Master Thesis

Machine Learning for All: a Methodology for Choosing a Federated Learning Approach

Guido Teunissen
Master Student Business Information Technology

Supervisors:

Dr. Adina Aldea
University of Twente

Dr. Mannes Poel
University of Twente

Kevin Bonnes, MSc
Topicus

Barthold Derlagen, MSc
Topicus

Deventer, 16-10-2020

UNIVERSITY OF TWENTE.

topicus

Abstract

Federated Learning is a new form of Machine Learning where a central model is trained decentrally on multiple distributed devices, while still keeping data on-device for privacy-preservation. Organizations who want to tap into the potential of having more data available for their predictive machine learning models, while still adhering to recent data protection regulations, will see a good fit in Federated Learning, as privacy-preservation is one of its main pillars. However, the research area is relatively new and the information fragmented. Therefore, this study provides a comprehensive review on the state-of-the-art in Federated Learning research. It sets an agreed-upon definition for Federated Learning, presents a comprehensive list of available Federated Learning algorithms, and purposefully investigates their main differences. All this information is then consolidated and used to design a methodology that supports organizations in making an informed decision in choosing among the myriad of Federated Learning algorithms available, based on their data-related characteristics, privacy-requirements, and business goals. This method has been successfully validated by means of a real-world case study in the financial industry, and positively been evaluated by means of a demonstration to experts. Also the resulting choice of the designed method, a Federated Learning algorithm, has been implemented by means of another case study. In order to show the practicality and partly validate the choice based on empirical results, not just on literature insights. All of this has been conducted in a methodological and scientific way. The overall study follows the design science research methodology (DSRM), the literature insights are collected methodologically by means of a Systematic Literature Review, the method is designed by means of a meta-methodology called Situational Method Engineering, and has been evaluated by using the Unified Theory of Acceptance and Use of Technology model. The resulting Federated Learning model has been developed by means of the CRISP-DM research methodology, a leading methodology in data science. This gives the study both scientific backing and practical relevance.

Preface

This report is the end result of my master thesis, which also constitutes the final phase of my master Business Information Technology at the University of Twente.

First of all, I would like to thank Topicus for facilitating this study. Topicus is a software development company, founded in 1998, with various locations throughout The Netherlands, and, as of writing, has more than a thousand employees. Topicus is divided into five departments: Finance, Healthcare, Education, Government / Social domain, and Core and is consequently also active in the first four similarly named sectors. For each of these four domains Topicus provides software and services to that respective market.

In particular, I would also like to thank my supervisors at Topicus, Kevin Bonnes and Barthold Derlagen, for their continued professional support in both improving the quality and the overall process of this study. Without them, this study would not have been possible.

Also, I would like to thank my supervisors at the University of Twente, Adina Aldea and Mannes Poel, for their support in shaping this study and their professional feedback. In this way I could improve the quality and academic relevancy of my work.

Guido Teunissen
Apeldoorn, October 2020

Structure

1. Introduction

This study is part of the researcher's final project in the master program Business Information Technology at the University of Twente in Enschede. The master thesis will be conducted in collaboration with the software company Topicus in Deventer. This chapter first introduces the problem context and motivation, introduces solution objectives, as it is a design study. From this, relevant research questions are devised. The mapping of these research questions to the remainder of this report and the associated research methodologies are given at the end of this chapter.

1.1 Problem Context and Motivation

LeCun et al's (2015) highly influential paper showed that the advent of increased data set sizes of eliminated most of the need of manual work in setting up and tuning conventional machine learning models, and basically started the concept of Deep Learning. Taking advantage of the larger amount of available data increased the usefulness and effectiveness of the machine learning models. So, having available more data is an advantage. However, unlike big corporations like Facebook and Google, which generate massive data sets on their own, other organizations are many orders of magnitude smaller and do not have these same capabilities. Other smaller organizations could, however, also make use of the same advantages stated before. By partnering up with similar organizations, they could construct larger available data sets and leverage the same advantages that these large corporations have.

However, when using traditional machine learning techniques, data need to be transferred from one party to another, usually to one central party, which will become responsible for this data (Yang et al, 2019). Consequently, constructing a joint data in such a way with traditional machine learning techniques generates additional privacy challenges, both from a legal and a competitive-interest perspective. Yang et al (2019) state that there is an increasing awareness of large companies compromising on data security and user privacy. In addition, they even affirm that emphasis on data privacy and security has become a worldwide major issue.

Many of the privacy concerns will have a legal origin. As of 2016 the European Union passed the General Data Protection Regulation (GDPR) (Zarsky and Tal, 2017). This regulation, among other things, impedes the sharing of data, and especially that of personal information. Which would complicate the traditional machine learning approach. Zarsky and Tal (2017) even call the GDPR incompatible with the advent of large data sets. Especially because they state that these data sets are mostly of a personal nature and the stringent data protection laws impede the flow of this data. In addition, they state that these laws will compromise the growth of the Big Data industry, and with it the added benefits.

Especially in healthcare this privacy aspect is important, as hospitals generate and store very personal and sensitive data, namely electronic health records. Also it is difficult to collect this medical data, as they exist in isolated spaces; essentially data islands, one for each hospital. Rumbold and Pierscionek (2017) raise concerns about the improvements in healthcare due to the strict data regulation laws, as the process of doing data science is impeded. But especially in this case, utilizing the joint information potential is crucial in improving healthcare predictions; hospitals on their own often have smaller data sets, are sometimes narrowly specialized, and have differences in their patient base (Deist et al, 2017). Yang et al (2019) even state that the insufficiency of data sources led to unsatisfactory machine learning model performances. The potential of learning from each other, and developing some sort of a joint data set is great. It could be a major technique in improving the performance of machine learning models. By combining the data sets, more accurate and robust machine learning models could be made, and, thus, better predictions can be made.

In addition, other industries face similar problems. Yang et al (2019) name the financial sector as a potential sector which could benefit from utilizing from a joint data set. As in the case of healthcare Yang et al (2019) state that also in the financial sector the data is isolated from each other, due to privacy and competitive-interest concerns. Lastly, the same goes for the mobile software industry. Hard et al (2018) seek a way to improve keyboard type prediction for its mobile

Google keyboard. However, the data is generated by many different users, all isolated from each other. The data generated can be of highly privacy-sensitive nature as the user can type personal information, passwords, and more.

Thus, given the fact that many industries struggle with data sharing concerns - due to privacy, legal, and competitive-interest considerations - organizations struggle in utilizing potential larger joint data sets for improving machine learning models.

As investigated in a preceding study (Teunissen, 2020), *Federated Learning* is a good solution fit to this problem context. Federated Learning can be defined as: *a form of distributed machine learning where a global model is trained on a central server utilizing multiple separate heterogenous edge devices, while still preserving privacy by not permitting the data to leave their origin devices*.

Especially because Federated Learning is focused primarily on privacy-preservation while still utilizing a distributed architecture, it addresses the previously raised concerns. Federated Learning does not permit data to leave its origin device by only sharing partial model updates to a central server. In this way, privacy sensitive information is protected, as no raw data is shared. Both addressing competitive-interest concerns, and data sharing prohibitions by GDPR.

However, the research area of Federated Learning is still relatively new, and the information is fragmented. Federated Learning methods (i.e. techniques) are usually introduced on a one-per-paper basis, making the information fragmented. Because there are a myriad of distinct Federated Learning methods it makes it especially difficult to choose the best approach. In addition, each Federated Learning method has its own characteristics, requirements, and performs better or worse depending on the data set used, its privacy requirements, and other facets.

Therefore, there is an apparent need for organizations and in research to consolidate this information and provide guidance in what Federated Learning method is suitable given a particular situation within the stated general problem context. This fragmented information introduces a problem for organizations who want to implement Federated Learning in the best way possible. Therefore, this study will create a method which guides organizations in the process of deciding upon the best suited Federated Learning method based on their organizational characteristics regarding its data and the privacy considerations of this data.

Concluding, the problem statement of this research can be summarized as: organizations are increasingly aware of challenges regarding privacy issues in machine learning, due to recent data protection regulations and competitive-interest considerations. While at the same time are aware of the potential of using larger data sets for machine learning, which are currently not accessible for data sharing due to privacy and competitive-interest considerations, i.e. the data are separated at different data silos. Also, the research area of Federated Learning, which is a good solution-fit for this problem context, is relatively new and fragmented. The best possible method per given company-specific situation is not clear without synthesizing the information in different sets of studies. Organizations who want to implement Federated Learning will have difficulty in choosing the best Federated Learning method that fits their specific situation.

1.2 Solution Objectives

In this section the solution objectives are described, i.e what artifact is to be created. These solution objectives are constructed based on the stated problem context. Also, in addition, a small stakeholder analysis is conducted, which is part of Wieringa's (2014) Design Science methodology.

Based on the problem statement the following to-be-designed artifact is chosen for this study: a method that organizations can use to decide upon which Federated Learning method fits their specific situation - regarding their objectives, data characteristics, and privacy - in the best possible way. These organizations are all scoped to be in a situation of having multiple separated data sites (i.e. data silos) [Appendix E: Definition 2] where data sharing is limited or even prohibited due to privacy and/or competitive-interest considerations, while still wanting to utilize the potential of the joint data as input for a machine learning objective.

Next are explicit exclusions from this stated scope of the study. This study does not concern itself with the more technical aspect of Federated Learning, such as communication protocols, technical infrastructure, technical implementation, and implementation costs of these Federated Learning methods. As machine learning and distributed machine learning already include a subset of the problems Federated Learning has, such as the implementation details, infrastructure, communication protocols, this is excluded in this study. Only the part that makes Federated Learning distinct from other distributed machine learning is considered: the properties of the fragmented data sets, the privacy aspect, and other differentiating characteristics of Federated Learning related to these aspects. Thus, the to designed method in this study should be seen to precede the actual implementation itself.

Next, Wieringa (2014) suggest to do a stakeholder analysis of the problem context. As the to-be-designed artifact will be evaluated on *utility*, there need to be one or more stakeholders on which this value can be measured. Wieringa states that: "a stakeholder of a problem is a person, group of persons, or institution affected by treating the problem". From this definition and the drawn problem context, the following list of stakeholders are identified:

- Domain experts (i.e. developers) in organizations who want to implement Federated Learning. These are classified as *normal operators* in terms of Wieringa's possible stakeholder list. They have a *technical conflict*, not having the knowledge to choose an appropriate Federated Learning method;
- The beneficiaries of those organizations' resulting applications and services, whose data will be used, and who will receive (part of) the benefits. (Can be the original organization itself, or a client). These stakeholders are classified as *functional beneficiaries* (indirect stakeholder);
- The subjects of the data, the data owners, whose data is being used. These stakeholders are classified as *negative stakeholders* (indirect stakeholder). The could have a *legal conflict* with the proposed solution.

1.3 Research Objectives

From the solution objectives stated before, research objectives can be constructed.

The research objective of this study is the following: to design a method that organizations can use to decide upon which Federated Learning method fits their specific situation - regarding their objectives, data characteristics, and privacy - in the best possible way. These organizations are all scoped to be in a situation of having multiple separated data sites (i.e. data silos) where data sharing is limited or even prohibited due to privacy and/or competitive-interest considerations, while still wanting to utilize the potential of the joint data as input for a machine learning objective.

From this main research objectives, several sub research objectives are drawn:

- To define what Federated Learning is (i.e., a definition) and what its defining characteristics are;
- To find out which Federated Learning methods are available in the literature;
- To find out the characteristics of and the differences between these Federated Learning methods;
- Designing a method to make an informed choice between these Federated Learning methods;
- Validation and evaluation of the designed method.

1.4 Research Questions

Main research question (MRQ):

What is an appropriate methodology to help organizations choose the most suitable Federated Learning method given their situation regarding data-related characteristics and privacy requirements?

Sub research questions (RQs):

Knowledge questions:

1. **What is the definition of Federated Learning according to the literature?**
2. **What Federated Learning methods exist in the literature?**

3. **What are the main differentiating characteristics of the Federated Learning methods found in the literature?**
4. **What are the differences in predictive performance among Federated Learning methods?**
5. **What is the effect of Federated Learning's consolidation technique of utilizing multiple data sites on predictive performance?**
6. **What is an appropriate method for identifying non-iid data sets in the context of Federated Learning?**

Design questions:

7. **How to design a methodology that fits the goal of the main research question?**
8. **How to evaluate the designed methodology?**

Next, the reasoning why these research questions are chosen is explained.

RQ1. The first research question is initiated to serve as background information to the topic, both for the reader and the researcher. Its goal is to investigate what the literature defines as Federated Learning and what its characteristics are, to set the basis for the remainder of this study. During the exploratory pre-mapping phase of the literature review (see the next chapter for this), it became apparent that the research area is still relatively new, the definition of Federated Learning differs, and the research area is fragmented. This research question will, therefore, provide context for the remaining research questions, providing a thorough and complete definition of what Federated Learning is in this study.

RQ2. For the second research question the most prevalent methods of Federated Learning will be identified by means of a systematic literature review. As it became apparent that many different methods exist in the literature during the pre-mapping phase, it is useful to make inventory of these methods. It is likely they have different characteristics and use cases. In order to be able to identify which Federated Learning method is the right fit for a particular (sub-)problem context, the first step is identifying which Federated Learning methods exist.

RQ3. The third research question is chosen because of the following. In order to be able to make an informed decision about the choice of a suitable Federated Learning method, their differentiating characteristics [Appendix E: Definition 3] need to be known. Only when you know the differences between available options, a decision can be made. More specifically, only the differentiating characteristics that are relevant to the organization using the to-be-designed methodology have to be considered. Relevant characteristics to these organizations are those which may limit options or impact the desired outcome regarding the organization's data-related characteristics and privacy considerations. The latter being a result of the set scope of the study in the introduction. Therefore, identifying these differentiating characteristics contributes to knowledge needed to create the to-be-designed methodology.

This research question is also answered by means of conducting a Systematic Literature Review (SLR), as described in the methodology section 2.3. For this specific research question, all studies which are mainly about introducing or describing one or more Federated Learning methods are included.

RQ4. After identifying which Federated Learning methods are out there, a comparison in terms of predictive performance among them is made in the third research question. Like stated before, the right method for a particular problem context is likely to be different, and their predictive performance is of utmost importance in this. In addition, a comparison to local-only methods is done where possible, to make a comprehensive comparison. Local-only methods refers to standard machine learning, learned on merely one local data set.

RQ5. The fourth and last research question was initiated because of the assumption made earlier that these different data sites may hold data of different nature. These data sites may contain the same type of data (fields) but may have significantly different characteristics in terms of size, distribution. For example, when a large disparity of number of data points exists between data sites, it may be the case that one data site overshadows another one. To investigate this assumption, the way in which Federated Learning methods consolidate methods is investigated, and its impact on predictive performance is reviewed.

RQ6. The sixth research question, and also last knowledge question is chosen because of the following reasoning. During the conduction of the Systematic Literature Review, none of the studies which were used to answer research questions 4 and 5 gave a clear definition of what they regarded as non-iid data. It was assumed to be implicit knowledge. The definition of what non-iid data is in the context of Federated Learning is not easily obtainable from the studies found. This information can be regarded as implicit knowledge in this research area. However, making this knowledge explicit not only gives readers from outside this area a better understanding on what they are reading, but it also forces studies on Federated Learning to be as clear as possible on what non-iid data exactly is. Also, only when having clearly defined what non-iid data is a potential method to identify it can be found or even developed.

An example of the confusion this implicit definition of non-iid data cause can be found in this very study. As found earlier in research question 5, there are contradicting claims on whether standard Federated Learning methods work well on non-iid data or not. A more clearly defined definition of what non-iid data is could make these contradictions easier to evaluate, as right now both sides could have a slightly different conception of what non-iid data is.

Therefore, this research question will make this definition more explicit. Therefore, this research question will aim to define explicitly what non-iid data is, in terms of a definition and its challenges. When explicitly defined, a method fragment which can identify whether the data is non-iid or iid can be constructed, which will contribute to the overall research goal of this study. Without a clear definition this is not possible.

1.5 Structure of the Study

In this section the structure of the study is described. First on a per chapter basis. Second, the phases of the overall research methodology of this study, Design Science Research Methodology of Peffers et al. (2008) (DSMR) are mapped to the chapters and other research methodologies used in Table 1.5.1. Lastly, a mapping of each research question to the used research methodology is made in Table 1.5.2. These research methodology will be explained in depth in the next chapter.

The mapping of this research methodology to the chapters in this report is the following:

- Chapter 1 describes the problem identification and motivation, the solution objectives, the research objectives, and the research questions;
- Chapter 2 elaborates on the research methodologies used;
- In Chapter 3 the knowledge questions are answered by means of a Systematic Literature Review;
- In Chapter 4 the artifact will be designed, i.e. the method will be constructed, based on Harmsen's Situational Method Engineering (SME);
- Chapter 5 will provide the validation of the designed method, by means of a case demonstration;
- Chapter 6 will provide the evaluation of the designed method, by means of the UTAUT model by Venkatesh et al. (2003), the Unified Theory of Acceptance and Use of Technology model;
- Chapter 7 provides an evaluation on the result of the designed method by means of implementing the resulting Federated Learning algorithm in a case study;
- Chapter 8 will provide the conclusion, discussion of the results, contributions of this study, its limitations, and possible future work.

Table 1.5.1 - Mapping of DSRM phases to Report Chapters and Other Research Methodologies

DSRM phase	Specific Research Methodology	Report
1. Problem Identification & Motivation	-	Ch.1
2. Define Objectives of a solution	-	Ch.1
3. Design & Development	Systematic Literature Review	Ch. 3
	Situational Method Engineering	Ch. 4
4. Demonstration	Case study demonstration (DSRM)	Ch. 5
5. Evaluation	UTAUT CRISP-DM	Ch. 6

Table 1.5.2 - Mapping of RQs to Research Methodologies

Research Question	Research Methodology	Type of question	Report
RQ1. What is the definition of Federated Learning according to the literature?	Systematic Literature Review	Knowledge question	Ch. 3.1
RQ2. What Federated Learning methods exist in the literature?	Systematic Literature Review	Knowledge question	Ch. 3.2
RQ3. What are the main differentiating characteristics of the Federated Learning methods found in the literature?	Systematic Literature Review	Knowledge question	Ch. 3.3
RQ4. What are the differences in predictive performance among Federated Learning and local-only methods?	Systematic Literature Review	Knowledge question	Ch. 3.4
RQ5. What is the effect on predictive performance effect of utilizing multiple data sites in Federated Learning by the means of consolidating this data?	Systematic Literature Review	Knowledge question	Ch. 3.5
RQ6. What is an appropriate method for identifying non-iid data sets in the context of Federated Learning?	Systematic Literature Review	Knowledge question	Ch. 3.6
RQ7. How to design a methodology that fits the goal of the main research question?	Situational Method Engineering	Design question	Ch. 4
RQ8. How to evaluate the designed methodology?	DSRM Case Study UTAUT CRISP-DM	Design question	Ch. 5,6,7

2. Research Methodology

In this chapter the research methodology of this study is stated. The research methodology is constructed by a multitude of methodologies, from coarse granularity for the overall structure to a more fine-grained and applied approach, supplementing each other. The overall research methodology will follow the Design Science Research Methodology (DSRM) of Peffers et al. (2008). It is supplemented by Wieringa's (2014) Design Cycle. A Systematic Literature Review by Kitchenham and Charters (2007) is used to answer knowledge questions in part of the design phase. For the Method Design phase Situational Method Engineering by Harmsen (1997) is chosen as a more detailed and applied approach. By combining these methodologies, each phase of the study will have the most relevant fit regarding both the objective and the granularity of the task at hand. In this chapter this approach is described in more detail starting with the overall research methodology: DSRM.

2.1 Design Science Research Methodology

This study will follow the research area of Design Science for its overall research methodology. More specifically, it will feature DSRM, complimented by Design Science by Wieringa. First, an explanation is given as to why Design Science is chosen, then the research methodology is briefly explained.

A research methodology with a good fit to the problem statement and the resulting research goal should be used in order to successfully conduct this study. For this, Design Science is chosen. Design Science is a good fit because of the following reasons. Firstly, the research goal calls upon creating an artifact which helps stakeholders in a specific problem context. This is primarily in the realm of Design Science. Secondly, there is not one solution design possible to solve this problem, but multiple. Therefore, a choice as to what is the best possible solution should be made, which is particularly the case in Design Science.

DSRM by Peffers et al. (2008) will be used as the overall research methodology of this study. This methodology features 6 phases. These phases are: (i) problem identification and motivation, (ii) define objectives for a solution, (iii) design and development, (iv) demonstration, (v) evaluation, and (vi) communication. See Figure 2.1.1 for a visual representation of this.

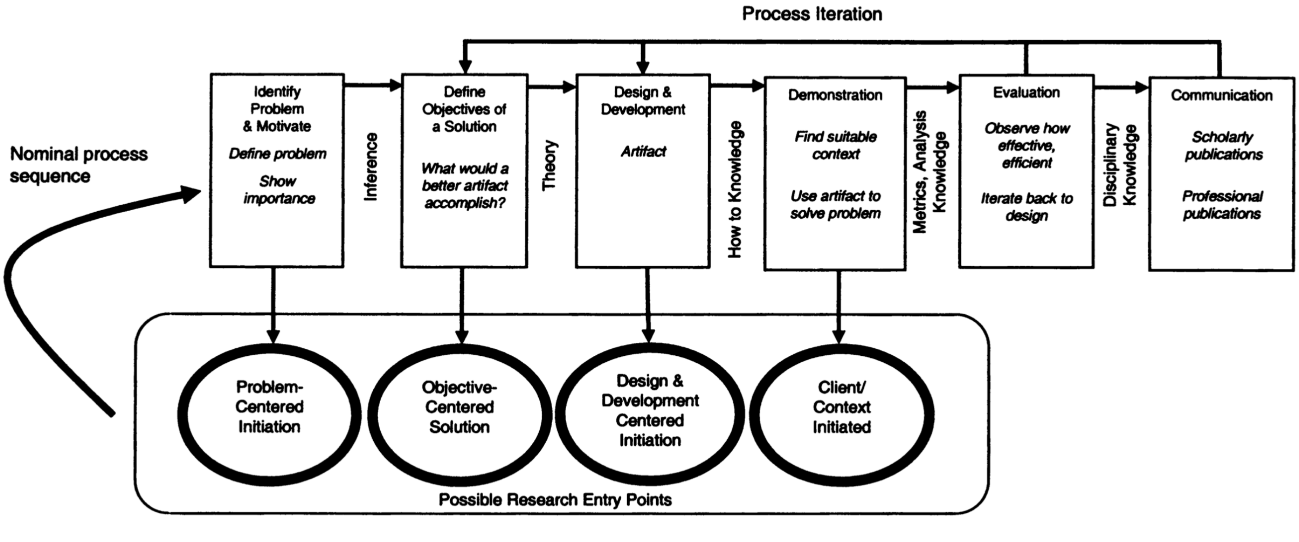

Process Iteration
Nominal process sequence
Identify Problem & Motivate
Define problem
Show importance
Inference
Define Objectives of a Solution
What would a better artifact accomplish?
Theory
Design & Development
Artifact
How to Knowledge
Demonstration
Find suitable context
Use artifact to solve problem
Metrics, Analysis Knowledge
Evaluation
Observe how effective, efficient
Iterate back to design
Disciplinary Knowledge
Communication
Scholarly publications
Professional publications
Problem-Centered Initiation
Objective-Centered Solution
Design & Development Centered Initiation
Client/ Context Initiated
Possible Research Entry Points

Figure 2.1.1 - Design Science Research Methodology (DSRM), Peffers et al (2008)

This methodology is supplemented by Wieringa's take on Design Science. Parts from Wieringa's theory which is used in this study is the following. It features a more detailed stakeholder analysis, a view that a to-be-created artifact should be designed by requirements, which in turn should have a contribution argument to the stakeholder or research goals. Lastly, Wieringa states that the artifact is evaluated by *utility*. From this view more detailed metrics and methods to validate and evaluate the to-be-designed method will be chosen. For this study the UTAUT model will be used as a way to evaluate the utility of the to-be-designed method.

2.2 Method Engineering

In the design phase an artifact has to be created. For this study it is chosen to design a method. In this section it is first defined what a method is, and then a more detailed (meta) methodology is chosen to guide the design of the proposed method.

In TOGAF, in the research area of Enterprise Architecture, a method or methodology is defined as: "a defined, repeatable series of steps to address a particular type of problem, which typically centers on a defined process, but may also include definition of content" (TOGAF, 2011). Important is this definition is that a method is a defined series of steps. Therefore, it is not only important what method steps are defined, but also the order of these steps is of importance. From this definition several parameters of a method can be stated: a set of method steps, the contents or process of each distinct step, the goal of each step, and the ordering of these method steps. However, from this definition alone it is still not well-defined how to design a method; a meta-methodology is needed.

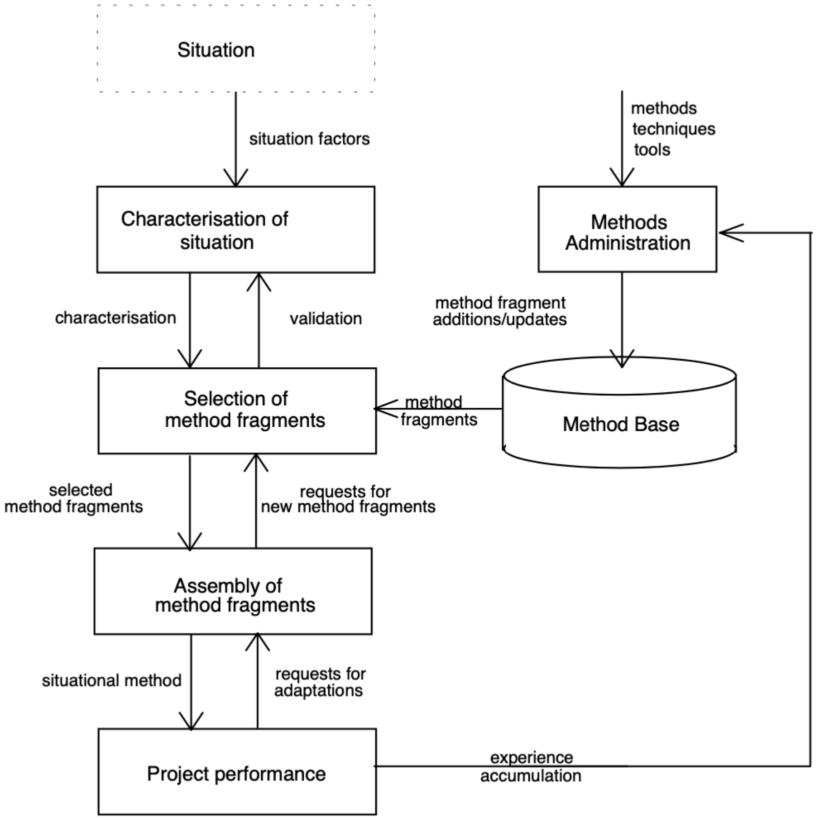


Figure 2.2.1 - Situational Method Engineering (SME), Harmsen (1997)

Harmsen (1997) provides such a meta-methodology and introduces the concept of Situational Method Engineering (SME). SME is used in this study for a more fine-grained implementation of the design phase. SME is summarized in Figure 2.2.1. It is briefly explained next.

The *Method Base* stores method fragments containing all types of method fragments, their relationships, properties, and constraints (Harmsen, 1997). It can be seen as a repository of all possible method fragments which can construct a new situational method. *Method fragments* can roughly be seen as (uncharacterized, 'template') parts of a method, i.e. the distinct steps in a method before the method itself is constructed. They should be able to describe every aspect of a method, it has relationships with other method fragments, e.g. processes may precede each other, products consist of other products, processes produce and require products (Harmsen, 1997).

Next is the *selection of method fragments*. The selection is based upon the *characterization of the situation* at hand. This situation characterization corresponds in this study to the Problem identification and the objective definition phases of the Design Science research methodology. The guidelines Harmsen gives for this situation characterization are, however, created specifically for information system (IS) project development and are not relevant for this study. Instead, the already established guidelines provided by Peffers et al. will be used for this.

Meaningful selection of the right method fragments require a thorough characterization of method fragments in a structured way in order to maintain comparability and consistency (Harmsen, 1997). For this, this study will devise a standard template which characterizes method fragments in terms of relevant properties. Because, with only a method fragment name and description selection cannot be standardized and consistent. For this study these properties are chosen to be the following: method fragment name, description, goal, input, prerequisites, actions to be undertaken, output. This is summarized in Table 2.2.1. Using these relevant properties, method fragments which support the solution objective can be selected.

Table 2.2.1 - Method Fragment Properties

Method Fragment Property	Explanation
Name	Name of the method fragment
Description	Description of the method fragment in freeform text
Goal	The goal of this method fragment. It should contribute to the overal solution objective goal
Input	Input needed for this method fragment, such as: data, knowledge, resources
Prerequisites	Required other method fragments which need to be completed before this method fragment
Actions	The actions this method fragment will undertake
Output	The output this method fragment produces. Such as: new insights, data, knowledge

The last relevant step of SME for this study is *method assembly*. Here the objective is to combine the method fragments and design the resulting method. Harmsen (1997) suggests using a strategy, guidelines, and *assembly rules* in order to perform method assembly in a consistent and sensible manner. In a general sense, the method should fit the situation (*suitability*), but also some quality criteria are used: *completeness, consistency, efficiency, soundness,* and *applicability*. In this way the method fragments can be used to design the resulting method in a structured an sound way.

The steps characterization of the situation and project performance can both be incorporated in the preceding and succeeding steps of the DSRM. In the problem identification and motivation, and the solution objectives definition steps, the characterization of the situation takes place, but merely has another name. In addition the project performance step is there to validate and improve the created methodology by validating it on a project basis. In this study, the project view is not relevant, but the validation part still stands, as it is also part of the DSRM. In this way the research methodologies can be consistently linked to each other. More on this is described in section 2.4.

2.3 Systematic Literature Review

The research methodology for this study is a Systematic Literature Review (SLR), as based on the paper of Kitchenham and Charters (2007). With this research methodology, the results of the study have several advantages over traditional studies. First of all, the results are less likely to be biased, and, second of all, the study is more transferable. As the SLR is based on a defined search strategy, which uses multiple sources, it is designed to give a more comprehensive picture of the current literature than standard literature studies, aiming to include as much relevant literature as possible. In addition, as the search is well-documented and systematic, the study becomes more transparent and replicable (Kitchenham and Charters, 2007). The SLR constitutes three phases: planning (i.e., design), execution, and results analysis. This process is summarized in Figure 2.3.1.

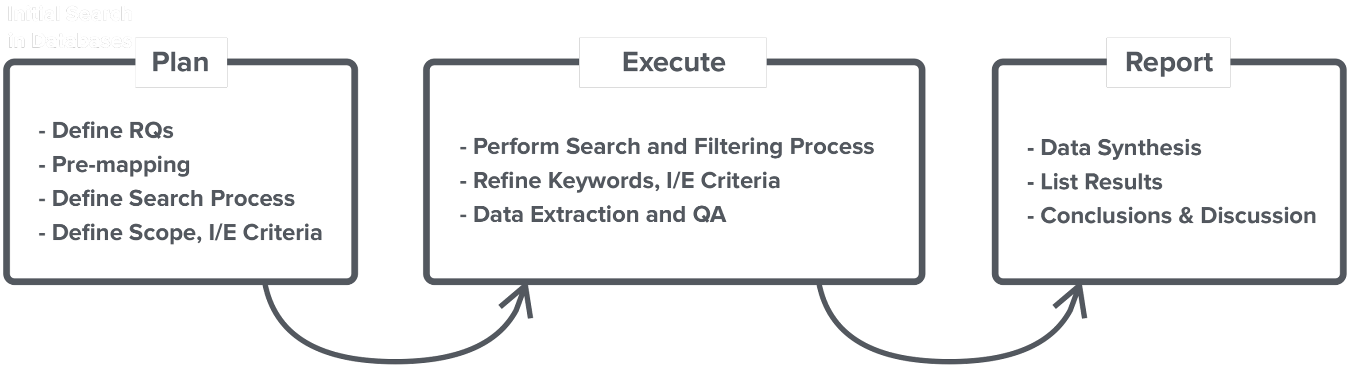


Figure 2.3.1 - the Systematic Literature Review (SLR) process

2.3.1 Pre-mapping phase

Kitchenham and Charters (2007) propose a pre-mapping phase, in order to make the reviewer more familiar with the topic, to help shape the research questions, provide a basis for the search keywords, and to help narrow down the search research space. The pre-mapping in this study includes an initial exploratory search in these scientific databases, reading relevant literature, both merely reading the abstract as well as reading the full text of the paper, and incorporating expert opinion. From this pre-mapping phase, the initial search keywords are defined.

The expertise of people knowledgeable in the field is utilized in the expert opinion incorporated in this pre-mapping phase. This is done in order to get a grasp on the research field and include papers and keywords that might be of interest. The interviews were informal, unstructured, and not transcribed, as this only serves as additional knowledge in a very early step of the research. The experts inquired were two people working at the company, and had domain knowledge about machine learning and the business problem, mentioned in the problem context, and two researchers of the university which facilitates this research.

