 3.4. With federated learning the approach is swapped from a data central to a data decentral method. This means that the data are kept local at each municipality and the model creation is performed locally as well. This approach copes with the challenge of GDPR restrictions in terms of data collection.

Federated learning collects and stores machine learning models at a central server or datacenter. However, by the work presented in section 3.2, it is known that even though the raw data is not collected, the models still poses a threat against the sensitive data used to train the models. By certain adversarial attacks, sensitive information about the training data, can be extracted from the model. This approach of course leaves a more private solution than the case is for a data central approach, but still leaves potential threats against privacy.

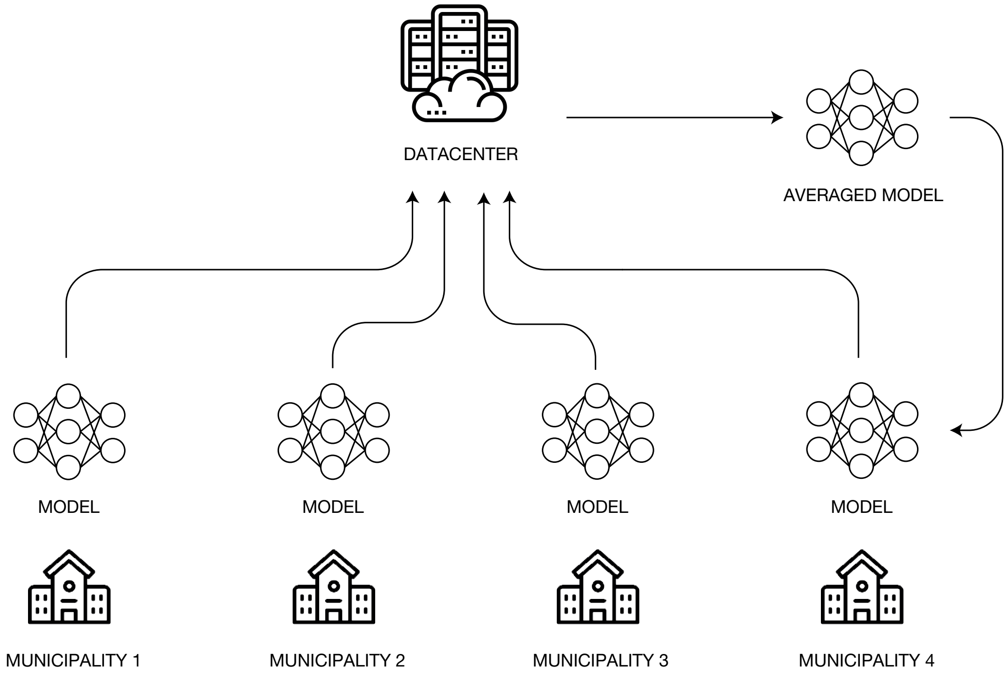

Figure 3.12: Conventional federated learning approach

3.7.3 Solution 3: Federated learning with DPSGD

The third solution is presented in figure 3.13 and has the same basis as the solution presented in figure 3.12. The challenge with conventional federated learning is that the collected models at the datacenter still poses a threat against privacy. With this third solution, the challenge is covered using differential privacy. By using differential privacy it is possible to add plausible deniability to all the answers extracted from the machine learning network, by adding different levels of noise to the queries. This means that no conclusive assumptions can be made about the people behind the training data, and therefore the privacy can be kept at a certain level. The noise is added to the gradients in stochastic gradient decent yielding the name differentially private stochastic gradient decent (refered to as DPSGD).

When using conventional differential privacy it is bound under what is referred to as normal composition. The amount of leaked privacy increases linearly with the number of queries against a network. In 3.3 a method called moments accountant is presented which leaves a more tightened bound on the composition of privacy leakage. However, even though moments accountant performs better, it still increases the leaked amount of privacy as the number of queries increases as seen from figure 3.7. This means that this third solutions do offer better privacy keeping (at the cost of performance), but still leaves potential privacy threats if adversarial are left with unlimited access to the models.

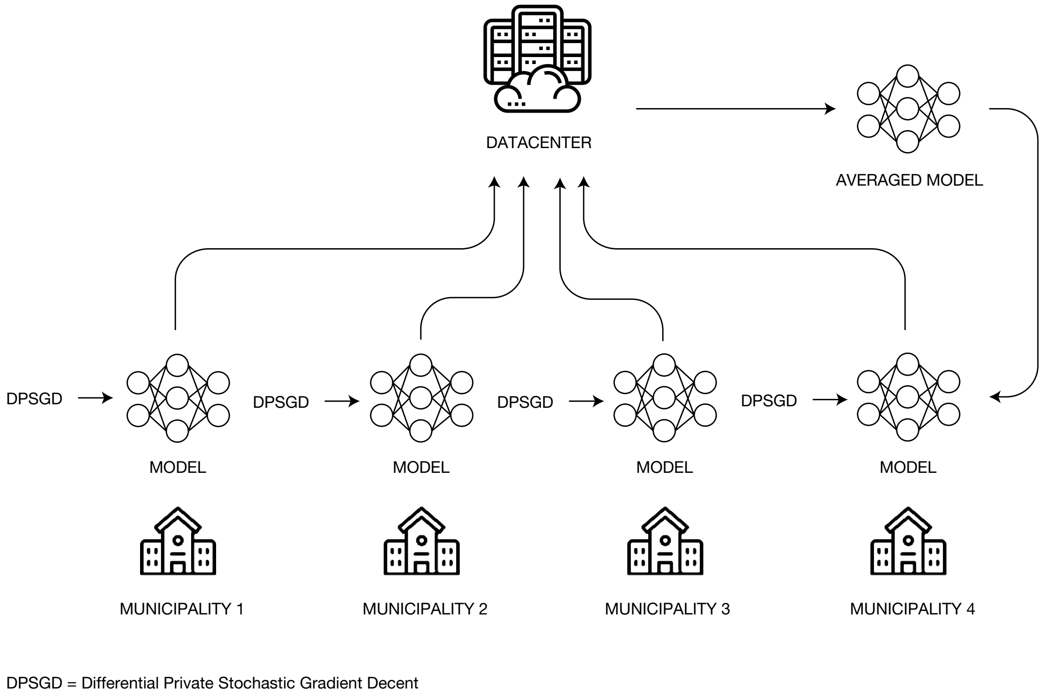

Figure 3.13: Federated learning with differential privacy

3.7.4 Solution 4: Federated learning with ASS

With the fourth solution presented in figure 3.14, the differential privacy is removed. Instead, the challenge with the second solution, of collecting models, is addressed in another way. The computation that is carried out at the datacenter is an averaging of models. This computation is, with this fourth solution carried out, using ASS.

By using ASS for the averaging process, none of the municipalities can decrypt information from other municipalities. This allows for computing the averaging function between municipalities without sharing information. This solution leaves both data and models local at each of the municipalities.

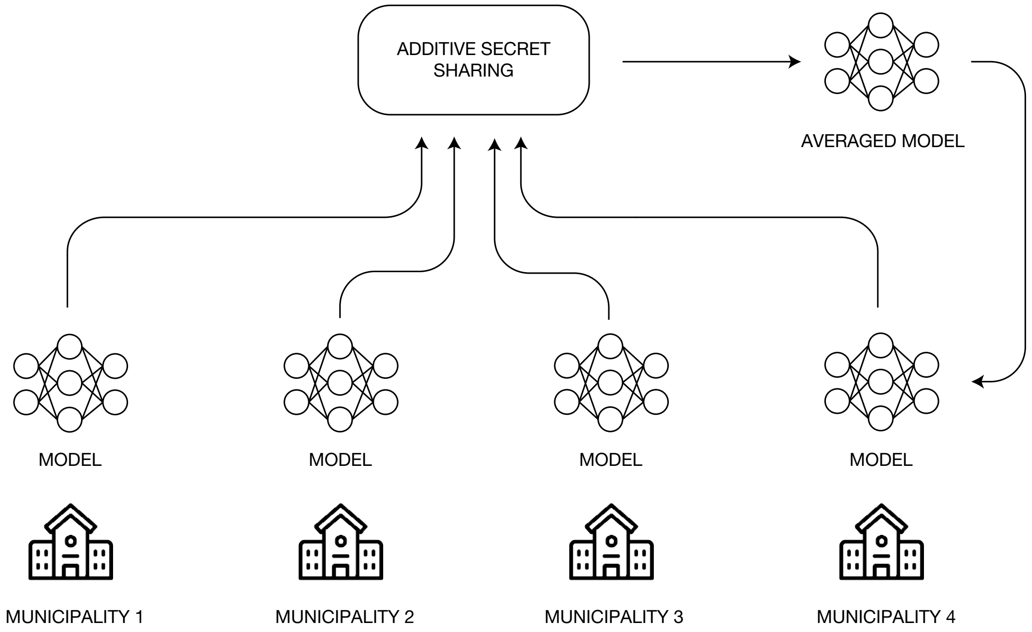

Figure 3.14: Federated learning with ASS

3.7.5 Solution 5: Federated learning with SMPC

The fifth and last solution, presented in figure 3.15, is again based on an encryption technique. Instead of using encryption only for the computation of the averaging function, a full SMPC network is created. With a SMPC network all the data and all the models from the different municipalities are ”shared” with each other, with the use of ASS. This encryption prevents any of the municipalities from decrypting any information, data nor models. This approach does also leave both data and models local because the encryption prevents the municipalities from retrieving complete information.

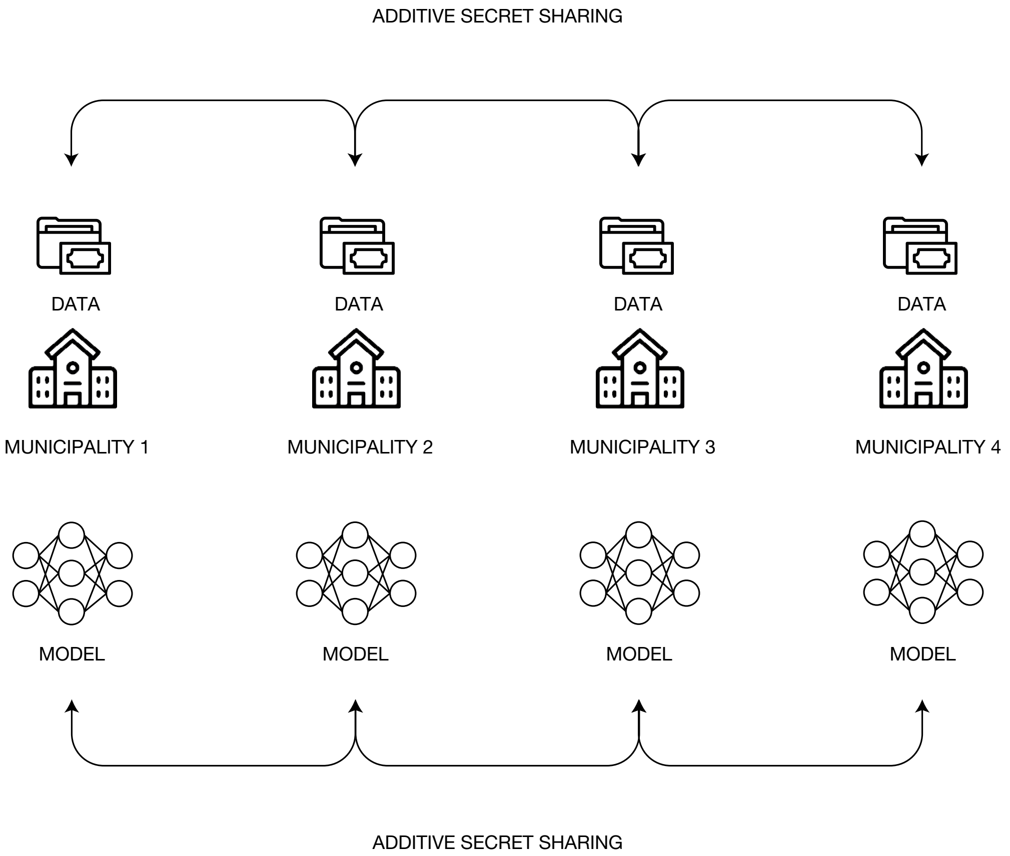

Figure 3.15: Federated learning with SMPC

Chapter 4

DATA

As mentioned in chapter 1, the direction of investigation for this thesis is how to optimize the process of fall-risk assessment by using privacy preserving machine learning initiatives. The data used to train machine learning algorithms within this area are naturally of private character and therefore sensitive to the people behind the data.

4.1 Data access

The data could potentially contain data features from two different databases. The first database contains data about various citizens usage of assistive devices. In the Danish society, and therefore in any given municipality, when a citizen is offered an assistive device, this information is logged and stored. This database will therefore both have information of which devices are being, but also a timeline for when the devices were given. This information is valuable in the fall-risk assessment process because the use of assistive devices contains information on the physical condition the citizens are in. If many assistive devices such as wheelchairs or assistive toilet seats are used, the balance and therefore the risk of failling will be increased.

The second database contains physical training data. DigiRehab which are delivering the data for this thesis, are already running a solution in which, they track medical and physical information from the citizens in the municipalities which are using the solution. DigiRehab are then offering training programs based on this tracking of information. At DigiRehab they are storing information about which citizens are participating in training sessions and how they are performing. This information could potentially be value in the fall-risk assessment because training history could reveal important information on balance and physical condition. If a citizen can complete a certain training program, the risk of him/her falling would decrease.

This second database, however, is only available for citizens in municipalities which currently are using the solution from DigiRehab whereas the information from the first database is available for all Danish citizens. This means that with the current state of information tracking, not enough citizens are available with training information, to be used in the machine learning creation process. When the model would be deployed, it would be biased against certain data that, for many citizens, would not be available.

Therefore, it is chosen only to use the database for assistive devices in the first iteration of machine learning model training. The information from the database is composed with general personal data linked to the CPR number such as gender and age.

4.2 Data analysis

CitizenId	Gender	Age	Cluster	1. ATS, 2. ATS ... N. ATS	NumATS	HasFallen
1002012383	1	78	3.0	1809, 1815 ... 1231	2	1
1002282161	1	83	16.0	1206, 1810 ... 1222	5	0
1002383031	0	69	5.0	2421, 1812 ... 1206	3	0
1002402973	1	95	9.0	1821, 1815 ... 1236	7	1
				. . .		

Table 4.1: Example of data structure

In table 4.1 an example of the data structure, can be seen. The first column is containing the ID of the citizen. These ID's are created by DigiRehab and are based on their social security number but are created by using a secret code such that the original social security number cannot be restored. This is the privacy measures that is being made currently to account for privacy, but as it has further been explained in section 3.2, removing the social security numbers are not a very effective way of preserving privacy, which reveals the potential for the work conducted in this thesis.

The second and third column are gender and age respectivly and is derived from the original social security number. The fourth column is named "cluster". Clusters are a representation of the timeline of assigned assistive devices. The clusters are generated preliminary to this thesis work. They are generated based on Kmodes which essentially is k-means clustering technique for categorical data features.