After the expert opinion, an exploratory literature review is conducted, incorporating the results of the expert opinion stage. The goal of this stage is to become familiar with the field of study, find an initial set of papers in order to extract relevant concepts and their accompanying keywords, which in turn will translate to the initial search queries.

2.3.2 Scoping the Research to Federated Learning

Some of the goals of the pre-mapping phase is to become familiar with the research area and to scope the research with this gained knowledge. For this problem context, a more general question was asked: what is the most prevalent method of utilizing multiple data sources in machine learning?

During the pre-mapping phase this question was answered. It quickly became apparent that Federated Learning is a prevalent method utilizing multiple data sources in machine learning. This is because early on in the exploratory literature review, and by provided studies from expert opinion, Federated Learning was identified as a clear and distinct research area, suitable for this problem context. Additionally, Federated Learning also concerns itself with privacy-preservation, which is also one of the mentioned aspects in the identified problem context. The privacy aspect was mentioned in all the papers found in the exploratory pre-mapping phase.

Therefore, this study is scoped to be solely concerned with Federated Learning (and, formerly known as, Distributed Learning) as the method of utilizing multiple data sources in machine learning. The research questions in chapter 1.2 have incorporated this.

2.3.3 Search Process

In order to find relevant literature, relevant sources should be selected. This SLR includes multiple sources in order to make the study more thorough and have more rigor. For this study, the following scientific databases are queried:

- Scopus;
- Science Direct (Elsevier);
- Web of Science.

The keywords query used in these databases is the following:

("Federated Learning" OR "Distributed Learning") AND "Machine Learning"

The query is quite broad and encompasses all research questions. The addition of Distributed Learning as a term is added because the concept of Federated Learning is sometimes also referred to as Distributed Learning. Before 2017 it was always referred to as a form of Distributed Learning. This comprehensive and broad search is possible because the research area is still relatively new and small, and in this way does not exclude potential papers for the sake of a more narrow and practical search. The results will next be manually filtered out based on exclusion criteria. The complete queries per search engine can be found in Appendix A.

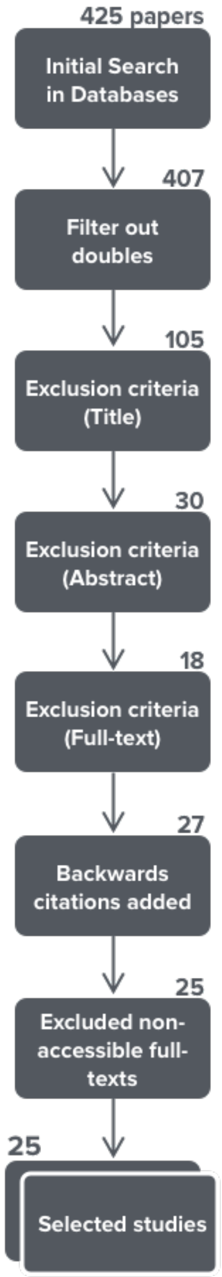

425 papers
Initial Search in Databases
407
Filter out doubles
105
Exclusion criteria (Title)
30
Exclusion criteria (Abstract)
18
Exclusion criteria (Full-text)
27
Backwards citations added
25
Excluded non-accessible full-texts
25
Selected studies

Figure 2.3.2
SLR search process

As there are multiple research questions, one can ask why there was only one search query used in this SLR. The reasoning behind this is as follows. Firstly, Federated Learning is a relatively new research area (as can also be seen in the histogram, Figure 2.3.2) and the number of papers are still very limited. It is therefore still practically possible to manually select studies based on the exclusion criteria, instead of using a more narrow search term. Secondly, an initial exploratory search showed that most papers found include: an explanation of a (new of existing) Federated Learning method, a definition of Federated Learning, a literature review, and an experiment or case study where this method is tested and evaluated. So there is an overlap in the papers' contents and the research questions. Thirdly, making the search term more narrow yielded in the exclusion of some of the earlier found relevant and valuable papers (in the exploratory search). Lastly, adding more keywords (like: data skew, local context, local sphere, feature consolidation, feature fusion, over-fitting, and more) did not expand the search to more found studies. Therefore, one broad search is conducted, which is later manually refined and categorized per research question, as can be seen in Figure 2.2.

Next to finding studies by means of query-based search, Wolfswinkel et al. (2013) additional propose to conduct a backward citation search to also include cited studies, which were not included in the initial search, but are relevant in answering the research questions. The process conducted in this SLR is as follows. While reading the full texts of the selected studies and extracting the information in the extraction form (mentioned in the next paragraph), relevant citations are added to the extraction form based on reading their title, then abstract and lastly the full text. Provided, of course, they meet the inclusion and exclusion criteria specified.

In the SLR the initial search of papers is filtered down to include only relevant papers to this study. In Figure 2.2 this process of filtering papers from the initial search to only relevant papers is shown.

2.3.4 Inclusion and Exclusion criteria

In this study papers which are relevant to the specified research questions are included, i.e. where the main topic of the research is Federated Learning. Especially those who include both a description of a (new of existing) Federated Learning method, and an experiment or case study which evaluates and/or compares this method. These studies provide the most comprehensive view and provide information for multiple research questions, and therefore take precedence. In the pre-mapping phase it became clear that experiments and case studies often both introduce a new method, compare it to other methods, and perform some evaluation, which is primarily the information this study is about.

In order to exclude non-relevant papers in the broad search specified before, exclusion criteria are specified. These criteria provide a systematic way for the researcher to exclude those papers not relevant to the research questions. This is done by either looking at the title, the abstract, or the full text of the paper, and is conduced in subsequent stages, each with a more in-depth view of the paper, for speed and practicality.

Next to the inclusion criteria, exclusion criteria should also be defined, as these are used to filter out papers which are not relevant to this study. The exclusion criteria in this SLR are defined as:

- Papers not related to the research questions;
- Publications which are leaflet papers;
- Papers not in English;
- Papers published before 2011;
- Duplicate papers;
- Very technical papers, related to:
  - Adapting a (sub-)algorithm for ML;
  - Image Recognition;
  - Constraint problems;
  - Communication efficiency, optimization problems;
  - Processor optimization;
  - Network optimization;
  - Wireless network efficiency, bandwidth optimization; and
  - Optimization for distributed processing;
  - Privacy-preserving algorithm development is the main topic; and
- Blockchain is the main topic;
- Big data is the main topic;
- Privacy considerations from a legal perspective is the main topic;
- Not related to Federated or distributed learning as the main topic of the paper in the title, abstract, full-text.

The reasoning for these exclusion criteria is the following. Leaflets are left out because they typically are very short and therefore provide not enough explanation. Non-english papers are left out because the researcher is not familiar with other languages.

In the pre-mapping phase it became clear that the research area of Federated Learning is still relatively new. The earliest mention of Federated Learning is from McMahan et al (2017). Papers before that did mention the concept of Federated Learning but the term was still different, i.e. a form of privacy-preserving Distributed Learning. The earliest paper for this found in the pre-mapping phase was from 2012, therefore it was of no use to include papers in the search from before that time. Also, these earlier papers mostly only contained information useful for historical context, the main interests of this research were mostly addressed from papers of 2015 and later. Therefore, papers before 2011 are not included.

During the filtering phases by title, abstract, and full-text it became apparent that many of the found papers were of very technical nature. Mostly about optimizing a part of an algorithm, processing optimization, (wireless) network optimization and more. These are excluded from the study as they focus too much on technical details not relevant to the research questions.

Next, papers primarily concerned with Blockchain and Big Data are also excluded, as they only mention Federated Learning as a side-case. They do not concern themselves with any of the research questions. The same goes for papers which take a primarily legal perspective on Federated Learning.

2.3.5 Quality Assessment

From the output of the previous step, a collection of selected studies, the next step of the SLR is conducted; the quality of each paper is assessed. This is done by evaluating these studies by making use of some quality assessment questions, which are an adaptation of the proposed questions of Kitchenham and Charters (2007). As stated in the previous section, studies which both include the definition of a Federated Learning method, and an experiment/case study which evaluates this method take precedence, as they provide information for multiple research questions. The quality assessment questions used in this SLR are:

- Relevance of study to the research questions. (yes, partial, little, none);
- How well are the practices or factors defined? (Yes, partial, not);
- How clearly is the research process established? (yes, partially, not);
- How clearly are limitations of the work documented? (yes, partially, not).

The quality assessment helps in a subsequent step, the data synthesis. When two conflicting statements are made, this quality assessment can be used to differentiate between statements, giving precedence to high quality studies. Additionally, while presenting the results, can be used as a form of discussion, doubting the results and validity of poor quality studies. The quality assessments are recorded in the data extraction form.

2.3.6 Data Extraction

The next step in the SLR is extracting the data in a systematic way. This process is done by making use of a data extraction form while reading the full text of each study, as suggested by Kitchenham and Charters (2007). The data extraction form template used can be found in appendix B.

In this data extraction form, the information of each research question is stated in a structured manner. In addition, to allow for some unstructured thinking, a freeform column is added to write down potential topics of interest about this paper. Next to this, for each paper, it is recorded what kind of study this paper encompasses, what the main goal, main contribution, and main finding of the study is, and what kind is research method is used. This structured way of extracting information gives a practical and systematic starting point for the next step, the data synthesis.

2.3.7 Synthesis and reporting

This study aims at structuring the results in a concept-centric way where possible, and fall back on the author-centric approach when concepts cannot be clustered in a more granular manner, which is a recommendation stated by Webster and Watson (2002). The approach in order to cluster the found papers in concepts and to provide a practical overview of the information in each paper per research question is chosen to be performed by using an extraction form (Kitchenham and Charters, 2007). It extracts information on a per-paper basis in a structured way,

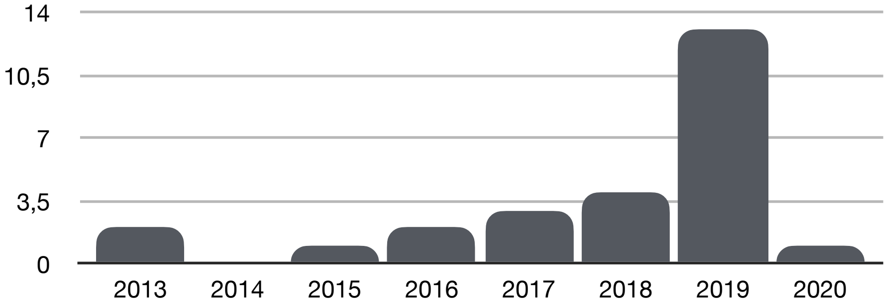


Figure 2.3.3 - Number of papers per year of publication

clustering relevant information in specified columns, in order to be able to answer the research questions (Rouhani et al, 2015).
In Figure 2.3.3 a histogram is plotted to show the distribution of the publication dates of the papers used. Most literature is from last year (2019) and only 12% of this research is based on books, which indicates the newness of this topic in research. The fact that only one included paper is from 2020 is because this study is conducted in March 2020, the papers of this year are still being written.

The spike in 2019 could indicate that the research area is gaining momentum. As the first study which formally defined Federated Learning was published in 2017 (McMahan et al, 2017), and can be seen as the formal start of this research area. It could be the case that from that point onwards other researched started building on top of this knowledge. The fact that 9 out of the 12 papers published in 2019 refer to McMahan et al (2017) strengthens this hypothesis. The two-year gap could be explained by the fact that research still had to be performed and published. It would be interesting to replicate this study at the end of the year, and see whether even more papers are published in 2020.

Table 2.3.1 - Study types

Study	Count	Percentage
Journal paper	18	69%
Conference proceeding	5	19%
Book section	3	12%

Next, in Table 2.3.1 the distribution and percentage of the used paper's study type are presented. As can be seen, the majority of the studies are journal papers, following by a small percentage of conference proceedings, and with even a smaller percentage book chapters. The low number of book chapters could be explained by the fact that the research area is still relatively new, and a book is usually published after the research area begins to mature. Also the publishing time of books could be longer and therefore are underrepresented. It is, however, peculiar that the number of journal papers overshadows the number of conference proceedings, given also the fact of the newness of the research area. One could argue that conference proceedings are published quicker than the more elaborate publishing in a journal. However, when diving deeper into the data, it shows that the papers from 2017 and before (8 cases) are mostly journal papers (6) and a book (1). While almost half (5) of the research published in 2019 are conference proceedings. With this more detailed breakdown the statistics confirming the newness of the research area is less peculiar.

2.4 UTAUT Model

The fifth phase of the DSRM constitutes the evaluation of the proposed method. The evaluation survey questions will be based on the UTAUT model by Venkatesh et al. (2003), the Unified Theory of Acceptance and Use of Technology model. This model provides a way to assess the likelihood for a new system to be accepted successfully in an organization and therefore fits the purpose of this evaluation. The UTAUT model and its usage in this study are described in Chapter 6.

2.5 CRISP-DM

In the case study, the method results in a choice for a Federated Learning algorithm. To validate the applicability and practicality of this result, another case study is executed where the this Federated Learning is implemented (alongside the development of two local Machine Learning models and a Centralized approach for comparison). This case study is presented in Chapter 7.

To execute the development of these Machine Learning model the CRISP-DM research methodology of Chapman et al. (2000) is used. This is a leading methodology for doing data science-related research (Kurgan & Musilek, 2006). It provides an academically-backed and

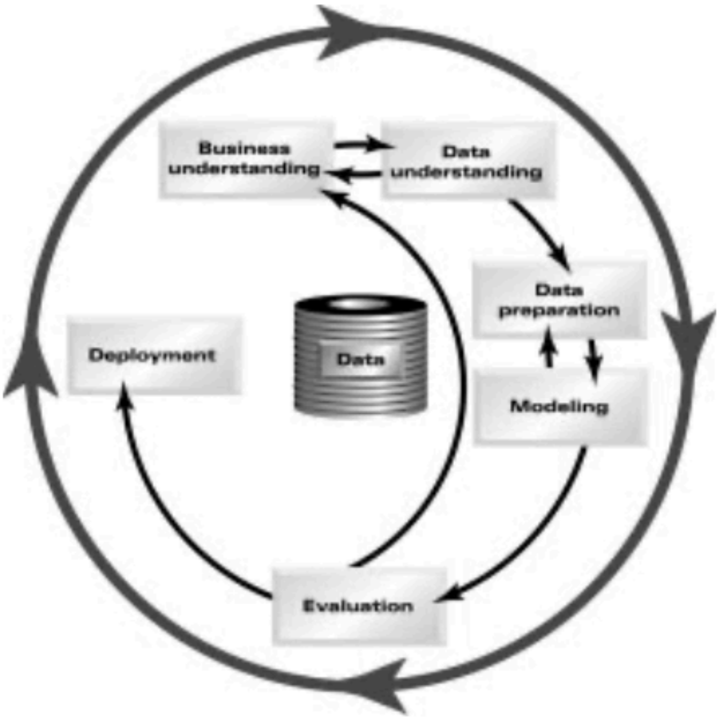


Figure 2.5.1 - CRISP-DM Cycle (Chapman et al., 2000)

structured way to perform data science-related research. It provides the researcher with a method to work in a systematic manner, both advancing the documentation and the reproducibility.

CRISP-DM consists of 6 phases: business understanding, data understanding, data preparation, modeling, evaluation, and deployment. The methodology model is shown in Figure 2.5.1. For each of the phases, guidelines are provided. The methodology does not follow a strict order; usually earlier phases are revisited as more knowledge has been obtained in later phases. The documentation of the method is, however, shown in the shown order for structure and readability.

Next, the 6 phases are shortly described (Chapman et al., 2000):
- **Business understanding**: focuses on the objectives from a business perspective;
- **Data understanding**: to get familiar with the data, and do a data quality assessment;
- **Data preparation**: to construct the input data set for the model from the raw data, this involves data transformation and data cleaning;
- **Modeling**: models are selected, applied, and their parameters are calibrated to attain optimal values;
- **Evaluation**: the model results are evaluated and discussed;
- **Deployment**: communicating the results to the target users (i.e. via this report).

Each of these phases are executed in the case study in Chapter 7.

3. Literature Review

In this chapter the results of the Systematic Literature Review (SLR) are presented. The used research methodology is that of Kitchenham and Charters (2007), which is described in detail in chapter 2.3. Each research questions is answered per paragraph. Each research question is first introduced, its additional reasoning is added, if available, then the results are given, and lastly each research question has a conclusion and discussion.

3.1 Research Question 1 - Federated Learning Definition

What is the definition of Federated Learning according to the literature?

As Federated Learning is quite a new area of research, definitions of the topic can still be less formally defined and some deviations from definitions may occur between papers, these matters will be discussed.

This research question will be answered by synthesizing and consolidating definitions stated in the papers included in the SLR. The aim is to understand what Federated Learning is from a definition standpoint. The definitions of multiple authors are laid out and synthesized. In addition, the major characteristics of Federated Learning are identified in papers who do not formally define Federated Learning. The purpose of this research question is to gain insight into this new field of study, to gain a clear and thorough definition, and to provide context to the next research questions. For this, first, context is added by means of investigating Federated Learning's history. In this way a comprehensive view of Federated Learning will be laid out in order to understand this area of research better.

3.1.1 Federated Learning history

Federated Learning is a relatively new term. In this section the history of Federated Learning is laid out.

The first paper to introduce and formally define what Federated Learning is that of McMahan et al (2017) and can therefore be seen as the origin. To be more thorough, Federated Learning goes even further back, as McMahan et al. also published an earlier paper that laid out the technical groundwork of Federated Optimization in 2015, but Federated Learning itself was yet to be defined. This earlier paper was mainly concerned with designing a communication-efficient optimization algorithm in a federated setting and is therefore not included further in this study.

Besides that, the concepts and practices preceding the formal definition of Federated Learning already existed before this paper was published. The academic need for what Federated Learning currently represents was already there, and the papers mentioned in the next paragraph indicate this. These papers laid out the preliminary groundwork of what would later become known as Federated Learning.

Before Federated Learning was formally defined by McMahan et al., 2017, it was known as a form of Distributed Learning. Distributed Learning can therefore be seen as a precursor to Federated Learning. Peteiro-Barral et al. (2013) perform a survey on existing algorithms in Distributed Learning, which are all predominantly concerned with how to combine separately learned models. Allende-Cid et al. (2013) show that Distributed Learning can be utilized to make better predictions in weather forecasting than local-only models, by designing their own distributed adaptation of an ensemble learner. Privacy protection of the data is not considered a concern still, but is, however, a major factor in the newly designed Distributed Learning approaches by Gong et al. (2016), Jochems et al. (2016). Gong et al, 2016, adapt an existing logistic regression algorithm, by decomposing it via a mathematical technique called the alternating direction method of multipliers (ADMM) [*Appendix E: Definition 7*]. With the usage of ADMM the model is split up into smaller subproblems which can be distributed over multiple clients. These authors are the first to be concerned about privacy-preservation of the distributed data, as it is healthcare data which is of highly sensitive nature.

The difference between Federated Learning and Distributed Learning is not clearly defined, but papers such as McMahan et al. (2015, 2017), suggest that Distributed Learning is mainly concerned with increasing processing power by parallel computation, which is a more technical perspective on Machine Learning. McMahan et al (2017) even lists four key properties that differentiate federated learning from distributed learning, which are: data is non-iid, unbalanced, massively distributed, and there is limited communication available with edge devices/clients.

Data which is iid means that the data is independent and identically distributed, whereas non-iid is the negative of this. This means that for iid data each data point is drawn without relation to the previously drawn data point(s).

3.1.2 FL definition

What exactly is federated learning? A good starting point is to look at the definition of federated learning. However, as multiple authors define this differently, the definitions of federated learning from multiple authors are listed and briefly discussed.

Yang et al (2019) give an informal definition of federated learning: "Their main idea [of federated learning] is to build machine learning models based on data sets that are distributed across multiple devices while preventing data leakage". To put it plainly, federated learning is a distributed form of machine learning which takes privacy considerations in mind. Stating both the distributed nature and the importance of privacy, which is a common theme in the literature.

Li and Smith (2019) define the federated learning problem as follows: "The canonical federated learning problem involves learning a single, global statistical model from data stored on tens to potentially millions of remote devices. We aim to learn this model under the constraint that device-generated data is stored and processed locally, with only intermediate updates being communicated periodically with a central server." and add that the goal is to optimize an objective function. This function makes a sum of the local objective function over all devices, and minimizes this function. They describe federated learning in a more technical way, and the importance of local computation and only sharing model updates, which is essentially a privacy concern.

McMahan et al's (2017) definition states Federated Learning as: "a learning task solved by a loose federation of clients which are coordinated by a central server. Each client has a local training dataset which is never uploaded to the server. Instead, each client computes an update to the current global model maintained by the server, and only this update is communicated". They later make the distinction with Distributed Learning by stating: "It's not completely distributed learning, as there is still a central server, and some trust in this central server is required."

It is also important to make the distinction between the term federated learning and Federated Optimization. Federated Optimization was first introduced by McMahan et al (2015). *Federated Optimization* is the task that solves the federated learning problem, presented in an algorithm. Formally, Federated Optimization is merely the algorithm and part of the concept of Federated Learning, the latter includes more facets like design, privacy, etc. The term Federated Learning is, however, also used instead of Federated Optimization in other papers.

This Federated Optimization problem can be seen as the cost function in traditional Machine Learning. So, in the same manner, a minimum for this cost function needs to be found. This is, for example, done by (stochastic) Gradient Descent in traditional Machine Learning, and some implementations of Federated Learning (Duan, 2019). In these terms, the cost function of Federated Learning can be explained in a simplified manner as: the summation of each local data site's cost function aggregated. In other words, a minimum should be found when the sum of each of the locally situated cost functions is at some minimum.

The formula of Li and Smith (2019) could make this clearer. They state that the goal of Federated Learning is to minimize the following objective function:

$$\min_{w} F(w) , \text{ where } F(w) := \sum_{k=1}^{m} p_k F_k(w)$$

where m is the total number of devices, w is the input parameter (i.e. input training data), F_k is the local objective function, and p_k specifies the relative impact of each device. This relative impact is usually set as: $p_k = 1/n$ or $p_k = n_k/n$, where n is the total number of training examples, and n_k is the number of training examples of a particular local device k. (Li and Smith, 2019)

Next to these given definitions, other characteristics of Federated Learning are identified by papers who do not give a formal, explicit definition of Federated Learning. This is done to reflect the overlap in the characteristics of all definitions in the literature. In the literature, Federated Learning is often typed by its topology and challenges, which can be easily translated into characteristics.

3.1.3 Topology

Almost all papers in this study mention a central server to be in the system design. This central server both coordinates the initialization, communication of the algorithm, and serves as the central place for the aggregation of the model updates. In this design the local nodes have some degree of trust in this central server, but still maintain independent and have their own degree on control of whether they participate and take ownership over their local data, the central server does not have access to the original local data.

Li and Smith (2019) confirms this and mention that a star topology, with one centralized coordinating server is the most common architecture used. However, they also mention that some decentralized topologies exist where no central server is present.

Sun et al (2019) explain what Federated Learning is by making the topology with a central server a central theme in the definition: "Usually in federated learning, a server moderates several data sites to carry out optimization iterations, like gradient descent updates, on each data site. Each data site then sends an intermediate result to the server. The server side aggregates the results and distributes it, so that each data site obtains an updated model."

A visual representation of the architecture design with a central server, created by synthesizing the information in the literature study, is given in Figure 3.1.1.

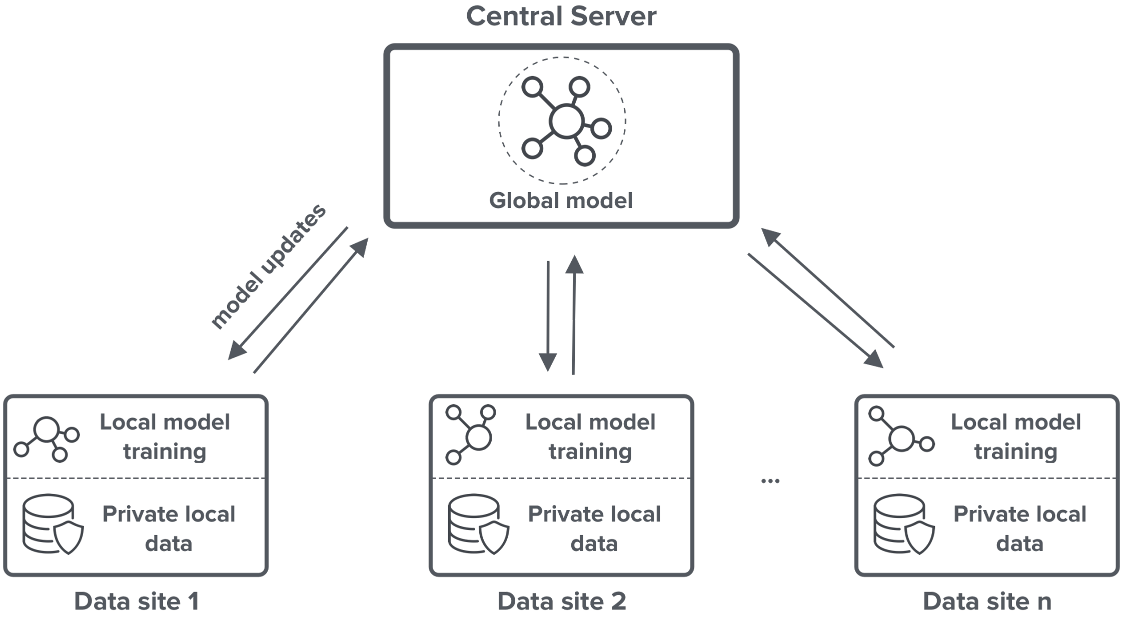


Figure 3.1.1 - Federated Learning's Architecture Design with Central Server

3.1.4 Core Characteristics

To paint a complete picture, major discerning characteristics of Federated Learning are identified in this section. To look at what the major characteristics are, this study takes a look at the core challenges, as these are often stated explicitly, clearly, and can be translated to core characteristics. Li and Smith (2019) list four core challenges associated with Federated Learning:

1. Expensive communication. As the communication overhead is networked and can be several orders of magnitude slower than local computation. Key is (i) to reduce the total number of communication rounds, or (ii) to reduce the size of the transmitted messages.
2. System heterogeneity. Systems, connection types included in the system have high variability.
3. Statistical heterogeneity. Data is often non-iid.
4. Privacy concerns. Cited to be often a major concern in federated learning applications.

While conducting the Systematic Literature Review, these core challenges were prevalently observed. These four challenges of Li and Smith reflect these findings and summarize them well. Below, each challenge is discussed and their support in this literature study is given.

Communication strain

McMahan et al (2017) names communication costs as the major constraint in Federated Learning. Although no primary data is shared there are still communication costs in Federated Learning. The 'data' that is shared are model (or parameter) updates in the training of the Federated model. A (Federated) Machine Learning model requires many iterations of training. For example, by finding the minimum of a cost function via gradient descent. Therefore, there are potentially many communication rounds, which also need to be communicated to potentially a large number of edge-devices. So, although the data itself is not shared, but merely the models updates, the communication costs are predominantly due to the many iterations of training and the the large number of devices this needs to be communicated to.

Li and Smith (2019) acknowledge this as one of the core challenges of federated learning, stating the importance of reducing the communication overhead. Zhao et al (2018) mention that communication cost in Federated Learning is a major challenge. The paper of Sattler et al (2019) goes a step further and adapts the FedAvg algorithm of McMahan to be even more efficient. They do this by designing a sparse ternary compression (STC) framework, which is specifically designed to better compress the communication rounds than standard FedAvg. Which also indicates that Sattler et al (2019) regard communication strain as a major factor in Federated Learning.

Contrary to this, Gong et al (2016), Jochems et al (2016), and Deist et al (2017) do not mention communication strain as a major factor explicitly. However, they do touch upon this issue as they adapt the standard machine learning algorithm to work in a distributed environment. But they all solve this relatively small issue themselves.

The pre-McMahan et al (2017) papers on federated learning (still called distributed learning at this point), like Gong et al. (2016) and Jochems et al. (2016), are case studies in the healthcare sector. The federated learning methods are often only concerned with only a handful of hospitals (the clients). In contrast papers published after McMahan are typically concerned with a vast number of clients. Hard et al (2018) mention their application being tested on 1.5 million clients, end-users' mobile phones in this case. Nilsson et al (2018) mention that clients can be very large in number. This large disparity in the number of clients may be the case why the papers pre-McMahan do not mention communication costs as a major concern, while the main design characteristic of McMahan et al's (2017) Federated Learning algorithm was concerned with reducing communication overhead. Also Sattler et al (2019) names communication overhead as the largest disadvantage of federated learning. To this we may conclude that Federated Learning is designed to work with a vast number of distributed clients, up to (at least) millions, and thus communication strain is a major challenge in Federated Learning.

System heterogeneity

Hard et al (2018) mention that the number of devices can reach up to millions and, in addition, these devices are heterogeneous in nature: they have different characteristics in hardware capabilities and connection types. Nilsson et al (2018) speak of 'a heterogeneous ecosystem of edge devices' as one of the motivations for the development of Federated Learning, confirming this as a major challenge.

In the literature, a distinction often made is that of horizontal and vertical partitioned (or distributed) data. Yang et al (2019), Gong et al (2016), and others touch upon this distinction. The following definitions can be set. *Horizontal distribution* means that the data at each data site have the same features, i.e. columns, but include different subjects, i.e. rows. *Vertical distribution* constitutes the opposite: the data of one subject, i.e. row, is present at multiple data sites. Each data site, therefore, has a different set of features, i.e. columns.

The importance of this distinction lies in what Federated Learning algorithms can be used for that particular case. Some Federated Learning methods include algorithms which can only work with one case, and some work with both. For example, Jochems et al (2016) develop a method which works only for horizontally partitioned data, as previous literature was limited to only vertically partitioned cases. While Gong et all (2016) develop a method which works with both horizontally and vertically partitioned data.

Given the fact that all papers mentioned above list system heterogeneity either explicitly or implicitly, system heterogeneity can be seen as a major challenge.

Statistical heterogeneity
Statistical heterogeneity is also referred to as the use of Federated Learning on non-iid data. It was seen as 'solved' by McMahan et al (2017), and one could therefore argue that it should be disregarded as a core challenge. However, as can be seen in chapter 3.4.5, Zhao et al (2018), Duan (2019), Verma et al (2019), and Huang et al (2019) all criticize this claim of McMahan and see statistical heterogeneity as a major challenge. A more in-depth review on this disparity of views in the literature should be performed to conclude which claim has the best support.

Privacy
Privacy-preservation is mentioned by all papers in this study as being an important factor in Federated Learning. McMahan et al (2017) says privacy is a major concern in sharing real-world data sets. So by providing privacy-preservation, in practice, more parties would be willing to share their data, and can be seen as a major distinguishing advantage for Federated Learning. Privacy is here preserved by removing the need of sharing the data of local nodes, and replacing this by only sharing necessary model training updates with a central server.

Gong et al (2019), Jochems et al (2016), Deist et al (2017, 2020), Brisimi et al (2018), and Huang et al (2019) are all concerned with Federated Learning in the context of healthcare. As this involves working with sensitive patient data, they all list the privacy aspect in their algorithms as of utmost importance.

Hard et al (2018), Huang et al (2019), Sattler et al (2019), Yang et al (2019), and Sun et al (2019) all mention the privacy aspect of Federated Learning too, by not making the data leave the original client. Hard et al (2018) is concerned with keyboard type prediction, and therefore privacy is important as clients can type sensitive information. Sun et al (2019) state: "this distributed model training process circumvents the bottleneck of data transmission and prevents private data from leaving the data center.", marking the central theme of privacy in Federated Learning. Yang et al (2019) even state that data privacy protection of the data owner is a major factor in discerning Federated Learning from Distributed Learning.

While all papers above mention that they do not permit data leaving the original device it’s on, some papers do mention the sharing of data. For example, Zhao et al (2018) mention that their adapted Federated Learning method does share data. However, this is only done to show the accuracy difference of sharing data in Federated Learning or not, and is discouraged to do so if not needed. Only Allende-Cid et al (2013) do not concern themselves with privacy. The data used in this research is, however, merely weather data that is from one organization. In addition, this is the oldest study in this literature review, all other papers after this one did concern themselves with privacy.

There is, however, some discussion on what is enough privacy. While some researchers argue that merely not letting data leave its origin provides enough privacy (by only sharing model updates), others counter this by stating that even model updates can contain sensitive information about the global context, e.g. when even aggregate data is sensitive information, and that additional privacy mechanisms should be added.

Gong et al (2016) suggests that privacy by relying on just model updates may not be enough, and privacy sensitive information could be extracted from these aggregated results, and therefore provides suggestions for additional privacy like a secure summation protocol or homomorphic encryption. Hard et al (2018) confirms this and states that Federated Learning is complementary to privacy-preserving techniques such as secure aggregation and differential privacy. The literature review study of Yang et al (2019) classify this as *indirect information leakage*, and argue that the intermediate results, like parameter updates, which are shared and communicated, provide no guarantee that sensitive information is protected. To mitigate this, Yang et al (2019) give recommendations of three additional security models: (i) Secure Multi-party Computation (SMC), (ii) Differential Privacy, and (iii) Homomorphic Encryption.

Thus, privacy preservation by not sharing edge devices' raw data, but only model updates is present in all Federated Learning methods. Additional privacy-preserving techniques are optional and complementary; their necessity is dependent on the specification of what privacy-sensitive information is and on implementation trade-offs.

Concluding, the core challenges listed by Li and Smith (2019) are a good representation of what is found in this study. These core challenges will therefore be seen as discerning factors and be included as main characteristics in defining Federated Learning. For this study, the latter two challenges are especially of interest, as the former two are of a more technical nature, which is out of scope for this study. Moreover, especially the third challenge about non-iid data is of main importance, as the research shows a discussion, and will be investigated in detail in research question 4.

3.1.5 Conclusion

Given the extracted information stated in this section, a summary can be made about what federated learning is. Here the main characteristics of Federation Learning are synthesized and summarized as the following characteristics:

(i) Federated learning is a distributed form of machine learning, typically utilizing a vast number of heterogenous client-side edge devices as data facilitators;
(ii) Federated learning is conducted by solving a federated optimization problem, by means of a federated optimization algorithm (like Google's FedAvg algorithm);
(iii) Privacy considerations are important, as it typically concerns a vast number of heterogenous clients, and also potentially making use of previously non-accessible data due to privacy concerns;
(iv) Data is kept locally at the client, only model updates are shared;
(v) Data can be horizontally or vertically distributed, Federated Learning methods are not necessarily suited to handle either or both;
(vi) Usually a star topology is used, where there is one central server which takes care of aggregating the model updates, and coordinating the communication with the edge devices;
(vii) The edge devices have some degree of autonomy to the central server, hence the term federated, e.g. autonomy over their own data, processing power;
(viii) An additional optional layer of privacy can be added, like Secure Multi-party Computation (SMC), Differential Privacy, or Homomorphic Encryption, in the case that even the aggregated information in the form of model updates may also contain privacy-sensitive information;
(ix) Next to privacy, communication overhead and data heterogeneity are major challenges in federated learning.

For this study, based on the aforementioned characteristics, topology, history, and other definitions the following definition of Federated Learning is synthesized [Appendix E: Definition 1]:

Federated Learning is a form of distributed machine learning where a global model is trained on a central server utilizing multiple separate heterogenous edge devices, while still preserving privacy by not permitting the data to leave their origin devices.

This study will refer to this definition when it talks about Federated Learning. It includes the origin (distributed learning), topology (central server), system heterogeneity (edge devices), and its major discerning factor from distributed learning: the privacy aspect by means of keeping data local at the edge client. The aspect of communication strain is implicitly included, as this is also inherent to distributed learning. In this way both the topology and all major characteristics mentioned before are included in this definition.

3.2 Research Question 2 - Federated Learning Methods

What Federated Learning methods exist in the literature?

As mentioned in the previous RQ, Federated Learning can be seen as the overarching term which constitutes the architecture design, a federated optimization algorithm, a way of preserving privacy, with sometimes an additional layer of privacy (e.g. differential privacy, homomorphic encryption) added, and more. Only in this context, as discovered in the previous RQ, this RQ can be answered. This RQ is answered by doing a systematic literature review, as stated in the methodology chapter of this report. From all relevant papers found, the Federated Learning method is extracted by summarizing and categorizing it via the previously mentioned extraction form.

After analyzing the results of the literature review mainly two types of methods can be identified, and this is inherent to the research area's short history, mainly: (i) Federated Learning which are either an adaptation of, improvement upon, addition of, or new application of the first proclaimed Federated Learning (Federated Averaging) method of McMahan et al. in 2017 at Google, or (ii) adapted versions of existing Machine Learning algorithms to make them work in a federated setting, but are not referring or building upon the first line of research. The second category, for sake of convenience, will be called proto-federated learning in this study, because the first papers where written before the introduction of the paper of McMahan et al. in 2017, or a continuation of that line of research already in place. The federated learning methods are either named, or on the cases that they are not named, named after the author of the paper.

3.2.1 Proto-Federated Learning Methods

Allende-Cid et al already spoke about '*building a general model by fusing in some manner distributed information*' for predicting wind speed forecasts in 2013, and is the oldest paper in this systematic literature review to do so. Their objective being to build a general model that outperforms local models that only have access to local data. For this they created an adapted regression algorithm to a (horizontally) distributed context. This algorithm is adapted by an ensemble approach, or more specifically: bagging. The algorithm works, broadly, as follows: all local models (or, optionally, just their outputs, to preserve privacy better) are shared with each local data site, these local models are then trained into a global model, where they - simply put - take the average of the preceding local models. This global model is trained by many iterations of model updates, which are computed locally per edge-device, to find the minimum of some cost function.

Next to weather forecasting, Federated Learning was already touched upon before its formal definition in healthcare research. This branch even continued to deviate from this branch of research after the publication of McMahan et al (2017).

Gong et al (2016) adapt a logistic regression algorithm into smaller subproblems, so it can be distributed over multiple clients and locally computed. This splitting into subproblems is based on the mathematical alternating direction method of multipliers (ADMM) technique. The algorithm's aim is to predict dichotomous outcomes in medical diagnosis and prognosis. The algorithm is reportedly also privacy-preserving, by the fact that the logistic regression problem is decomposed in a way that the central server only has to simply average over local classifiers. In order to preserve privacy even more, in the case that even the averages are privacy-sensitive, Gong et al propose two additional techniques: a naive approach and a modified approach. The naive approach preserves privacy more by using a secure summation protocol. The modified approach involves homomorphic encryption, which increases privacy-preservation even more. Moreover, the algorithm is evaluated empirically, with real-world medical data, and show similar results with a centralized approach, after 40 iterations.

Deist et al (2017) also base their adapted approach on ADMM, but applied to a support vector machine (SVM), and later continues this line of research by also using an adapted (based on, again, ADMM) logistic regression model to predict post-treatment survival in lung cancer patients (Deist et al, 2020). The algorithm also preserves privacy by ensuring that no data leaves the local data sites. They evaluate their algorithm empirically, and show that it provides similar results compared to a centralized approach, while preserving privacy.

Jochems et al (2016) take on this problem without using ADMM as a basis to make the machine learning algorithm work in a distributed context, like the previous research line, but instead adapt a Bayesian network model themselves to be suitable for distributed use. The aim is to predict dyspnea, with data like tumor location, lung function tests, and more. Just as in the previously presented algorithms, Jochem's algorithm also states that it provides privacy by not letting the data leave local data sites (i.e. the individual hospitals). The algorithm is evaluated empirically, on medical data, and compared to locally trained models. The area under the curve (AUC) performance is 0.67. In addition, the results show that the model performs better in some hospitals (Eindhoven and Liege), and worse in others (Maastro, Aachen, Jessa).

Lastly, Brisimi et al (2018) also do not base their adapted SVM for distributed usage on ADMM, but instead use their own custom approach. They develop an iterative cluster Primal Dual Splitting (cPDS) algorithm for solving the large-scale sSVM problem in a decentralized fashion. The aim is to solve a binary classification problem to predict hospitalizations for cardiac events.

3.2.2 Federated Learning after McMahan

McMahan et al (2017) first introduced the formal definition of Federated Learning. Important is the separation made between Federated Learning as the overarching term, and Federated Optimization, the optimization task, embodied in an algorithm, which is inherent to Federated Learning. It performs the logic of training the local models, communicating and combining the training steps into a global model. In this RQ Federated Optimization is seen as the method.

The method that McMahan et al (2017) introduce is called FederatedAveraging (FedAvg). It is an algorithm that combines stochastic gradient descent (SGD) on each client with a server that performs model averaging, such that the largest stated bottleneck of Federated Learning, the communication overhead, is reduced by a factor of 10 to 100. The FedAvg optimization algorithm is applied to a neural network model in this case. Hard et al (2018) use this same FedAvg algorithm and apply it to a practical case of mobile keyboard prediction, in which is shows better prediction recall than the set baseline. As stated in the introduction paragraph, most other papers are adaptations of this FedAvg algorithm, which will be discussed next.

Next is the Federated Stochastic Block Coordinate Descent (FedBCD) method of Liu et al (2019). The researcher developed a method which is one of the only Federated Learning methods to support vertically partitioned data. They evaluated the results on multiple data sets with good AUC results (84% and 99,7%).

Nilsson et al (2018) discuss both Federated Stochastic Variance Reduced Gradient (FSVRG) and CO-OP as alternative methods to FedAvg. FSVRG is based on the idea that it performs one expensive full gradient computation centrally, followed by many distributed stochastic updates on each of the local clients. A stochastic update is then performed by iterating through a random permutation of the local data. CO-OP differentiates itself from both FedAvg and FSVRG as being an asynchronous approach, instead of a model with synchronous model updates. This approach immediately merges any received client model update with the global model, where the age difference of the model keeps track of how to compute the weights.

Sun et al (2019) develop a rather different approach where local data influence is seen as most important for the training process. They introduce the Restrictive Federated Model Selection

(RFMS) method in which the local data sites are only trained on their local data, but a federated approach is used for hyper-parameter optimization. They justify this approach in stating that empirical results confirm the importance of globally tuned hyper-parameters, as locally tuned hyper-parameters generalize poorly over data sites. This is one of the few papers to give precedence to local training, which merely uses Federated Learning as a preliminary step. A direct comparison on predictive performance to conventional Federated Learning methods is not made.

A federated recommender system is developed by Jalalirad et al (2019). The main difference with other federated learning methods is that they seek to achieve a new balance between local and global training, essentially shifting more responsibility to the local clients by allowing them to give more importance to the local client's data. After the global training is completed, each edge device trains a local version too, fine-tuning the globally trained vector based on personalized data. The paper states that their model outperforms traditional federated learning methods. In the case of non-iid (non-independent and identically distributed) data, the method does not perform at the same level of accuracy as a centralized method, indicating that the increase in privacy comes at a cost of accuracy in this case.

This performance problem with non-iid cases is also addressed by Zhao et al (2018). They developed an adapted federated learning method, with the main aim of addressing this reduced accuracy in non-iid settings. In short, the method initializes, before the start of the FedAvg algorithm, an initial well-balanced model is used, instead of a random one. This creation of this initial model involves a preliminary step of creating a globally, well-balanced, data set which they then mix into local data sites which have a different data distribution. This method does unfortunately introduces the need to share data once again, and may not be suitable in many cases where privacy is important.

Other researchers also acknowledge this problem and create their own adapted federated learning method. Like Shaoxiong et al's (2019) Attentive Federated Aggregation algorithm (FedAtt), Duan's (2019) automatic self-balancing Astraea framework (sic, method is a better term), Huang et al's (2019) CBFL clustering approach, and Verma et al's (2019) unnamed model to better perform on skewed data. In addition, the papers of Sattler et al (2019), Schmid et al (2019), and Wang et al (2019) also address the problem of non-iid setting in federated learning. This topic of non-iid settings in federated learning deserves to be investigated in more detail, and will therefore be addressed in the next and especially in the fourth research question, as it is better suited there. The found Federated Learning methods are listed and summarized in Table 3.2.1.

3.2.3 Higher-level methods

Additionally, there is a category identified in this study which does not develop a new federated learning algorithm itself, but concerns itself with the method around it, essentially providing a higher-level overview of how to improve existing or to-be developed federated learning algorithms. For completeness these higher-level methods are also discussed in this subsection.

A high-level method, although named a framework by the authors, to improve federated learning models is presented by Ilias and Georgios (2019). They identify existing problems with federated learning, which are: privacy, management of participating nodes, and (data) integrity. For each they provide a methodology in how to address this problem and a recommended practical solution. For privacy they recommend the usage of homomorphic encryption, in addition to the current practice of not sharing data beyond its owner. Next, they address the management problem by using smart contracts in blockchain. Lastly, on the data integrity problem they also suggest using blockchain as a solution. In this way a distributed validation system can be created in order to test whether false data is added, or wether a node is not participating equally in the processing power distribution. The paper does, however, not evaluate its method empirically, questioning its real-world usefulness.

Next, the paper of Malle et al (2017) proposes a workflow architecture to better design future federated learning methods and systems. The main elements Malle et al consider to be important, and should be included into the workflow are: (i) client-side machine learning, (ii) privacy, (iii) interactive machine learning, and (iv) distributed bagging. The main contribution of this paper is, however, the introduction of a local sphere. Which originated from the old technique of bagging, but adapted to a distributed context. This method is, like the previous one, not evaluated

Table 3.2.1 - Federated Learning (FL) Methods

Method	Description	Type of FL
Allende-Cid et al's (2013) method	Adapted regression algorithm, for distributed use. Adapted by an ensemble approach	Proto-Federated Learning methods
Gong et al's (2016) method	Logistic regression model (local ML model) adapted for distributed use by ADMM	
Deist et al's (2017) method	Support vector machine, adapted for distributed use by ADMM	
Deist et al's (2020) method	Logistic regression model, adapted for distributed use by ADMM	
Jochems et al's (2016) method	Bayesian network model, adapted for distributed use (no ADMM)	
Brisimi et al's (2018) method	Adapted SVM for distributed usage	
Federated Averaging (FedAvg) - McMahan et al (2017)	First formal method. Used as baseline in most papers	FL methods for iid data
Federated Stochastic Block Coordinate Descent (FedBCD) - Liu et al (2019)	One of the few Federated Learning methods that primarily supports vertically partitioned data.	
Federated Stochastic Variance Reduced Gradient (FSVRG) - Nilsson et al (2018)	Main difference with FedAvg: one expensive full gradient is computed centrally, then next local iteration randomly	
CO-OP - Nilsson et al (2018)	Asynchronous algorithm	
Restrictive Federated Model Selection (RFMS) - Sun et al (2019)	local data sites trained only on local data, only federated approach for hyperparameter optimization	
Federated recommender system - Jalalirad et al (2019)	New balance made between local and global training, shifting more responsibility to the local clients	
Zhao et al's (2018) method	Adapted FL method for non-iid data settings. Needs data sharing	FL methods for non-iid data
Astraea method - Duan (2019)	Adapted for non-iid usage. No data sharing needed	
Attentive Federated Aggregation algorithm (FedAtt) - Shaoxiong et al (2019)	Adapted for non-iid usage. No data sharing needed	
CBFL - Huang et al (2019)	Clustering approach for non-iid settings	
Verma et al's (2019) method	Method adapted to work for skewed data	

empirically. The overlap between the two methods is, however, that the current privacy preserving approaches of federated learning methods are lacking, and should be improved.

3.2.4 Conclusion and discussion

Concluding, there are a myriad of federated learning methods, each developed in their own context. The discovered methods are listed in Table 3.2.1. The plethora of federated learning methods can be divided into origin: the proto-federated method, and the federated learning methods after McMahan. The proto-federated methods can be described as existing machine learning methods, decomposed and adapted for distributed usage. Privacy in these methods is usually preserved by not sharing the data of the edge devices (local data sites), but merely sharing the local model updates, which can also be said about the federated learning methods after McMahan. An additional, optional, layer of privacy via homomorphic encryption or secure aggregation is also often introduced, also mentioned in both types of federated learning.

The federated learning methods after McMahan, are typically an adapted version of the original FedAvg algorithm. Also, to make these federated learning methods work better with non-iid data, which is an often mentioned problem, an extra initial step which includes some type of balancing mechanism can be added. Federated Learning can therefore be seen as a three-layered system: (i) an optional initial balancing mechanism (to suit non-iid data better), (ii) the federated optimization algorithm itself, either before of after McMahan, and (iii) an optional privacy-layer of homomorphic encryption or secure aggregation, which alters the algorithm. Lastly, high-level methods can be used to develop better federated learning algorithms in the future, but, however, are not yet tested empirically so their real-world value still has to be tested.

It should also be noted that typically each author claims that their newly-developed federated learning method performs at least similarly or better than some other method, and usually evaluate and test their method on a newly introduced data set. The latter is exactly the case why these contradicting claims can be made; the performance is likely dependent on the data set used. Therefore, it is essential for the evaluation of these methods to be tested by a third-party on the same data set. More on this comparison of performance will be discussed in the next research question. Also, this research question was intended to only identify existing methods in the literature. It is advised to identify more relevant characteristics of these methods in a later study.

3.3 Research Question 3 - Differentiating Characteristics Introduction

What are the main differentiating characteristics of Federated Learning methods found in the literature?

This research question is answered by means of conducting a Systematic Literature Review (SLR), as described in the methodology section 2.3. For this specific research question, all studies which are mainly about introducing or describing one or more Federated Learning methods are included. These are, therefore, the same set of studies used in research question 2. The relevant information is extracted and documented in three rows of the data extraction form, each extracting information of one of the 3 differentiating characteristics identified. Those differentiating characteristics are identified and described next.

3.3.1 Differentiating Characteristics

First, the differentiating characteristics types need to be identified. For this, a clear definition has be be constructed first. The *differentiating characteristics of Federated Learning methods* are defined as: characteristics of Federated Learning methods which both *(i) limit options or impact the desired outcome regarding a organization's data-related characteristics and privacy considerations, i.e. those that are relevant to the to-be-designed method, and* (ii) have variation in implementation among the Federated Learning methods, i.e. not all Federated Learning methods have the same implementation regarding this characteristic. *[**Appendix E: Definition 3**]*

This definition contributes to the overall research goal of designing a method to choose the most suitable Federated Learning method, as a choice should be relevant. Relevant characteristics to these organizations are those which may limit options or impact the desired outcome regarding the organization's data-related characteristics and privacy considerations, which is addressed by criterium (i). Also, when there is no variation in the implementation of these characteristics, they can be regarded as static and inherent to Federated Learning itself. They are, then, not relevant to making an informed choice, which is addressed by criterium (ii). Given this definition the differentiating characteristics can be identified in the literature.

Looking back at the results of research question 1, the definition of Federated Learning, four core challenges of Federated Learning were identified: (i) expensive communication, (ii) system heterogeneity, (iii) statistical heterogeneity, (iv) privacy concerns. These four challenges can be regarded as differentiating Federated Learning, as a concept, from traditional machine learning. This will be the starting point of identifying the differentiating characteristics. From this several alterations are made. The first alterations are to expand this list of potential differentiating characteristics, as it might not be inclusive enough for its purpose. After that, all found characteristics are tested to the definition drawn and those who do not meet the definition are excluded.

The first addition will be the underlying machine learning model and machine learning problem type. It was prevalently mentioned in each of the used studies from the literature review used to answer Research Question 2. Looking at the definition, it fits both criteria. There is found to be a large variation among the used machine learning models used, i.e. many different machine learning models are used to developed Federated Learning methods, satisfying the second criterium. Next, there is also an impact on the choice, as an organization will be limited in its choice if it wants to solve a specific machine learning problem. Then, not all machine learning methods support this problem type (e.g. when it wants to solve a clustering problem, a linear regression model will not suffice).

The second addition that was considered was the supported topology of the Federated Learning method. As found in Research Question 1, there are two main types of topologies in Federated Learning: centralized (star topology) and decentralized. It fits the first requirement of a differentiating characteristic: it has impact on the choice as it can limit options. However, all found Federated Learning methods in this study used the centralized star-topology and it does, therefore, not satisfy the second requirement of a differentiating characteristic of having variation. Therefore, topology is not added as a differentiating characteristic.

The third addition that was considered was the characteristic of communication costs in Federated Learning. It was, however, excluded because of the following. Although this characteristic will have an impact on performance, results do not show any variation among Federated Learning methods of this characteristics. Essentially all Federated Learning methods cope with expensive communication with largely the same impact. So, it is of no use to choose one Federated Learning method over the other because of large communication costs. Therefore, it does not satisfy both requirements of differentiating characteristics and is excluded.

Next, *system heterogeneity* is about differences and variation among used edge devices and *data set partitioning*, as identified in research question 1. For the purpose of refitting it to a differentiating characteristic, only the data set partitioning is extracted from this. This is because this facet can be clearly defined and therefore the differences can also be stated clearly. Which is not the case in variation among edge devices. Furthermore, data set partitioning has a clear impact on the choice of a Federated Learning method, as data set partitioning can be seen as a static, unchangeable characteristic of an organization's data landscape, i.e. an organization cannot easily change the current partitioning type, especially not if data sharing is limited or prohibited, identifying what type of data partitioning is present is crucial in the development of the artifact of this study. As options to what Federated Learning method is then available will be limited. Also, there is variation observed in which type of dat partitioning each Federated Learning method supports.

Next up is *statistical heterogeneity*, which can also be stated as *non-iid data support*. Non-iid data stands for non-identically distributed and independent data, this will be elaborated more later in this study. It fits both criteria, as there are Federated Learning methods which are found to be better suited for non-iid data sets in terms of predictive performance or having a higher accuracy of the resulting model (addressed both the impact and variability criteria). *Predictive performance* will, with the same reasoning, and because it is tied to non-iid data support, also be added as a differentiating characteristic.

Lastly, *privacy concerns* are tested to the definition of differentiating characteristic. There are Federated Learning models which are found to violate the non-data sharing guarantee which almost all Federated Learning methods do support, which organizations expect to rely on to protect their data. Which addresses both criteria.

Given this analysis, the following list of differentiating characteristics of Federated Learning methods is constructed:

1. Data partitioning, i.e. system heterogeneity;
2. Underlying machine learning models;
3. Privacy guarantees;

4. Performance (accuracy, predictive performance);
5. Non-iid data support, i.e. statistical heterogeneity.

The first three differentiating characteristics (1-3) are investigated in this research question and chapter, as they can be observed independently. Characteristics 4 and 5 are observed to have some interrelationship with each other; a trade-off can be made to satisfy one or the other more. There is not a Federated Learning method which completely supports non-iid data, but there are methods which perform, to some extend, better than others. Therefore, these characteristics will be investigated in their own research questions. For these characteristics, an analysis will also be conducted how it relates to the consolidation technique, which is unique and defining to Federated Learning, to explain this interrelationship between performance and non-iid data better.

The remainder of this section is structured as follows: first, definitions for the differentiating characteristics (1) data partitioning, (2) underlying machine leaning model, and (3) privacy guarantee, will be constructed. After that, the results are described, summarized in a table, and conclusions are drawn.

3.3.2 Differentiating Characteristic 1: Data Partitioning

This section discusses the first differentiating characteristic of Federated Learning: data partitioning. First, the reasoning as to why data partitioning is a differentiating characteristic is shortly revisited. Second, the methodology of information extraction for the type of data partitioning is described. Third, a definition and distinction between two types of data partitioning in Federated Learning is given. Lastly, in the next section the results of the systematic literature study are shown.

As already identified, data partitioning can be classified as a differentiating characteristic of Federated Learning methods. It both: has an impact on the limiting of options organizations have, and have variation in implementation among the Federated Learning methods, satisfying the requirements of the definition. This is because some methods, such as Jochems et al's (2016) Federated Learning method, only work with one type of partitioned data (horizontal or vertical). The data characteristics of an organization should therefore be explicitly defined, as this characteristic is defining in which Federated Learning methods are available to this organization. In this way a new method fragment can be constructed, which contributes to the overall research goal of helping the organization to choose a good fit between the Federated Learning method available and the organization's specific situation.

Before the information about data partitioning can be extracted, a clear definition has to be given. Only with a clear definition the data partitioning information can be extracted. Studies could, for example, not mention it explicitly or use another term for the same concept. Relying on just those specific keywords alone will not suffice. Therefore, a definition of horizontally and vertically partitioned (or sometimes called: distributed) data in the context of Federated Learning is given. This definition is synthesized by the definitions of Yang et al (2019) and Gong et al (2016).

Definition *horizontally partitioned data*: [Definition 5.1, Appendix E]
Horizontally partitioned data means that the data at each data site have the same features, i.e. attributes or columns in traditional data base terms, but include different subjects, i.e. rows. For example, imagine a software company who sells software to small businesses to conduct their administration digitally. Each subject (i.e. small business) uses exactly the same type of software and generated the same types of data, i.e. the columns of the domain model are the same.

Definition *vertically partitioned data*: [Definition 5.2, Appendix E]
Vertically partitioned data constitutes the opposite: the data of one subject, i.e. row, is present at multiple data sites. Each data site, therefore, has a different set of features, i.e. columns or attributes. For example, in the case of hospitals, where a patient's health records are scattered across many hospitals. One hospital has data about his blood work, while another specialized hospital only stores the results of a lung scan. The different hospitals store data about the same subject (the patient), but have different data features or columns of that patient.

3.3.3 Differentiating Characteristic 2: Underlying Machine Learning Model

This section discusses the second differentiating characteristic of Federated Learning: the underlying machine learning model. First, the reasoning as to why privacy is a differentiating characteristic is shortly revisited. Second, the methodology of information extraction is described. Third, an inventorization and categorization of underlying machine learning models in Federated Learning is given. Lastly, in the next section the results of the systematic literature study are shown.

The underlying machine learning models of each Federated Learning method and their inherent supported machine learning problem types will limit the options organizations have. Therefore, it is relevant as a differentiating characteristic.

In order to extract information regarding this differentiating characteristic, a list of all options regarding underlying machine learning models and problems is constructed. This is done by both citing what the literature lists as option, and by doing categorization by means of listing all types of machine learning models and problem (types) found in this study's systematic literature review.

Verma et al (2019) list the following machine learning models in Federated Learning: "among these models are: decision trees, clustering, rule-engines, Gaussian Mixture Models, SVM [Support Vector Machines], NN [Neural Networks]". In addition, by looking at the studies found in this study's systematic literature review the following models were also encountered: linear models and Bayesian Network models. From this the following lists of options for the data extraction form is synthesized. These options are added to the template of the data extraction form, see Appendix A - Extraction Form 1. The machine learning problem types are synthesized by listing and categorizing all encountered problem types in the studies found in this SLR.

Underlying Machine Learning model list: Linear model, Bayesian network model, Decision Tree, Clustering model, Rule-engines, Gaussian mixture models, Support Vector Machine (SVM), Neural Network (NN). [*Definition 4.1, Appendix E*]
Machine learning problem type list: Linear/Regression problem, Classification problem, Rule-learning problem, Clustering problem (unsupervised), Language modeling problem. [*Definition 4.2, Appendix E*]

These options will be used to categorize the found underlying machine learning model and problem for each Federated Learning method in the data extraction form.

3.3.4 Differentiating Characteristic 3: Privacy Guarantee

This section discusses the third differentiating characteristic of Federated Learning: the privacy guarantee. First, the reasoning as to why privacy is a differentiating characteristic is shortly revisited. Second, the methodology of information extraction for this privacy guarantee is described. Third, a categorization of privacy guarantee levels in Federated Learning is given. Lastly, in the next section the results of the systematic literature study are shown.

The privacy guarantee is a differentiating characteristic, as there is variation in the levels of privacy guarantees among Federated Learning methods and the choice of an organization is impacted by this. For example, an organization's distributed data set could be privacy sensitive in such a way that no data is allowed to leave its original location. In this way it limits the choices this organization has regarding Federated Learning methods.

In this section the privacy guarantees of Federated Learning are investigated. This in done in the following way. For each of the aforementioned Federated Learning methods and the studies they are introduced in, claims and descriptions about privacy of that method are extracted. After all information is extracted, the results are synthesized and common themes are deduced. Then, broad privacy guarantee categories are chosen to categorize each Federated Learning method. This categorization will be based on practicality and relevancy to the overall goal of the study: to fit an organization's needs relating to the privacy considerations of their data.

This study categorizes three broad levels of privacy guarantees in Federated Learning (in order):

1. *Violates the no data sharing principle*
   In this category the Federated Learning methods give no guarantee whatsoever about the no data sharing pillar of Federated Learning. Although Federated Learning is founded on the idea of no data sharing, in this category, each local data site does not have control over their data. The data of each local data site can be shared with other data sites or the central server.
2. *Privacy by no data sharing*
   This category can be seen as a standard of Federated Learning. Federated Learning is founded by the principle of local data sites having ownership over their own data. In this category privacy is upheld by not allowing the data itself leave each local data site. Instead, only aggregates in the form of partial model updates (i.e. parameter updates during model training) are shared with a central server.
3. *Additional privacy mechanism*
   In addition to privacy by no data-sharing, some studies indicate that this privacy guarantee level is still not enough. In fact, they claim that even the aggregate data, i.e. the parameter updates that are communicated with the central server, are private information. In addition, when the central server of other local data sites cannot be trusted, additional privacy mechanisms are also of importance. Both claimed by Gong et al (2016) and Jochems et al (2016). The additional privacy mechanisms mentioned in the studies used in this systematic literature review are: Anonymization, Differential Privacy, Secure Multi-Party Computation, Homomorphic Encryption. Each of these are described later in this section.

These levels can also be found in [*Definition 5, Appendix E*].

These three privacy guarantee levels are categorized in this way because of the following. The second privacy guarantee can be seen as a standard for Federated Learning. As seen in the definition research question, one of the main pillars of Federated Learning is privacy. Privacy is by standard guaranteed in a 'Federated' manner; where each of the local data sites has ownership over their own data, the raw data will not be permitted to leave the original data site. Therefore, this will be the standard category. However, two deviation are made from this standard, both in a decreased and heightened privacy guarantee level.

First, some Federated Learning methods try to up the predictive performance, especially in non-iid settings, by sharing data among data sites (e.g. Zhao et al’s (2018) method). This violates the no data sharing principle of Federated Learning and these method should be categorized differently. See category 1, violates no data sharing principle.

Second, some studies argue that privacy by not sharing the raw data with others may not even be enough in some cases. That, in fact, additional privacy measures should be implemented. In Federated Learning methods, although no raw data is shared, local models updates are shared, which are comprised of aggregates of this data, where a local minimum is to be found. If it is the case that not only the primary data is privacy sensitive, but also aggregates, then the no data sharing principle is not enough. For example, one could imagine a data set of a company’s clients. This stores a client’s file: its name, category, revenue. From this an aggregate could be the number of clients this data set has, or the total revenue generated, which could in fact be privacy sensitive and could be shared to a central server in the form of parameter updates. Gong et al (2016) has the same concerns, in the healthcare domain, and mentions that "local regression parameters are trained on individual private data, and may leak sensitive information about patients". Jochems et al (2016) confirms this by saying that the central server should be a trusted party, otherwise the privacy guarantee does not always hold. Also, Yang et al (2019) address this indirect information leakage, and argue that the intermediate results, like parameter updates, which are shared and communicated, provide no guarantee that sensitive information is protected. In addition, Gong et al (2016) mentions collusion attacks, where multiple data sites may collude together to infer previously communicated private data about patients. So, this indicates that just relying on the no data sharing principle alone may not be enough in all cases. Additional privacy mechanisms could be used to mitigate this.

There are several additional privacy mechanisms mentioned in the studies of Gong et al (2016), Liu et al (2019), Yang et al (2019), Jalalirad et al (2019), and Shaoxiong et al (2019), which can mitigate such privacy concerns. The additional privacy mechanisms mentioned in the studies used in this systematic literature review are: Anonymization, Differential Privacy, Secure Multi-Party Computation, Homomorphic Encryption. Each of these are described, in short, next.

Anonymization
Described by Gong er al (2016) as a popular way to preserve privacy. It anonymizes the data by hiding the identity of the data source. It can be classified as an easy implementation. However, concerns. are also raised, as it is possible to re-identify the data source in some cases.

Differential Privacy
The main approach of this technique is to add noise to the data during model computation (Gong et al, 2016) (Yang et al, 2019). In this way, a higher level of privacy is reached where the aforementioned problems are addressed. On the flip side, it is said by Gong er al (2016) and Yang et al (2019) to impact the predictive performance of models in a negative way, albeit relatively small. So, it does come with a cost, and therefore this trade-off between predictive performance and privacy should be taken into account.

Secure Multi-Party Computation
Gong et al (2016) describes Secure Multi-Party Computation as: "Secure multi-party computation-based approach is a conventional approach to training classifiers based on private data owned by multiple parties. A combination of cryptographic techniques is used to compute a function of their private data. This approach usually guarantees that none parties can learn anything beyond what is contained in the final result.". But also states that the algorithm is not efficient and computation costs will be high.

Homomorphic Encryption
Yang et al (2019) describes it as follows: "Homomorphic Encryption is also adopted to protect user data privacy through parameter exchange under the encryption mechanism during machine learning". It is also said to be a stronger form of privacy preservation as the data cannot be guessed like in the case of differential privacy. However, Gong er al (2016) mention that homomorphic encryption is not efficient, especially when the training data set size increases.

Summarized, additional privacy mechanisms are all founded upon the assumptions that the central server and/or the other local data sites cannot be trusted or that aggregate data, contained in model updates to the central server may also be of sensitive nature. In those cases, an organization should attempt to choose a Federated Learning method which supports an appropriate additional privacy mechanism. An organization should look at these categorizations, reason what the relation is to their data sets and privacy requirements, and ask what privacy type is appropriate for their use case.