The fifth column is called ATS and is a string containing all the assistive devices given in chronological order. The assistive devices are labeled by a HMI number, referring every potential assistive device that is available in Denmark. The HMI number is originally 8 digits, but the last 4 digit are removed to lower the resolution. That is done because, in the algorithm creation, it is not desired to know the specific wheelchair model, but only weather or not a citizen is in need of a wheelchair. The resolution of 4-digit HMI numbers was decided in collaboration with DigiRehab. All HMI numbers can be tracked at HMI-basen [3].

The ATS column contains 23 instances since the citizen in the dataset with most assistive devices has 23 devices. For citizens with less devices the remaining instances contains zeroes. The sixth column contains the number of assistive devices. This number does not necessarily map to the number of non-zero entities in the ATS string since a citizen potentially can have multiple types of a given assistive device. This would only be noted by one instance in the ATS string but yield two in the number counting in the sixth column.

The final column contains binary labeling of whether the citizen has experienced a fall accident within the time of tracking.

4.3 Data manipulation

The ATS string is difficult to embed in a deep learning model, and therefore an encoding scheme is used to adapt the categorical information to a numerical representation. With ordinary label encoding each assistive device (each HMI number) would be assigned a value starting from 0 and increasing. This however is a problem when used with deep learning networks. Since the labels assigned by label encoding increases, the deep learning network could potentially "think" that a higher label means a higher numerical value and therefore higher importance in the algorithm. Since the information is categorical this is not the case.

To avoid this problem, one hot encoding is used [9]. One hot encoding creates columns for each potential category and assigns ones in the columns which is contained in the ATS string and zero in the others. After one hot encoding the data structure would be presented as in table 5.1.

CitizenId	Gender	Age	Cluster	1809, 1812 ... xxxx	NumATS	HasFallen
1002012383	1	78	3.0	1, 0, ... 1	2	1
1002282161	1	83	16.0	0, 0, ... 1	5	0
1002383031	0	69	5.0	0, 1, ... 0	3	0
1002402973	1	95	9.0	0, 0, ... 1	7	1
				. . .		

Table 4.2: Example of data structure after one hot encoding of ATS string

4.4 Data segmentation

As covered in chapter 1, the purpose of this thesis is to uncover methods for privacy preserving machine learning. The case involves fall-risk assessment in the Danish municipalities. The dataset delivered by DigiRehab with assistive device information, is collected from the municipalities which already uses DigiRehabs screening system. However, the goal with the system, uncovered in this thesis is to be used with all Danish municipalities.

Because the dataset, delivered from DigiRehab, is not split into municipalities, and because the uncovered system must be used with numerous municipalities, the dataset is chosen to be divided into five portions, each representing an individual municipality. Furthermore, the dataset will, before municipality segmentation, be split into a training and a test segment. The test segment is kept for the experimental phase, where the various models will be evaluated. The data segmentation process can be seen from figure 4.1.

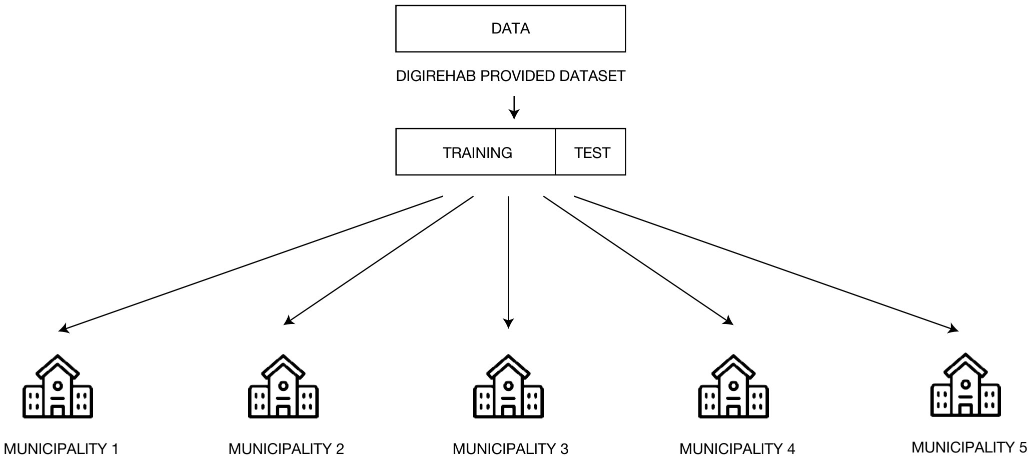

Figure 4.1: Illustration of data segmentation process

Chapter 5

EXPERIMENTS

In this chapter different experiments will be carried out to validate the performance of the various techniques uncovered in previous chapters. The experiments will serve as basis for the evaluation of privacy preserving measures in machine learning. In figure 5.1 an overview of the different experiment phases can be seen.

The experiments are divided into three phases: ”conventional data central model”, ”privacy preserving initiatives” and ”machine learning explainability”. Each experiment is labeled according to their phase with the abbreviation: ”CDCM”, ”PPIN” and ”MLEX” respectively. Within the phases, subcategories are established to group different experiments. These groups are labeled with the letters ”A” and ”B” while chronological numbers represents each single experiment. An example could be CDCM-B-2. Explanations of the subcategories can be seen from figure 5.1.

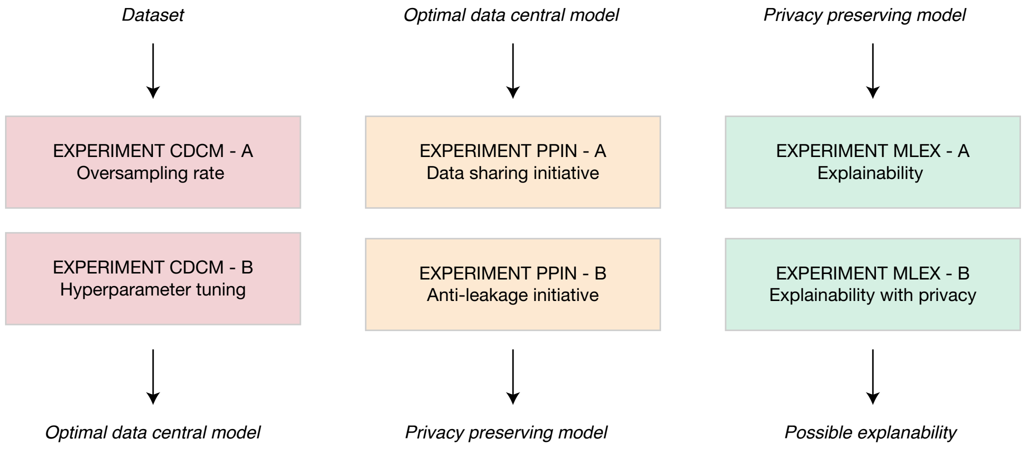

Figure 5.1: Overview of the different experimental phases with inputs and outputs

5.1 Benchmark of model performance

In the following experiments, the model performance will often be referred to. In order to compare model performance for different models and different datasets, a benchmark needs to be defined. Six different metrics are defined to compare the model performance in different scenarios which are elaborated below. When model performance are referred in the following experiments, these six metrics are what is referred to. The only exception is (ε, δ)-differential privacy which is only used with experiment PPIN-B-2.

Sensitivity is a statistical performance measure of a binary classification. Sensitivity is calculated as the percentage of true positives correctly identified as positive according to the model. That is, sensitivity measures the model's ability to correctly identify positives. Sensitivity is calculated as:

$$sensitivy = \frac{true\_positive}{true\_positive + false\_negative} \tag{5.1}$$

Specificity is closely related to sensitivity. It is also a statistical performance measure of a binary classification. Specificity is the opposite metric to sensitivity. That is, specificity measures the model's ability to correctly identify negatives. Specificity is calculated as:

$$specificity = \frac{true\_negative}{true\_negative + false\_positive} \tag{5.2}$$

ROC AUC score is an accuracy metric for model predictions. Conventional accuracy metrics don't take the rate of false positives into account which can lead to skewed results. The ROC AUC score is calculated based on the ROC curve. The ROC curve is a curve plotted with the "false positive rate" on the x axis and "true positive rate" on the y axis. A random guess suggests an equal increase in "false positive rate" and "true positive rate", while a perfect classifier has zero "false positive rate". The ROC AUC score is the area under the ROC curve and lies between 0.5 and 1 with 1 being perfect preditction.

CPU time consumption (train and test) is a measure used to determine the computational cost of the process. The metric states the walltime used for both the training and the test.

Memory usage is a measure used to determine the amount of memory used to train and test a given model. Some of the privacy preserving initiatives might demand more memory, which is why this metric is tracked.

(ε, δ)-differential privacy is the introduced privacy measure from chapter 3. This metric will be used when privacy leakage is measured for differential privacy.

5.2 Experimental environment and tools

The implementation of the findings from chapter 3 and the code for the carried-out experiments will be made with Python 3. As extensions to Python several different tools are offered within the field of privacy preserving machine learning and explainability. The tools used for the experimental phase is presented in figure 5.2.

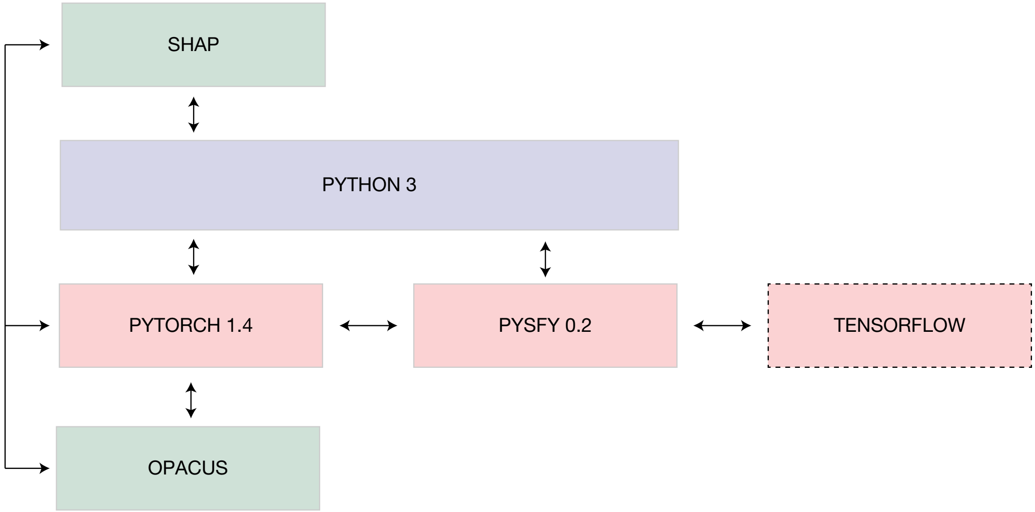

Figure 5.2: Overview of the implementation environment. Frameworks presented are used during the experiments.

PyTorch is an open-source machine learning library build for python. With PyTorch, the creation of machine learning models, especially deep learning models, is made easy. PyTorch is an alternative to TensorFlow, which may be known by more people. The choosing of PyTorch as library for machine learning model creation is due to the well build privacy extension Opacus, which is only offered for PyTorch models.

Opacus is a library build to enable PyTorch models to be trained with differential privacy. It is built as an extension in a way that ensures that the code for the model does not need to be changed in order to add differential privacy to the model building process. Opacus offers a privacy engine which needs to be created with a set of parameters. This privacy engine can then be attach to the optimizer in the model, and thereby add the desired amount of noise. The adding of noise with Opacus builds on the DP-SGD approach (see chapter 3.3.4)

Another tool used with the privacy initiatives is PySyft. PySyft is a python library build for decentralized learning. That means that PySyft offer possibilities for creating virtual machines that mimics decentralized servers in which model creation can be done. PySyft decouples sensitive data from computational servers, which allows for using initiatives in federated learning and cryptographic approaches such as ASS and SMPC.

The last tool used is a package build for Python which implements the ideas from SHAP (see 3.6). The SHAP package offer functions for the calculations of SHAP values as well as functions for building the SHAP plots. The SHAP package can be used with many different python machine learning models, including PyTorch models and Opacus extended models.

5.3 Conventional data central model

This section covers the first phase of the experiments (left part of figure 5.1). The experiments in this phase are carried out to determine the optimal settings for a conventional data central model. This is needed to ensure an optimal reference model against the privacy initiatives which will be covered in the next two phases of the experiments. The output from the experiments in this phase is an optimal data central machine learning model for fall detection.

In order to tune the hyperparameters for the data central model, as is the purpose with the first phase of the experiments, an architecture needs to be established. This architecture will be used for all the models throughout the experimental phase. Figure 5.3 illustratrates the chosen architecute which has 4 fully connected layers of 72, 128, 128 and 1 neuron(s). Furthermore, it uses a dropout layer of 0.2 and a sigmod function to convert the output logits (log-odds) to probabilities. For loss computation, a binary cross entropy function is used as shown.

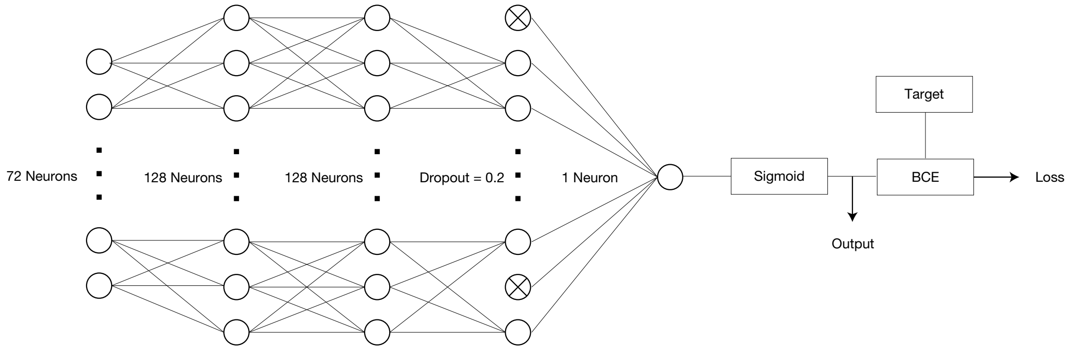

Figure 5.3: Overview of deep learning model architecture. This architecture is used throughout all the experiments.

5.3.1 Experiment CDCM-A: Oversampling rate for unbalanced dataset

Purpose: Examining what level of oversampling rate for the unbalanced dataset yields the best model performance.

Procedure: In chapter 4, the data for the experiments are presented. The dataset contains 34635 samples with 2874 of these samples being samples where a fall was recorded. This yields an imbalanced dataset with one class (the not falling class) being around 11 times more present in the dataset. This would naturally result in a model biased towards the negative class since it would be exposed to more samples of that class during training. In order to compensate for that factor, oversampling is used. Oversampling means replicating samples from the small class and resampling them into the dataset, resulting in a more balanced dataset. This experiments aims at determining the best level of oversampling rate. The different rates used for the experiment can be seen from table 5.1.