3.3.5 Results

This research question can be summarized by Table 3.3.1, found at the end of this section. It lists, for all Federated Learning methods, their first three differentiating characteristics: (i) data partitioning, whether it supports horizontally partitioned data (HPD) and/or vertically partitioned data (VPD), (ii) underlying machine learning model and inherent machine learning problem type, and (iii) privacy guarantee. The extracted results are presented in Table 3.3.1, the results, where relevant, are described in more detail below.

The results for data partitioning are extracted by either looking for the exact terms HPD or VPD, or, if those terms are not mentioned, by identifying the concept they represent. This is done by identifying characteristics in the data model descriptions or looking at how the Federated Learning algorithm works, and linking them to working with either or both horizontal or vertical partitioned data. The results regarding the underlying machine learning model were more straightforward. Each study did mention it explicitly. The results could be easily extracted by making used of the earlier constructed categorization. Therefore, the results are only stated in Table 3.3.1. Regarding the privacy guarantee level, each of the studies' Federated Learning methods are evaluated to have one of the three identified privacy categorizations: (1) *Violates no data sharing principle, (2) Privacy by no data sharing, (3) Additional privacy mechanism*. First, data partitioning results are described and the privacy guarantee results are next.

One of the first Federated Learning methods of Allende-Cid et al (2013) works only with HPD. It is, however, not explicitly mentioned, as the definition of HPD and VPD were not introduced yet. This information is, thus, deduced by looking at the domain models for the distributed data sites; which have exactly the same structure, i.e. columns, indicating HPD.

Next are the studies related to Federated Learning in healthcare. Gong et al (2016) and Jochems et al (2016) mention explicitly what partitioning their Federated Learning methods support. Both HPD and VPD for Gong's method, and only HPD for Jochem's method. The other healthcare studies do not mention it explicitly, but all only support HPD. This fact is deduced in the case of Deist et al. (2017) by its data description: all hospitals supply the same variables as data input, i.e. all have the same data attributes. The same goes for their 2020 study, which mentions that the data sites (the hospitals) have to agree upon a common data model. Lastly, Brisimi et al (2018) also only support HPD, which is deduced from the fact that the patients' (the subjects') data is not fragmented across data sites. So, one subjects' data is only present at one data site.

The popular FedAvg method of McMahan et al (2017) supports only HPD. While this paper was published after other studies already explicitly defined HPD and VPD, it does not give an explicit description of which data partitioning type it supports. Deducing from the algorithm description and the performance results evaluation (which mentions the data used) it becomes clear that it only supports HPD. Because Zhao et al (2018) and Duan (2019) base their method on FedAvg and use similar evaluation, they can also be categorized to only support HPD.

Federated Stochastic Variance Reduced Gradient (FSVRG) also only supports HPD. This is deduced from the algorithm description and especially based on the characteristics of the used data set (MNIST) for evaluation, which is standardly HPD. The same goes from the CO-OP method, based on the study of Nilsson et al (2018).

The Restrictive Federated Model Selection (RFMS) by Sun et al (2019) also only supports HPD. Deduced from the used data set, the GEO database. This database features breast cancer data. The researchers filtered out duplicate patients across artificially created distinct data sites, therefore they did not support VPD. The same reasoning can be used for the Federated recommender system by Jalalirad et al (2019). Deduced from the evaluation experiment description. They used the movieLens100k dataset, which contains movie ratings from users. The study partitioned the data per user, so this indicates HPD.

The Attentive Federated Aggregation algorithm (FedAtt) of Shaoxiong et al (2019) did not mention which type of data partitioning they support. It could also not be easily deduced from the data description or the algorithm description. An estimated guess would be HPD, as it is the most common and apparently more easy to implement. Also it compares it results directly to FedAvg who only supports HPD.

The CBFL method of Huang et al (2019) only HPD. Which is deduced from data description and the partitioning description. The data sites contained the same features, indicating HPD. Lastly Verma et al's (2019) method also only support HPD. Also deduced from description on how the data is partitioned.

3.3.6 Conclusions and discussion

Differentiating characteristics

In this research question the term differentiating characteristics for Federated Learning methods is defined. Differentiating characteristics both (i) are relevant to the to-be-designed method, i.e. those facets which may limit options or impact the desired outcome regarding a organization's data-related characteristics and privacy considerations, and (ii) have variation in implementation among the Federated Learning methods. These differentiating characteristics will define the relevant differences between Federated Learning methods, and contribute to the overall research objective of designing a method to make an informed choice among these Federated Learning method.

The resulting list of differentiating characteristics of Federated Learning methods is constructed:

1. Data partitioning, i.e. system heterogeneity;
2. Underlying machine learning models;
3. Privacy guarantees;
4. Non-iid data support, i.e. statistical heterogeneity;

5. Performance (accuracy, predictive performance).

Of this list, the first three differentiating characteristics (1-3) are addressed in this research question. The remainder are addressed in separate research questions, as they require more knowledge and are larger.

Data Partitioning

A distinction between two data partitioning types is made: horizontally partitioned data (HPD) and vertical partitioned data (VPD). By looking at the results, the following can be concluded. Not all Federated Learning methods support both (horizontal and vertical) data partitioning methods, confirming the need to identify what data partitioning type a specific situation has. Moreover, VPD support is very rare, and is only observed twice. Which means that the options for organizations with VPD have limited options. Future research should investigate whether there is a need for more VPD support in Federated Learning, and if so, it is recommended that more Federated Learning methods be developed which support VPD. Consequently, the overwhelming majority supports HPD.

Next, the relation to a potential method fragment is discussed. Given the fact that a organization's data landscape is static, i.e. an organization cannot easily change the current partitioning type, especially not if data sharing is limited or prohibited, identifying what type of data partitioning is present is crucial in the development of the artifact of this study. As options to what Federated Learning method is then available will be limited. Because it limits options and makes the organization aware of their specific landscape, it contributes to the overall goal.

A method fragment which main goal is to identify the data partitioning type is therefore recommended. This identification will be based upon the synthesized definitions of HPD and VPD, and the method of extracting the information (by either explicitly defined terms of by looking for characteristics similar to either definition) in this research question.

Underlying Machine Learning Models

Each federated learning method has an underlying machine learning model. The following common themes can be extracted from the results. First, contrary to the Federated Learning methods after McMahan, the proto-Federated Learning methods use a wide variety of machine learning models, such as: linear models, support vector machines, and bayesian network models. This could be because in this stage Federated Learning was still in its infancy and many different approaches were tested, as no common standard was available yet. Also, more than half of the found Federated Learning methods are concerned with classification. This could be explained because these studies are concerned with healthcare, where classification of diseases is prevalent. Second, the Federated Learning methods after McMahan are predominately based on neural networks. Mainly because many methods are based on FedAvg of McMahan, which is itself based on a neural network model.

Next, regarding the overall research goal of this study, the following can be said. The used underlying machine learning models and their inherent supported machine learning problem types will limit the options organizations have regarding the usage of available Federated Learning methods. Therefore, it is relevant to design a method fragment which categorizes the desired solution to one of these categories.

However, this should not be seen as an inherently static characteristic of an organization's specific situation. Because an organization could be in the stage of discovery where it is still exploring its options regarding Federated Learning and has no predetermined goal. Then, a bottom-up approach could be preferred: looking at the available data landscape and from there determine what is possible and feasible. If, however, the specific machine learning objective is already set (or if the data does not support a specific problem), then it should be seen as a static characteristic. Which will limit the organization's options.

Privacy Guarantees

In this study three privacy guarantee categorizations are distinguished: (1) *Violates no data sharing principle, (2) Privacy by no data sharing, (3) Additional privacy mechanism*. Regarding the third type, several additional privacy mechanisms are used in Federated Learning, namely: Anonymization, Differential Privacy, Secure Multi-party Computation, and Homomorphic

Encryption. Each of the Federated Learning methods identified are categorized in their respective privacy guarantee level. From this, common themes are extracted.

The vast majority of Federated Learning methods support the privacy by no data sharing principle. This is not surprising, as it can be seen as the default privacy level upon which Federated Learning was founded. Only a few deviations are spotted, both to a lower and higher privacy level. The privacy level was lowered, by violating the no data sharing principle in studies that were concerned with low accuracies for non-iid data sets. To improve the predictive performance, data sharing among data sites was introduced. On the other hand, some studies argued that privacy by no data sharing was not even enough, that, in fact, even the communicated model/parameter updates to the central server, which are aggregates, could contain sensitive information. A few Federated Learning methods, therefore, support additional privacy mechanisms, such as homomorphic encryption and differential privacy.

These privacy levels are of importance to the main goal of the research. This is because it allows organizations to categorize their needs regarding the privacy requirements of their data sets in the context of Federated Learning. This organization can then, by means of the extracted and summarized information in Table 3.3.1, choose an appropriate Federated Learning methods that suits their needs.

Table 3.3.1 - Differentiating Characteristics (1-3) of Federated Learning Methods

FL Method	Data partitioning	Type of ML model & problem	Privacy Guarantee
Allende-Cid et al's (2013) method	HPD	Linear model. Linear/regression problem. (Regression)	Privacy by no data sharing
Gong et al's (2016) method	HPD & VPD	Linear model. Classification problem. (Logistic regression)	Additional privacy mechanism (homomorphic encryption)
Deist et al's (2017) method	HPD	SVM. Classification problem.	Privacy by no data sharing
Deist et al's (2020) method	HPD	Linear model. Classification problem. (Logistic regression)	Privacy by no data sharing
Jochems et al's (2016) method	HPD	Bayesian network model. Linear/regression problem.	Privacy by no data sharing
Brisimi et al's (2018) method	HPD	SVM. (Binary) classification problem.	Privacy by no data sharing
Federated Averaging (FedAvg) - McMahan et al (2017)	HPD	Neural Network. Classification & linear/regression problems.	Privacy by no data sharing
Federated Stochastic Block Coordinate Descent (FedBCD) - Liu et al (2019)	VPD	Neural Network. Classification & linear/prediction problems.	Privacy by no data sharing
Federated Stochastic Variance Reduced Gradient (FSVRG) - Nilsson et al (2018)	HPD	Based on FedAvg. So NN. Classification & linear/regression problems.	Privacy by no data sharing
CO-OP - Nilsson et al (2018)	HPD	Based on FedAvg. So NN. Classification & linear/regression problems.	Privacy by no data sharing
Restrictive Federated Model Selection (RFMS) - Sun et al (2019)	HPD	Bayesian model. (Binary) classification problem.	Privacy by no data sharing
Federated recommender system - Jalalirad et al (2019)	HPD	Neural Network. Classification & linear/prediction problems. (Recommender system)	Privacy by no data sharing
Zhao et al's (2018) method	HPD	Neural Network. classification & linear/regression	Violates no data sharing principle
Astraea method - Duan (2019)	HPD	Neural Network. Classification & linear/regression	Privacy by no data sharing
Attentive Federated Aggregation algorithm (FedAtt) - Shaoxiong et al (2019)	Not mentioned. Estimated guess: HPD*	Neural Network. Language modeling problem.	Additional privacy mechanism (differential privacy)
CBFL - Huang et al (2019)	HPD	Clustering model. Classification problem. (Unsupervised learning)	Privacy by no data sharing
Verma et al's (2019) method	HPD	Neural Network. Classification problem.	Privacy by no data sharing

3.4 Research Question 4 - Predictive Performance Differences

What are the differences in predictive performance among these Federated Learning and local-only methods?

Google introduced Federated Learning as a formal definition through the paper of McMahan et al (2017) and proposed their FedAvg algorithm to be a new standard for performing Federated Learning. However, how well does this method actually perform in relation to the other (newly developed) Federated Learning methods, to centralized methods, and to local-only methods? To examine this, the predictive performance of Federated Learning methods will be discussed and compared in this section.

3.4.1 Federated Learning Comparisons

Allende-Cid et al (2013) state that their ensemble distributed method outperforms a local model in almost every case, as measured by the standard deviation from the actual weather forecast predictions. The data is stated to be identically distributed and from the same data distribution. Meaning that in this case more available data for the global model makes every local prediction perform better.

Gong et al (2016) states that their custom federated algorithm achieves the same accuracy as a centralized approach, while their algorithm preserves privacy by not sharing data among data sites. Thus it can be stated that in this case the same accuracy as a centralized approach can be achieved while preserving privacy. They also make a comparison with a local-only approach and show that the prediction accuracy is worse than with the globally trained model. However, this comparison simply states that a particular locally-trained model does not generalize well to other data sites, they do not compare a locally trained model to that respective local data site, which would be a more practical comparison.

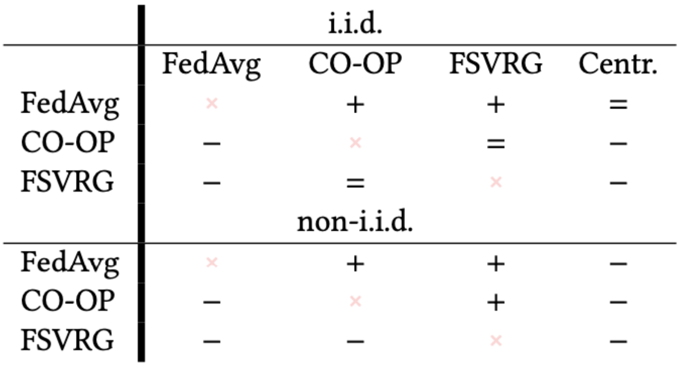
Brisimi et al (2018) develops a Federated Learning algorithm for the purpose of solving a binary classification problem, predicting hospitalizations for cardiac events. A comparison between a centralized approach is made based on the Area Under the Curve (AUC) accuracy, which states that similar accuracy is achieved. Confirming Gong et al (2016) and Deist et al's (2017) findings.

Deist et al (2017) developed their own federated learning algorithm, instead of using a version of FedAvg. The main contribution of this paper to this research question is the provision of a detailed comparison between the custom federated algorithm and a centralized approach (where all local data is transferred to a central server and a model is trained centrally). The results show that the federated and centralized methods both show very similar results (0.66 AUC), indicating that the federated approach does not impede predictive performance.

In the field of keyboard prediction, Federated Learning even achieves better performance than earlier used methods. Hard et al (2018) state that the baseline method of making keyboard type predictions (a word n-gram finite state transducer, which was up to this point the practical choice for doing keyboard type predictions) performs worse than the Federated Learning method, FedAvg in this case. This would indicate that, instead of performing similarly, as stated before, Federated Learning does indeed perform better. However, this claim is arbitrary, as it compares it to a very use case specific method, not to a standard centralized approach.

Nilsson et al (2018) compare three Federated Learning methods: FedAvg, Federated Stochastic Variance Reduced Gradient (FSVRG), and CO-OP. These methods were already identified and explained in research question 2. Of these methods FedAvg performed better than both FSVRG and CO-OP in both iid and non-iid settings. However, a centralized approach performed better in all cases, except for the comparison between FedAvg in a iid setting, there the centralized approach performed similar to FedAvg. Meaning that a centralized approach, based on only the predictive performance measure, is always preferred in a non-iid setting, and both FedAvg and a centralized approach are equally as good in a iid setting. These results are summarized by Nilsson et al (2018) in the Table 3.4.1. More about the effect of non-iid will be discussed next.

Table 3.4.1 - Summary of algorithm comparisons, showing if the algorithm in a row is better (+), worse (–), or practically equivalent (=) compared to the algorithm in a column. Nilsson et al (2018)



	i.i.d.			
	FedAvg	CO-OP	FSVRG	Centr.
FedAvg	×	+	+	=
CO-OP	–	×	=	–
FSVRG	–	=	×	–
		non-i.i.d.		
FedAvg	×	+	+	–
CO-OP	–	×	+	–
FSVRG	–	–	×	–

3.4.2 Non-iid and Imbalanced Data Set Considerations

A research branch within Federated Learning questions the statements made by McMahan et al (2017) that Federated Averaging (FedAvg) algorithm also performs well on non-iid and imbalanced data sets. A more in-depth analysis on this topic will be discussed later, in the next research question, as it is more suited there. For now, a brief overview of the numerical results will be discussed next.

Zhao et al (2018) show that non-iid data can reduce the accuracy of a federated neural network by up to 55%, contradicting McMahan et al's (2017) claim. Duan (2019) shows a more modest potential decrease in accuracy due to imbalanced training of 7.92% compared to FedAvg, but a decrease nonetheless. With his Astraea framework he managed to improve the accuracy by 5.59% on the imbalanced EMNIST data set and 5.89% on the imbalanced CINIC-10 data set. Therefore both claiming that predictive performance gains can be made with their proposed solutions.

Shaoxion et al (2019) makes a comparison between their Attentive Federated Aggregation (FedAtt) method and the already existing methods FedAvg and Federated Stochastic Gradient Descent (FedSGD) in a non-iid setting. The results show that their FedAtt algorithm either has similar predictive performance to FedAvg and FedSGD to in some cases better predictive performance. This means that FedAtt performs worse than the previously mentioned algorithms, who already perform better than FedAvg.

Jalalirad et al (2019) developed a custom Federate Learning recommender system, which reportedly outperformed earlier developed mainstream Federated Learning methods. However, the data set used was non-idd. The system was evaluated and compared to a centralized approach. They state that "[their] distributed algorithm does not reach the lowest error rates reported by centralized algorithms on the same dataset". Which means that in cases where the data is non-iid and accuracy is of utmost importance, a centralized method is preferred over a Federated Learning method.

3.4.3 Discussion

Federated Learning's most common method, FedAvg, is stated by its creators to work well with both iid and non-iid data. Although this statement is made and tested by means of an evaluation on some artificial data sets, many papers dispute this and validate this finding by showing (potential) predictive performance losses when applied to some other data sets. The reason for this being that the data distributions of particular data sites are not the same. When adding more data to a global model from different distribution does not necessarily produce better models for all. This could also be because almost all Federated Learning methods consolidate this data in a naive manner, by simply giving all data points from all data sites the same weight in the model. The fact that there are multiple empirically-backed papers that dispute the claims made for FedAvg working well on non-iid data is, could mean that this claim for FedAvg is unfounded.

Federated Learning methods adapted specially for non-iid settings exist, which claim to perform better than FedAvg. Those are therefore preferred over FedAvg in non-iid settings. The methods proposed by Zhao et al (2018) still include some data sharing. Although they claim only a small percentage of data need to be shared, it still circumvents a main pillar of Federated Learning: the strict non-data sharing privacy aspect. It is therefore of little value if privacy-preservation is important. Zhao's method does claim a much higher improvement in accuracy (of up to 55% compared to FedAvg) than Duan's (2019) Astraea method (of about 6% improvement) in non-idd settings. However, the Astraea method does not need data sharing, unlike Zhao's method.

Next, it has been shown that centralized training performs better than FedAvg in non-idd settings. A comparison between a centralized approach and the aforementioned (non-iid) adapted methods is, unfortunately, not made, and no clear recommendation between the two can be made. However, in non-iid settings the data can be highly skewed at some data sites. A centralized approach provides no balancing mechanism like the non-iid adapted federated learning methods provide. A calculated assumption can me made that the non-idd federated learning methods perform better. This should, however, be investigated in the future research to confirm this.

Lastly, it has become apparent that many studies both develop a new Federated Learning method, and then perform an evaluation and comparison with another Federated Learning method. It is always claimed to be better than the compared method. Most of these comparisons are made by predominantly 'primary studies'; studies which both first introduce the novel Federated Learning method and then also evaluate and compare it themselves. These primary studies usually use a non-standard data set in evaluating their newly developed method. This, of course, could lead to potential bias, as the method is developed in order to attain the best results on that particular data set, and not on other standard data sets. For a proper comparison to investigate which Federated Learning method is truly the best for a given context, more external comparisons should be made, preferably on the same data set(s). Only a few studies found in this literature review did that, so it is difficult to state the best performing Federated Learning method with certainty.

A good example of such an external study is that of Nilsson et al (2018), which compares multiple Federated Learning methods, and a centralized method. This research is more independent; it did not develop the compared algorithms itself, but rather evaluates others' algorithms. Therefore this research shows less bias than papers who are presenting and evaluating their own algorithm, which is highly dependent on the data set used. It is, therefore, a good source to evaluate the performance of these Federated Learning methods. Nilsson states that FedAvg is the best performing Federated Learning algorithm compared to FSVRG and CO-OP. However, for non-iid settings, a centralized approach performs even better than FedAvg, for non-iid settings FedAvg and a centralized approach perform similarly.

3.4.4 Conclusion

Concluding, a common theme is that most Federated Learning methods (custom ones and FedAvg) achieve similar predictive performance results compared to a centralized approach. Meaning that the privacy-preserving mechanisms of non-data sharing in Federated Learning do not significantly impede model accuracy. Also, multiple researchers claim that their Federated Learning method outperforms a local-only approach. Lastly, among the Federated Learning methods, the best known results are that of the FedAvg method, at least, in an iid data context.

As the last caveat made clear, a clear distinction should be made between non-iid and iid data sets in comparing the predictive performance. As many studies evaluate their Federated Learning method on an iid data set, the comparisons are not transferable to all real-world settings. There are specialized Federated Learning methods which improve model accuracy in non-iid settings. Of these methods Zhao et al's (2018) method shows the greatest improvement in accuracy of up to 55%, but does, however, require data sharing. The Astraea method does not require this, but only shows a modest increase in accuracy of about 6%. A more detailed investigation of the effect of non-iid data on Federated Learning is included in the next research question, in Section 3.5.

Therefore, it is dependent on the specific scenario, as the best performing methods differ. One of these scenarios is whether it involves iid or non-iid data. For iid data, the FedAvg method is the best choice. For non-iid data it depends on whether data sharing is accepted, to what extend, how much data and how often. If it, for example, can share data then Zhao et al's (2018) method is the preferred choice in non-iid settings, as it shows much higher accuracy gains than the Astraea method which does involve data sharing. As the latter is still not clearly explained enough, the effect of non-iid data on predictive performance in Federated Learning models will be investigated in the next research question in more detail.

These findings are briefly summarized in Table 3.4.2.

Table 3.4.2 - Best Performing Federated Learning Methods per Situation

Method	Situation
FedAvg, McMahan (2017)	Best-performing method in an iid data context.
Zhao et al's (2018) method	Best-performing method in a non-iid data context (55% increase in accuracy compared to baseline). Requires data sharing between data sites.
Astraea method, Duan (2019)	Adapted method for non-idd data context. Shows a 6% increase in accuracy compared to baseline. Does not require data sharing.

3.5 Research Question 5 - Predictive Performance and Non-iid Data

What is the effect on predictive performance effect of utilizing multiple data sites in Federated Learning by the means of consolidating this data?

3.5.1 Introduction

This research question is answered by, again, the results of the systematic literature review, and can be viewed as the main contribution of this study in a theoretical sense. For each paper deemed relevant to this research question the relevant information of that paper was extracted into the aforementioned extraction form. This was done by either looking at how the researchers described the federated optimization algorithms workings, or by looking at the algorithm itself if it was provided. In this section the extracted data is listed, analyzed, and conclusions are made from these findings.

To understand this research question better, the problem formulation of Li and Smith (2019) is revisited. The goal of federated learning is to minimize the following objective function:

$$\min_{w} F(w)\,, \text{ where } F(w) := \sum_{k=1}^{m} p_k F_k(w)$$

where *m* is the total number of devices, *w* is the input parameter (i.e. input training data), F_k is the local objective function, and p_k specifies the relative impact of each device. This relative impact is usually set as: $p_k = 1/n$ or $p_k = n_k/n$, where *n* is the total number of training examples, and n_k is the number of training examples of a particular local device *k* (Li and Smith, 2019).

This objective function, and especially the relative impact term, is the main topic of this research question. As this is the technique how federated learning consolidates local information into a global context. Where the relative impact term *1/n* gives each local data site the same importance, independent of the number of training examples provided, and n_k/n gives each local data site an importance based on the number of training examples provided. Both relative impact terms can be viewed as taking a (simple) average. The hypothesis of this research question is that this simple averaging may not be the most effective method for every federated learning use case in terms of maximizing the predictive performance.

This section is divided as follows. First, the proto-Federated Learning algorithms will be discussed. Next, the Federated Learning method from the line of McMahan will be examined. As most researchers develop their own Federated Learning algorithm, instead of evaluating a new one, the examination of each algorithm will be mainly on a paper-to-paper basis. After the assessment of the Federated Learning algorithms' mechanism to consolidate local data into a global context, an extensive discussion is presented, which is the main contribution of this research question.

3.5.2 Federated Learning Methods' Approach to Consolidation

Federated Learning deviates from traditional machine learning, as been made clear in the previous research questions, in that it is a form of a distributed approach. A central server learns a global model based on the partial model updates each local data site provides to it. The mechanism with which these model updates are consolidated into the global model potentially has a large influence on the resulting model and its predictive performance. Questions like how Federated Learning deals with particular data sites with much more data than others, or how it deals with data sites who have a different data distribution are particularly interesting. How exactly these model updates are consolidated in the global model is discussed in this section, starting with the proto-Federated Learning methods.

Allende-Cid et al's (2013) distributed algorithm consolidates the weather prediction data by simply aggregating the training examples with equal weights. A local data site's contribution to the global model is thus proportional to the number of training examples it has, data sites with more data examples have, therefore, a higher impact on the global model. The data is, however, iid and has the same data distribution over all data sites.

All other proto-Federated Learning algorithms, except Deist et al's (2020) method, also use simple aggregation, proportional to the number of data points contributed to the global model, as a means of consolidating data to the global model. Jochems et al's (2016) custom distributed Bayesian network model performs simple averaging, as does Gong et al's (2016) custom distributed algorithm, and Brisimi et al's (2018) does the same. Deist et al's (2017) first developed an adapted SVM for distributed use, by using ADMM, as mentioned earlier. This method does also performs simple averaging as a means of consolidation, however show that the performance per data site has a high variation. In a later paper, Deist et al (2020) mention data skew as a potential problem in their approach and mention calibration as a means to alleviate this problem. However, they do not share the workings of the used calibration method, apart from that it is manual work.

Next, FedAvg is discussed, the mainstream method presented by McMahan et al (2017). Both from the papers of McMahan et al (2019) and Nilsson et al (2018) it is clear that FedAvg consolidates data by taking an average, as can be deduced from the provided algorithms. In McMahan's paper n_k, the number of training examples added by a local batch, is divided by n, the total number of training examples already added to the central server, to determine the (weighted) contribution of this local batch to the global model. This creates a ratio exactly proportional to the number of training examples added per local client, a local client with many training examples has therefore a higher influence on the model than local clients with few training examples.

Next is FSVRG. The aim of FSVRG is being primarily concerned with sparse data, meaning that it concerns itself with features that are poorly presented in the dataset. (Nilsson et al, 2019) However, looking at the algorithm (algorithm 2, line 12), the update to the global model just simple averages out the new training examples to the global model like in the case of FedAvg. No weighting or balancing is applied.

CO-OP is one of the only Federated Learning methods that does appear to apply balancing, instead of just taking the average of all provided data samples. However, this is due to the algorithm's nature of being asynchronous. It appears that the balancing mechanism is only there on a technical level, not tot balance the data set itself. So on a fundamental level, this algorithm also just averages the training examples provided by each local data site.

3.5.3 Questioning the Effectiveness of Federated Learning on Non-iid and Imbalanced Data Sets

Given these results, the majority of the Federated Learning methods consolidate the local context into a global one by taking the average of all locally added data examples. Does, however, simply taking the average produce the best results in real-world settings, where data can be messy, unevenly distributed, non-iid, or highly skewed? McMahan et al (2017) state with confidence that two of the fundamental properties of Federated Optimization are that it works in non-iid contexts and on unbalanced data. A clear empirical demonstration of this is, however, not given at this point. In many cases, an artificially distributed dataset is generated instead of a real-world one for evaluation. For example, McMahan (2017) and Nilsson (2018) use the MNIST dataset to evaluate the effectiveness of Federated Learning, instead of a real-world setting. Other papers that are mentioned in the previous research question are adaptations of McMahan's algorithm also make this (implicit) assumption that Federated Learning works well with non-identically distributed or heavily skewed data.

It turns out that there are researchers questioning and even contradicting these claims. One of the more fundamental and, in this study, oldest statements is made by Allende-Cid et al (2013), who state: "the general model is built under the assumption that there is a global context, and that [another assumption, this general] model is valid in every one of the distributed sources". Indicating that there may exist cases where there is no one overarching general model, and that the general model is not always applicable to all local contexts. Although this paper predates the formal definition of Federated Learning in 2017, the assumption is rarely discussed or even mentioned in later papers. Some more recent papers, published after 2017, revisit this assumption and contest the claim made by McMahan (2017) that Federated Learning is always suitable in a non-iid and unbalanced data context, and will be discussed next.

3.5.4 Data Skew Problem Already Noted by Proto-Federated Learning Papers

Deist et al (2017) predates the mainstream FedAvg algorithm, and therefore developed their own federated learning algorithm instead of an adapted version of the FedAvg algorithm. The main contribution of this paper to this research question is the provision of a detailed comparison between the custom federated algorithm and a centralized approach (where all local data is transferred to a central server and a model is trained centrally). The results show that the federated and centralized methods both show very similar results (0.66 AUC), indicating that the federated approach does not impede predictive performance. However, the authors also gave a breakdown of the validation AUC performance by excluding one data site (in this case hospitals) at a time. By doing this, the variation in the AUC performance became substantial, giving a spread of 0.57 AUC for one case to as high as 0.77 AUC for another. Such a high variability by only excluding one data site shows the sensitivity of learning a global model (either centrally or federally). This reflects back on the assumption discussed earlier by Allende-Cid et al (2013). If the results vary this much by only removing or adding one data site, it might be that some local data sites do not gain benefit from a global model, because their data distribution differs too much from this global model.

In a later study of Deist et al (2020), they revisit these considerations and conduct another study based on their earlier developed custom federated learning algorithm. The results in this study show similar results. The spread between predictive performance results between different local data sites is again high: with 0.61 AUC to 0.85 AUC for different data sites gives a spread of 0.24. In addition, as can be seen in Table 6 of Deist et al (2020) the variation in data points is large: 706 for one data site up to 16.260 for another. One can imagine that the former data site has less influence on the global model than the latter. If the data distributions of both data sites differ substantially, it may be the case that the former data site yields worse results than if they just trained a local model, given that simple aggregation is used like suggested by the FedAvg approach. However, as the number of data points are low for the former data site, it also has less impact on the the validation results, which are often an average of the whole federated system

Deist et al (2020) acknowledge this problem. They argue that data skew and bias based on combining data from all data sites is indeed a problem. They also show this by describing their data. The statistical differences between data sites are significant, and show this by providing insight into the data distributions of each data site. For example, they mention that. some data sites have an excess of a certain type of cancer, that can skew the results of that particular sites as an average is used in the federated approach. Deist et al mention calibration of each data site as a solution. The method they proposed is, however, manual work and not automatic, and should be performed each time data is added, diminishing the practically of this solution.

Jochems et al (2016), also from a healthcare perspective, see a similar problem in their results. They mention that the AUC performance is 0.67 across the line, there are, however, major differences in per-site local performance. They state that the model used in this study performs better in some hospitals (Eindhoven and Liege), but worse in others (Maastro, Aachen, Jessa). To which we may conclude that training a global model does not necessarily lead to better local performance. On top of that, if the local sites would have relatively low number of data points, their significance may be overshadowed by other data sites, as validation of the results are ultimately captured in an average.

To summarize, these papers not following the mainstream FedAvg approach provide in-depth results on predictive performance with a breakdown on a per-site local basis. Other papers usually only give an average of all sites together, and don't provide this detailed breakdown. Only when performing this in-depth breakdown are the large variations of predictive performance between local data sites visible. This is not merely due to Federated Learning, as Deist et al (2017) show that a centralized approach yields very similar results, and also contain this spread in variation between data sites.

3.5.5 Criticisms on Claims Made by FedAvg Advocates

Zhao et al (2018) provide a comprehensive and empirically backed paper on this topic of the impact of non-iid data sets in Federated Learning. Zhao et al show that non-iid data can reduce the accuracy of a federated neural network by up to 55%, directly contradicting the claim made by McMahan (2017), and therefore accuracy gains can be made. For this, they develop a new method, based on the FedAvg algorithm. The main difference is that initially, at the start of the FedAvg algorithm, instead of a randomly initialized model, a centrally trained initialization model is used, based on a shared data set.

The paper of Zhao et al (2018) demonstrates that there is a trade-off between the test accuracy and the size of the globally shared dataset (G), and also with the weight, i.e. relative importance, of G and the test accuracy of the model. It states that an increase in the importance of a globally shared dataset in the algorithm can improve accuracy results. A good trade-off point of only sharing 5% global data in order to increase accuracy by 30% is found, providing a large jump in accuracy for only sharing a relatively little amount of (potentially privacy-sensitive) data.

However, as a result of the introduction of a globally shared model, one of the pillars of federated learning is breached; data sharing between data sites is again introduced. Although this sharing of data is done manually and as a preliminary step even before the initialization of the algorithm, this approach may not be a suitable practical solution for many federated settings where privacy is of utmost importance. Nonetheless, it does show, on a more fundamental level, that Federated Learning does not always provide the best results in a non-iid setting, directly contradicting McMahan's (2017) aforementioned claim.

Duan (2019) builds upon this insight of Zhao et al (2018) and develops a self-balancing framework, called Astraea, which among others, eliminates the need for manually creating a globally shared dataset. Duan shows, experimentally, by making use of an imbalanced subset from EMNIST dataset, a more modest potential decrease in accuracy due to imbalanced training of 7.92% compared to FedAvg, but a decrease nonetheless. With his Astraea framework he managed to improve the accuracy by 5.59% on the imbalanced EMNIST data set and 5.89% on the imbalanced CINIC-10 data set.

The Astraea framework is described by Duan (2019) as follows: "the Astraea framework counterweights the training of Federate Learning with imbalanced datasets by two strategies. First, before training the model, Astraea performs data augmentation to alleviate global imbalance. Second, Astraea proposes to use some mediators to reschedule the training of clients according to the KLD between the mediators and the uniform distribution. By combining the training of skewed clients, the mediators may be able to achieve a new partial equilibrium". KLD here means the Kullback Leibler divergence, a measure to test the difference between two data/probability distributions.

Duan (2019) essentially revisits the same assumption made by Allende-Cid et al (2013). Duan states that existing studies make the assumption that the global data distribution of a federated network is balanced, even though the volume of data on the devices may be disproportionate. In real-world scenarios, however, the global data distribution can be imbalanced.

On top of empirically demonstrating the accuracy loss by class imbalance on imbalanced data sets, Duan also gives mathematical proof that the imbalance of distributed training data can lead to a decrease in accuracy of Federated Learning.

To sum up, Zhao et al (2018) and Duan (2019) still use the simple averaging of FedAvg in the end. However, due to the preliminary (and intermediate) balancing steps it cannot be simply stated that they take an average as to how they consolidate the local context into a global model. On top of this, they preliminarily balance local clients by making use of a globally shared data set, and, in the case of Duan, take an intermediate step by making use of mediators to balance the local clients even more when necessary.

Federated Learning in non-iid settings: a visual explanation
A visually illustrated error of training a federated model with a skewed data set is demonstrated by Verma et al (2019), see Figure 3.5.1 below. The problem arises in a situation where different sites are estimating the function on only a (concentrated) portion of the data range, which can happen with skewedly distributed data. The estimated local function does not represent the ground truth function, and simply averaging the two out will result in an erroneous estimated function.
To combat this problem, Verma et al (2019) propose a new federated learning method which differs from traditional federated learning in two ways. The first is bounds-aware fusion, where the aggregation estimates the bounds of each local data site, trying to find overlap with other data sites, in order to estimate functions based on their aggregate bounds. The second is bounds expanding data exchange, where data is shared among local data sites, in order to expand their data range, mitigating the aforementioned problem of concentrated data ranges in particular data sites.

The mentioned custom federated model is, however, not documented in detail, and not empirically tested with a real-world data set. This questions the validity of the claims made about the effectiveness of their new federated learning method, and should be investigated in further research.

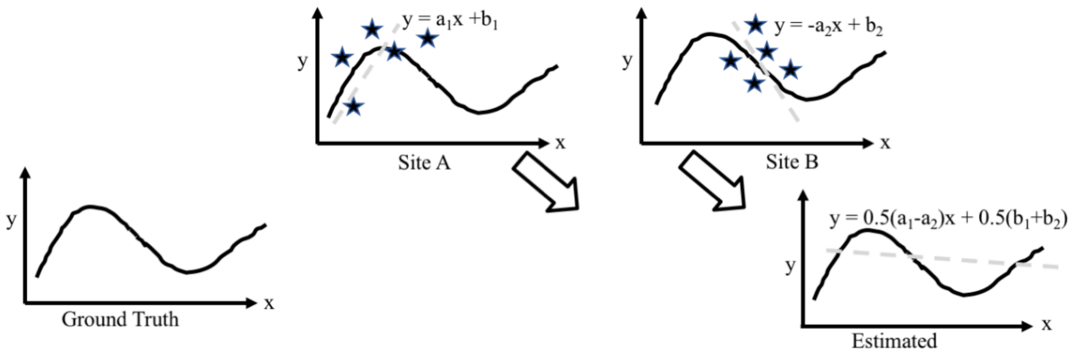


Figure 3.5.1 - Problem with function estimation in Federated Learning with data skew. By Verma et al. (2019)

A clustering approach
Another alternative approach to consolidating local data comes from the healthcare sector. As hospitals are particularly prone to non-identically distributed data, due to geographic placement

and medical specialization, Huang et al (2019) proposed a novel community-based federated learning (CBFL) algorithm. They acknowledge that Federated Learning may underperform when data is non-identically in-dependently distributed, which is especially the case in hospitals. To mitigate this they add a preliminary step which clusters hospital data into several similar communities (based on patient age, ethnicity, and more), and a separate model is trained on each cluster. (As opposed to earlier alternative methods, without data sharing.) The reasoning behind this is similar as discussed earlier in this section: it is in this context easier to train multiple models per cluster, than to learn one general model which over-fits or under-fits several local data sites. They evaluate their model and concluded that CBFL outperformed baseline Federated Learning, confirming their hypothesis of the underperforming of federated learning in non-iid settings.

Keyboard type prediction revisited and wrap-up
Shaoxiong et al (2019) also questions the assumption made by regular Federated Learning of aggregating all local client's data by simply averaging the results on the aggregation server. They state that, in the case of keyboard type prediction, as is also the case study in Hard et al's paper (2018), this assumption could lead to worse predictive performance, because the clients can have very subjective and specific behavior. They provide a solution to this by adapting the FedAvg algorithm, by making use of attentive weights. "The method minimizes the weighted distance between the server model and client models by iteratively updating parameters while attending to the distance between the server model and client models", as stated by Shaoxiong et al (2019).

Lastly, the two remaining authors in this study who question the approach of simple averaging are stated next. Wang et al (2019) propose a control algorithm that determines the best trade-off between local update and global parameter aggregation to minimize the loss function. Schmid et al (2019) question whether just consolidating data of various clients is actually improving the quality of the objective (i.e. classification or prediction). It states that the quality improvement is not a guarantee, and depends on the aggregation method used, but in general it does hold true that more data input is better for learning. In this case the authors suggested using ensemble methods, like bagging and boosting, in the context of federated learning.

To summarize, the alternative methods which take into account the data skew problem, still perform, in most of the cases, simple averaging of local data into the global model. However, this averaging is often preceded by a preliminary step of data sharing among the local clients, clustering of similar local data sites, or by some other balancing mechanism which may or may not involve data sharing.

3.5.6 Conclusion and Discussion

Concluding, almost all Federated Learning methods consolidate features by merely taking the average of each local data site's contribution to the global model in terms of data points (i.e. number of data points provided by a particular data site divided by the total number of data points in the global model). Studies mentioned in the earlier part of this section proclaim that these methods of consolidating local data also work well on non-iid and imbalanced data. However, a substantial number of papers, discussed later in this section, differ from this approach and mention that just taking an average is not be the right course of action for imbalanced data sets. Predictive performance can actually be improved if another balance between local and global context is set and should be taking into account when working with non-iid and imbalanced data sets.

On top of that, one should take the objective of performing machine learning - federated or not - in mind. As concluded earlier, based on the papers of Deist et al (2017, 2020) and Jochems et al (2016), the per-site predictive performance shows a large spread, both in the federated and centralized case. If a local data site (e.g. a hospital) where to embark on a journey of implementing machine learning in order to perform some prediction (e.g. on predicting cancer), they should seriously question whether a centralized or federated approach would be useful for them. While the assumption of federated learning is that more available data leads to a better model is true in many cases, it is not for a local data site whose data distribution does not match that of the global model, i.e. if there is data skew. In this case a locally trained model could be a better fit. To test this, an in-depth review of the results, and in particular a breakdown of the data distributions and the results (after training the model) on a per-site basis, should be performed.

Whether Federated Learning is the right approach is context dependent. First a separation should be made between iid and non-iid contexts. The majority of the findings state that Federated Learning, and especially FedAvg, has similar predictive performance compared to a centralized approach in iid settings, and better performance compared to a model trained on only local data, due to the increased data set size. Therefore, the advantages of using Federated Leaning are clear: it performs similarly to a centralized approach and due to its privacy-preserving mechanisms even has the potential of attaining data that were previously not accessible. For non-iid settings it is, however, a different story, as standard algorithms like FedAvg perform worse than a centralized, or even a local approach in some cases. For non-iid settings the choice is dependent on importance of privacy-concerns. A Federated Learning method adapted for non-iid usage like Zhao et al's (2018) method, the Astraea framework, or others mentioned in this section has preference over FedAvg. A centralized method is preferred over FedAvg if there are no privacy concerns and data is available readily. A comparison between a centralized approach and the non-iid Federated Learning methods has not been made yet, and should be investigated in the future. Also, a local-only approach should be considered in some cases, as discussed in the previous paragraph.

3.6 Research Question 6 - Non-iid Data Identification

RQ6:

What is an appropriate method for identifying non-iid data sets in the context of Federated Learning?

3.6.1 Introduction

The importance of this research question is clearly demonstrated in the previous research question. Where it became clear that non-iid data has a significant impact on the predictive performance of Federated Learning methods. When faced with non-iid data sets, Federated Learning methods which are tailored to working with non-iid data should be used as they yield better predictive performance overall. Therefore, it is important to find a method which can detect non-iid data in a Federated Learning context.

To answer this research question, first, a definition of non-iid data in the context of Federated Learning will be constructed, as it is yet to be clearly defined. Only when a clear definition is drawn, a method to identify non-iid data sets can be found. When explicitly defined, a method fragment which can identify whether the data sets are non-iid or iid can be constructed, which will contribute to the overall research goal of this study. Without a clear definition this is not possible.

The first part is answered by the following method. It is conduced by extracting already found papers in the SLR, and filtering them based on the inclusion criterium of 'having relevance to iid, non-iid data'. First based on the title and abstract (9 studies), then on full-text (4 studies). Also, this process is repeated by backwards citation search, which brings the total number of studies used to answer this research question to 7. From these filtered relevant studies, relevant information about non-iid data is extracted and synthesized by making use of a data extraction form (See Appendix B - Data extraction form 2), as well as searching for methods or references to methods which can identify non-iid data in a Federated Learning context. For the second part, a regular literature study is conducted. For this, references found in the studies about non-iid data in the already done SLR are added. In addition, the studies used to construct the non-iid data definition in Federated Learning are also used. If they describe or reference a method to determine non-iid data, it is also included.

3.6.3 Non-iid Definition in Federated Learning

In this section a definition of non-iid data sets in the context of Federated Learning will be constructed. The definition of non-iid data in the general sense is not applicable to a Federated Learning context because of its infrastructure and privacy-properties. Whereas in the general sense a data set can be assessed to be non-iid, in Federated Learning this is not the case. There is by definition not one (global) data set in Federated Learning; there are merely several separated data sets, which cannot be combined due to the main purpose of Federated Learning: privacy preservation. Combining the data sets to assess non-iidness would defeat the foundation of why Federated Learning was founded. Therefore, the existing techniques and methodologies cannot be used to assess non-iidness in the context of Federated Learning. Instead, it should be assessed incrementally on a per-data set basis, comparing each data set individually to the others, to hypothesize what a combined data set would entail. Next, to understand the basics, the definition of non-iid is briefly discussed from a general perspective first, then the definition is tailored to what is relevant to the context of Federated Learning.

In statistics, iid data is data where observations are independent and identically distributed. It is a very common assumption that is often made in the field of statistics and data science (Clauset and Aaron, 2011). Here, independence means that the probability of observing two values (x_1 and x_2) is the same as the probability of observing this one observation multiplied by the other
$P(x_1, x_2) = P(x_1) * P(x_2)$ (Clauset and Aaron, 2011). Basically, the covariance of the observations are, and should be by this rule, zero. In simpler terms, this means that observing the first observation will not influence the probability of the subsequent observation(s). Identically distributed, then, means that two observations are from the same probability distribution. Non-iid data is then the opposite, i.e. data sets that are not identically distributed and independent. This definition is, however, very general and applied in statistics. Next, the definition of non-iid data is reflected to the context of Federated Learning.

Similar as in statistics, the field of traditional machine learning is also, in theory, built upon the assumption of having random variables that are iid (Dagstuhl, 2015). This also confirms the findings in the previous research question, which states that many authors assume that their Federated Learning methods will be used only on iid data sets. Cao (2015) confirms Dagstuhl's statement by stating that: "most of the classic theoretical systems and tools in statistics, data mining and machine learning are built on the fundamental assumption of IIDness, which assumes the independence and identical distribution of underlying objects, attributes and/or values". Indicating that, as in the definition from Clauset and Aaron (2011), data is non-iid when two observations come from different data distributions.

Does, however, this assumption of iidness hold in Federated Learning? Dagstuhl (2015) is clear on this and states that the assumption can only be made in theory. In practice it is often violated, where training data is likely to to come from different distributions, because the used data is Federated Learning is likely to come from a heterogeneous set of devices. This means that the data distributions of different devices depends on their usages [in data generation], which are likely to be different (Duan, 2019). (Usage here references to data generation patterns, for example the usage of a mobile phone keyboard will influence the data generation pattern in terms of words typed and the frequency.) This means that in Federated Learning iidness of data set cannot be assumed, and in some use cases with heterogeneous devices is even likely to be non-iid as opposed to iid. This indicates that there is a need to identify non-iid data sets in the context of Federated Learning.

To identify non-iid data sets the term non-iid will be divided into smaller problem parts. Such a practical list is constructed by Duan (2019). Duan states that in Federated Learning non-iid data sets can be caused by:

1. Size Imbalance, where the data size on each device (or client) is uneven;
2. Local Imbalance, where each device does not follow a common data distribution;
3. Global Imbalance, means that the collection of data in all devices is class imbalanced.

This list is more specific to Federated Learning, and overlaps with Dagstuhl's (2015) earlier statement that non-iid data sets come from different data distributions (overlaps with point 2). Also, it is likely for point 1 to occur in Federated Learning. As just stated earlier, the heterogeneous nature of devices in Federated Learning is likely to cause different data generation patterns, due to the fact that the edge devices have different usage patterns. In more practical terms, some edge devices could generate much more data than others. Due to the fact that Federated Learning weights each data point equally at the consolidation, as investigated in the previous research question, a few devices may influence the model disproportionally. Lastly, Sun et al (2019) confirm both the local imbalance and the global imbalance as a problem in Federated Learning. Global imbalance, or class imbalance, occurs when features with categorial values are heavily skewed towards one class, such that one class has a large number of examples and the other only a few (Japkowicz & Stephen, 2002)

This list of criteria can be used to assess non-iidness in Federated Learning. It can be used to compare (sample) data sets that will be used in a Federated Learning system, in an incremental way. In this way the new data set is to be assessed and compared by the existing data sets to be similar in size, having no different data distribution, and contain no class imbalance.

To conclude, *non-iid* data sets in the context of Federated Learning are primarily concerned with differences in data distributions between the data sets of edge devices, as the individual data sets will form a theoretical consolidated data set in the end. As has been made clear, the assumption that data is iid cannot be made in Federated Learning. More practically, non-iid data sets can be caused in Federated Learning by adding new data sets which area assessed to be adversarial to the existing data sets on the basis of three factors: (1) size imbalance, (2) local imbalance, and (3) global imbalance. Now the definition of non-iid data in Federated Learning is clear, a method which can detect non-iid data sets in Federated Learning is introduced.

3.6.4 Identifying Non-iid Data Sets

In this section the method of Rabanser et al. (2018) is presented. It is a high-level methodology which guides in identifying non-iid data sets in the context of Federated Learning. The methodology is supplemented by the earlier drawn definition, some more technical and lower-level methods, and the three criteria by Duan (2019).

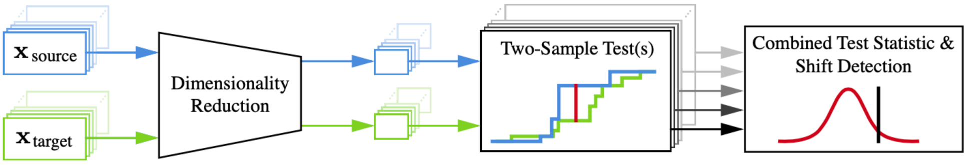

$\mathbf{x}_{\text{source}}$
$\mathbf{x}_{\text{target}}$
Dimensionality Reduction
Two-Sample Test(s)
Combined Test Statistic & Shift Detection

Figure 3.6.1 - Method for identifying distribution differences between two data sets. Rabanser et al. (2018)

The method of Rabanser et al. (2018) contains three steps: (1) Dimensionality Reduction, (2) a Two-Sample Test(s), and (3) Making an assessment and a conclusion from these insights. The method is visualized in Figure 3.6.1. The first step, dimensionality reduction, or feature selection, is used to filter out unnecessary features from the data set, i.e. features which are assessed to not contribute to predictive performance of the model. Feature selection is important to help in understanding the data, but also to increase the predictive performance of the prospective model, as stated by Chandrashekar and Sahin (2014). This is also done for practicality, as too many features would make the assessment time intensive.

The second step is performing a two-sample test. These tests result in a metric which determines the likelihood of the two data sets to be of the same or of a different distribution. This is what determines whether the data sets are non-iid or iid in the context of Federated Learning. Rabanser et al (2018) suggests several tests, such as: the Maximum Mean Discrepancy (MMD) test, Kolmogorov-Smirnov (KS) Test, and the Chi-Squared Test.

To provide an alternative to the two-sample test, Nilsson (2018) and McMahan et al. (2016) show another approach. They divide the data into *shards* and plot these shards on a simple bar charts (histograms) and compare them visually to determine non-iidness. These *shards* are constructed in the following way. The data is sorted, then it is divided up into equally sized shards, and lastly a randomly assigned number of these is given to each client (Nilsson, 2018). The bar charts used are shown in Figure 3.6.2. This test could also be used to determine non-iidness in data sets for Federated Learning application. It could be used as an alternative the two-sample tests.

To make this second step of Rabanser more general, a different approach is presented. Next to merely testing whether the newly assessed method differs in data distribution from the existing method via the aforementioned tests, the three criteria of Duan (2019) are used in this step. These criteria are: (1) size imbalance, (2) local imbalance, and (3) global imbalance. Here the second

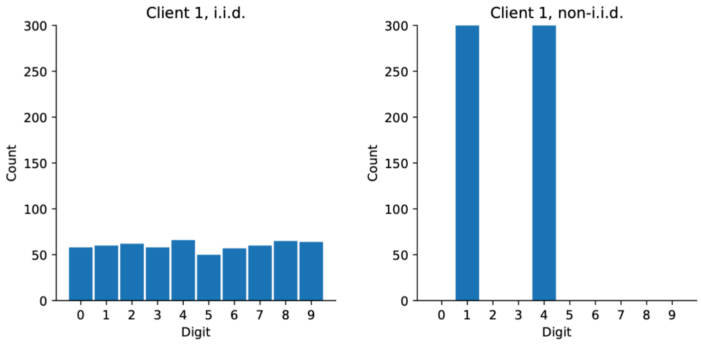


Figure 3.6.2 - Determining non-iid data sets by shards.
Nilsson (2018)

criterium coincides with the two-sample tests, as they test differences in distribution, which is another term for local imbalance.

Lastly, the method is concluded by making an professional assessment and taking conclusions on the insights provided in the previous step.

A shortcoming of this method for Federated Learning is that is only compares two data sets at a time. Whereas Federated Learning has the potential of having a large number of separate data sets. The number of combination might grow large very quickly as more data sets are added. A possible solution could be to merely take a sample of the data sets involved and/or select data sets which have a higher likelihood of differing from the other distributions.

3.6.5 Conclusion

To conclude, non-iidness in the context of Federated Learning has a slightly different definition and way of assessing than in the traditional definition. As in Federated Learning there is not one data set, but multiple separated data sets which by principle cannot be combined, non-iidness cannot be assessed by means of an evaluation on one data set. Instead, it should be assessed by comparing each potential new data set incrementally to the existing data sets. For this, the methodology of Rabanser et al. (2018) is used and altered. The proposed method consists of three steps:

1. Dimensionality reduction, i.e. feature selection, to reduce the number of features to a manageable number with only relevant features;
2. A three-criteria test on non-iidness; and lastly,
3. A final assessment, where conclusions are drawn based on the findings in the previous step.

For the second step the three criteria of causes of non-iidness in Federated Learning by Duan (2019) is used:

1. Size Imbalance, where the data size on each device (or client) is uneven;
2. Local Imbalance, where each device does not follow a common data distribution;
3. Global Imbalance, means that the collection of data in all devices is class imbalanced.

In this way, an assessment on the non-iidness of a newly added data set to a Federated Learning system can be made.

4. Method Design

In this chapter the artifact, i.e. the method, will be designed. The method will be designed according to the research methodology Situational Method Engineering (SME) of Harmsen (1997), as also specified in chapter 2.2. This will also constitute the third phase of the overall research methodology DSRM: design and development.

4.1 Characterization of Situation

A *situation* is the combination of circumstances at a given organization (Harmsen, 1997). The to be designed method will, then, be tailored to this specific situation to achieve its set goal. It is, therefore, an important first step to define the situation, as all other parts of this research methodology are dependent on this. Looking at the already executed first two phases of DSRM, problem identification & motivation and define objectives of a solution, a similarity can be observed between the characterization of situation step in SME. Moreover, the similarity reaches to such an extent that this phase will make use of the already drawn problem context.

The situation where this method will be applied can be characterized by the following description: This method is intended for organizations who (i) want to implement Federated Learning, (ii) are unaware of the best possible Federated Learning method regarding their data-related characteristics, (iii) have a data landscape with multiple distinct data silos, and (iv) data sharing between data silos is limited due to privacy considerations. These privacy considerations have their origin in a legal and a competitive-interest perspective. Organizations who can identify with this situation characterization are a potential fit for this method.

The goal of the method is to find the best solution-fit for this organization given the organization's data-related characteristics (and privacy requirements, which can also be seen as a data-related characteristic) which are relevant to the choice of a Federated Learning method. This method will precede any actual implementation of a Federated Learning method, but does, however, lay the groundwork for an informed choice. As it requires technical expertise in the field of machine learning and data science, the actors executing this method should have professional skills relating to these fields. These actors are already identified in the brief stakeholder analysis in the introduction of this report.

The situation characterization is scoped to be narrow and not too complex, which is exactly what Harmsen (1997) suggests as a good practice. The project characterization should not be to include all possible factors as it would make the project characterization process too complex. However, this characterization does include several key factors, as suggested by Harmsen: the goal of the situation, the skill of the actors needed, project related factors, and knowledge and expertise of the users. In this way all four situational factor types are included in this characterization.

4.2 Selection of Method Fragments

In this section the next step of Harmsen's Situational Method Engineering is conducted: the selection of method fragments, and as an extension also the method base is described. The characterization of the situation, drawn in the previous section, is used as a basis for the selection and creation of method fragments. Meaningful selection of the right method fragments require a thorough characterization of method fragments in a formal way in order to maintain comparability and consistency (Harmsen, 1997). To account for this, method fragments are defined by using the already drawn template in Chapter 2.2, Table 2.2.1.

Relation to Literature Review

The already conducted literature review is used as a basis for creating and selecting method fragments. More specifically, the earlier determined differentiating characteristics of Federated Learning are used as the main way of shaping the method. This is used as a basis because of the definition of differentiating characteristics. Differentiating characteristics are characteristics of Federated Learning methods which both (i) are relevant to the to-be-designed method, i.e. those which may limit options or impact the desired outcome regarding a organization's data-related characteristics and privacy considerations, and (ii) have variation in implementation among the Federated Learning methods, i.e. not all Federated Learning methods.

These differentiating characteristics are used as the basis for the method fragments, because they are in line with the goal of the method: to make an informed choice between Federated Learning methods. These differentiating characteristics allow for a sensible and definition-based decomposition of the problem at hand. This makes sure that the reasoning is clear, explicit, reproducible, and, above all, that it contributes directly to the overall goal. In addition, this decomposition makes sure that the methods do not overlap but also, together, provide a complete solution, as is suggested by Harmsen (1997).

The five differentiating characteristics of Federated Learning methods, determined earlier, are:
1. Data partitioning, i.e. system heterogeneity;
2. Underlying machine learning models;
3. Privacy guarantees;
4. Performance (accuracy, predictive performance);
5. Non-iid data support, i.e. statistical heterogeneity.

Next, the relation between the method fragment selection process and the already conducted literature study is explained.

Research question 2 takes inventory of existing Federated Learning methods in the literature. This list determines what the possible options are in making an informed choice, it is used as the basis for the remainder of the selection process.

Research question 3 (and in extension, 4 and 5) determines the relevant and differentiating (data-related) characteristics of Federated Learning. It asks what are unique and relevant characteristics of Federated Learning methods which have an impact on the to be made choice. In this way an informed selection can be made of the characteristics of Federated Learning which truly matter to achieve the goal of this method. Instead of including all possible characteristics of Federated Learning, which may not be relevant, a small but relevant selection is made. Each of the identified differentiating characteristics can be translated to (selected) method fragments, as the relevancy is already determined.

Research Question 6 is more fine-grained. The impact of non-iid data on Federated Learning is so significant that it is appropriate to find an additional, lower-level method for determining non-iidness. As non-iidness determines to a large extend which Federated Learning method is more suitable for that situation. Some Federated Learning methods are created specifically to work better with non-iid data. Therefore, a way of determining the non-iidness of a data set should be included. For this, a small literature study is conducted, to make a selection within the myriad of possible ways of identifying non-iidness in Federated Learning. This identification of non-iidness is then used as a method fragment, but on a lower and more fine-grained level than the previous five.

Method Fragments

In this section the method fragments, which will be the building blocks of the overall method, are described. The method fragments are selected mainly based upon the 5 identified differentiating characteristics of Federated Learning methods. The template for each method fragment was given in Chapter 2.2 in Table 2.2.1, and is repeated here for convenience in Table 4.2.1. The goal of the method fragment also represents the contribution argument, in terms of Wieringa's Design Cycle (Wieringa, 2014). It is a way of reasoning why it is feasible that this method fragment would contribute to stakeholder and research goals and, in turn, the overall goal of the methods.

Table 4.2.1 - Method Fragment Properties Template

Method Fragment Property	Explanation
Name	Name of the method fragment
Description	Description of the method fragment in freeform text

Method Fragment Property	Explanation
Goal	The goal of this method fragment. It should contribute to the overal solution objective goal
Input	Input needed for this method fragment, such as: data, knowledge, resources
Prerequisites	Required other method fragments which need to be completed before this method fragment
Actions	The actions this method fragment will undertake
Output	The output this method fragment produces. Such as: new insights, data, knowledge

Method Fragment 1: Determine Underlying Machine Learning Problem Type

Table 4.2.2 - Method Fragment 1: Determine Underlying Machine Learning Problem Type

Method Fragment Property	Description
Name	Determine Underlying Machine Learning Problem Type
Description	In this method fragment the underlying Machine Learning Problem Type will be either chosen, if there is still flexibility in this, or determined
Goal	To determine the Machine Learning Problem Type
Input	- Business objective regarding Federated Learning system goals - (Provided by the method) Machine Learning Problem Type List [Definition 4.1 and 4.2]
Prerequisites	Business objective for Federated Learning system is set
Actions	Investigate or make an informed choice about the underlying Machine Learning model of the prospective Federated Learning application
Output	Knowledge about the Machine Learning problem type is clear

The goal of this method fragment is to determine or choose the Machine Learning problem type that underlies the prospective Federated Learning application. As each Federated Learning method only supports a certain Machine Learning model, the choice to what is suitable for a given situation is limited. For example, if the underlying machine learning model only supports the problem type of classification it is of no use if the intent was to solve a linear/regression problem. Therefore, it is of significance to know which problem type should be supported.

The method fragment requires the organization to already have set the business objective regarding the prospective Federated Learning system. This is also the input of the method fragment. This business objective is required because it determines the Machine Learning problem type.

The users of the method fragment are tasked to determine the machine learning problem type that the prospective Federated Learning application should support. The users are supported by provided definitions. As provided input, several types of Machine Learning model types and problem types are listed, which are extracted from the literature. These can be found in *Appendix E:* definition 4.1 and 4.2.

The user of the method fragment should, by means of analyzing the problem statement of the prospective Federated Learning application, and by means of this list, identify which problem type supports the defined problem objective best. This result will later be used to figure out which Federated Learning method supports this problem type.

Method Fragment 2: Data Partitioning

Table 4.2.3 - Method Fragment 1: Determine Data Partitioning Type

Method Fragment Property	Description
Name	Determine Data Partitioning Type
Description	This method fragment is used to determine the data partitioning type (HPD or VPD) of the data sets which will be used in the prospective Federated Learning model.
Goal	To determine the Data Partitioning type: HPD or VPD (Horizontally or Vertically Partitioned Data), which will limit options.
Input	- Prospective data set(s) - Domain model (optional) - Data partitioning definitions and guidelines [Definitions 4.1 and 4.2, Appendix E]
Prerequisites	-
Actions	The data set is analyzed and determined, by their definitions, to be HPD or VPD.
Output	Knowledge about the type of data partitioning

The goal of this method fragment is to determine whether the data set that is intended to be used in a prospective Federated Learning application is either HPD or VPD. This is an important piece of knowledge needed to determine which Federated Learning methods are applicable to this data set. Some Federated Learning methods have restrictions in what type of data partitioning they support; most support only HPD, some only VPD or both. Therefore, this piece of knowledge is vital to the overall goal of the method.

The actors of this method fragment will analyze the (distributed) data set, intended to be used for a prospective Federated Learning application, and will determine whether the data set is horizontally (HPD) or vertically (VPD) partitioned. This analysis is accompanied by the already drawn definitions of HPD and VPD. These definitions will provide the the user of this method the knowledge to determine themselves what the data partitioning type of the data sets are. As the situation characterization already stated, the executers of this method will be professionals in the area of data science and machine learning. It can therefore be assumed that the users of this method will be able to determine the right type of data partitioning of the data sets by using their professional expertise supplemented by the definitions of HPD and VPD. These definitions can be found in Appendix E: Definition 5.1 (HPD) and Definition 5.2 (VPD).

The method fragment uses the prospective data sets as input. Characteristics of these data sets will be used to test against the definitions of HPD and VPD. If a domain model (diagram) is available it can also be used as an addition, as it can speed up the process. Otherwise a domain model can also be constructed by inference from the data sets itself. As a guideline, if the domain models per data site are (practically) identical, then it is an indication that each data site stores the same features for different subjects. This is then a strong indication for HPD. However, the data itself should also be analyzed to confirm this assumption, by means of identifying whether subjects are indeed not fragmented across data sites.

It is recommended that the data partitioning type is documented for later purposes, it represents the knowledge output of this method fragment.

Method Fragment 3: Determine Privacy Guarantee

Table 4.2.4 - Method Fragment 3: Determine Privacy Guarantee

Method Fragment Property	Description
Name	Determine Privacy Guarantee
Description	This method fragment is used to determine the privacy guarantee level required by the organization on data sets that will be used by a prospective Federated Learning method.
Goal	To determine the required privacy guarantee level of the data set(s) used
Input	- Prospective data set(s) or Domain Model (optional) - Privacy Level Definitions and Guidelines [Definition 6, Appendix E]
Prerequisites	Organization has assessed privacy requirements of the data sets
Actions	Data set(s) get categorized in their privacy guarantee level
Output	Knowledge about the Privacy guarantee level of a data set

The goal of this method fragment is to determine what the required privacy guarantee level is of the data set(s) used. This, in turn, contributes to the higher-level goal of the overall method: it ultimately helps the user of the method in making a choice between the Federated Learning methods available. For this, the privacy guarantee is important. It is a hard requirement, i.e. it cannot be changed by the user of the method itself, as it originates from a legal or competitive-interest source. Therefore, only Federated Learning methods which can at least support the minimum required level of privacy can be considered viable options.

The users of this method fragment will assess the privacy guarantee level of the data sets involved in the prospective Federated Learning application. As the users' have experience in data science and/or machine learning, they will have sufficient knowledge on how to assess this on a technical level. In addition, in Federated Learning three levels of privacy guarantees are distinguished and described. This piece of knowledge together with the already inherent expertise of the users will provide enough context to determine the privacy guarantee level, given legal and competitive-interest requirements are set. These requirements are assumed to be non-changeable as the user of the method has nog influence over them, they are set by management or the legal system.

In the previously conducted literature study 3 privacy guarantee levels were identified in Federated Learning. These levels can be found in Appendix E: Definition 6.

These categories and their descriptions should provide information for the user to determine the appropriate category. To further help users determine the privacy level of the data sets, the following guidelines supplementing the descriptions of the three privacy guarantee levels are given next.