Class 0 samples	Class 1 samples	Sensitivity	Specificity	ROC AUC
31761	2874	67,30	94,00	80,65
31761	5748	77,84	90,68	85,26
31761	11496	81,24	89,62	85,43
31761	20000	83,62	87,76	85,69
31761	25000	84,98	85,90	85,44
31761	31761	83,43	83,21	83,32

Table 5.1: Experiment CDCM-A results. With a higher oversampling rate, the ROC AUC score increases until overfitting hits. That is seen from the last tried oversampling rate, where the ROC AUC score begin decreasing.

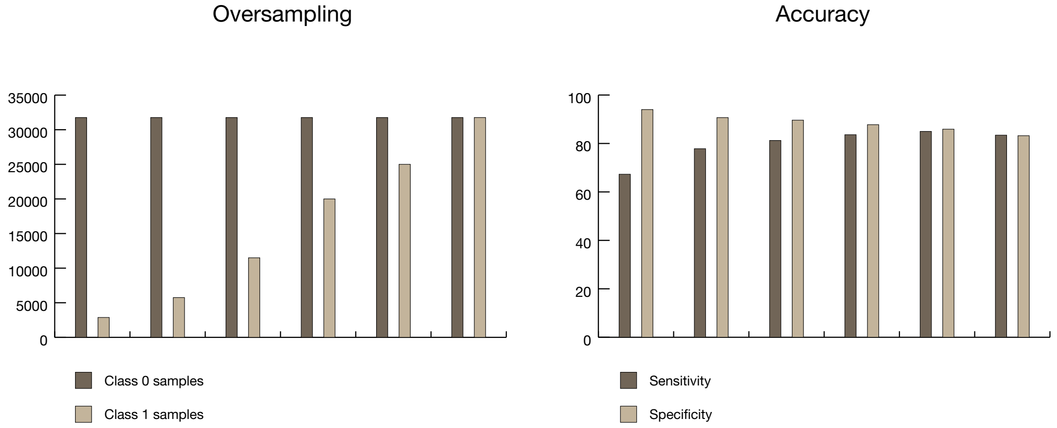

Figure 5.4: Experiment CDCM-A result plot. The tendencies from table 5.1 can be seen here.

Results: The results of the experiment are presented in figure 5.4 and table 5.1. It can be seen how the original dataset without oversampling results in a high specificity but a relatively lower sensitivity. That is, the model is good at identifying negatives but not as good at identifying positives, due to the lack of exposure to the positive class. by increasing the oversampling rate, the sensitivity and specificity alligns more, making the model better for both classes. From the last oversampling rate, which equals the number of samples in the two classes, a small drop in both sensitivity and specificity and therefore also ROC AUC score is seen.

Discussion: With oversampling, the model is exposed to more equal samples during training. This has the upside of equaling the performance for all classes but have the downside of possibly overfitting the model to the training data yielding a drop in performance during test. This is what can be seen from the last 2 oversampling rates where the performance drops (the ROC AUC score begins decreasing). The experiment shows how the third to last oversampling rate which yields 31761 samples of the negative class and 20000 samples of the positive class should be used.

5.3.2 Experiment CDCM-B-1: Learning rate tuning on data central model

Purpose: Examining which learning rate yields the best model performance.

Procedure: During creation of the data central reference model, parameters must be tuned, the first one being the learning rate. The learning rate determines how fast the model converges towards the minima of the loss function. This means that with a higher learning rate, the model converges faster. A tradeoff exists because with a lower learning rate, the model is less likely to overshoot the minima since the steps are smaller. This experiment aim at determining the optimal learning rate in terms of model performance by running the model with different learning rates and keeping track of the model performance in the process.

Learningrate	ROC AUC	Training Time (s)	Testing Time (ms)	Memory usage (MiB)
0,001	86,51	31,17	620,0	217,08
0,005	87,65	30,78	419,0	217,00
0,010	86,84	29,29	391,0	216,76
0,050	79,90	29,11	339,0	216,76
0,100	79,85	28,72	380,0	216,71

Table 5.2: Experiment CDCM-B-1 results. This experiment shows that the ROC AUC score is optimal at a learning rate at 0.005 while the training time only decreases by a small amount. Therefore the optimal value of 0.005 is chosen.

Results: The results from the experiment is presented in table 5.2 and figure 5.5. The training time drops as the learning rate increases. However, this also yields a drop in the ROC AUC score which means a lower accuracy. The test time reveals a drop when increasing the learning rate while the memory usage is kept steady.

Discussion: From the curves presented in figure 5.5, the tendencies in the metrics can be investigated. It is seen how the drop in training time is relatively small by only a few percentages. Because of this relatively small benefit gained within training time, a higher ROC AUC score is preferable.

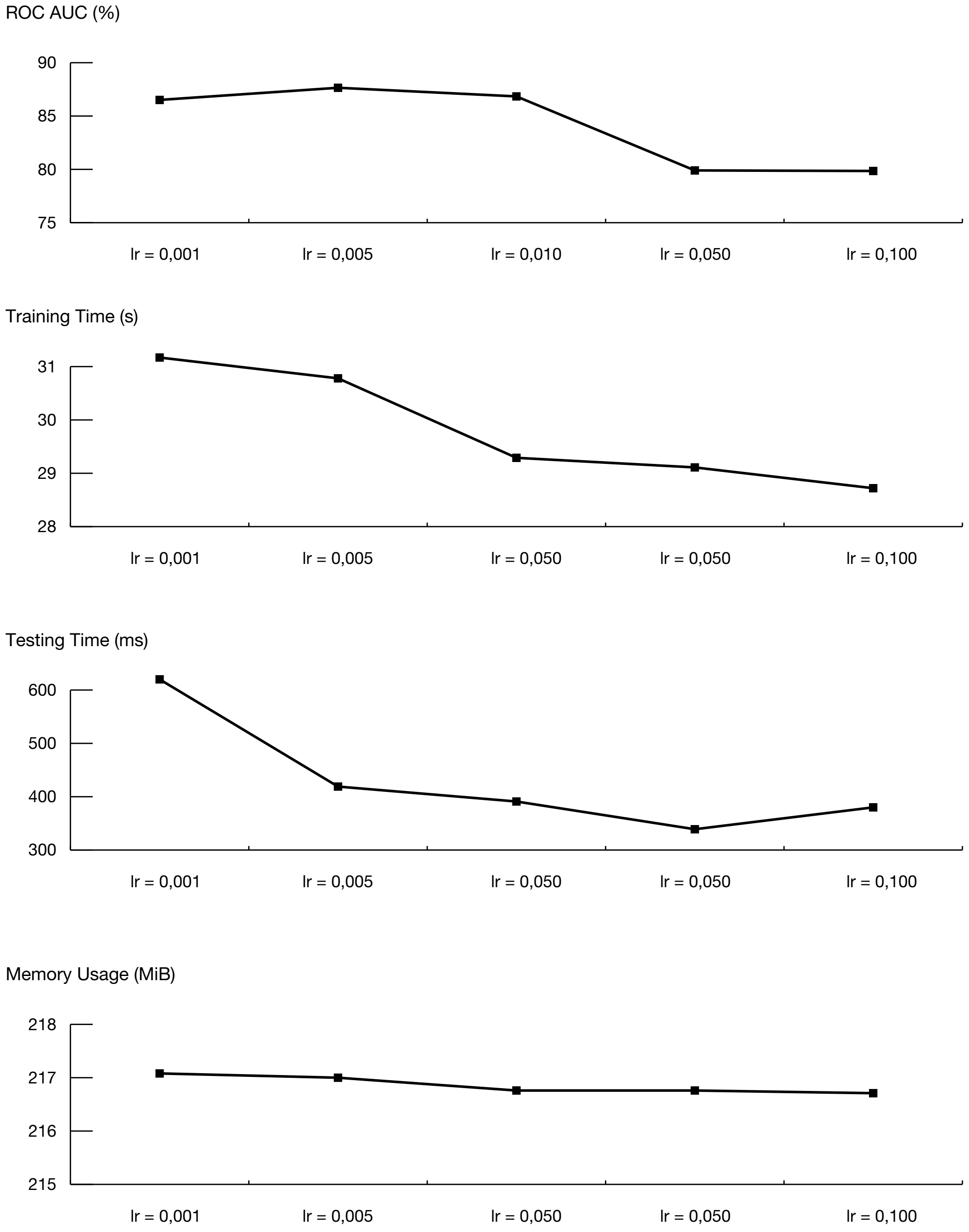

Figure 5.5: Experiment CDCM-B-1 result plot. Illustrates the tendencies from table 5.2

5.3.3 Experiment CDCM-B-2: Batch size tuning on data central model

Purpose: Examining what batch size yields the best model performance.

Procedure: When training a machine learning model, the batch size is a setting determining the number of samples utilized in each training iteration and has large impact on the model performance. With this experiment, the batchsize is changed for different training and test runs to determine the optimal batch size.

Batchsize	ROC AUC	Training Time (s)	Testing Time (ms)	Memory usage (MiB)
2	82,16	150,9	641,0	319,73
16	86,85	93,82	541,0	331,70
32	86,89	67,76	430,0	333,46
64	87,10	42,62	411,0	333,02
128	87,57	30,91	258,0	332,72
256	87,40	27,95	238,0	334,03
512	86,27	20,48	261,0	333,94
1024	86,11	18,10	255,0	332,39

Table 5.3: Experiment CDCM-B-2 results. The experiment shows that the ROC AUC score is increasing along with the batch size but settles after a batch size of 128. The training time follows the same tendencies. 128 is chosen as the optimal batch size

Results: The results from the experiment is presented in table 5.3 and figure 5.6. With an increasing batch size it can be seen how the ROC AUC score generally increases, making the model better at predictions. At the same time, the training time and the test time decreases because the calculations in the training and testing phase must be run fewer times. The memory is kept at a steady level regardless of the batch size.

Discussion: With in increasing batch size the model performance increases both in terms of ROC AUC score and time used in both training and testing. The downside to operating with larger batch sizes is that the updates to the model are less frequently because the computations are heavier with larger batch sizes. This however is compensated by less batches needed. With a smaller batch size it would potentially be possible to stop the model training early because the model updates are more frequent. However, for this baseline model, the performance enhancement is the aim which means that a batchsize of 128 is chosen.

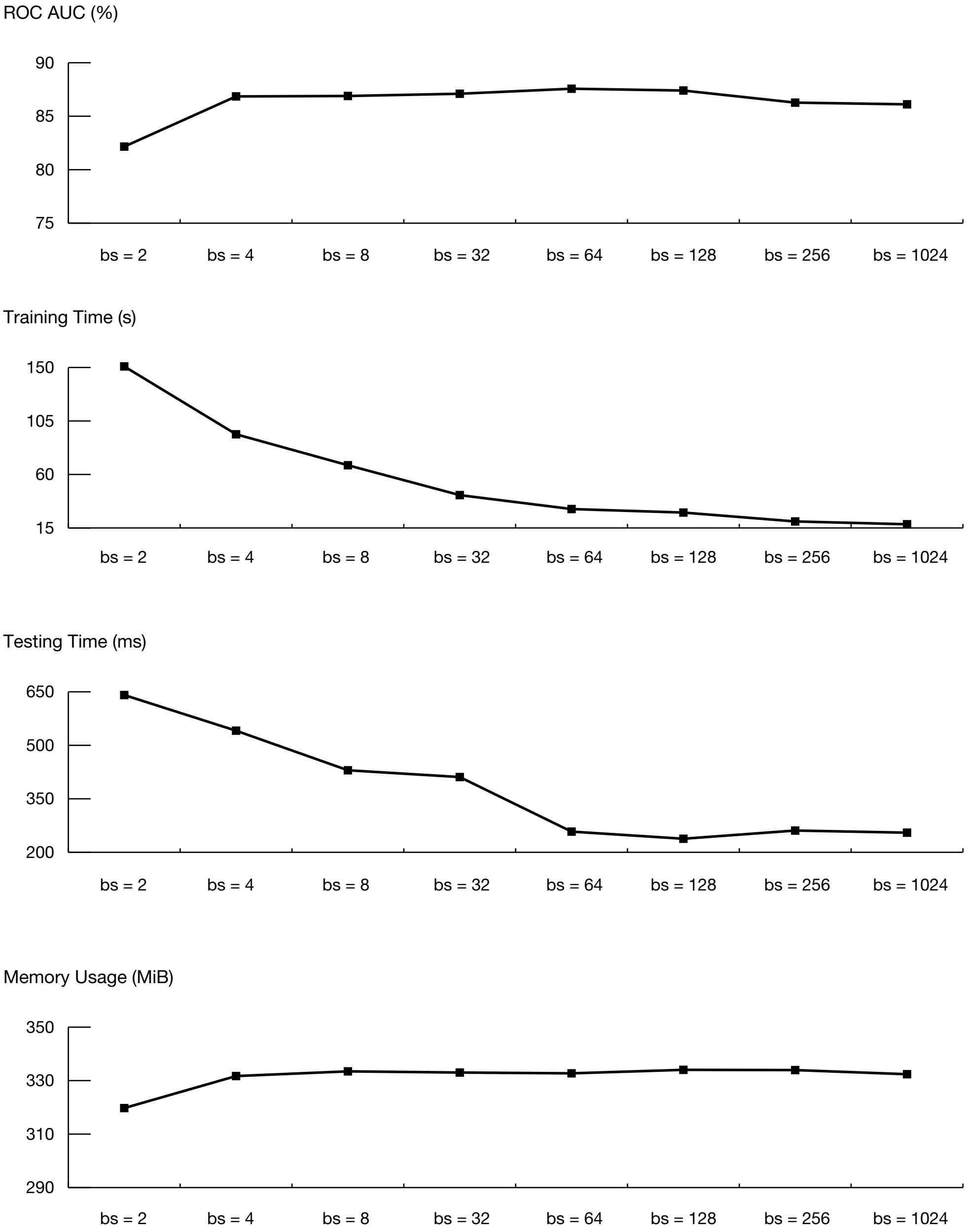

Figure 5.6: Experiment CDCM-B-2 result plot. Illustrates the tendencies from table 5.3

5.3.4 Experiment CDCM-B-3: Epoch tuning on data central model

Purpose: Examining what number of epochs yields the best model performance.

Procedure: When training a machine learning model, the model converges against reaching the minima of the loss function. To reach this minima a certain amount of iterations is needed. When the minima is reached, the model performance will not increase further. This experiment aims at determining the optimal amount of iterations needed to reach the minima. The optimal number of iterations is found when the curves are showing a peak in model performance.

Epochs	ROC AUC	Training Time (s)	Testing Time (ms)	Memory usage (MiB)
2	81,87	3,300	243,0	337,09
4	83,46	6,990	228,0	337,71
6	85,17	11,00	206,0	337,71
8	87,14	13,55	197,0	337,71
10	87,20	17,30	193,0	357,51

Table 5.4: Experiment CDCM-B-3 results. The experiment shows that with an increased amount of epochs the ROC AUC score is increasing with it. The tendency settles at 8 epochs. With more epochs the gain in ROC AUC score is very small, and the cost in training time is large. Therefore the number of epochs is chosen to be 8.

Results: With increasing number of epochs the ROC AUC score is increasing drastically up until 8 iterations. The subsequently increase to 10 iterations reveals a very small increase in ROC AUC meaning the minima is very close to be found. The training time naturally increases along with the increase in iterations. Both the testing time and memory usage is kept very steady during the experiment, which is indented because the number of iterations does not have any effect on these metrics.

Discussion: The ROC AUC score flattens after 8 epochs, yielding only a very small increase in accuracy at further increase in epochs. The experiment could have continued to more epochs, but with the accuracy score flattening and the training time drastically increasing, a compromise of 8 epoch was chosen as the optimal number of iterations.

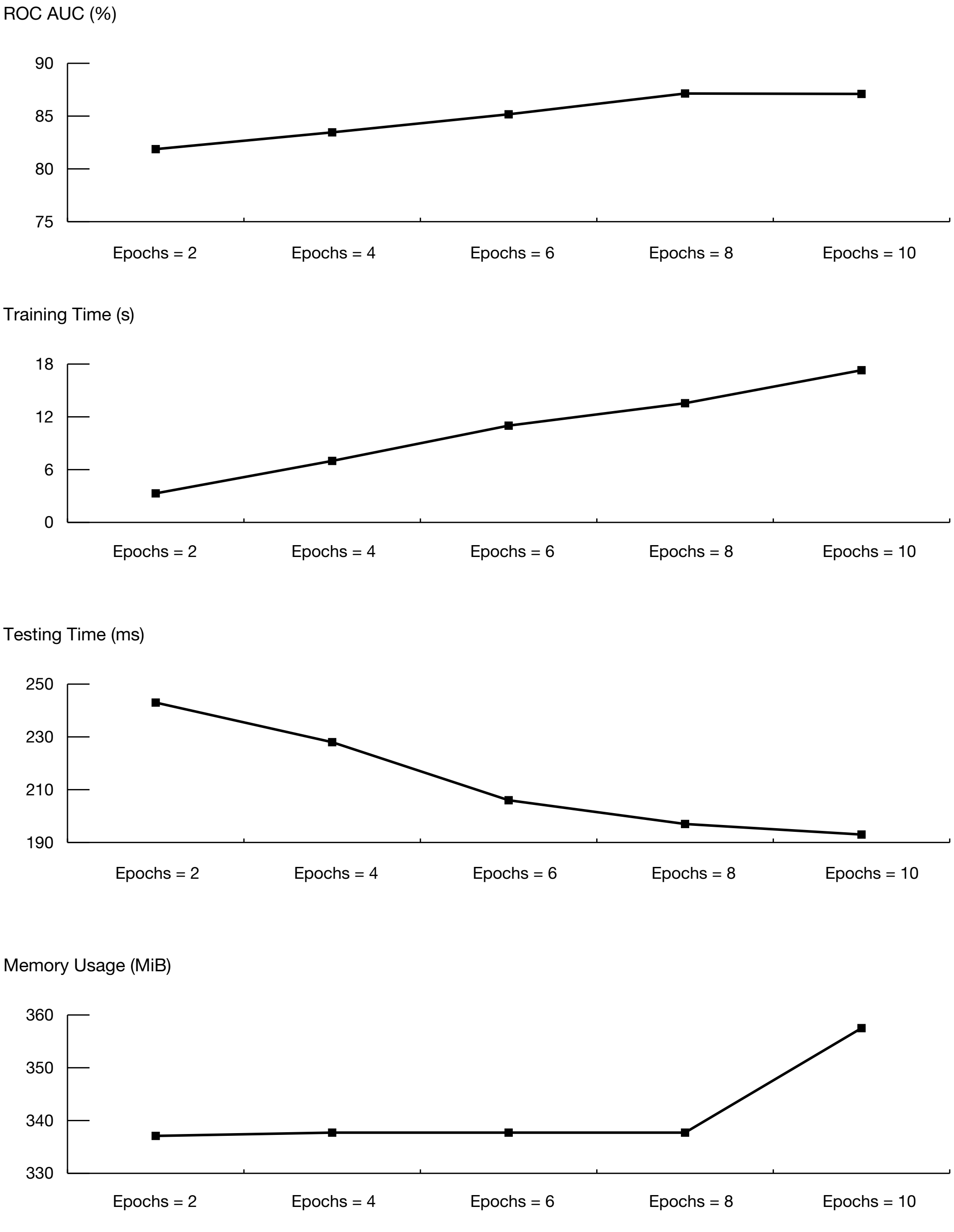

Figure 5.7: Experiment CDCM-B-3 result plot. Illuatrates the tendencies from table 5.4

5.4 Privacy preserving initiatives

This section covers the second phase of the experiments (middle part of figure 5.1). The experiments in this phase is carried out to investigate how well the privacy initiatives, uncovered in chapter 3 perform against a conventional data central model. The output of this phase is on the one hand different privacy preserving machine learning models and on the other, knowledge about how the privacy initiatives affect the model performance.

5.4.1 Experiment PPIN-A: Impact of data sharing initiatives

Purpose: Examining performance decay when using federated learning for decentralized learning.

Procedure: From chapter 3 it is known how federated learning creates a way of training machine learning models locally without sharing data and still benefit from information found at different locations. For this experiment the data is split between municipalities as described in chapter 4. By using the PySyft library 5 virtual machines are created to mimic the local municipalities. Each of these ”municipalities” gets a data share allocated mimicking a local dataset unknown to other ”municipalities”. A local data central model, with the parameters from the first experimental phase is created and allocated at each of the ”municipalities”. Each model can then be trained on local data and transferred back from the ”municipalities” for a local computation of the federated models. The models are combined according to the findings from chapter 3 and tested against the local test dataset.

Federated learning

Epochs	Batchsize	Learningrate	ROC AUC	Training Time (s)	Testing Time (ms)	Memory usage (MiB)
8	128	0,005	77,22	171,60	457,0	784,96

Table 5.5: Experiment PPIN-A results

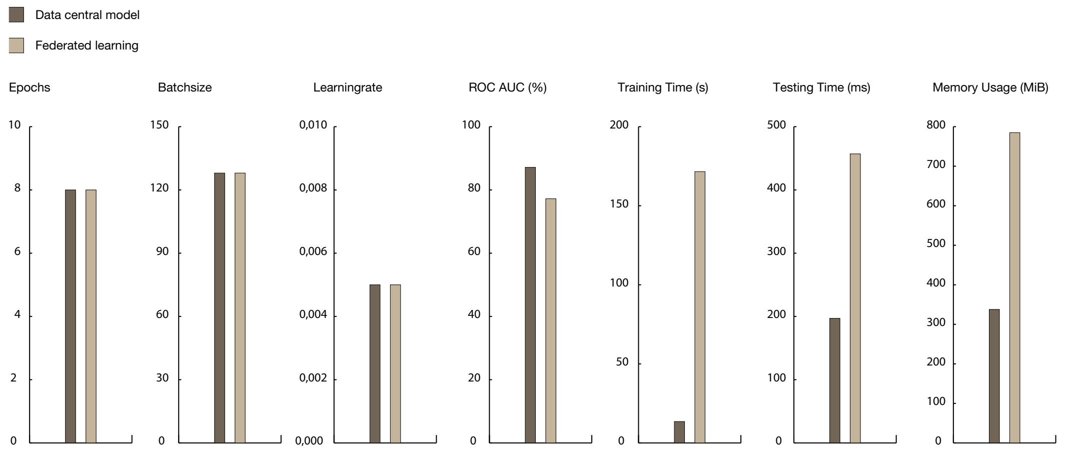

Figure 5.8: Experiment PPIN-A result plot. The experiment reveals how federated learning as privacy initiatives has costs in both ROC AUC score, training time, test time and memory usage.

Results: The results from the test run of the federated model are shown in table 5.5 and figure 5.8. In figure 5.8 the model performance of the federated model is shown along with the corresponding metrics from the data central model. The first three column pairs show how the settings of the two models are equal. The ROC AUC score of the federated model reveals a drop of around 10 percentage points compared to the data central model. The next column pairs reveal a drastically increase in both training time testing time and memory usage.

Discussion: Federated learning has the benefit of decentralized learning where private data are kept locally. However, this benefit comes at a cost with a drop in model performance across all metrics. It is important to note that this drop in model performance is expected, partly because of the averaging process needed to combine models, and partly because of the communication overhead needed to share models between "municipalities". The most drastic drop in performance is with training time. If training time is not a critical factor, the drop in ROC AUC must be considered when using this initiative. Another important note is that these results comes from using PySyft for federated learning. The communication overhead may be lowered if more effective tools can be found.

5.4.2 Experiment PPIN-B-1: Impact of anti-leakage initiatives (ASS)

Purpose: Examining performance decay when using ASS for model security.

Procedure: The first experiment (PPIN-A) covered the model performance of normal federated learning. As previously covered in the proposed solutions in chapter 3, normal federated learning still leaves privacy threats, because the models are shared between municipalities and the feature vectors can possibly be extracted from the models. With ASS, encryption techniques are used to compute an average model without any municipality getting complete knowledge of other models. For this experiment the setup is the same as in PPIN-A where data is shared between "municipalities" by using PySyft. A data central model is allocated on each "municipality" and trained on local data. Instead of transferring the models back for a local computation, the models are shared between all "municipalities" by using ASS as described in chapter 3. This allows for an encrypted computation of the averaged model. This encrypted average model is tested against both the conventional data central model and the normal federated learning model.

Federated learning with additive secret share

Epochs	Batchsize	Learningrate	ROC AUC	Training Time (s)	Testing Time (ms)	Memory usage (MiB)
8	128	0,005	78,56	179,40	477,0	826,54

Table 5.6: Experiment PPIN-B-1 results

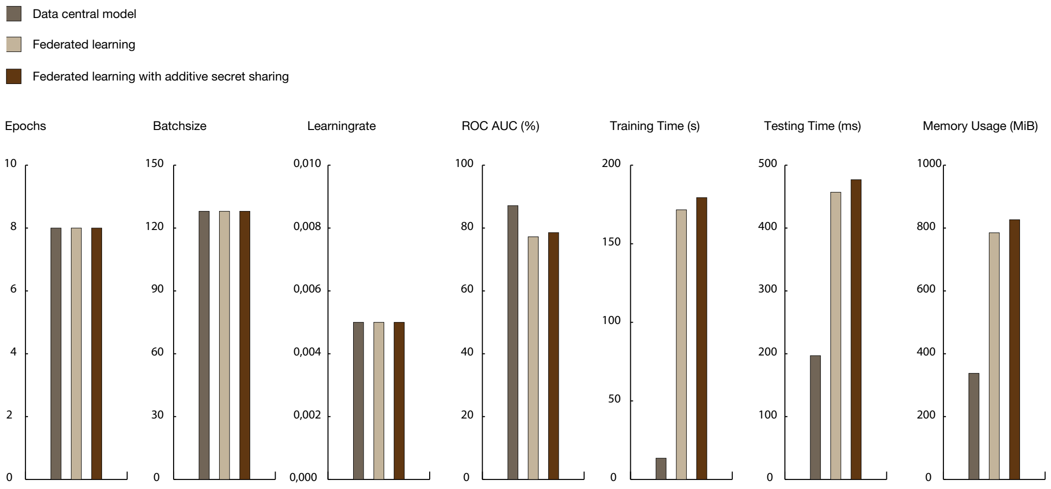

Figure 5.9: Experiment PPIN-B-1 result plot. The experiment shows how the model performance when using ASS is close to equal to the normal federated learning approach. The added privacy from ASS proves to have almost no cost.

Results: The results from the test run of the federated model with ASS are shown in table 5.6 and figure 5.9. In figure 5.9 the model performance of the federated model with ASS is shown along with the corresponding metrics from the data central model and the normal federated model. The general picture of the performance shows that federated learning with ASS performs a little worse than normal federated learning with an increase in both training time, testing time and memory usage; with roughly the same ROC AUC score.

Discussion: By using ASS in the averaging process from normal federated learning, the added security of prevention against model inversion attacks is gained. This comes at a small cost across most model performance metrics. This small cost is expected because the overhead in communication increases when adding ASS. Even though this privacy initiative prevents against a certain threat it still leaves other privacy concerns (covered in the following experiments).

5.4.3 Experiment PPIN-B-2: Impact of anti-leakage initiatives (Differential privacy)

Purpose: Examining performance decay when using differential privacy for model security.

Procedure: As described in chapter 3, the federated learning initiatives does not prevent against all types of privacy threats against machine learning models. If the averaged model is available certain information could still be extracted by adverserials. Differential privacy adds certain amounts of noise to the gradients at each update and clips the gradients to avoid overfitting against certain samples. This noise enables the model to raise a question of whether the output from a query to the model was the actual output computed from sensitive information or if the result was skewed by noise. From this experiment the model performance is tracked at various amount of noise used in a differentially private model. The differentially private model is created by adding a privacy engine from Opacus to the normal data central model. This privacy engine takes gaussian noise and uses it in the training process. The noise level is controlled by what is referred to as a noise multiplier. Model performance is tracked at different levels of the noise multiplier.

Noise Multiplier	ROC AUC	Training Time (s)	Testing Time (ms)	Memory usage (MiB)	Epsilon
0,1	86,63	131,9	276,0	337,81	9058,2
0,3	82,39	134,6	427,0	382,65	55,705
0,5	80,14	136,0	293,0	383,14	9,6400
0,8	79,32	133,2	281,0	383,11	2,3310
1,5	78,30	135,9	421,0	383,41	0,6666

Table 5.7: Experiment PPIN-B-2 results. The experiment shows how added privacy from differential privacy has a clear cost in ROC AUC score while other metrics are kept steady. Note that the leaked ε increases drastically when the noise is lowered.

Results: The result can be seen in table 5.7 and figure 5.10. It can be seen how the differential privacy initiative does not have any serious effect on neither training time, testing time or memory usage. The last two metrics from table 5.7 however, is interesting. The general tradeoff between accuracy and privacy uncovered in chapter 3 can be seen from the results. When the noise multiplier is raised both the ROC AUC score and the ε value are decreasing, yielding lower accuracy but better privacy. An important note is that the δ value in (ε, δ)-differential privacy is kept at $1 * 10^{-5}$, regardless of the level of noise added and thereby the level of ε.