First the legal ramifications are discussed. The literature references to the European GDPR prevalently (Yang et al, 2019; Sun et al, 2019; Liu et al., 2019; Nilsson et al., 2018), indicating that this piece of legislation is new and strict. The GDPR will therefore be used as a leading example for privacy requirements from a legal perspective. Yang et al. (2019), Nilsson et al. (2018), and Sun et al. (2019) indicate that the no data sharing principle of Federated Learning does not pose any additional violations of this regulation, and can therefore be seen as compliant to GDPR. This is under the assumption that the organization was already compliant with the GDPR before the implementation of Federated Learning. A full GDPR compliance assessment is out of scope of this study. This study only considers the implications Federated Learning has on these regulations. This line of reasoning, therefore, only holds when the organization was already GDPR compliant in the first place. Concluding, the need for compliance with privacy regulations is well-addressed by the standard Federated Learning privacy guarantee level (2) of no data sharing between data sites, given that privacy regulations were already met before the implementation of Federated Learning in the first place.

This also implies that the need for a higher level of privacy guarantee, the additional privacy mechanism, does not originate from general privacy regulations, but from competitive-interest considerations or specific privacy laws for specific industries. The information that has the potential of being leaked at the privacy level up to this one, is merely aggregate data instead of raw data. If competitive-interest considerations are of such importance that even aggregate data is of utmost importance to remain private (to other parties/data sites involved in the prospective Federated Learning application), then this additional privacy mechanism level is the right choice.

Lastly, the least protective privacy guarantee is discussed. Given these previous guidelines, it seems as if the first privacy guarantee level (violates the no data sharing principle) does not apply at all. However, it is recommended that an organization chooses the least restrictive privacy guarantee, as this limits choices of Federated Learning methods the least. An example of a context in which this level is viable would be when the organization has control over all data sites involved, but does not want to transfer all data to a central server. In this way, both the competitive-interest considerations and the privacy regulations do not apply.

It is recommended that the resulting privacy guarantee level is documented for later purposes, it represents the knowledge output of this method fragment.

Method Fragment 4: Non-iid Data Identification

Table 4.2.5 - Method Fragment 4: Non-iid Data Identification

Method Fragment Property	Description
Name	Non-iid Data Identification
Description	The prospective data sets are tested whether they classify for non-iidness in the context of Federated Learning
Goal	To determine whether the prospective data sets are non-iid or not
Input	- Prospective data sets - Non-iid definition, sub-method, and guidelines (provided by method)
Prerequisites	Privacy guarantee level known (output of method fragment 3) & data partitioning type known (output of method fragment 2)
Actions	The data sets are assessed to be non-iid or idd, via a three-step sub method.
Output	Knowledge about whether the data sets are iid or non-iid

The goal of this method fragment is to determine whether the data sets used for the prospective Federated Learning application are non-iid or iid. The contribution argument that is made for the justification of this method fragment is the following. There are several Federated Learning methods which perform sub-optimally in terms of predictive performance in a non-iid context. Whereas several other Federated Learning methods exist which are specialized to optimize the predictive performance results in a non-iid context. Therefore, to make an informed choice of the Federated Learning methods, which is the overall goal of the method, it is of added value to know whether the data sets in question are non-iid.

The users of this method fragment will analyze the prospective data sets and determine whether they are non-iid or iid in the context of Federated Learning. As non-iidness in the context of Federated Learning is not something that can be assumed to be knowledge the user possesses, a definition of non-iid data sets is extracted from the literature study.

The provided input of this method, the three-step non-iid identification sub-method, alongside the definition of non-iid data in the context of Federated Learning can be found in the conclusion of research question 6, in Chapter 3.6.5.

This method requires an iterative approach when more than two data sets are used in the evaluation, as the method only compares two data sets at a time. This can quickly become very time consuming as the number of combinations quickly rises the more data sets are involved. For this scenario, a sample of some of these data sets can be used.

The prerequisites of this method fragment are that the privacy guarantee level is known and that the data partitioning type is known. This is important for the sake of the method's efficiency, as the outcomes in terms of the most applicable Federated Learning methods are limited to such an extend that the non-iid identification method fragment does not provide more useful information to make an informed choice.

Method Fragment 5: Predictive Performance Trade-Off

Table 4.2.6 - Method Fragment 5: Predictive Performance

Method Fragment Property	Description
Name	Predictive Performance Trade-off
Description	In this method fragment the importance of predictive performance is assessed. If the data sets are non-iid a trade-off analysis between performance and privacy is to be assessed.
Goal	To choose the Federated Learning method which, according to the literature, has the best predictive performance opportunity for this scenario. A trade-off between privacy and accuracy is made for non-iid data contexts.
Input	- Machine Learning Problem Type - Data partitioning type of the data sets - Privacy guarantee level - Non-iid assessment - Federated Learning methods performance lookup table (provided by method)
Prerequisites	- Underlying Machine Learning Problem Type known (output of method fragment 1) - Data partitioning type known (output of method fragment 2) - Privacy guarantee level known (output of method fragment 3) - Data sets are analyzed to be non-idd or iid (output of method fragment 4)
Actions	Trade-off between privacy and accuracy is made, and related to whether the data sets are iid or non-iid. The most applicable Federated Learning method is chosen for the situation.
Output	Decision of the most applicable Federated Learning method

The goal of this method fragment is to assess the importance of predictive performance and possibly make a trade-of with privacy.

This method fragment is only applicable for situation where it is assessed to have data partitioning type HPD, an underlying Machine Learning problem type which is supported by Neural Networks, and no additional privacy mechanism is needed. If these criteria are met, a more informed choice regarding predictive performance can be made. For the other cases the literature is not clear, and nothing valuable can be said about the expected predictive performance of the Federated Learning method compared to others.

First, the literature study shows a clear distinction between non-iid and iid data sets regarding performance. It shows that standard Federated Learning methods are not well-adapted to work well with non-iid data. If the data is non-iid traditional Federated Learning methods do not create a model that is generalizable among all data sites, resulting in lower accuracy. This is due to the fact that the data sites have different data distributions, thus a common model is not feasible. Specialized Federated Learning methods, such as Zhao et al's (2018) method or the Astraea method of Duan (2019) use balancing mechanisms to mitigate this problem. It is, therefore, recommended to implement such a specialized Federated Learning method for a non-iid context.

In the case of the data sets being non-iid, two Federated Learning methods should be considered: Zhao et al's (2018) method and the Astraea method of Duan (2019). Both apply balancing techniques to attain higher predictive performances in non-iid contexts. To decide between these two a trade-off should be made between privacy and predictive performance. If the goal is to build the best-performing model possible and privacy is not an issue, i.e. the privacy requirements allow for data sharing among data sites and to a central server, then Zhao et al.'s (2018) Federated Learning method should be chosen. It is the best-performing methods in a non-iid context, up to a 55% increase in accuracy compared to baseline. If, however, data sharing is not permitted and this requirement is non-changeable, then the Astraea method of Duan (2019) should be chosen, as it does not violate the no data sharing principle of Federated Learning. In addition, it does show a modest increase in accuracy of 6% in non-iid context compared to baseline.

If the data sets are found to be iid, then the FedAvg method of McMahan (2017) is to be selected. This is the Federated Learning method which in a comprehensive and fair comparison shows the best performance according to the literature, compared to other Federated Learning methods which support HPD data partitioning only, are built upon a Neural Network model, and do no provide an additional privacy mechanism. It is therefore recommended to choose this Federated Learning method in this scenario. See Appendix F - Table F.1 Best Performing Federated Learning Methods per Situation for a summary of the above.

Method Fragment 6: Lookup Table

Table 4.2.7 - Method Fragment 6: Lookup Table

Method Fragment Property	Description
Name	Lookup table
Description	This method fragment generally constitutes the last step of the method, where the resulting Federated Learning method is chosen by means of a lookup table
Goal	To choose the Federated Learning method which matches the situational factors, i.e. matches the organization's data- and privacy-related characteristics and requirements.
Input	- Machine Learning Problem Type - Data partitioning type of the data sets - Privacy guarantee level - Non-iid assessment - Federated Learning methods lookup table (provided by method)
Prerequisites	- Underlying Machine Learning Problem Type known (output of method fragment 1) - Data partitioning type known (output of method fragment 2) - Privacy guarantee level known (output of method fragment 3)
Actions	A Federated Learning method will be chosen by the user according to the earlier determined characteristics
Output	Decision of the most applicable Federated Learning method

The goal of this method fragment is to make the final decision for the most applicable Federated Learning method for the organization's particular situation. For this method fragment the user should use the knowledge gained in the previous method fragments regarding the data partitioning type, the underlying machine learning problem type, and the minimum privacy guarantee level.

For situations which do not identify with the previous section, the look-up Table F.2 in Appendix F should be used. Here the user is tasked with going through all listed Federated Learning methods and comparing their differentiating characteristics with the knowledge gained from previous method fragments regarding the data partitioning type, the underlying machine learning problem type, the privacy guarantee, and whether the data sets are iid or non-iid. The user should pick the Federated Learning methods which complies with all previously stated. Per differentiating characteristics some guidelines are set for this process:

- For data partitioning, a Federated Learning method rows should be selected which include the determined data partitioning type of the prospective data sets.
- Mapping should be conducted for the Machine Learning problem type. As certain Machine Learning models can solve multiple problem types (e.g. a linear model can be adapted to work both with linear/regression problems and classification problems. But a classification-type model cannot be used to solve a linear/regression problem.) It is assumed, by means of the prerequisite knowledge of the user of this method, that the user can make this mapping itself.
- For the privacy guarantee, Federated Learning method which are at least on the identified privacy guarantee level or higher should be selected.

The result of this method fragment is the selected and most applicable Federated Learning method given the organization's data- and privacy-related characteristics and requirements, which matches the overall goal of the full method.

Now all method fragments are identified and selected, the method assembly is discussed next.

4.3 Method Assembly

In this section, the method assembly is conducted. The objective here is to combine the method fragments and design the resulting method, but in such a way that it does not contain any defects or inconstancies. This is done by using a strategy, guidelines, and assembly rules, to perform it in a consistent and sensible manner. In general, the method should fit the situation (suitability), i.e. all steps should contribute to the overall goal. To help achieve this, Harmsen (1997) formulated situational dependent and situational independent quality criteria. The situational dependent criteria are formulated in the S^3 model: success, situation, scenario. The independent quality criteria are: completeness, consistency, efficiency, soundness, and applicability. The resulting method is assessed according to these quality criteria in this section.

4.3.1 Situational dependent criteria

Situational dependent quality criteria are formulated according to the S^3 model, as suggested by Harmsen (1997). The S^3 model is, however, adapted to work well for Information System project design, not necessarily design science. A similar concept as the S^3 model can be seen in Design Science. The S^3 model's goal is to justify the design steps of the method. As stated earlier, in Design Science the artifact - in this case the method - is evaluated by *utility* (Wieringa, 2014). To justify the method design beforehand, contribution arguments are made before the evaluation takes place. A contribution argument takes the form of: <context assumptions C> AND <requirements R) IMPLY <contribution to stakeholder G> (Wieringa, 2014). This is similar to the S^3 model, only the project success factors are substituted by contribution to stakeholder goals. This study will therefore use the contribution arguments by Wieringa instead of the S^3 model. These contribution arguments are already included in each method fragment, as the goal of each method fragment is stated explicitly. These method fragment goals, can then be seen as sub goals which realize the overall goal of designing the method, which are all evaluated by utility.

4.3.2 Situational independent criteria

Situational independent quality criteria are formulated in terms of: completeness, consistency, efficiency, soundness, and applicability. These criteria are used as assembly rules for the resulting method.

Completeness

Completeness is the requirement that the situational method contains all the method fragments referred to by the method fragments in the situational method. It can be split up into: input/output completeness, contents completeness, process completeness, association completeness, and support completeness (Harmsen, 1997). Each of these requirements will be discussed next.

Input/output completeness is adhered to by making sure that the method fragments which require pieces of knowledge or product inputs are documented. The method fragments use a template which included an input and an output column. For example, method fragment 2, determining the data partitioning type, requires as input the prospective data sets of the Federated Learning application. Whenever a product is involved in the action column of the method fragment, this product is also included in the input column. Which is also the justification for both *association completeness* and *support completeness*.

In the same manner, *contents completeness* is adhered to by ensuring that each method fragment's required provided content is present in the description. The contents of this method fragment are all pieces of knowledge, definitions, and sub methodologies. These are all added in text form.

Process completeness is adhered to in the following way. The resulting outputs, i.e. the products, are all pieces of knowledge in this method. To ensure that each method fragment's output is produced, each method fragment has an associated goal, which is then realized by the resulting actions from this goal. This way of working ensures the process completeness.

Consistency
Consistency of a situational method addresses the requirement that situational methods do not contain contradictions and are thus mutually consistent. This criterium can be fragmented into multiple sub-criteria (Harmsen, 1997). The justification of the resulting method for this is given next.

The method does adhere to *precedence consistency*, i.e. method fragments are placed in the right order. This is guaranteed by the inclusion of the *prerequisite* column in the method fragment template; any method fragment that requires any other method fragment to be completed first is documented. In this way the order of the method fragments is secured to have no consistency breaches. For example, method fragment 5 cannot start unless all other method fragments have completed.

Support consistency and *perspective consistency* are adhered to by including the right tools and pieces of knowledge per method fragment when it cannot be assumed anymore that the user can execute this task by prerequisite knowledge alone. For example, in method fragment 1, the definition of data partitioning in the context of Federated Learning is given as a piece of knowledge, to assist the user in making a assessment of the data sets. Without this piece of knowledge these consistencies would be broken; the user of the method would be able to complete the task.

Granularity consistency is secured in this method by making all method fragments to be of similar granularity. The method fragments are roughly equal in size (time to complete) and have the same technical level. One challenging method fragment regarding this criterium was method fragment 4. It contains a sub methodology to identify non-iid data. This method fragment encapsulates these sub methodologies, which are of lower granularity and of a more technical nature. The latter is also another example of *support* and *perspective consistency*.

Lastly, *Concurrence consistency* is guaranteed, because no parallel execution of the method fragments are introduced.

Efficiency
Efficiency addresses issues that can be decided on without taking into account the situation. When it is taken into account time and money can be saved. *Efficiency* is the requirement that the situational method fulfill its duty at minimal cost and effort (Harmsen, 1997).

The resulting method takes efficiency into account in the following way. It takes shortcuts in the decision process where applicable. As there are only a finite number of Federated Learning methods available, each with a limited number of combinations of supported differentiating characteristics, not all differentiating characteristics need to be known in certain situation. Thus, not all method fragments need to be executed to come to an informed decision, saving time. For this a coupling with the data and the method is made, i.e. the method is dependent on the data.

Let’s illustrate this with an example. There are only two Federated Learning methods which support the data partitioning type of VPD. Because of this, the resulting differentiating characteristics (underlying machine learning model, privacy guarantee, non-iidness, and performance) have a limited number of remaining combinations available. For this example, there is no difference (at least known difference) between in the performance of non-iid vs iid data sets. Both methods are not characterized to be adapted to work well with non-iid data sets. Given this, it is no benefit to execute method fragment 4, identifying whether the data sets are non-iid or iid. Which can be a time consuming method. In this way efficiency is improved. This way of improving efficiency by coupling the method with the data is used where applicable.

Soundness

Soundness is the requirement that the situational method is semantically correct and meaningful (Harmsen, 1997). In other words, the input and output of a method fragment should fit with each other.

In the resulting method soundness is guaranteed by using the *input* and *output* columns in the method fragment tables, and only sequentially combining them when these fit (e.g. by not ignoring the prerequisites of the method fragment). Each association of the method fragment is tested to have compatible input and output parameters.

Applicability

Applicability is the requirement that actors are able to apply the situational method. Thus, for each technical method fragment, there should be at least one actor capable of working with it (Harmsen, 1997).

For the designed method applicability is adhered to by designing all method fragments in such a way that prerequisite knowledge of the actors is taken in mind. The actors, i.e. the users of this method, are the same for every method fragment. They are professional experts in the field of data science or machine learning wanting to apply Federated Learning. They therefore do already have knowledge about building and implementing traditional machine learning models, and all indirect associated knowledge. Therefore, all method fragments are designed to take into account this assumption. Thus, whenever a user is tasked with a concept which requires knowledge that is particular to Federated Learning, pieces of knowledge, definitions, or sub methods are provided to give context. In this way, applicability is adhered to.

Given the justification of the method assembly process by means of all of these situational dependent and independent quality criteria, the method is designed by making use of these assembly rules and the earlier constructed method fragments. The resulting method is introduced in the next section.

4.4 Resulting Method

In this section the resulting method is presented visually. The method fragments identified in section 4.2, together with the method assembly rules and quality criteria from section 4.3 are used to design this resulting method. The method is visualized in BPMN, Business Process Model and Notation (bpmn.org), and can be seen in Figure 4.4.1 at the end of this chapter.

Each method fragment has been converted to a *task activity* in terms of BPMN. Each task activity is further connected via *access-relations* to (data) *input* and *output objects*. There correspond to the earlier identified input and outputs in the method fragment tables. For example, the first method fragment *Determine Underlying Machine Learning Problem Type* has two input objects (Federated Learning Business Objective & Machine Learning Problem Type List) and one output object (Machine Learning Problem Type). The grey fill color of some of the input objects, such as the Machine Learning Problem Type List input, indicate that these inputs are provided by the method itself.

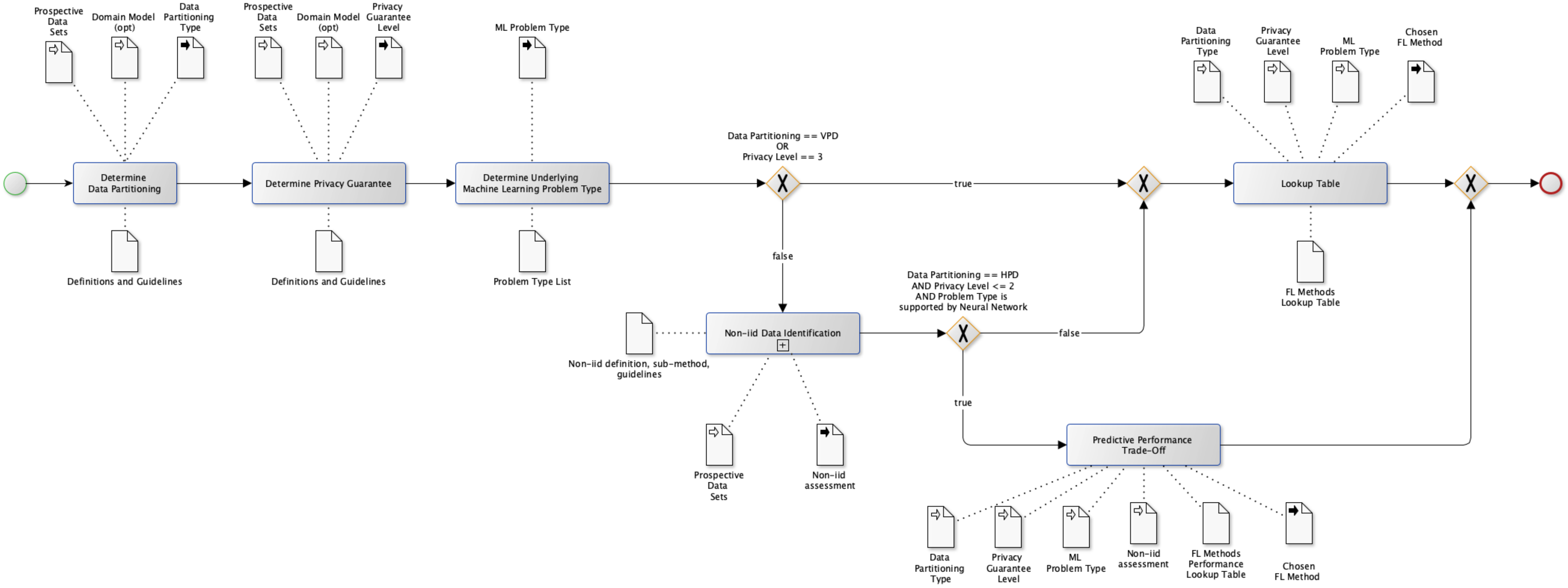


Figure 4.4.1 - Resulting Method in BPMN

The first logic gate check whether the Data Partitioning type is VPD or if the organization requires the privacy level to be of the highest level (3). This distinction is made because of the efficiency situational independent criterium. Situations with these characteristics are so unique that one a very limited number of Federated Learning methods are available; further characterizing the situation (e.g. for it to be non-iid or the performance trade-off task) does not provide more relevant information. Therefore, those unnecessary tasks are skipped.

If the logic gate follows the false path an assessment will be made regarding the non-iidness of the data sets. This task activity is comprised of a sub method. A sub method is provided as input by the method, i.e. the method of Rabanser et al. (2018).

The second logic gate check whether the Machine Learning Problem Type, identified earlier, is supported by using a Neural Network. This is because the literature supporting the predictive performance trade-off step is scoped to only say something meaningful in the context of neural network Federated Learning models. Out of this scope, the predictive performances have not been thoroughly compared, and no meaningful choice based on better predictive performance can be made. Therefore, the other cases default to the general Federated Learning methods lookup table. Both task activities result to the output of a chosen Federated Learning method for the organization’s specific situation, which is also the end of this method.

5. Evaluation - Demonstration by Case Study

In this chapter the designed method is demonstrated by means of a real-world case study at the software company Topicus. This intent of this demonstration by case study is to show that the method can work and is useful in a real-world setting. This demonstration constitutes the fourth phase of DSRM and comprises the validation of the artifact.

The case study is structured as follows. First, a general description of the company is given to provide context. Then, the company-specific problem statement is given and compared to the general problem statement and scope of this research. After that, the designed method is executed and described. Real-world data sets are included in this case study; several Dutch banks provided mortgage application data for this case study.

5.1 Company Description

The company where this case study will be executed is Topicus. Topicus is a software development company, founded in 1998, with various locations throughout The Netherlands, and, as of writing, has more than a thousand employees. Topicus is divided into five departments: Finance, Healthcare, Education, Government/Social domain, and Core and is consequently also active in the first four similarly named sectors. Topicus provides software and services to each of these four domains.

The department in which this case study takes place is the Finance department. The main customers of the Finance department are some of the big financial institutions (for simplicity also referred to as banks) in The Netherlands. At Topicus Finance a software product called the Force Product Suite (FPS) is developed and provided to these banks. The FPS incorporates a myriad of services, but the common denominator is that it supports the lending processes at these banks, especially the mortgage application process. The software provides mostly back-end, but also some front-end solutions to support the mortgage application process, like: data entry, integration with third-party services, decision making, but also analytics and reporting.

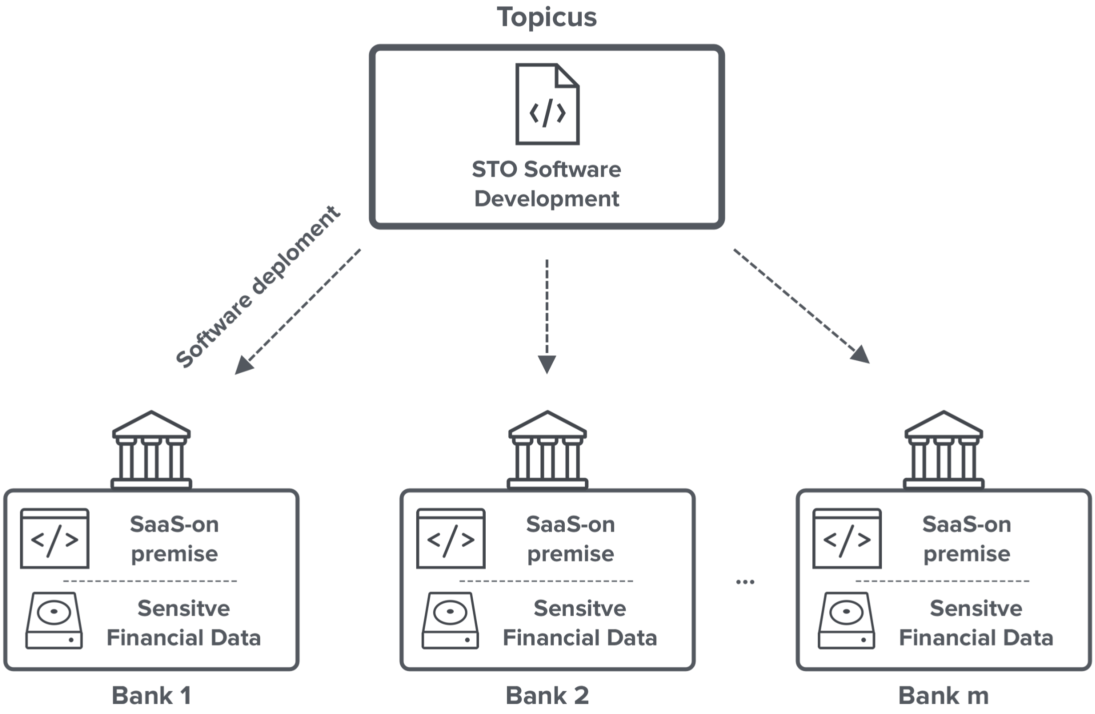


Figure 5.1.1 - Problem context at Topicus. Multiple separate data silos with private data

One of the services FPS offers is Staging-out (STO). The FPS generates operational and heterogeneous data. Because of this, the data is vast and not readily useable for the clients to use for (internal and external) reporting and analysis. To help in this need, STO is developed. STO is a data warehousing service which provides structured and relevant input for reporting and analytics tools. The sub-department where STO is developed is where the case study is situated.

Infrastructure

The FPS is typically hosted at the client itself, as a SaaS-on premise (Software-as-a-Service). Consequently, the software at these clients are not identical. Each instantiation can contain a subset of the complete landscape of the software services FPS offers; some banks only acquire some parts of the services provided. In addition, different versions of the software are made, with minor modifications to suit the business context of a client better. All in all, the main features and fundamentals of the software suite will roughly still be the same, but not completely identical.

The applications and data are, due to privacy considerations, strictly separated. These privacy considerations constitute both the company's own interests as juridical, as they feature sensitive (financial) customer data. This is also part of the reasoning why the infrastructure is separated, by means of the SaaS-on premise concept. In Figure 5.1.1 a visual representation of this infrastructure is shown.

Data Description

A sample of the data in the data warehouse of this STO service is obtained, and the data model of this is shown in Figure 5.1.2. This data model is a blueprint for each STO service and is instantiated almost identically at each of Topicus' clients. Important to note is that the data is strictly separated and resides in distinct data silos at each client. Topicus does not store this data on its own servers, it only pushes new versions of the domain model and the software to those clients.

5.2 Problem Statement

At Topicus Finance, next to providing operational software to mortgage lenders (i.e. banks), Topicus also develops reporting and analytics services. On the frontier of these analytical services are the development of predictive models based on machine learning. Especially, there is an apparent wish at Topicus to predict the lead times of mortgage applications (i.e. the time it takes for a mortgage application to finalize from the moment is is initialized). As of right now, these predictive models are being built merely locally at one mortgage lender at a time. However, Topicus seeks to utilize the potential of combining the vast amount of data of all its mortgage lenders, making a more reliable 'super' model, not just from one data site. They assume that more data leads to better predictive models in this context.

However, in this problem context, combining data from multiple banks is not possible in practice: the financial data is privacy-sensitive. Privacy considerations from both the legal sphere, mainly GDPR, and from competitive interest considerations, banks do not want to share customer information with competitor banks, play a role. These privacy considerations prevent the extraction of the data. Also, no known algorithm in traditional machine learning accounts for the usage of training on this heterogeneous distributed network. Next, the assumption that more data leads to better results can be questioned because the data is generated in different quantities at those mortgage lenders: some produce more quantities of data and possible have another distribution (different types of workflows or customers). A concern here is that some mortgage lenders' data will overshadow the others' and produce worse results for some mortgage lenders instead of better results. Topicus is therefore searching for a method which can make use of these separated data islands and train a global model while also not violating the privacy concerns.

Federated Learning is a good fit for this problem context. It fits both with the need of utilizing multiple separated data sites for training a global model, and doing so in a privacy-preserving manner; the two main requirements Topicus has. However, at Topicus the data scientists have only been recently introduced to the concept of Federated Learning and are unaware of the possibilities in Federated Learning.

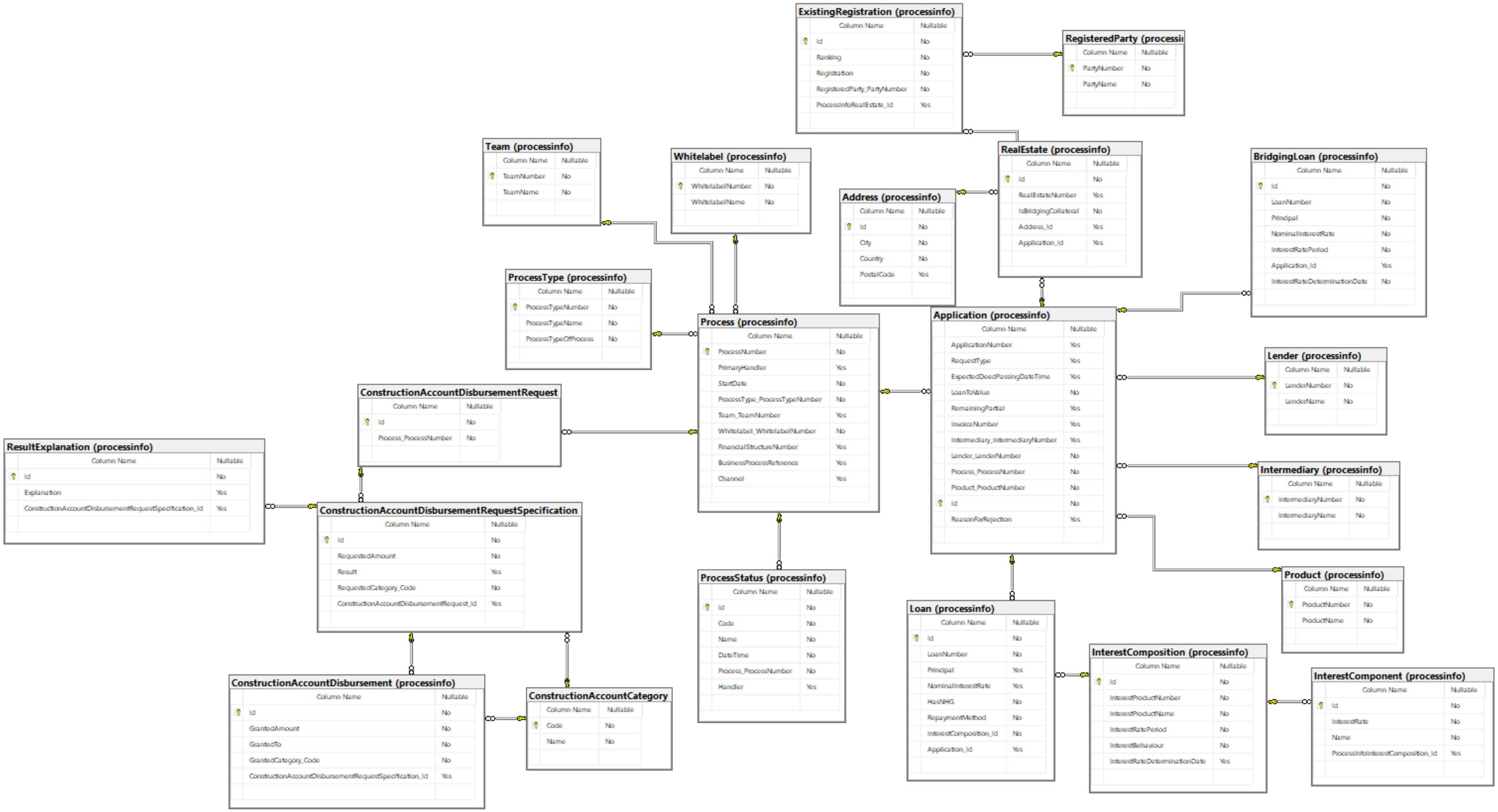


Figure 5.1.2 - Domain Model at Topicus

Given this problem statement, Topicus is a perfect candidate for the proposed method of this research. It has an apparent need to utilize the potential of using multiple separated data sites in a privacy-preserving manner through Federated Learning. However, it is not familiar with Federated Learning and its myriad of different Federated Learning method. This is exactly what this research is for.

5.3 Case Study Execution

The company has provided the researcher with the resources needed to execute the method. It has provided real-world data from two of its clients, two mortgage providing banks active in The Netherlands. In this section the designed method is applied to and executed on this case study. The researcher is executing the method at the company. It is split up in each method phase. Starting with the Determine Underlying Machine Learning Problem Type phase and continuing along the specific path of this instance of the method.