Discussion: Differential privacy affects the model performance. With increasing amounts of noise added the promise of privacy is getting stronger while the accuracy is getting worse which yields a tradeoff. The level of privacy needed, must be uncovered before employing this privacy initiative. That question, however, is very hard to answer. The required level of ε depends on both the data, the features and the desired probability of getting the correct answer. As previously discussed in chapter 3 ε is calculated based on the sensitivity of the data. The sensitivity changes when looking at different features and more importantly changes when new datasets with corresponding data are made available. That is because differential privacy promises its privacy regardless of what other information is available. When sensitivity is calculated, the maximum distance from the same query against adjacent databases is calculated but finding "the other database" can be very hard since no one knows exactly what other data is available in the world. Because of this problem, ε should often just be used as a relative measure of the privacy provided and the level of noise used in DPSGD should be set based on the required accuracy.

From figure 5.10 the decreasing values of both ROC AUC and ε can be seen. It is important to note the scale of the y axis in the ε plot. This is a logarithmic scale, in order to fit all the values in the same plot. The ε value is increasing very drastically when the noise is lowered.

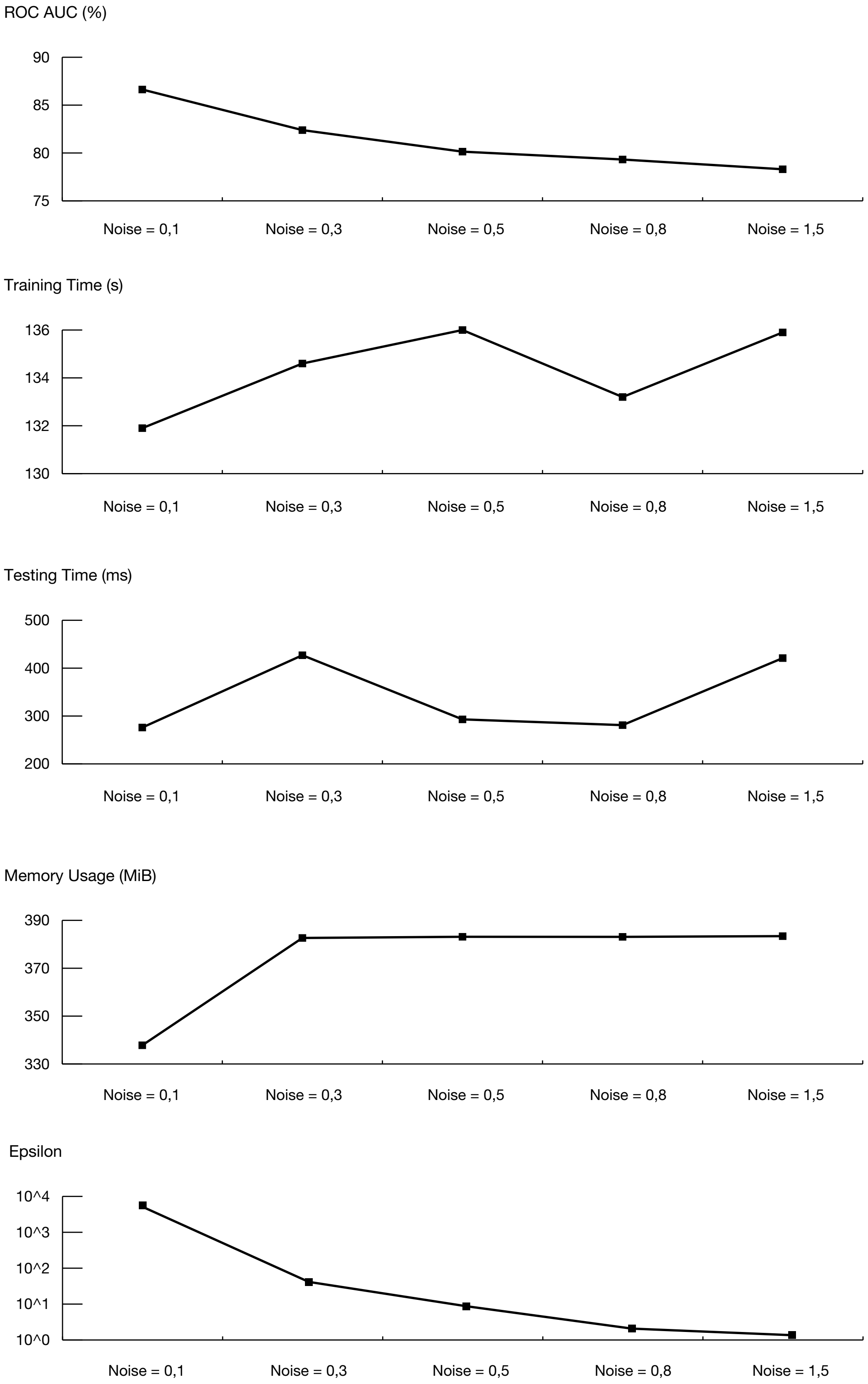

Figure 5.10: Experiment PPIN-B-2 result plot

5.4.4 Experiment PPIN-B-3: Impact of anti-leakage initiatives (SMPC)

Purpose: Examining performance decay when using SMPC for model security.

Procedure: As described in chapter 3 SMPC is another approach to gaining benefits from other municipalities data without directly sharing data. With this experiment, the same procedure as in experiment PPIN-B-1 is used. That is, ASS is used, to divide information between municipalities for jointly computations. The difference from experiment PPIN-B-1 is that with SMPC both data and models are shared with ASS, meaning all computations of both predictions, loss calculations and averaging are jointly computed. By using PySyft, models and data are shared between municipalities and a model is jointly trained. The model performance is tested in the same way as with previous experiments.

Results: Due to limitations in RAM on the PC used for the experiments, SMPC with PySyft cannot be run, and therefore no results are achieved.

Discussion: Due to the sharing of large amount of data between virtual machines in PySyft, not enough RAM was available. This was the case when using PySyft, but other frameworks might offer more streamlined operations, allowing these SMPC setups to be run. For this thesis work, the solution with SMPC will not be further covered because no results were achieved.

5.5 Machine learning explainability

This section covers the third phase of the experiments (right part of figure 5.1). The goal of the experiments is to determine how explainable the data presented in chapter 4 is and how the different privacy initiatives affect this potential explainability. The explainability is found by calculating SHAP values and plotting these for each of the features in the dataset.

5.5.1 Experiment MLEX-A: Explainability of feature vectors

Purpose: By using SHAP framework, examine if the feature vectors in the dataset are explainable.

Procedure: With the conventional dataset presented in chapter 4, a fixed amount of 128 samples, that is one batch, is extracted from the test set and the SHAP values for each feature is calculated for each sample. These values are then plotted with the twenty most significant features based on the distribution of SHAP values for all the samples. After plotting, the SHAP values are investigated based on the feature and the HMI number for the ATS features. The explainability is determined by a qualitative analysis based on the intuitive understanding of how the different features should affect the probability of falling. (An example could be that a walking stick would indicate an increase in probability of falling while assistive devices for better writing would indicate the opposite).

Feature	Description	Feature	Description
1: NumberAts	Number of devices	11: 1803	Assistive tables
2: 0433	Devices to keep the tissue intact	12: 0907	Body stabilizing devices
3: 2227	Devices for surveliance	13: 1809	Assistive sitting devices
4: 0933	Devices for assisting baths	14: 1206	Walking assistive devices
5: 1222	Manual wheelchairs	15: 1810	Accesories for assistive sitting devices
6: 0912	Devices for assisting toilet visits	16: 2421	Devices for extended reach
7: Gender	Gender	17: 0906	Body-worn protection
8: 2803	Assistive furniture	18: 2230	Reading assistive devices
9: 1812	Assistive beds	19: 2218	Assistive devices for sound
10: BirthYear	Birthyear	20: 3018	Assistive devices for images

Table 5.8: Description of 20 most significant SHAP features

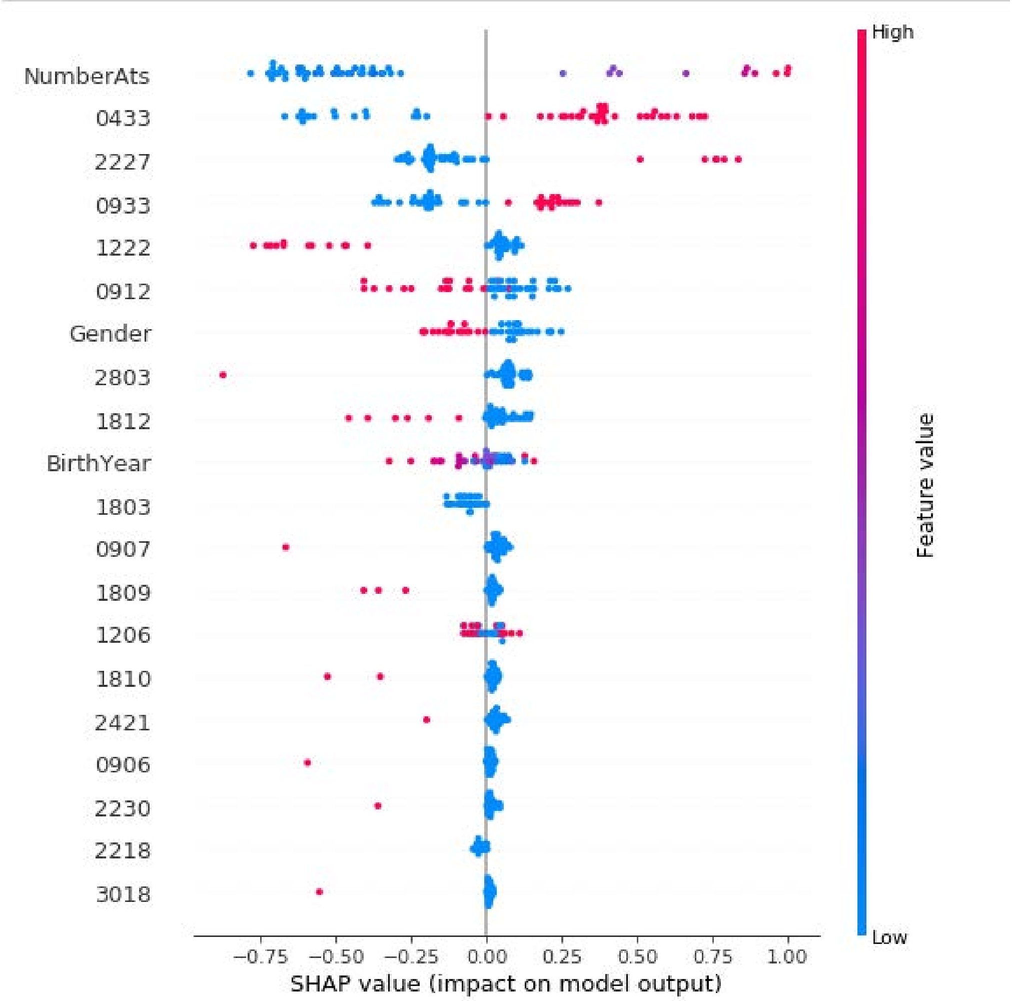

Figure 5.11: SHAP summary plot of 20 features with 128 samples. The plot shows how the features of the dataset are clearly explainable

Results: The results can be seen from the SHAP summary plot presented in figure 5.11. The 20 most significant features for determining the probability of a fall is presented in table 5.8.

Discussion: The SHAP values presented in figure 5.11, reveals that by intuitive understanding, the features are highly explainable. The most significant feature is the number of assistive devices, if you have many devices allocated, you are likely to fall. The second most significant feature is devices to keep the tissue intact, which could be special pillows for sitting or lying in bed. These devices would only be allocated to people who spend most their time not standing up, which indicates high risk of falling when they actually do stand up. Further down the list, features like assistive devices for sound and images are found, which intuitively does not affect the probability of falling much.

5.5.2 Experiment MLEX-B: Privacy preserving mechanisms impact on explainability

Purpose: Examine if / how the privacy initiatives from experiments PPIN-A and PPIN-B affect the explainability.

Procedure: With the same dataset that was used for the calculations of SHAP values for experiment MLEX-A, SHAP values are calculated for the machine learning models which were trained with privacy preserving initiatives. Two models are chosen for investigating the effect of privacy initiatives on the explainability. The two models are the normal federated learning model and the differential privacy model. By examining the SHAP values and the corresponding summary plots for the privacy models once again a qualitative analysis is carried out to determine whether the privacy initiatives have any effect on the explainability.

Feature	Description	Feature	Description
1: NumberAts	Number of devices	11: 2230	Reading assistive devices
2: 0933	Devices for assisting baths	12: 0912	Devices for assisting toilet visits
3: BirthYear	Birthyear	13: 1231	Devices for changing positions
4: 2227	Devices for surveliance	14: 1206	Walking assistive devices
5: 0433	Devices to keep the tissue intact	15: 0436	Devices for perception training
6: 1812	Assistive beds	16: 1222	Manual wheelchairs
7: 1212	Assistive devices for cars	17: 0909	Devices for assisting clothing
8: 1203	Walking assistive device (one arm)	18: 1809	Assistive sitting devices
9: 1810	Accesories for assistive sitting devices	19: 1803	Assistive tables
10: Gender	Gender	20: 3018	Assistive devices for images

Table 5.9: Description of 20 most significant SHAP features federated learning

Feature	Description	Feature	Description
1: NumberAts	Number of devices	11: 0427	Stimulators
2: 1222	Manual wheelchairs	12: 3018	Assistive devices for images
3: BirthYear	Birthyear	13: 1810	Accesories for assistive sitting devices
4: 2803	Assistive furniture	14: 0933	Devices for assisting baths
5: 0907	Body stabilizing devices	15: 1231	Devices for changing positions
6: 0912	Devices for assisting toilet visits	16: 1203	Walking assistive device (one arm)
7: 2227	Devices for surveliance	17: 0433	Devices to keep the tissue intact
8: 1236	Devices for lifting people	18: 2224	Devices for distance communication
9: 1206	Walking assistive devices	19: 0930	Urin absorbing devices
10: 0436	Devices for perception training	20: Gender	Gender

Table 5.10: Description of 20 most significant SHAP features differential privacy