5.3.1 Determine Underlying Machine Learning Problem Type

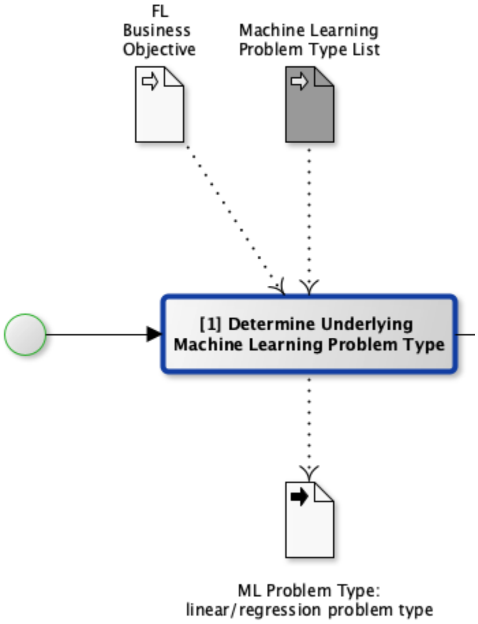


Figure 5.3.1 - First Execution Step of the Method in this Case Study

The goal of the first phase of the method is to determine the machine learning problem type from the problem context. It is important to know this, as Federated Learning methods are all built upon an underlying machine learning model (for example a Neural Network). Because of this, the Federated Learning method are limited in only solving problem types which are supported by these machine learning models.

As specified in the problem context description of this case study, Topicus aims at providing predictive machine learning models for its clients. More specifically, in this case study it aims at predicting mortgage application lead times. This is the time it takes for a mortgage application process to reach the status 'Binding Offer Sent', starting from the instantiation of this application process, which has the status 'Start New Application'. From this description a machine learning problem type is chosen.

The document that supports this method phase provides a list of machine learning problem types. These problem types are: Linear/Regression problem, Classification problem, Rule-learning problem, Clustering problem (unsupervised), Language modeling problem. The user of this method is assumed to be a professional in the field of data science and/or machine learning and should be able to identify the problem type by this list alone.

The each problem type in this list is compared to the prospective aim of predicting mortgage application lead times. First, the problem type is not a classification problem. There is a need to predict a real number, with possibly infinite output options, not a finite set of classes. A rule-learning problem is also not applicable, as it also outputs a finite set of classes. It could be a clustering problem, as it could be useful to identify different kinds of features which drive deviations in mortgage lead times. However, the apparent wish of Topicus is to provide predictions beforehand, which is not the purpose of this problem type. Also a language modeling problem is not applicable as it does not concern text but numeric data. The option that fits best is the linear/regression problem type. It fits the aim of predicting a real number, i.e. a lead time.

The resulting machine learning problem type of this case study is therefore found to be the linear/regression problem type.

5.3.2 Determine Data Partitioning Phase

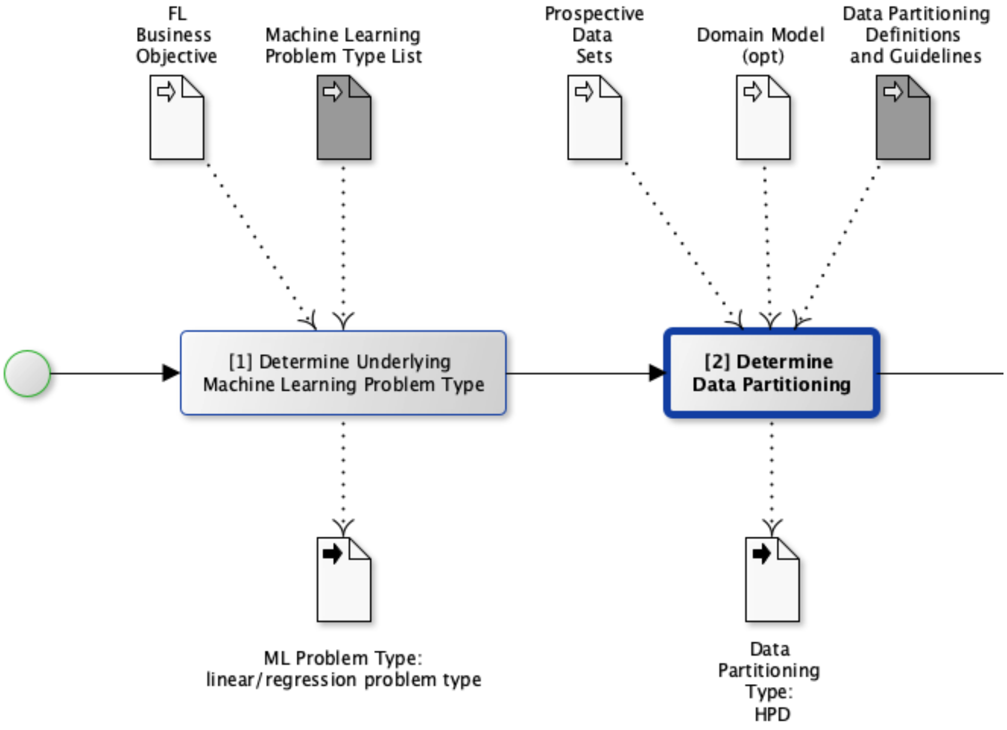

FL Business Objective
Machine Learning Problem Type List
Prospective Data Sets
Domain Model (opt)
Data Partitioning Definitions and Guidelines
[1] Determine Underlying Machine Learning Problem Type
[2] Determine Data Partitioning
ML Problem Type: linear/regression problem type
Data Partitioning Type: HPD

Figure 5.3.2 - Second Execution Step of the Method in this Case Study

In this phase the data partitioning type is determined. Its goal is to determine whether the data sets used in this case study are HPD (horizontally partitioned data) or VPD (vertically partitioned data). This is an important fact to know, as it has implications on what Federated Learning methods are available for Topicus.

The data partitioning type is determined by comparing the prospective data sets used with the definitions of HPD and VPD (provided by the method as a document). The definition that fits best with the characteristics of the data sets will constitute the type of data partitioning. These data sets are required as input for this method phase. The characteristics of this data set will determine the data partitioning type. In addition, the optional input, a domain model, is also available and will be used too.

First, the domain models are analyzed. A domain model diagram is available at Topicus' documentation repository. It is shown in Figure 5.1.2. The domain models at each data site are found to be identical for the 2 banks in question. This implies that the features of all subject types are the same across all data sets. Which then implies the data being horizontally partitioned (HPD). For example, the subject Consumer in the domain model has the same features across all banks, all have the features: id, ConsumerNumber, ConsumerRole, TaxObligationsAbroad, DeathBenefitAmount, and Application_id. This is found to be true for all tables (subject types) in the domain model.

The data sets in question are stored as a relational database in MS SQL Server. A database client is used to gain access to these databases. Because the documentation of this database at Topicus included a domain model diagram it does not need to be inferred. Instead, a check is performed whether the data acknowledges the finding that the domain models are indeed the same and do not store different features of the same subjects.

It is, however, possible that the same customer could have multiple mortgages at different banks. No instances where found in these data sets, as the data sets were anonymized to only include database specific customer id's, not cross reference information. Nevertheless, this is not a cause for invalidating the previous findings. Practically, the customers (who can be classified as subjects) are different subjects, as they instantiate a new mortgage and the data cannot be linked.

Concluding, the data partitioning type is determined to be HPD (horizontally partitioned data). The same features are stored across the banks' data sets and data of an individual subject is stored at one data site only.

5.3.3 Determine Privacy Guarantee Level Phase

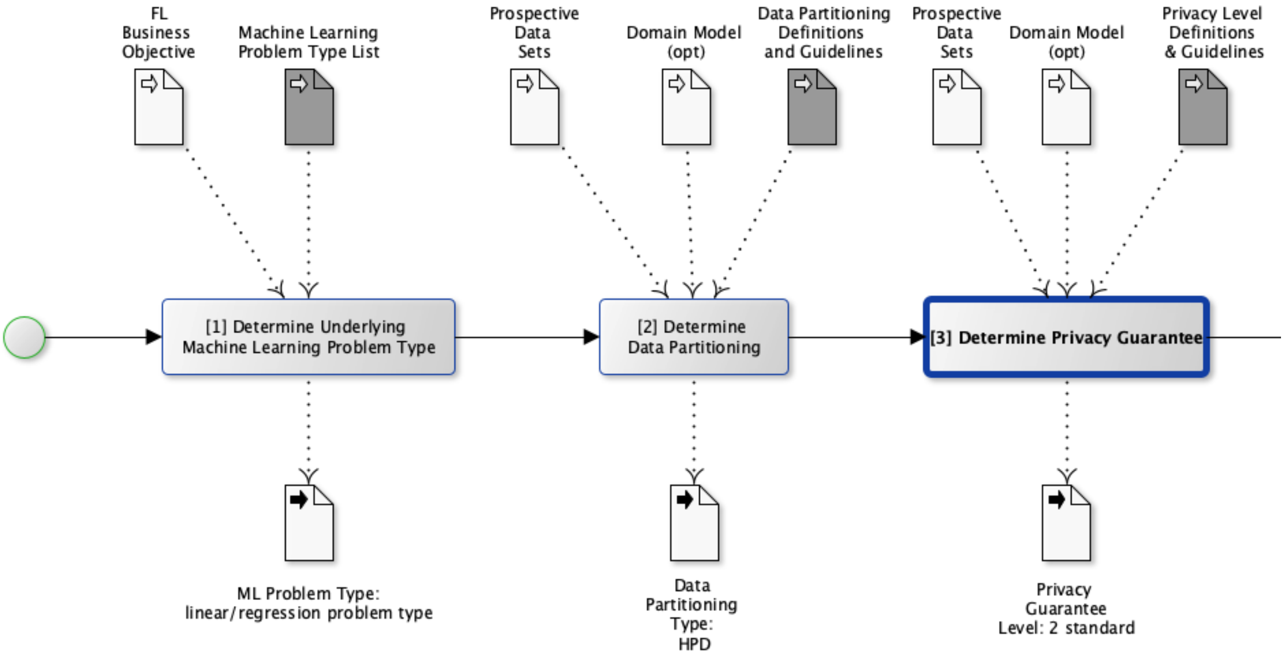

FL Business Objective
Machine Learning Problem Type List
Prospective Data Sets
Domain Model (opt)
Data Partitioning Definitions and Guidelines
Prospective Data Sets
Domain Model (opt)
Privacy Level Definitions & Guidelines
[1] Determine Underlying Machine Learning Problem Type
[2] Determine Data Partitioning
[3] Determine Privacy Guarantee
ML Problem Type: linear/regression problem type
Data Partitioning Type: HPD
Privacy Guarantee Level: 2 standard

Figure 5.3.3 - Third Execution Step of the Method in this Case Study

This phase of the method is intended to determine the privacy guarantee level of the data sets. The data sets can be at one of three levels: (1) violates the no data sharing principle, (2) privacy by no data sharing, or the most stringent, (3) additional privacy mechanism. It is important to have clear what privacy level of the data sets are, because some Federated Learning methods may not guarantee the level of privacy required.

In Federated Learning there are three levels of privacy guarantee distinguished. These levels and their definitions are provided by the method as a document. These definitions provide information to the user of the method to distinguish the three privacy guarantee levels and compare it to its situation. These definitions can be found in section 4.2, method fragment 3, and are not repeated here. In addition, the provided guidelines are followed were applicable.

The only input of this phase are the prospective data sets. These data sets are analyzed for containing privacy sensitive data, both from a legal as a competitive-interest perspective. The case study description already mentioned the presence of sensitive personal data. This can be acknowledged by testing it to the GDPR definition of personal data: "personal data means any information relating to an identified or identifiable natural person ('data subject'); an identifiable natural person is one who can be identified, directly or indirectly, in particular by reference to an identifier such as a name, an identification number, location data, an online identifier or to one or more factors specific to the physical, physiological, genetic, mental, economic, cultural or social identity of that natural person" (EU GDPR Regulation, 2016). As the customer data in the data sets contains identification numbers and location data, it is classified to be personal data.

Therefore, it is subjected to limitations in processing and sharing this data. Processing in this regulation is defined as: "processing means any operation or set of operations which is performed on personal data or on sets of personal data, whether or not by automated means, such as collection, recording, organisation, structuring, storage, adaptation or alteration, retrieval, consultation, use, disclosure by transmission, dissemination or otherwise making available, alignment or combination, restriction, erasure or destruction" (EU GDPR Regulation, 2016). Looking at this definition, training a model which shares data as specified in level 1 of the privacy guarantee levels does imply data processing, which is not compliant to this regulation. As specified in the guidelines of this method phase, the privacy level should at least be level 2: privacy by no data sharing.

This level also suits the competitive interest considerations of the participating banks. At this privacy level their raw data does not leave their premises. As no additional requirements are set by Topicus or the company in terms of encryption or any other additional privacy mechanisms, the privacy level is not elevated any higher. It is, therefore, determined that the privacy guarantee level is (2) privacy by no data sharing.

For the next step, a conditional branch is encountered. As the data partitioning type is found to be HPD and the privacy level 2, privacy by no data sharing, the 'false' branch of the method is followed. So the next phase will be the Non-iid Data Identification phase.

5.3.4 Non-iid Data Identification Phase

For this phase an assessment will be made whether the two data sets would be non-iid when combined. The data sets are used as input for this method step. Alongside this, the non-iid definition and the the three-step method devised in Chapter 3.6 (research question 6) are used as provided inputs.

The data sets are from 2 banks in The Netherlands, and stored in MSSQL relational databases. It includes real-world mortgage application data, and are both from the same time period. The data are extracted by means of SQL queries. Also, the data sets are anonymized, they do not include any identifiable customer data, identifiable addresses, and the contract dates are changed by a random value.

Next, the 3-step sub-method is executed.

1. **Dimensionality Reduction**

The first step in the submethod of this phase is dimensionality reduction. The goal of this step is to reduce the number of features that are potential inputs for the model. Only features which have a predictive performance potential are included in this assessment. This feature selection is important to help in understanding the data, but also to increase the predictive performance of the prospective model (Chandrashekar & Sahin, 2014). This predictive performance expectancy assessment has been conducted with the help of an expert working on the STO application at the company.

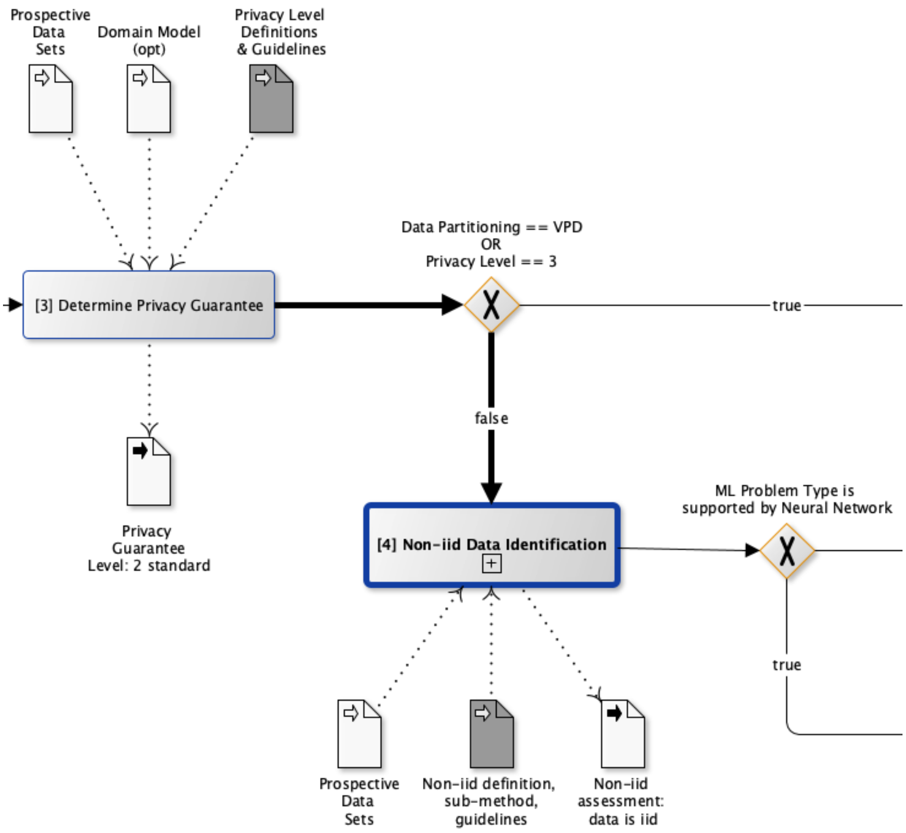

Prospective Data Sets
Domain Model (opt)
Privacy Level Definitions & Guidelines
[3] Determine Privacy Guarantee
Data Partitioning == VPD OR Privacy Level == 3
true
false
Privacy Guarantee Level: 2 standard
[4] Non-iid Data Identification
ML Problem Type is supported by Neural Network
true
Prospective Data Sets
Non-iid definition, sub-method, guidelines
Non-iid assessment: data is iid

Figure 5.3.4 - Fourth Execution Step of the Method in this Case Study

The objective of a future model is to predict mortgage lead times, i.e. the time it takes for a mortgage application to reach the completion status. To find relevant features in this domain model, all features which have an impact on the underlying business rules and business processes of the mortgage application process are assessed. Thus, features which have an impact on the mortgage application lead time. The included features are assumed to have the potential of being predictive features. For example, the application process will be different for a borrower who is self-employed compared to a borrower who has a stable employment contract at a company, which will have impact on the lead time. Other features which do not meet this criterium are disregarded. This assessment is done by an expert at the company, who has the knowledge about the business rules and business processes of the mortgage application. This domain knowledge is important in understanding the impact features have on a machine learning model.

Next, each relevant table in the domain model is discussed. Each relevant feature is briefly described and the reasoning why it is assumed to be a predictive feature is given.

Process

In the *Process* table, *PrimaryHandler* is a potential predictive feature for a machine learning model. This feature represents the employee who issues and administrates the mortgage application. It would have been a potentially predictive feature in a normal machine learning model. However, this cannot be used to determine differences in distribution or non-iidness in a Federated Learning setting, as the employees are unique for each bank. There is no overlap in *PrimaryHandler* instances in both data sites. Therefore, this feature is excluded in this assessment. Therefore, no features are used from this table.

No features from this table are extracted as relevant features.

ProcessType

ProcessTypeName is a potentially predictive feature, as this denotes the type of mortgage. For example, whether it is a first time mortgage or an extension. Due to the business rules, for example, an extension has to undergo fewer steps in assessing the risk than a first time mortgage. Therefore, this is an expected predictive feature.

Extracted features:
ProcessTypeName

Process Status
The column *Code* denotes the step this process is at in the overall application process. The application process, simplified, starts with the status code 'Start New Application' and ends with 'Binding Offer Sent'. Each of these have available a date-time stamp. By taking the differences between the date-time stamp of the beginning of the process and the end of the process, the lead time is constructed. The lead time will be used in the assessment as a calculated feature, which will constitute the predicted value of the model. The lead time feature is also the target feature of this model.

Extracted features (the target feature):
LeadTime

Application
This table denotes the mortgage application itself. It has several features which are assessed to be potentially predictive. First is the column *RequestType*, which denotes whether the borrower is self-employed or on payroll (two options: *OndernemerInPrive* or *Particulier*). This is assumed to have a impact on the lead time, as the mortgage application process has different (and additional) steps for self-employed borrowers. Next is the *LoanToValue* column. This column denotes the ratio between the borrowed amount and the value of the property, a value of 1,0 means that the entire amount is borrowed with no own money added from the borrower itself. As this feature impacts the risk of an application (1,0 having a high-risk), and mortgage applications are primarily used to standardize risk mitigation, this feature is assumed to have an impact on the lead time. A higher risk induces the need for extra risk management, which prolongs the application process. Next, the *RemainingPartial* feature denotes the principal amount of the mortgage. This is also assumed to have an impact on the process and therefore the lead times.

The Product_ProductNumber is the type of mortgage. This is assumed to be a predictive feature in a machine learning model, as the type of mortgage also impacts the application process. However, the actual categories are tied to the data site itself, there is no overlap between banks. Therefore, this feature cannot be used in a Federated Learning setting.

Extracted features:
RequestType
LoanToValue
RemainingPartial

Consumer
This table includes features about the consumer, i.e. the borrower of the mortgage. No relevant features are identified here. However, one mortgage application could be associated with multiple consumers/borrowers. The business rules imply that each borrower is evaluated separately. Therefore, the number of borrowers per mortgage application is also an impactful feature, as it prolongs the application process.

Calculated field:
Number of consumers per application

Income
GrossIncome denotes the gross income of the mortgage borrower. The *IncomeType* is the type of income the borrower has. Such as the employment contract (flexible or guaranteed hours), whether the borrower is an entrepreneur, or might be receiving their pension income already. Both are features expected to be predictive of a machine learning model, as they both have an impact on the business rules regarding the acceptance of the application.

Extracted features:
GrossIncome
IncomeType

Determined Market Value
Assessed to not be predictive, as the information is already incorporated in the loan to value feature in the Application table.

Interest Composition
InterestRatePeriod is the period the interest component is active. This is assessed to be a potentially predictive feature, as it has impact on the business rules in the application process.

Extracted features:
InterestRatePeriod

RealEstate
The property which is used as collateral on the mortgage. Here it is assessed that the features YearOfConstruction and PurchaseAmount are potentially predictive features, as both have an impact on the associated risk of the lender.

Extracted features:
YearOfConstruction
PurchaseAmount

Next, each of the extracted relevant features are input for the next phase, the testing phase. For practicality, only a subset of the extracted features is tested.

2. Non-iidness Three-Criteria Test
In this phase, each of the extracted features are tested to be non-iid. For this, three criteria are used. First an assessment based on Duan's (2019) specification of non-iidness in Federated Learning is conducted. It is checked, for the relevant features whether they have a (1) size imbalance, a (2) local imbalance, and a (3) global imbalance. For the local imbalance an assessment based on visualization via box plots is performed. As there are only two distinct data sets in this case study, only one comparison needs to be made.

First, it is checked whether there is a *size imbalance* between the two data sites of the two banks. This is done by assessing the difference in the number of rows in the primary tables Process and Process Status. The difference between data points generated between the two banks is a factor 4,2 and 3,3 respectively. Other features follow a similar difference ratio. This indicates that bank A has a larger impact on the model than bank B with traditional Federated Learning methods. However, although the there is no 1:1 ratio, the difference is still below an order of magnitude of 10. A size imbalance is not overly clear in this case, but probable. See Figure 5.3.5 for a visual representation of this, the x-axis represents the number of data points per attribute. The count values on the x-axis are redacted due to privacy considerations.

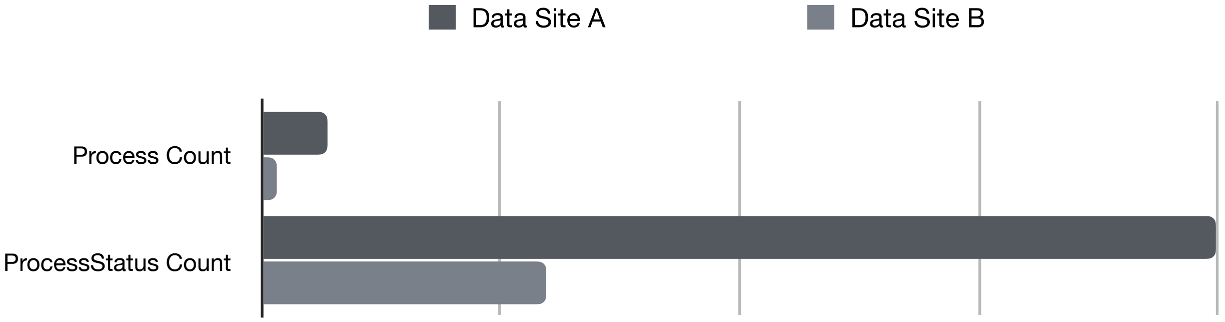


Figure 5.3.5 - Size Imbalance Check on Process and ProcessStatus Attributes

Second, it is checked whether there is a *local imbalance*, i.e. if there is a difference in data distributions between the data sets. As the user of this method is not an expert in statistics, the suggested two-sample statistical test skipped, and the assessment will be solely based on visual plots. These distribution plots are made for the feature PurchaseAmount.
When plotted in a histogram, both data sets follow a similar pattern. See Figure 5.3.6 and 5.3.7 for this. (Both the frequency and bin sizes are redacted, as it contains competitively sensitive information). Next to the distribution shape, the median values are also almost the same. The only differences are observed are at the minimum and maximum values. However, when excluding only 6 data points, which can be considered outliers, the minimum and maximum values of both data sets are similar again. Therefore, it is assessed that both data sets follow a similar data distribution regarding PurchaseAmount. This means that the houses the clientele of both banks purchase is roughly the same regarding purchase prices. This indicates that, when combined, the data sets will be iid.

Lastly, *global imbalance*, i.e. *class imbalance*, in both data sites is checked. Class imbalance occurs when features with categorial values are heavily skewed towards one class, such that one class has a large number of example and the other only a few (Japkowicz & Stephen, 2002). In this case study, relevant extracted features with categorial values are tested for class imbalance. The features which have categorial values are RequestType from the Application table, IncomeType from the Income table, and ProcessTypeName from the ProcessType table.

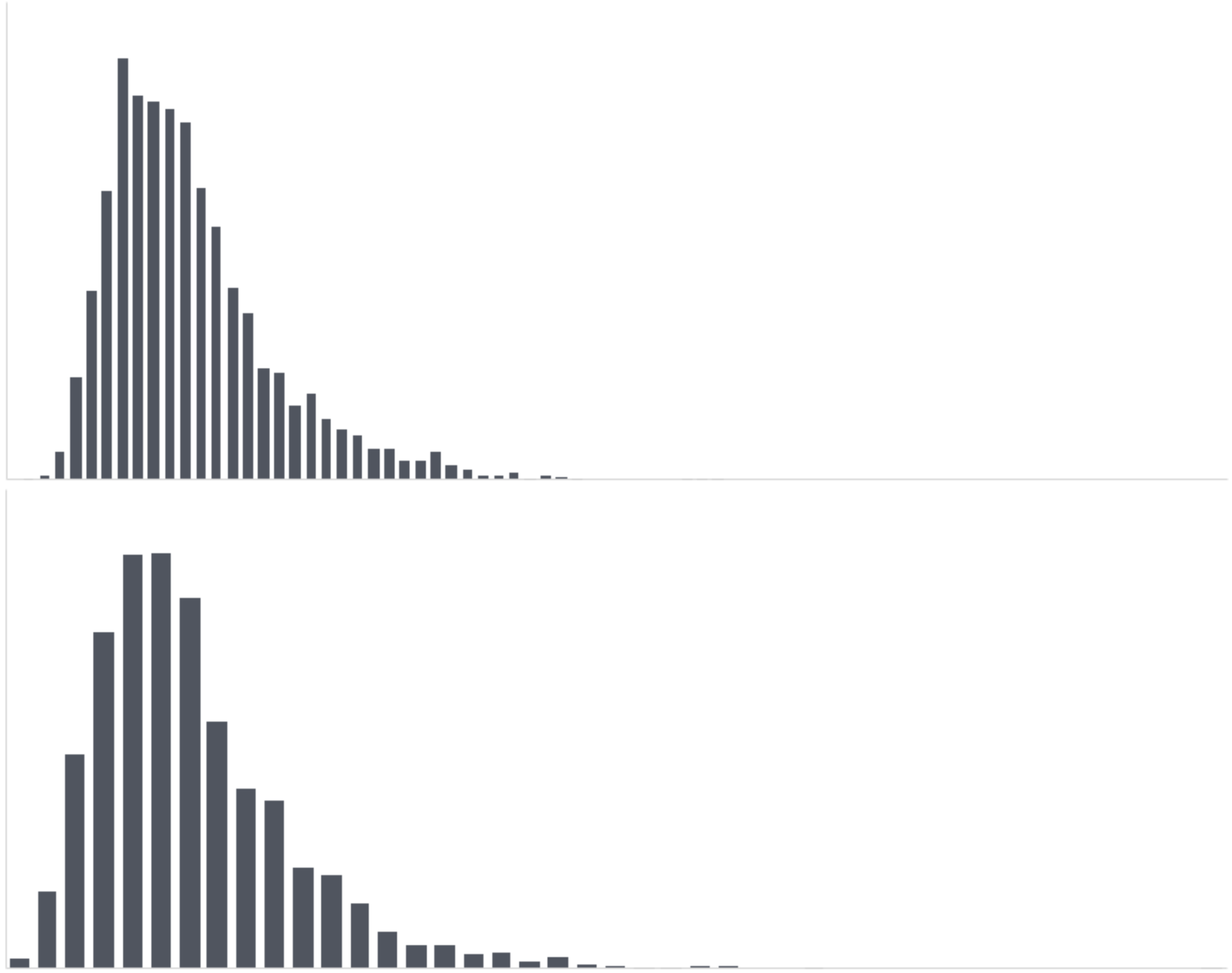


Figure 5.3.6 and 5.3.7 - Histograms of the PurchaseAmount Attribute in Data Set A and Data Set B

The results are obtained by querying the two databases (via a group by clause on the categorial attribute). The results are visualized in Figure 5.3.8 for the feature RequestType, which has 2 categorial values, and Figure 5.3.9 for the feature ProcessTypeName, which has 5 categorial values. The count values are represented on the x-axis. RequestType does not have a class imbalance, as the differences in count for the categorial values exists, but is not overwhelming. For ProcessTypeName the differences are clearer. There is a heavy skew for the categorial value 'Acceptance', constituting almost all values. For some of the categorial values, the count is even so low that it barely shows on the figure. A class imbalance is certainly observed here. However, due to the nature of the feature, the model can be scoped to only train on 'Acceptance' applications. IncomeType is not visualized, as it contains 36 categorial values. Here, no class imbalances are observed here.

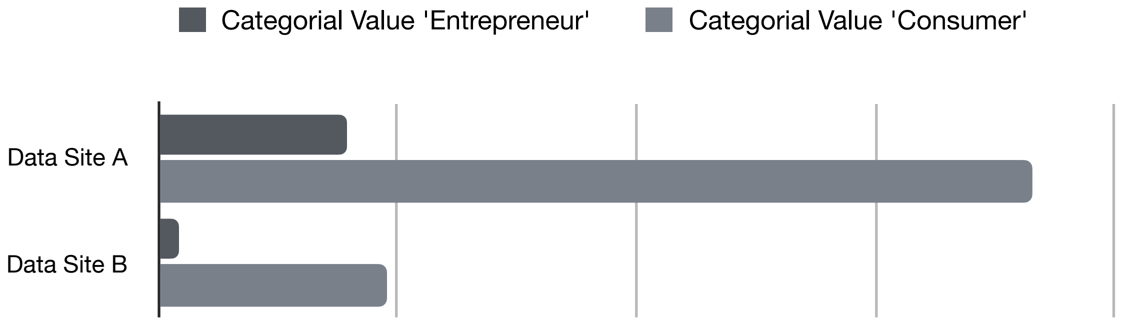


Figure 5.3.8 - Class Imbalance Check for Attribute RequestType

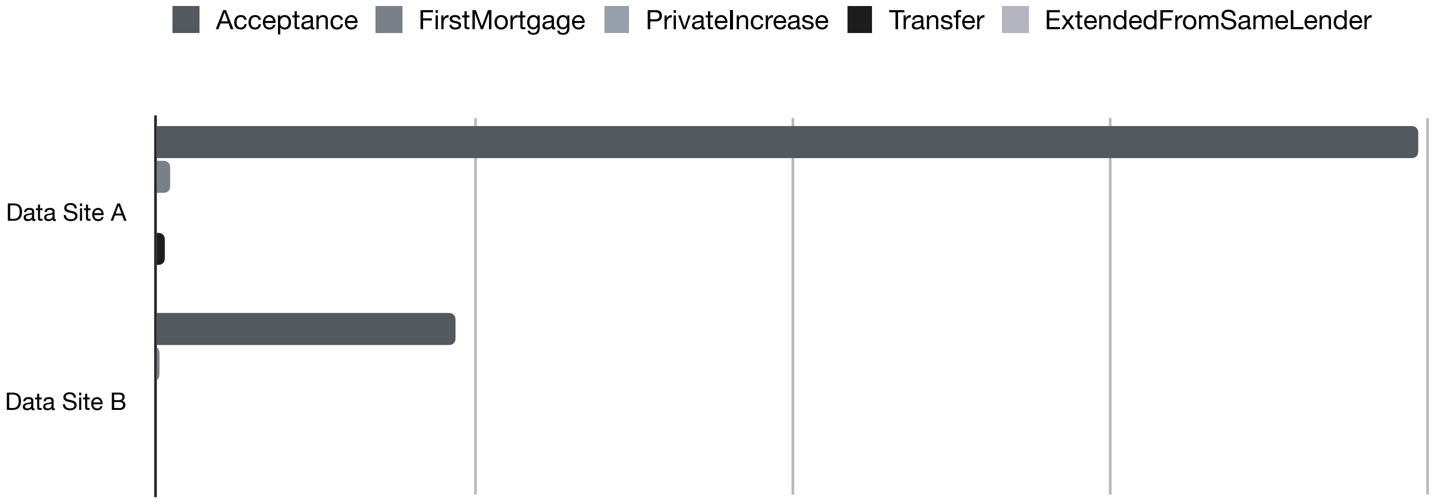


Figure 5.3.9 - Class Imbalance Check for Attribute ProcessTypeName

3. Assessment

Given the tests performed in the previous steps, the results need to be interpreted to come to a final assessment whether the combined data set will be non-iid or iid.