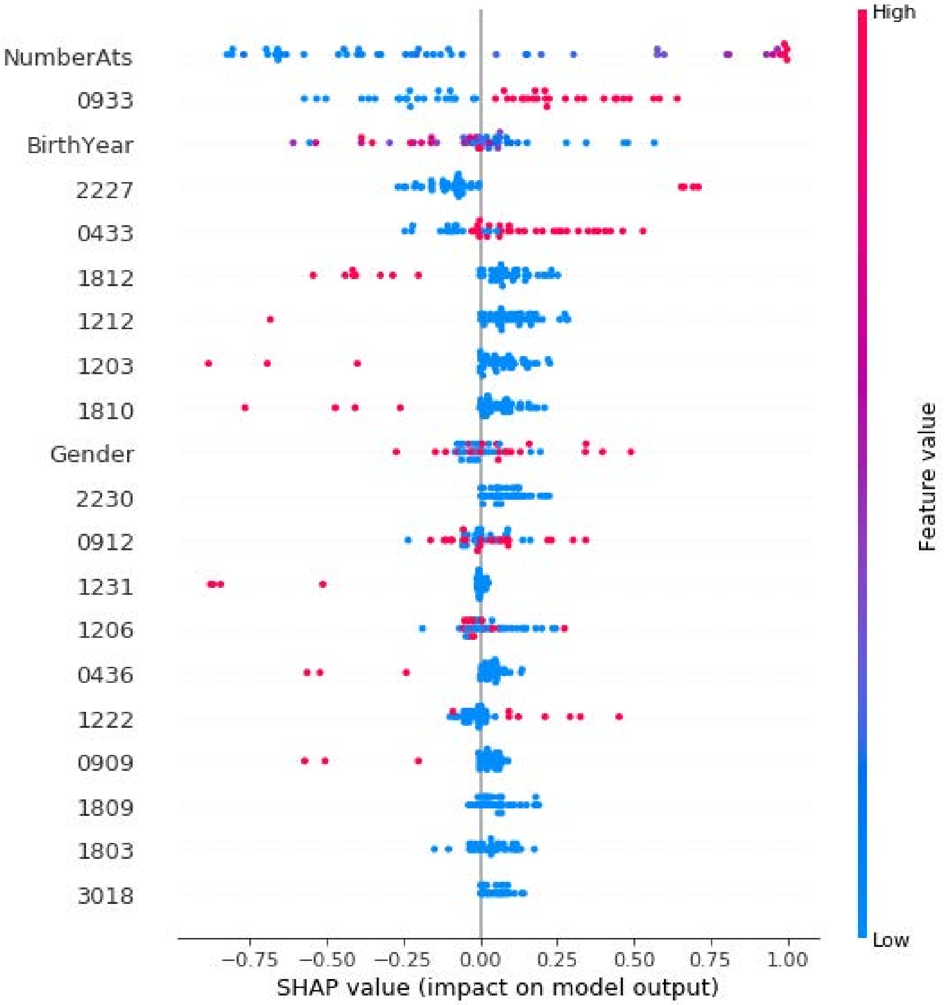

Figure 5.12: SHAP summary plot of 20 features with 128 samples for federated learning model

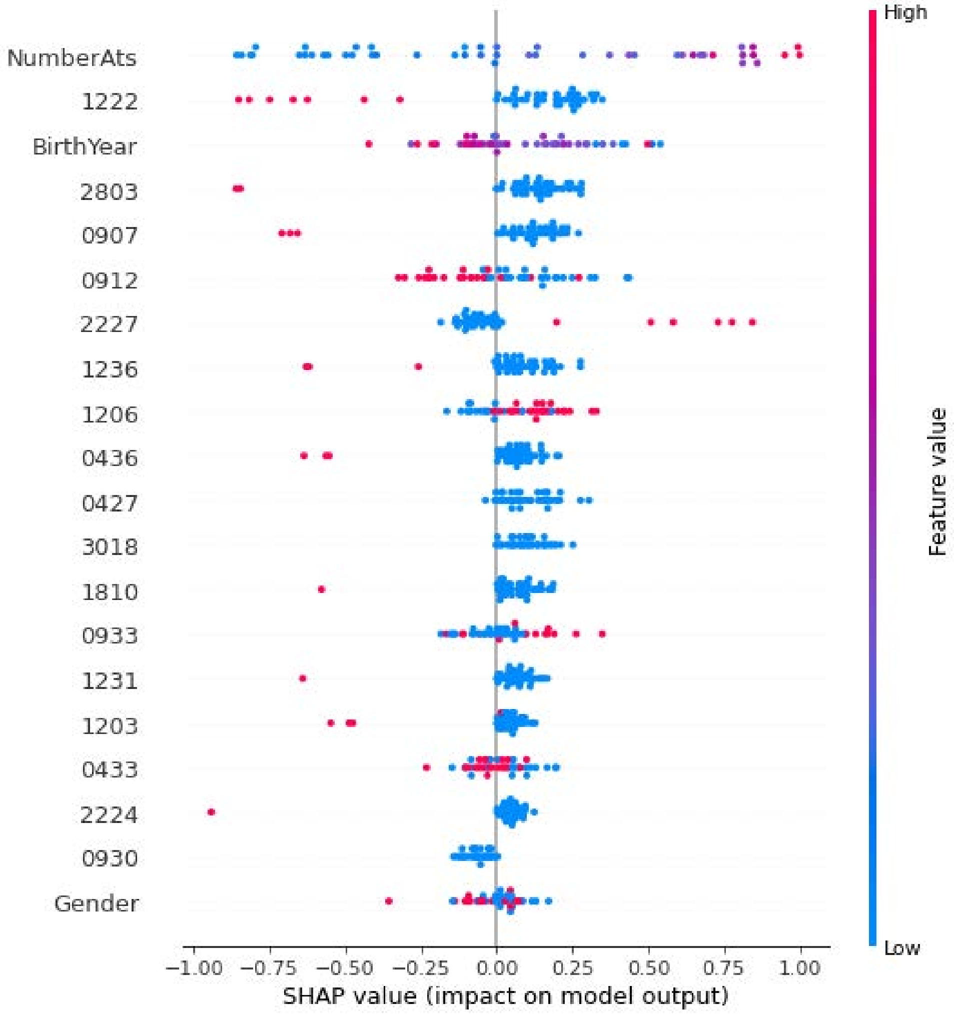

Figure 5.13: SHAP summary plot of 20 features with 128 samples for differential privacy model

Results: The results can be seen from the SHAP summary plots presented in figure 5.12 and 5.13. The 20 most significant features for determining the probability of a fall is presented in tables 5.9 and 5.10.

Discussion: From the results of this experiment it can be seen how the privacy initiatives does influence the explainability of the machine learning models. This is due to the fact that the features included in the 20 most significant SHAP features changes when using privacy models. However, it is interesting to note how federated learning does not affect the model nearly as much as differential privacy. An illustration of this can be seen from figure 5.14. It shows that the conventional data central SHAP features has 4 of the first 5 features in common with the federated learning SHAP features. Contrary, the differential privacy model only has 2 of the first 5 SHAP features in common with both the data central SHAP features AND the federated learning SHAP features. This is due to the fact that federated learning does not change the model as much in the averaging process as differential privacy does with noise addition. If explainability is important for the use case, this result should be taking into consideration.

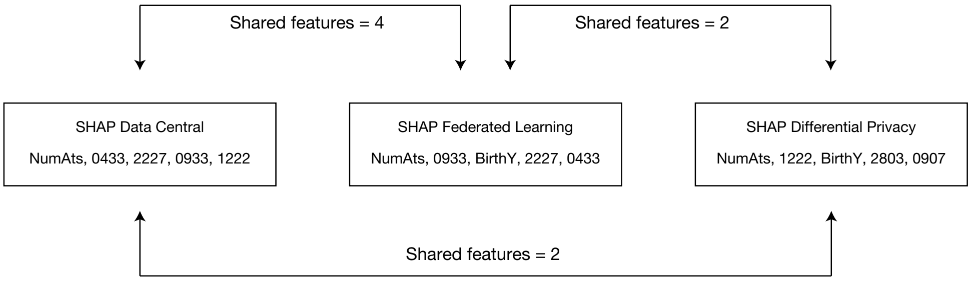
Shared features = 4
Shared features = 2
SHAP Data Central
NumAts, 0433, 2227, 0933, 1222
SHAP Federated Learning
NumAts, 0933, BirthY, 2227, 0433
SHAP Differential Privacy
NumAts, 1222, BirthY, 2803, 0907
Shared features = 2

Figure 5.14: Illustration of the shared features for data central SHAP values, federated learning SHAP values and differential privacy SHAP values

Chapter 6

DISCUSSION

This chapter discusses and evaluates the proposed solutions for a privacy preserving machine learning approach described in chapter 3, based on the outcomes from the experimental phase carried out in chapter 5. The evaluation is made based on the performance of the different solutions within the domain.

The first proposed solution, data central deep learning, outlies the exact problem domain which lead to this thesis. Data cannot be shared between municipalities due the general data protection regulations. This fact states how a data central approach is not possible when handling sensitive information and therefore the demand for privacy preserving initiatives arises.

The second solution, conventional federated learning, is based on the idea of creating models locally and sharing the model parameters instead of sharing data. The experiment PPIN-A demonstrated this approach where a machine learning model was shared between municipalities. However, it was also shown from PPIN-A how the federated learning approach comes with both limitations and disadvantages. The performance of a federated learning model yields a decay in both accuracy and time consumption. For the health care domain, the performance decay in processing time may not be critical because the model training only happens when a new dataset becomes available. This would be limited to, at most, a few times a year, which means a higher processing time would be acceptable. The performance decay on accuracy, however, would be more critical. With a lower accuracy, fall risk assesment would be more error prone. This could ultimately lead to deficient fall prevention.

The third solution, federated learning with DPSGD, covers one of the core limitations with federated learning. Federated learning models might still reveal sensitive information because of the risk of model inversion attacks. With the use of differential privacy, this risk is lowered. From experiment PPIN-B-2 it was shown how differential privacy can be added to the machine learning model training phase to allow for a more secure model. PPIN-B-2 revealed how the amount of noise used in training creates a tradeoff between privacy and accuracy. This means that the privacy concern must be analyzed before deploying models with differential privacy as privacy initiative. However, one of the key results from PPIN-B-2, is that when increasing the noise, the decrease in accuracy is way smaller than the increase in privacy. That means that if a minor drop in accuracy is not critical, a large gain in privacy is possible.

A core limitation with the privacy guarantee stated from differential privacy is the exhaustion of the privacy budget. These privacy machine learning models, would upon deployment be made available for unlimited use by various medical staff and caretakers. The unlimited use makes a potential exhaustion attack possible for an adversary with access to the model. By making large number of queries against the model, the noise distribution can be subtracted from the answers and reveal sensitive information. If differential privacy were to be used, this concern must be considered.

The fourth solution, federated learning with ASS , also covers a limitation with federated learning. Instead of sharing the models between municipalities which potentially leads to sharing of information because of model inversion attacks, ASS offers an encryption-based approach. From experiment PPIN-B-1 it is shown how the addition of ASS does not affect the model performance much in any of the tracked metrics, compared to conventional federated learning. This finding is important because it states that if federated learning is chosen as a privacy initiative, then the addition of ASS should definitely also be considered.

From the last experimental phase with experiments MLEX-A and MLEX-B it is shown how it is possible to make complex deep learning models explainable by calculating SHAP values. This is an important finding, since it enables caretakers to make a more qualitative approach when communicating results from the models. Instead of only being able to tell a citizen that they must engage in a fall prevention program because the system states they should, they are able to tell the citizens the exact reason behind it. Which factors in their lives has them placed in a given risk group? This ability to explain how a system is working, will increase the trust in the system and potentially the results found from using it.

From experiment MLEX-B it is important to note how differential privacy has a larger effect on the explainability than the case is for federated learning approaches. Since the explainability is an important factor within the domain, this should be taken into consideration when choosing privacy initiatives.

Chapter 7

IMPROVEMENTS

This chapter presents areas within privacy preserving machine learning that with future work could be explored to improve the results, presented and evaluated in chapter 6. The areas covered in this chapter is found based on limitations discovered during the experimental phase in chapter 5. That means that the areas are outside the scope of this thesis work.

7.1 Larger batch size for DPSGD

During experiment PPIN-B-2 the use of differential privacy as privacy initiative was explored. The findings showed how the addition of noise in the stochastic gradient decent step resolved in better privacy protection, at the cost of accuracy and time consumption. The time consumption for DPSGD was found to be increased by a factor of 10 for the training phase. This increased training time could potentially be critical for the usage of differential privacy as privacy initiative.

In order to improve time consumption, the algorithm for DPSGD presented in chapter 3 is analyzed. The algorithm states how a batch of training samples is taken from the training set after which every sample within the batch has a gradient computed with the addition of noise. The reason for the increased time consumption with differential privacy is because each sample must be computed individually, which basically corresponds to a batch size of 1. If the algorithm for DPSGD could be extended to handle noise addition on larger batch sizes the time consumption and accuracy could be improved.

7.2 Renyi differential privacy

During the uncovering of differential privacy in chapter 3 one of the key limitations of differential privacy was found to be the potential exhaustion of the privacy budget. This means that if a model is available for numerous queries the privacy promise will eventually brake. The solution to this limitation is to disallow for queries when the budget is emptied.

However, use cases might exist where the denial of numerous queries could make the system unusable. A new definition of differential privacy has been discovered in [29] named Renyi Differential Privacy. Renyi differential privacy proposes a new definition with a relaxation of differential privacy based on Renyi Divergence. In [29] they

”demonstrate that the new definition shares many important properties with the standard definition of differential privacy, while additionally allowing tighter analysis of composite heterogeneous mechanisms”. This could potentially lead to a larger privacy budget because of the tighter bound under composite mechanism.

7.3 Federated matched averaging

During experiments PPIN-A and PPIN-B-1 it was uncovered how federated learning does deliver a method for decentralized learning, but that it comes with a cost in terms of model performance. This reveals that the idea behind federated learning is very promising, but there are areas which could be improved.

The main reason for a decrease in accuracy is the averaging process. In order to improve this process, a new method has been proposed in [40] called Federated Learning with Matched Averaging (FedMA). This method, according to the authors, ”indicate that FedMA not only outperforms popular state-of-the-art federated learning algorithms on deep CNN and LSTM architectures trained on real world datasets, but also reduces the overall communication burden.”

By exploring FedMA the federated learning approach could potentially reduce the drop in model performance compared to a data central approach, leading to more applications where federated learning could be applicable.

7.4 Super convergence

From experiments PPIN-A, PPIN-B-1 and PPIN-B-2 it is seen how privacy initiatives, no matter which, does affect the time consumption of both training and testing machine learning models. As previously covered, this could potentially lead to privacy initiatives being deferred.

A phenomenon which was named super-convergence has been discovered in [36]. Super-convergence is a method for adjusting the learning rate during training in a way that allows for much faster convergence. According to the authors ”neural networks can be trained an order of magnitude faster than with standard training methods”. If super-convergence was used in collaboration with some of the privacy initiatives presented in this thesis, perhaps the model performance would not decrease as much, and the privacy models might be better suited for time critical use cases.

Chapter 8

CONCLUDING REMARKS

In this thesis the impact of privacy preserving initiatives on machine learning training has been evaluated against sensitive information acquired from the Danish healthcare system. In this chapter the found results will be evaluated against the thesis goals presented in chapter 1. Furthermore, this chapter will present my personal outcome from the thesis work, as well as list the contributions presented throughout the thesis work.

8.1 Achieved results

The thesis goals which was presented in chapter 1 were:

> **Goal 1:** Provide a method for sharing data between municipalities without compromising the privacy of the individual data samples.
>
> **Goal 2:** By using privacy preserving machine learning techniques, provide a method for ensuring that sensitive information is not leaking from the trained models.
>
> **Goal 3:** By using privacy preserving machine learning, provide an assistive tool to be used for caretakers, which explains the conclusions made from the machine learning models.

8.1.1 Goal 1

During the uncovering of privacy preserving machine learning fundamentals, the concept of federated learning was investigated. During both chapter 3 and 5, the theory behind federated learning was elaborated and the performance was tested. The results state that federated learning does offer possibilities for training machine learning models on distributed datasets without the need for sharing information between entities, in this case municipalities. When the data is not shared between the municipalities the general data protection regulations are not violated and the privacy of the sensitive information in each municipality is not compromised. The results, however, also showed that the use of federated learning does reveal a performance drop, most critically in the terms of accuracy and time consumption.

8.1.2 Goal 2

During the uncovering of fundamentals of privacy preserving machine learning, several initiatives for ensuring model security was uncovered. The key outcome was that differential privacy or the addition of ASS for federated learning was the optimal concept for model security. SMPC was also investigated, but due to limitations in computing power, experiments could not be carried out and the concept was therefore not further discussed.

The results showed how the addition of ASS does provide a method for securing the federated model against model inversion attacks, without decreasing the model performance of conventional federated learning significantly. Furthermore, the results showed how differential privacy can be used to add privacy to machine learning models, but at the cost of accuracy. The privacy delivered by differential privacy correlates with the drop in model performance. The findings also states how the concept of differential privacy can be hard to implement due to the limitations in knowledge from the world. Without access to exact knowledge on what correlating data is available, the needed value of ε cannot be determined. This leaves differential privacy as a trial and error approach.

With both federated learning using ASS and differential privacy, two different methods for ensuring that models do not leak sensitive information is provided. The federated learning approach makes sure that the other entities in the network cannot access private models, while differential privacy ensures that the models are not exposed even to the users of the system. That leaves the choice of privacy initiatives to a tradeoff between model exposure and model performance.

8.1.3 Goal 3

During the uncovering of fundamentals of privacy preserving machine learning, a concept called SHAP was investigated. SHAP delivers a method for gaining explainable values for each feature in the deep learning model and by that indicating which features are "dragging" the model in which direction. In the experimental phase, SHAP was used to calculate explainable values for the features in the dataset. The experiments showed how this method delivers explainability even for complex deep learning models. However, the experiments also showed how the privacy initiatives does influence the explainability, because they potentially can change which features are most important when making predictions.

With the SHAP calculations an assistive tool is provided which can be used on any machine learning model, in order to gain explainability. With this tool, the conclusions made from using machine learning models can easier be communicated, towards non-technical people.

8.2 Personal outcome

From this thesis work I, personally, have gained better technical understanding in several areas (machine learning, privacy, data handling) as well as better insights into the challenges of handling sensitive information. Many of the techniques used in the technical world today are not taking privacy into account. Due to a more and more strict definition of privacy, and stronger and stronger regulations within privacy, it is very much something that need to be considered more.

From the thesis work I have had to work in close collaboration with both my supervisor at the university, and the people from DigiRehab, who delivered domain knowledge and data. From the close collaboration I have gained a deeper insight into how to implement an optimal workflow with people from the industries.

Lastly, this thesis work was carried out as a part of AIR, a large research project at the university. This project constellation have helped me gain insights into how research is carried out at a larger scale, and how different entities can work together as a part of research.

8.3 Contributions

In this section the most significant contributions from the work carried out in this thesis is presented:

- The definition of an implementation on how one-hot encoding can be used on DigiRehab provided datasets to prepare them for machine learning algorithms. This allows for easier data manipulation for future projects provided with DigiRehab data.

- An in-depth analysis on the concept of federated learning presented with the potential use cases and limitations. It was found that federated learning does offer the possibility of training machine learning models on decentralized datasets but that it comes at the cost of a decrease in model performance.

- An in-depth analysis on how differential privacy works, along with its potential use cases and limitations, found from technical experiments. It was found that while differential privacy can deliver better privacy protection, it comes at the cost of decreased model performance, and complex model configurations.

- An investigation of how the encryption based method ASS, can be used to avoid potential model inversion attacks with federated learning. It was found that ASS offers higher model security and that it comes at nearly no performance decrease.

- It was shown how model explanation is possible, even for complex deep learning models, by calculating SHAP values and presenting these along with the model features. This allows for transparency of the machine learning models.

- It was presented how 4 different ideas (Larger batch size for DPSGD, Renyi Differential Privacy, Federated Matched Averaging and Super Convergence) potentially can improve on some of the limitations found from previous contributions.

8.4 Recommendations

Based on the conclusions drawn in section 8.1, recommendations can be made for organizations who wants to embed privacy initiatives into their machine learning setup. These recommendations are established based on the empirical experience gained from the thesis work.

The type of privacy initiative to choose for better data protection, depends highly on the potential threats and / or the regulations that must be met. If regulations state that data can't be shared between entities and this sharing is a necessity, federated learning is an opportunity for gaining the same benefits as if data was shared, without breaking regulations. It must be kept in mind how federated learning does lower performance of the machine learning setup, but if the performance decay is within acceptable range, it is recommended to use this privacy initiative.

If regulations instead state that even machine learning models cannot be shared due to the risk of model inversion attacks, other measures must be considered. The addition of ASS encryption techniques offer an increased protection, because models are never fully shared between entities. This initiative comes at a low performance cost and it is therefore recommended to use this technique together with federated learning.

The addition of ASS does leave exposure against membership inference attacks and, if unlimited access is available, model inversion attacks. If these threats are critical to the system operations, differential privacy can be used. Differential privacy adds plausible deniability, meaning an adversary can't be certain that received information is correct. However, differential privacy is a complex configuration, and the needed privacy protection can be challenging to calculate. Therefore, the recommendation is to only use differential privacy, if regulations clearly states that other initiatives are insufficient.

Furthermore, differential privacy affects the explainability, which means that if explainability is important for the system operations, the use of differential privacy should be considered even more.

Appendix

Appendix A

DICTIONARY

In this appendix a dictionary is presented, explaining domain specific terminology.

Assistive device: A device delivered to citizens embedded in a healthcare program. These devices help citizens with tasks that their personal abilities no longer can handle.

Caretaker: A broad definition of personal embedded in a healthcare program.

HMI index: A reference number for assistive devices. Original resolution is 8 digits. For this thesis a resolution of 4 digits is used.

Municipality: A section of the Danish society. All municipalities are driven under the same regulations but work as separate entities.

Healthcare system: The Danish system for initiating healthcare programs and delivering personal.

Healthcare program: A specific program enabled to help a citizen with certain task(s).

SOSU worker: A more specific definition of a caretaker. SOSU workers are personal who visit citizens and help them with their currently assigned healthcare program.

Rehabilitation: A process of bringing a citizen ”back to normal standards” after illness or accidents.

Training program: A program used in the rehabilitation process to help strengthen the citizen.

Sensitive data: Data which is protected under the GDPR regulations. Sensitive data cannot be shared between municipalities.

Datacenter: An entity apart from the municipalities. Could be a university or a server company.

Additive Secret Sharing (ASS): An encryption technique used for shared computation of functions.

Secure MultiParty Computation (SMPC): An encryption scheme based on ASS for complete encryption of data and computation.

Differentially Private Stochastic Gradient Decent (DPSGD): Implementation of stochastic gradient decent with addition of noise and gradient clipping.

General Data Protection Regulation (GDPR): European data regulations. Limits how to access, collect and data.

Epoch: Each run of during machine learning training.

Learning rate: Rate for how large steps to converge during machine learning training.

Batch size: Number of samples to process at each step when training machine learning algorithms.

Appendix B

HMI INDEX OVERVIEW

In this appendix, a complete overview of the HMI indecis used in the dataset is found. The original resolution is 8 digit, but since 4 digit resolution is used in the Master thesis, the overview is presented at that aswell.

Feature	Description	Feature	Description	Feature	Description
0433	Devices to keep the tissue intact	1222	Manual wheelchairs	2227	Devices for surveliance
0933	Devices for assisting baths	1812	Assistive beds	2803	Assistive furniture
0912	Devices for assisting toilet visits	1803	Assistive tables	2421	Devices for extended reach
0907	Body stabilizing devices	1809	Assistive sitting devices	2230	Reading assistive devices
0906	Body-worn protection	1206	Walking assistive devices	2218	Assistive devices for sound
0436	Devices for perception training	1810	Accesories for assistive sitting devices	2224	Devices for distance communication
0909	Devices for assisting clothing	1212	Assistive devices for cars	3018	Assistive devices for images
0427	Stimulators	1203	Walking assistive device (one arm)	NumAts	Number of devices
0933	Devices for assisting baths	1231	Devices for changing positions	Gender	Gender
0433	Devices to keep the tissue intact	1236	Devices for lifting people	BirthYear	Birthyear
0930	Urin absorbing devices				

Figure B.1: Overview of the used HMI indices

Appendix C

SOFTWARE OVERVIEW

In this appendix, overviews of the different software components, developed for the experiments in this thesis, are presented. The intention with these overviews are to support the understanding of the flow in the developed software. When exploring the software, the figures presented in this chapter, should be read along with it.

In figure C.1 the flow of python script 1_data_central.ipynb is presented.

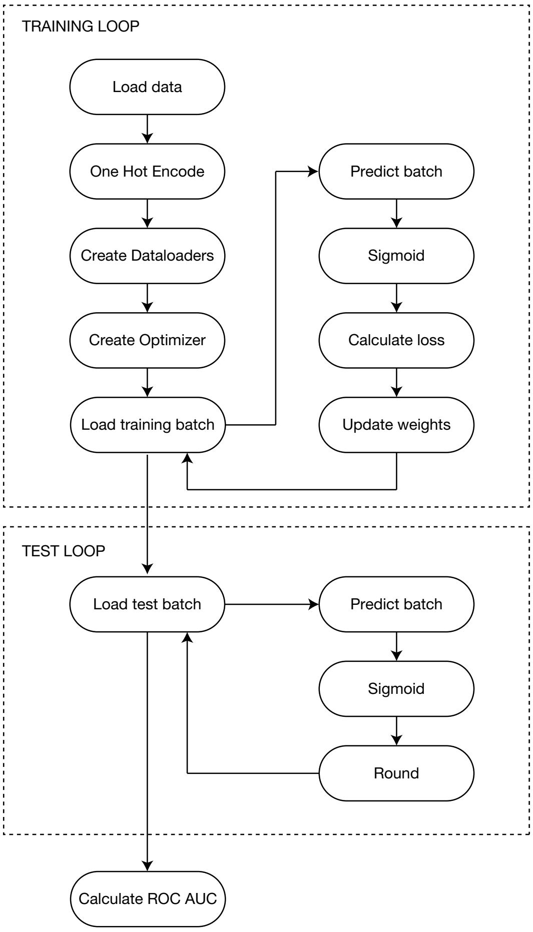

Figure C.1: Overview of software flow when training and testing a data central model. The dotted lines mark sessions of the flow which will be present in figure C.2 and C.5

In figure C.2 the flow of python script 2_federated_learning.ipynb is presented.

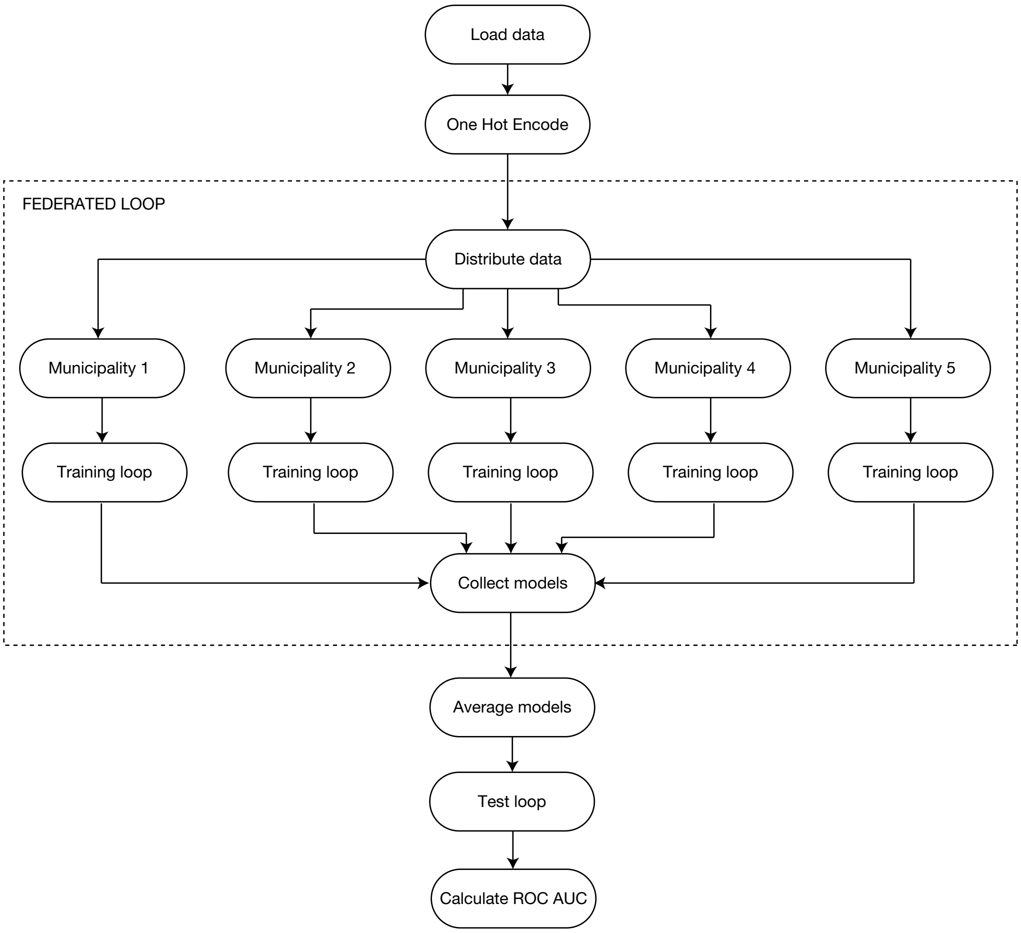

Figure C.2: Overview of software flow when training and testing a federated learned model. The dotted lines marks the federated part of the flow, which will be present in figure C.3

In figure C.3 the flow of python script 3_federated_learning_ass.ipynb is presented.

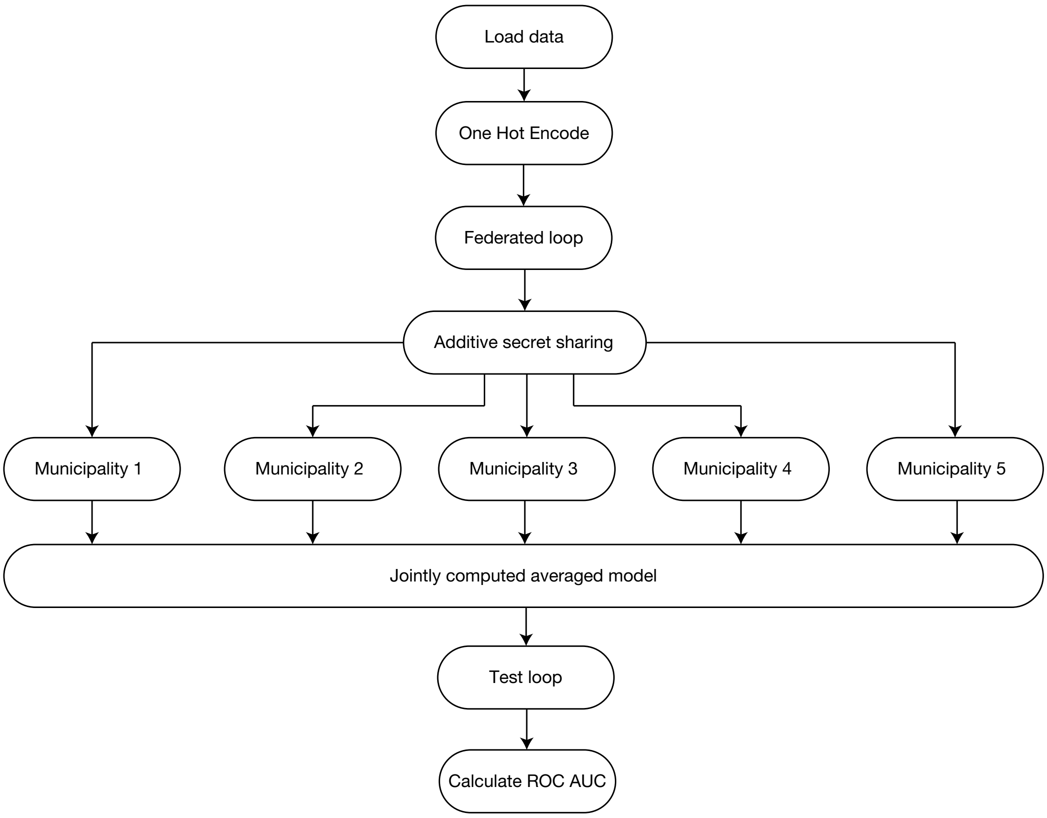

Figure C.3: Overview of software flow when training and testing a federated learned model with additive secret sharing for model averaging.

In figure C.4 the flow of python script 4_differential_privacy.ipynb is presented.

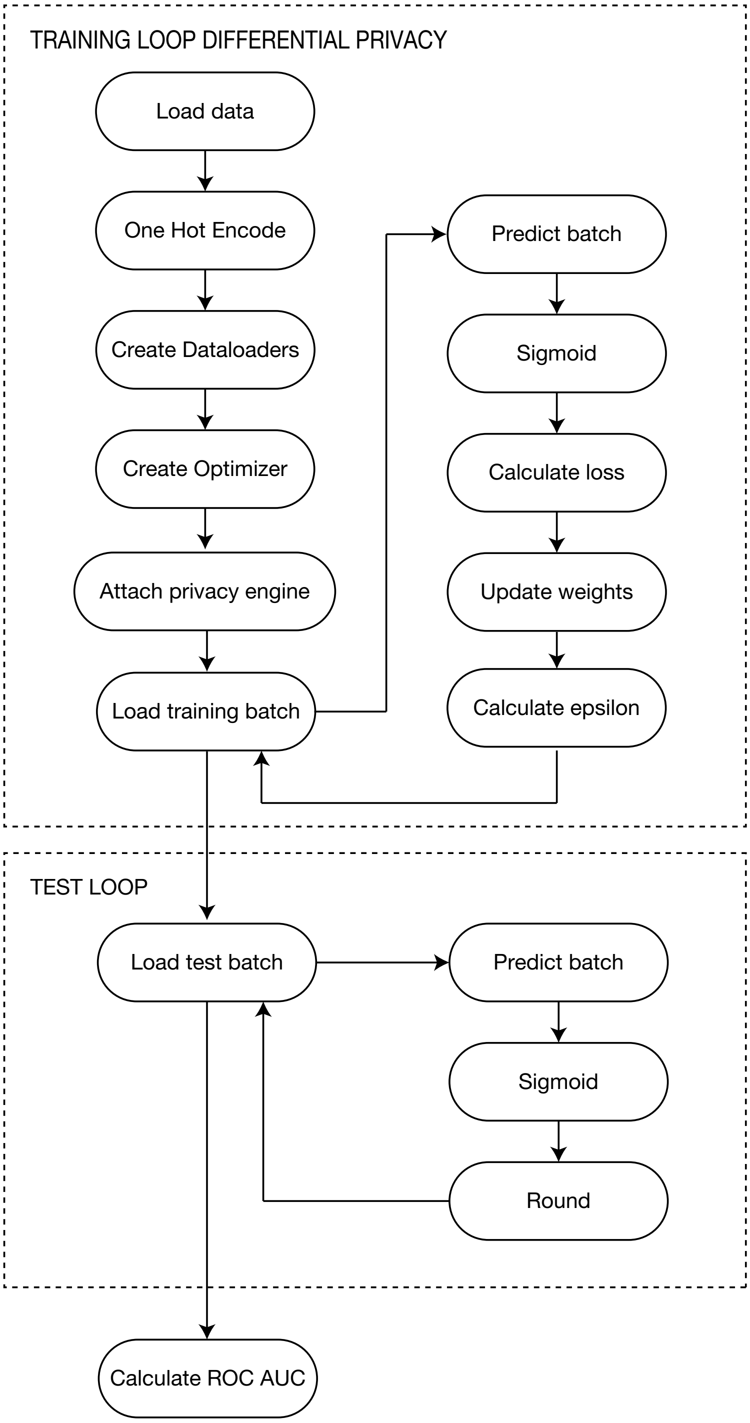

Figure C.4: Overview of software flow when training and testing a data central model with the use of differential privacy.

In figure C.2 the flow of python script 5_shap_calculations.ipynb is presented. The same procedure from this figure is present for the python scripts 6_shap_calculations_federated.ipynb and 7_shap_calculations_differential.ipynb.

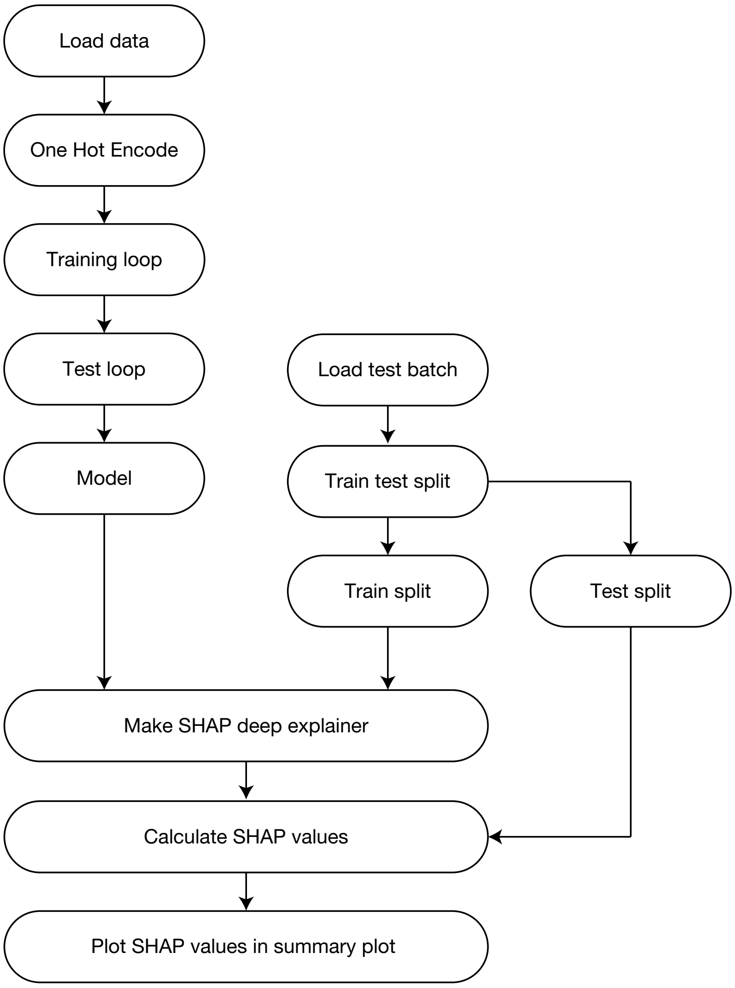

Figure C.5: Overview of software flow when training and testing a data central model and running SHAP value calculations. Same ideas as is presented in this figure could be adapted for federated learning and differential privacy models.

LIST OF FIGURES

LIST OF TABLES

References

[1] Accuracy of noisy counting. https://georgianpartners.shinyapps.io/interactive_counting/. Accessed: 2020-10-16.

[2] Federated learning: Collaborative machine learning without centralized training data. https://ai.googleblog.com/2017/04/federated-learning-collaborative.html. Accessed: 2020-10-16.

[3] Hjælpemiddelbasen. https://hmi-basen.dk/. Accessed: 2020-10-16.

[4] Hvad er tvÆrspor?? https://www.tvaerspor.dk/hvad-er-tvarspor/. Accessed: 2020-10-16.

[5] Interpreting complex models with shap values. https://medium.com/@gabrieltseng/interpreting-complex-models-with-shap-values-1c187db6ec83. Accessed: 2020-10-16.

[6] Local vs. global differential privacy. https://desfontain.es/privacy/local-global-differential-privacy.html. Accessed: 2020-10-16.

[7] Udacity course on secure and private ai. https://www.udacity.com/course/secure-and-private-ai--ud185?irclickid=VzXTSiQ0NxyORSgwUx0Mo3ERUkiQpKVtnzkY080&irgwc=1&utm_source=affiliate&utm_medium=ads_n&aff=259799. Accessed: 2020-10-16.

[8] What is secure multi-party computation? https://medium.com/pytorch/what-is-secure-multi-party-computation-8c875fb36ca5. Accessed: 2020-10-16.

[9] Why one-hot encode data in machine learning? https://machinelearningmastery.com/why-one-hot-encode-data-in-machine-learning/. Accessed: 2020-10-16.

[10] Aarhus University. AIR (AI Rehabilitation). https://projekter.au.dk/air/, 2020. Online; accessed 24 September 2020.

[11] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In

Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, pages 308–318, 2016.

[12] Mohammad Al-Rubaie and J Morris Chang. Privacy-preserving machine learning: Threats and solutions. *IEEE Security & Privacy*, 17(2):49–58, 2019.

[13] Eugene Bagdasaryan, Omid Poursaeed, and Vitaly Shmatikov. Differential privacy has disparate impact on model accuracy. In *Advances in Neural Information Processing Systems*, pages 15479–15488, 2019.

[14] Aurélien Bellet, Amaury Habrard, and Marc Sebban. A survey on metric learning for feature vectors and structured data. *arXiv preprint arXiv:1306.6709*, 2013.

[15] R. Bhardwaj, A. R. Nambiar, and D. Dutta. A study of machine learning in healthcare. In *2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC)*, volume 2, pages 236–241, 2017.

[16] Peter Bjerregaard and K Juel. Middellevetid og dødelighed i danmark. *Ugeskrift for Laeger*, 155(50):4097–100, 1993.

[17] A. Callahan and N. Shah. Machine learning in healthcare. 2017.

[18] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407, 2014.

[19] Jianjiang Feng and Anil K Jain. Fingerprint reconstruction: from minutiae to phase. *IEEE transactions on pattern analysis and machine intelligence*, 33(2):209–223, 2010.

[20] J Frankenfield. Artificial intelligence (ai). *Investopedia udgivet*, 13(6):19, 2019.

[21] 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*, pages 1322–1333, 2015.

[22] JINHO Kim[1], BS Kim, and Silvio Savarese. Comparing image classification methods: K-nearest-neighbor and support-vector-machines. In *Proceedings of the 6th WSEAS international conference on Computer Engineering and Applications, and Proceedings of the 2012 American conference on Applied Mathematics*, volume 1001, pages 48109–2122, 2012.

[23] Keith E Kolekofski Jr and Alan R Heminger. Beliefs and attitudes affecting

intentions to share information in an organizational setting. *Information & management*, 40(6):521–532, 2003.

[24] Jaewoo Lee and Chris Clifton. How much is enough? choosing ε for differential privacy. In *International Conference on Information Security*, pages 325–340. Springer, 2011.

[25] Jaewoo Lee and Daniel Kifer. Concentrated differentially private gradient descent with adaptive per-iteration privacy budget. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1656–1665, 2018.

[26] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60, 2020.

[27] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60, 2020.

[28] Peng Liu, YuanXin Xu, Quan Jiang, Yuwei Tang, Yameng Guo, Li-e Wang, and Xianxian Li. Local differential privacy for social network publishing. *Neurocomputing*, 391:273–279, 2020.

[29] Ilya Mironov. Rényi differential privacy. In *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*, pages 263–275. IEEE, 2017.

[30] Arvind Narayanan and Vitaly Shmatikov. Robust de-anonymization of large sparse datasets. In *2008 IEEE Symposium on Security and Privacy (sp 2008)*, pages 111–125. IEEE, 2008.

[31] B. Pabst. Vi skal bruge flere penge på pleje af ældre. 2020.

[32] Elizabeth A Phelan, Jane E Mahoney, Jan C Voit, and Judy A Stevens. Assessment and management of fall risk in primary care settings. *Medical Clinics*, 99(2):281–293, 2015.

[33] Danske Regioner. Flere ældre betyder flere patienter i sundhedsvæsenet. 2019.

[34] Saumya. Miniai: Differential privacy in 2 mins. 2002.

[35] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18. IEEE, 2017.

[36] Leslie N Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*, volume 11006, page 1100612. International Society for Optics and Photonics, 2019.

[37] sundhed.dk. Fald og faldtendens hos ældre. `https://www.sundhed.dk/sundhedsfaglig/laegehaandbogen/geriatri/symptomer-og-tegn/fald-og-faldtendens-hos-aeldre/`, 2018. Online; accessed 24 September 2020.

[38] ITGP Privacy Team. *EU General Data Protection Regulation (GDPR).* IT Governance Limited, 2017.

[39] Anamaria Vizitiu, Cosmin Ioan Nită, Andrei Puiu, Constantin Suciu, and Lucian Mihai Itu. Applying deep neural networks over homomorphic encrypted medical data. *Computational and Mathematical Methods in Medicine*, 2020, 2020.

[40] Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris Papailiopoulos, and Yasaman Khazaeni. Federated learning with matched averaging. *arXiv preprint arXiv:2002.06440*, 2020.

[41] Deidre Wild, US Nayak, and B Isaacs. How dangerous are falls in old people at home? *Br Med J (Clin Res Ed)*, 282(6260):266–268, 1981.

[42] Lizhi Xiong, Wenhao Zhou, Zhihua Xia, Qi Gu, and Jian Weng. Efficient privacy-preserving computation based on additive secret sharing. *arXiv preprint arXiv:2009.05356*, 2020.