It is assessed that there is no substantial size imbalance between the data sets, which indicates that the combined data sets will likely be iid. Next, the data distribution of the feature PurchaseAmount follows a similar data distribution among the data sets. It is therefore unlikely that there will be a local imbalance which can cause non-iidness when combining the data sets. Lastly, there is no class imbalance found for 2 of the 3 assessed features. As the feature ProcessTypeName does contain a class imbalance, it is recommend to limit the scope of the Federated Learning model to only work with 'Acceptance' data. Given these points, it is concluded that the data sets, when combined, will likely be iid.

5.3.5 Predictive Performance Trade-Off Phase

The next phase in the method is 5: the Predictive Performance Trade-Off phase, as the earlier identified machine learning problem type is a linear problem. This problem type is assessed to be supported by a Neural Network, and therefore the true path of the decision node is followed.

Here the Federated Learning methods Performance lookup table in Appendix F.1 is used as provided input. As the data partitioning type is HPD, the privacy guarantee level is 2, the Machine Learning problem type is supported by a Neural Network, and the data is assessed to be iid, the resulting most applicable Federated Learning method is Federated Averaging (FedAvg) of McMahan. Which is the output of this phase and also constitutes the end of the method.

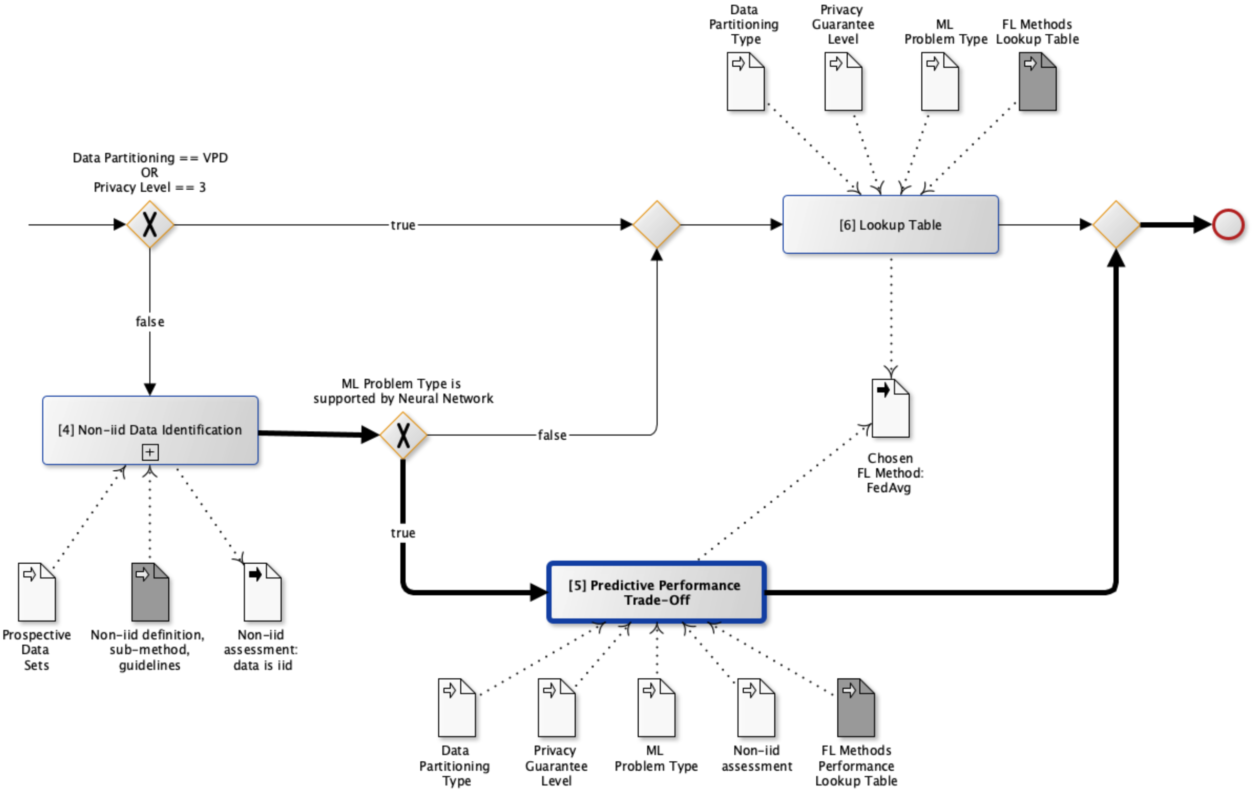

Data Partitioning Type
Privacy Guarantee Level
ML Problem Type
FL Methods Lookup Table
Data Partitioning == VPD OR Privacy Level == 3
true
false
[6] Lookup Table
[4] Non-iid Data Identification
ML Problem Type is supported by Neural Network
false
Chosen FL Method: FedAvg
true
[5] Predictive Performance Trade-Off
Prospective Data Sets
Non-iid definition, sub-method, guidelines
Non-iid assessment: data is iid
Data Partitioning Type
Privacy Guarantee Level
ML Problem Type
Non-iid assessment
FL Methods Performance Lookup Table

Figure 5.3.10 - Fifth and Last Execution Step of the Method in this Case Study

5.4 Conclusion

In this chapter the designed method was validated to work in a real-world scenario by means of a case study. The case study was executed at Topicus, which has a problem context fitting to the method's scope and goal. The case study has shown that the method can be applied to a real-world case. Also, in this way a practical example of the execution of the method is given, which further enhances the understanding of the method.

6. Evaluation - Case Study Demonstration

This chapter presents the evaluation of the proposed method, and thereby constitutes the fifth phase of the DSRM. The goal of this phase is to evaluate the utility of the method, as an artifact's assessment should be based on utility according to Wieringa (2014). This evaluation is done by conducting a workshop at a company which fits the problem context of this study. In this workshop the method and its usage by means of the case study is presented to experts. Afterwards, the experts are asked to evaluate the utility of the method by means of a survey. The evaluation survey questions are based on the UTAUT model by Venkatesh et al. (2003), the Unified Theory of Acceptance and Use of Technology model. This model provides a way to assess the likelihood for a new system to be accepted successfully at an organization and therefore fits the purpose of this evaluation.

In this chapter, first, the UTAUT model and its relation to this study are explained in more detail. Next, the workshop set-up is presented. Lastly, the results of the survey are described and discussed.

6.1 UTAUT model

The UTAUT model is a unified theory which aims at predicting the usage intention and actual usage of some system. The model formulates four determinants (Performance Expectancy, Effort Expectancy, Social Influence, Facilitating Conditions) and four moderating factors on those determinants (Gender, Age, Experience, and Voluntariness of Use). The interrelationships between these determinants and factors are visualized in Figure 6.1.1. Each determinant and factor in the model is measured by means of standardized survey questions to prospective users (Venkatesh et al, 2003). Due to its relevant characteristics, this model is used in determining the utility of the designed method. Next, each of the determinants are explained in more detail.

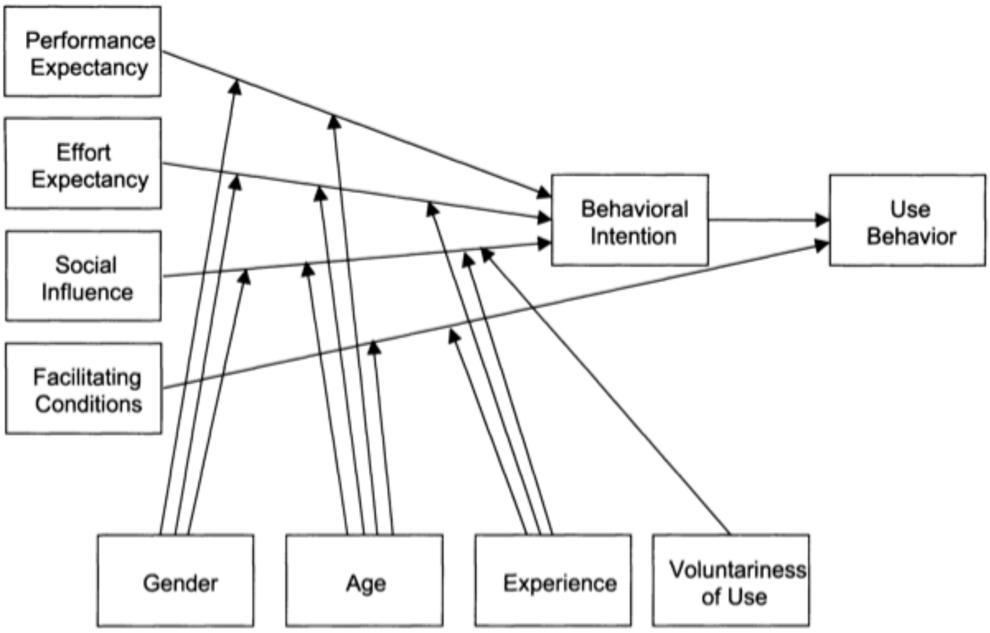


Figure 6.1.1 - UTAUT Research Model (Venkatesh et al, 2003)

Performance Expectancy
Performance expectancy is the degree to which a prospective user believes that using a system will help them gain attain gains in job performance (Venkatesh et al, 2003). It is the strongest predictor of behavioral intention. Performance expectancy is moderated by the factors gender

and age. The effect of this determinant will be larger for men and especially for younger men (Venkatesh et al, 2003).

The survey items that are used in this study to measure this determinant are listed in Table 6.1.1.

Table 6.1.1 - Survey Items to Measure Performance Expectancy

Item code from original study	Survey item
U6	I would find the method (or: way of working) useful in implementing a Federated Learning system
RA1	Using the method enables me to accomplish tasks more quickly
RA5	Using the method increases my productivity when implementing a Federated Learning system

These are the questionnaire items that are used and validated in the original study of Venkatesh et al (2003). The item code denotes the item code from the original study. The items are altered to be as specific as possible for this use case, substituting the generic terms of system with method, and specifying the company name where the generic term organization is used. The items will be measured on a 1-5 Likert scale, with the values: strongly disagree, disagree, neutral, agree, and strongly agree.

Effort Expectancy
Effort expectancy is the degree of ease associated with the use of the system (Venkatesh et al, 2003). It is moderated by gender, age, and experience. Those indicate that the effect of effort expectancy is larger when someone is female, is younger, and has little experience (Venkatesh et al, 2003).

The survey items that are used to measure this determinant are listed in Table 6.1.2.

Table 6.1.2 - Survey Items to Measure Effort Expectancy

Item code from original study	Survey item
EOU3	My interaction with the method would be clear and understandable
EOU5	It would be easy for me to become skillful at using the method
EOU6	I would find the method easy to use
EU4	Learning to use the method will be easy for me

Social Influence
Social influence is the degree to which an individual perceives that important others believe they should use the new system. It is moderated by all factors, gender, age, experience, and voluntariness of use. The effect of this determinant is stronger for women, particularly older women, and especially in mandatory settings when experience is low (Venkatesh et al, 2003).

The survey items that are used to measure this determinant are listed in Table 6.1.3.

Table 6.1.3 - Survey Items to Measure Social Influence

Item code from original study	Survey item
SN1	People who influence my behavior think that I should use the method
SN2	People who are important to me think that I should use the method
SF2	I expect my seniors/management at Topicus to be helpful in the use of the method

Item code from original study	Survey item
SF4	In general, I expect the organization to support the use of the method

Facilitating Conditions
The last determinant is facilitating conditions. It is the degree to which an individual believes than an organization and technical infrastructure exists to support the use of this system. It is the only determinant that does directly influence use behavior and not the behavioral intention. Next is is moderated by the factors age and experience. The effects are especially stronger for older workers with more experience (Venkatesh et al, 2003).

The survey items used to measure this determinant are listed in Table 6.1.4.

Table 6.1.4 - Survey Items to Measure Facilitating Conditions

Item code from original study	Survey item
PBC2	Topicus will provide the resources necessary to use the method
PBC3	I have the knowledge necessary to use the method (given the guidelines and input documents provided)
PBC4	I have the resources necessary to use the method
PBC5	The method is compatible with other systems or ways of working I use
FC3	Support from an individual/a group, or a service is available when problems are encountered using this method

Next to the main determinants, also survey items are used to measure the intermediate determinant Behavioral Intent and the moderating factors: gender, age, experience, and voluntariness of use. These items can be found in the full survey in Appendix C.

6.2 Workshop Set-Up

The method is demonstrated to experts in the field of Data Science at the company Topicus. These experts were selected by means of their job description, relating to data science or machine learning. A workshop is organized where the participants are introduced to the earlier performed case study. Here the problem context is drawn, essential concepts are explained, such as Federated Learning, non-iid data, and other definitions, and the method is explained. To help the participants prepare, the case study description, the method description, alongside a list of definitions are sent to them. The researcher was available to provide further explanation and context on the problem statement and the nature of the study, not for explaining the method more than described in the provided document. This document is a slightly altered version of Chapters 4 and 5.

At the end of the workshop a survey is held where the aforementioned UTAUT model survey items are presented. The survey questions, adapted for this study, can be found in Appendix C. The following is altered. All questions including a negative form were turned to a positive form question, so the answers are on the same scale. Also all instances of the term 'system' in the questions are substituted by the term 'method', as this represent a more specified term of system that is being investigated. Also where a generic organization is mentioned, it is substituted by the specific company name.

6.3 Results

In this section the survey results are described and discussed. First, an analysis on the respondents' descriptive statistics is given, alongside the moderating factors. Second, the results of the UTAUT survey items are analyzed and discussed. The full survey results can be found in Appendix G.

Descriptive Statistics

The workshop and the corresponding survey was executed by 5 respondents. To check the selection criteria of having experience in the field of data science and/or machine learning, some questions about (work) experience are asked. These also serve to measure the moderating factor Experience.

Respondents were asked about their experience, by posing questions about their formal education, work experience, and familiarity with the concepts of machine learning and data science. All respondents are well educated, having a college or university degree in a related field. Also all respondents have a job description relating to the experience criterium of this study. The median work experience bracket is 3-5 years work experience, which is not particularly high. Next respondents were asked about their familiarity with the concepts of data science and machine learning. With a Likert-scale of: no experience, beginner, medium, senior, expert, which are then translated to numerical values of 1-5. The average for data science is 3,4, which is slightly above the medium value. The average for machine learning is lower, it is 2,8, which is slightly below the medium value. Meaning that the respondents are less familiar with machine learning. Next, the respondents frequency of working with a machine learning (related) concept is on average 3,0, which represents 'sometimes'. All in all, the experience level is assessed to be of medium value. Visual representations of these results can be found in the Figures 6.3.1 and 6.3.2 below.

Figure 6.3.1 - Work Experience Distribution (n=5)

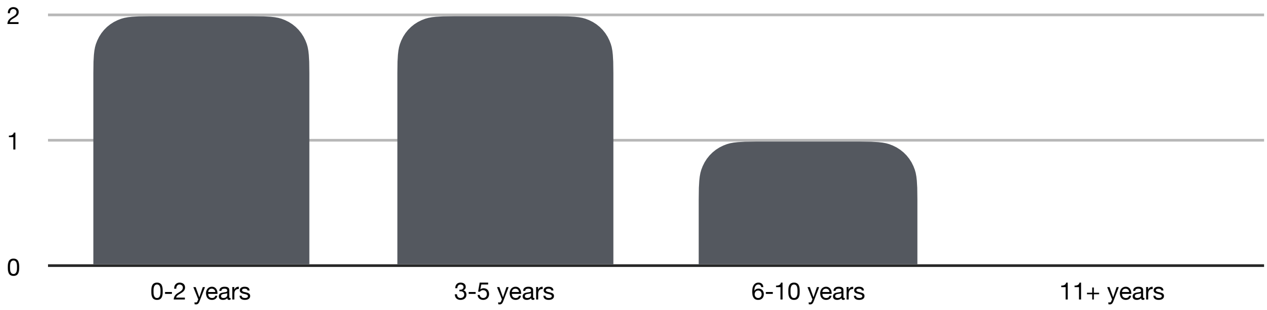
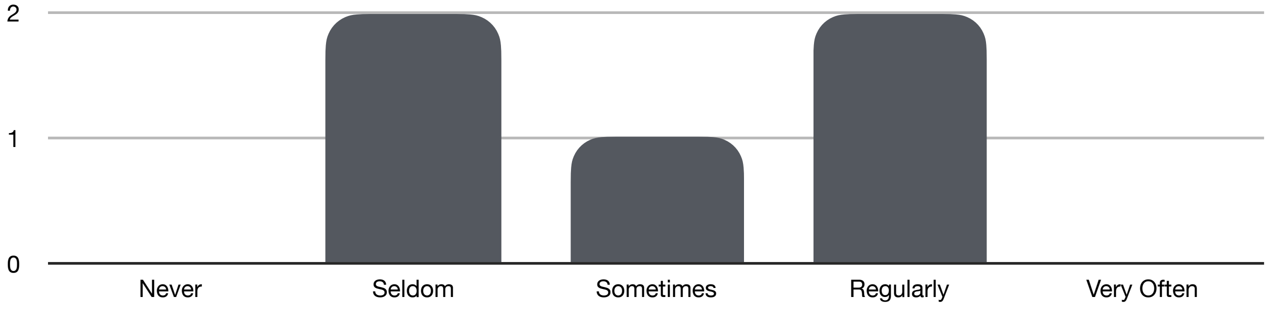


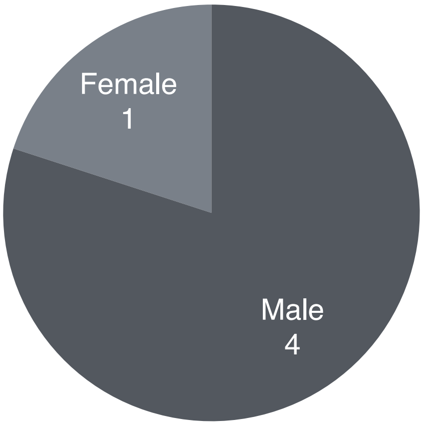
Figure 6.3.2 - Frequency of Working with Machine Learning Concepts Distribution



The survey was filled in by 4 males and 1 female. This shows an overwhelming majority of the respondents being male. This has implications on the determinants performance expectancy, effort expectancy, and social influence, as each of them are moderated by gender. The effect on performance expectancy is higher for males, while the effect on effort expectancy and social influence is higher for females. See figure 6.3.3 for a visual representation of this.

Next is age. The average age of the respondents is low. All respondents were between the age of 21 and 35 years old. The same number of respondents were in the age brackets of 21-25 years old as 31-35 years old. See Figure 6.3.4 for the distribution. This has implications on all determinants, as all are moderated by age. This means that the expected effect of performance expectancy and effort expectancy will be higher, and lower for social influence and facilitating conditions. This is favorable, as will be presented later in Table 6.3.1, because the average scores

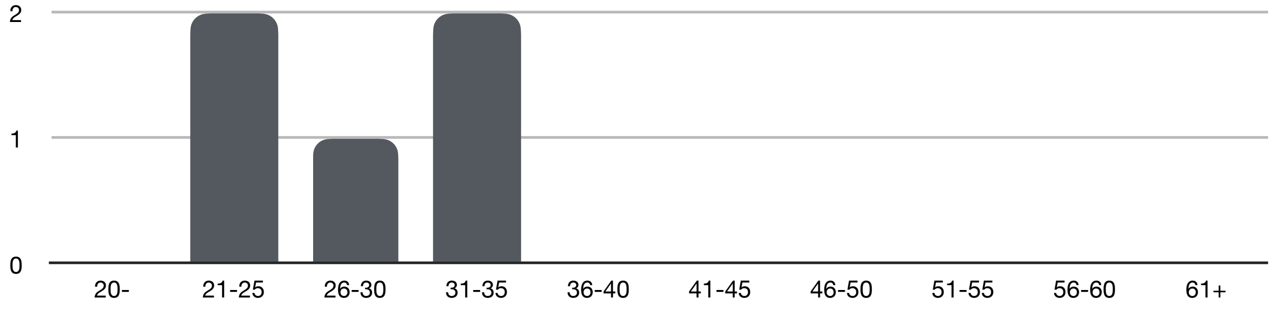
Figure 6.3.3 - Male to Female Ratio



for performance expectancy and effort expectancy are the highest, and social influence and facilitating conditions the lowest. In other words, the highest scoring determinants get leveraged up and the lowest scoring determinants have a lower than normal impact on the use behavior, all due to the demographics of the respondents. This of course, only translates to a real-world setting if the respondents are a representative group, which was not the setup for this survey.

UTAUT Determinants
Next, the aforementioned survey items for measuring the determinants in the UTAUT model are given. Each of these are measured on a 5-point Likert scale, with the values: strongly disagree, disagree, neutral, agree, and strongly agree. These values are, for the sake of analysis, transformed to numerical values ranging from 1 to 5. Aggregated and summarized data of the 5 respondents are given in Table 6.3.1. These results and the impact is has on the evaluation of the method are discussed next.

Figure 6.3.4 - Age Distribution



Performance Expectancy and Effort Expectancy show the highest average (mean) values of 4,33 and 4,45. An interesting remark to be made is that, looking at the raw survey data, both experienced and less experienced respondents mark the method as high in performance expectancy. This can be derived from the low standard deviation of 0,30, and the minimum and maximum values of 4 (agree) and 5 (strongly agree) in performance expectancy. This could indicate that the method both provides enough depth and perceived productivity gain for more experienced users, while still being easy enough to be understood and provides enough guidelines for less experienced users. The least experienced respondent (also a female), however, scored an average of 3,75 on effort expectancy, while higher values are attributed to more experienced respondents. This is in line with the theory, which suggest that the effect on performance expectancy is higher among users which are younger, female, and have little experience (Venkatesh et al, 2003). This shows that the method does indeed have a higher chance of actually being used by more experienced users, but less experienced users are predicted to still to use this method, as they still rate it favorably.

Table 6.3.1. UTAUT Evaluation Survey Summarized Results
(n=5. PE: Performance Expectancy. EE: Effort Expectancy. A: Attitude Towards Technology. SI: Social Influence. FC: Facilitating Conditions. BI: Behavioral Intent.)

Determinant	Item Code	Average	Min	Max	St. Dev.
PE	U6	4,8	4	5	0,45
	RA1	4,2	4	5	0,45
	RA5	4	4	4	0,00
PE avg		**4,33**	**4,00**	**4,67**	**0,30**
EE	EOU3	4	3	5	1,00
	EOU5	4,6	4	5	0,55
	EOU6	4,6	4	5	0,55
	EU4	4,6	4	5	0,55
EE avg		**4,45**	**3,75**	**5,00**	**0,66**
SI	SN1	3,8	3	4	0,45
	SN2	3,6	3	4	0,55
	SF2	4,2	3	5	0,84
	SF4	4	3	5	0,71
SI avg		**3,90**	**3,00**	**4,50**	**0,63**
FC	PBC2	4,4	4	5	0,55
	PBC3	4,4	4	5	0,55
	PBC4	3,4	3	5	0,89
	PBC5	3,4	2	4	0,89
	PC3	3,4	3	4	0,55
FC avg		**3,80**	**3,20**	**4,60**	**0,69**
A	A1	4,6	4	5	0,55
	AF1	3,6	2	5	1,14
	AF2	4,2	3	5	0,84
	Affect1	4,4	4	5	0,55
A avg		**4,20**	**3,25**	**5,00**	**0,77**
BI	BI1	4,2	4	5	0,45
	BI2	4	3	5	0,71
	BI3	4,4	4	5	0,55
BI avg		**4,20**	**3,67**	**5,00**	**0,57**

Of the four determinants, Facilitating Conditions has the lowest average score of 3,8. Essentially scoring between a 'neutral' and an 'agree' level. This could indicate that the experts are not familiar with the concepts behind the method and the company does not provide enough support in that case. Which is not surprising, given the fact that Federated Learning is a new research area. This also further confirms the need for the literature review part of this study; showing that there is indeed a need to consolidate information and resources regarding Federated Learning. Also, to further improve the score on this facilitating conditions determinant, a more comprehensive guideline could be written to give the experts better tools at hand, as item PBC4 - relating to having available the right resources - had a low score. Also, item PBC5 was particularly low, with an average score of 3,4, indicating that the way of working in practice at the company does not directly fit with the method's way of working.

Social Influence has an average score of 3,8, which is the second lowest score, just behind facilitating conditions. This is largely due to the low average score on item SN2 of 3,6, stating that superiors oblige the usage of the method, i.e. a mandatory environment. As Federated Learning is largely new, it could be the case that the potential of Federated Learning has not reached their attention, and therefore are also not starting (investigative) projects on it. The scores have higher values for the items citing a hypothetical situation where a Federated Learning project would start. So, essentially, the average score of social influence is lower, because there is no mandatory environment yet. The respondents give higher scores in the hypothetical situation if a (mandatory) project were to be undertaken.

Next, the intermediate determinant Behavioral Intent, influenced by the determinants performance expectancy, effort expectancy, and social influence, show a similar score. The average of the latter combined being 4,23, and of the former 4,20. As these determinants are related and the three mentioned determinants influence behavioral intent, they should score similarly. If the two would deviate greatly, it could indicate that the survey was not properly understood or executed. As they do show similar results here, it provides extra confidence in the validity of the results. As all of these determinants show a high score, it is expected that the method will be used in a real-world setting, as users see utility in this method.

All in all, in general, the results are positive and further confirms the utility of the designed method by evaluating that the method would indeed be likely to be used in the real world by experts. Although the results are positive overall, the method could still be improved by better adapting to the practical way of working at companies and providing more resources and guidelines, as the score on facilitating conditions is the lowest. These two lower scoring determinants are, however, of less importance than the higher scoring because the respondents are predominantly male and young. These descriptive statistics leverage the effect on the high scoring determinants performance expectancy and effort expectancy, while having a lowing effect on the low scoring determinants social influence and facilitating conditions. This effect is only applicable if the survey respondents are a representative sample for other organizations.

7. Case Study - Local Neural Network & FedAvg Implementation

In the previous chapters the method to choose a Federated Learning approach has been designed, presented, and evaluated. During the evaluation by case study the method was executed. Its resulting choice for a Federated Learning algorithm for the case study company was Federated Averaging (FedAvg) by McMahan et al. This choice has been solely based on insights from the literature study. But, does the result of the method also translate to be a practical implementation in the real world? To answer this, this chapter presents a case study to implement the resulting Federated Learning algorithm at the same company with the same problem context where the method was executed.

The goal of this is to show the applicability and practicality of Federated Learning at an organization which fits the earlier stated problem context. Earlier, the method itself was evaluated for utility by stakeholders, not on the results of the method. Therefore, this supplemental case study serves as a (partial) empirical validation on the results of the designed method. Next, this also presents the opportunity to perform a comparison between locally trained Machine Learning models, a joint Centralized approach, and a Federated Learning model, as has been identified gaps in research. To accomplish this in a structured and academically-backed manner, this chapter will follow the research methodology of CRISP-DM; a leading methodology for doing data science-related research (Chapman et al., 2000; Kurgan & Musilek, 2006).

This case study will not attempt to build an exhaustive and fully optimized model due to time constraints. The scope here is to perform a pilot study which shows the feasibility and practicality of implementing the decided Federated Learning algorithm. Also it serves to compare FedAvg, local Machine Learning models, and a Centralized approach to each other. As FedAvg is Neural Network based, only a Neural Network model has been chosen, to provide a fair comparison.

7.1 Business Understanding

This section describes the first phase of CRISP-DM, the business understanding phase.

This case study is executed at the same company as the previous case study in Chapter 5. Consequently, the same company description and problem statement are used. These are not repeated here. Only additional information needed for the purpose of this CRISP-DM phase is presented.

7.1.1 Business Objective

The business objective of this case study is synthesized to be the following. At Topicus there is an apparent wish to provide their clients (mortgage lenders, i.e. banks) with better information regarding the mortgage application process, which in turn will be used to optimize the business processes at these clients.

One aspect of this process optimization is regarding the lead time of the mortgage applications. The lead time is defined as: the time it takes for a mortgage application to reach an (practical) end status, counting from the start of the process. Topicus' role is to help the banks provide better customer service by giving an indication of the expected lead time of an application, to provide better insight in upcoming workload, and to help pave the way to shorten this application process. To accomplish this, Topicus wants to predict the lead times of the mortgage application process. In addition, Topicus is exploring options to create a global model by means of Federated Learning, combining information from different banks, in order to improve the individual models at each back.

Business objective: Predict mortgage application process lead times based on customer and mortgage application data to provide better services to the clients of the business.

7.1.2 Situation Assessment

To accomplish this Topicus has provided the researcher with resources and tools. First of all, domain experts are readily available for informal interviews and other forms of knowledge gathering. Next, the researcher has access to the official documentation of the STO application, the business processes generating the data for this application, and the data models. Lastly, the researcher has been given access to real-world data from two of Topicus' clients: two banks operating in The Netherlands. Some attributed of the data have been anonymized, such as all information that could identify an individual consumer. Hardware and software to access and interface with this data has also been provided by means of a remote desktop which hosts a MS SQL database.

The scope of this case study is to show the applicability and practicality of predicting lead times at Topicus with Machine Learning and Federated Learning, not to create a fully optimized model. This is due to the initial set scope for this study and the limited time frame. This case study should be seen as a pilot study which the company can further iterate on.

Assessed situation: A short pilot study to assess the applicability and practicality of Machine Learning and Federated Learning. Necessary resources to accomplish this task are available.

7.1.3 Data Mining Goal

From the business goal, a more technical data science-related goal is formulated. To accomplish the stated business objective two local Machine Learning models are developed at each bank. Next, a Federated Learning is developed to utilize the combined information of both banks. Also, a centralized model is trained to make a comparison with the Federated model. As the objective is to predict lead times this problem can be categorized as a regression problem in the context of Machine Learning. Only models that support this type of problem should be selected. For this purposes of this study only a Neural Network is chosen, as the resulting FedAvg algorithm is Neural Network based. This makes the comparison more valuable.

Data Mining Goal: Develop a local Machine Learning model to predict mortgage application lead times at each bank and compare the results to a Federated Learning model on the joint data sets.

7.1.4 Tool Selection

Chapman et al. (2000) suggest to decide upon which toolset to use early on in the process. Therefore, tools are already selected in this stage. Also, it is worth noting that the choice of a toolset is not of major importance, other than that it fits the purposes of the set goals. The toolset choice is mainly based on the personal preference of the researcher, as the differences between the tools are minimal (Meka and Patil, 2015). The choice of tools are documented for reproducibility.

For data extraction, SQL will be used as the interfacing language, as the data are being stored on a MS SQL database server. The provided data model to purposefully execute these queries is available. Next, SQL will also be partly be used for data cleaning. As SQL has practical limitations, also the programming language Python will be used, with the Pandas library, a Python data analysis library (https://pandas.pydata.org). To develop both the Machine Learning and Federated Learning models, Python is again used. In addition to this, TensorFlow, the open source machine learning platform (https://www.tensorflow.org) is used as an additional library for Python. TensorFlow is used to develop machine learning models more systematically and in a quicker way. Also, TensorFlow has the *FederatedAveraging* (*FedAvg*) algorithm and a way to simulate Federated Learning on one device built-in, in the TensorFlow Federated library.

7.2 Data Understanding

In this section the actual data and data model of the STO data warehouse are described, put intro context by domain knowledge, and the quality is assessed.

7.2.1 Data Description

The data made available for this case study is production data from two instances of the STO database at two Dutch banks. The data are anonymized such that no data is traceable to individual consumers. Also, the two obtained data sets are from the same time period, so they are comparable. As already described in the case study in Chapter 5, there is a data size imbalance between the two data sets in the order of magnitude of a ~4:1 ratio.

Next the STO database is described. STO is a data warehouse that consolidates and summarizes operational data from the application Force (the product suite used for administrative purposes for the mortgage application process). Each bank hosts its own instance of their own STO database, these are thus strictly separated.
At Topicus, extensive documentation on STO is available. The most important part of this documentation is the data model and its supplementary description. A visual representation of the data model can been found in Figure 5.1.2. This data model is described next.

The main table of this data model is regarded to be the Process table, i.e. the main fact table. The process status changes can be deduced by the associated ProcessStatus label, which includes the status code and a timestamp. From this the lead time can be constructed. Also included in this table is the Primary Handler, the unit that is responsible for processing this application. Next, the Team table denotes the team at the company that is responsible for this process. The Whitelabel table is used to differentiate between different brands present within one company.

Next is the table ProcessType. This denotes the distinction between 'Acceptation'/'Acceptatie' and 'Administration'/'Beheer' processes. The Acceptation processes should be seen as the main processes of the mortgage application process. The Administration process denotes supplementary processes to alter an existing mortgage contract after it has been initialized. These do not follow the standard process path and are therefore not representative. The scope of this case study is, thus, set to only include the Acceptation processes.

The table Application denotes the mortgage application itself. One of the most interesting features in this table is LoanToValue. It is a features which incorporates the ratio between the mortgage amount and the market value of the piece of real estate. When this ratio is high it indicates a higher risk for the bank, and could therefore prolong the process with extra steps to mitigate this risk.

Associates tables to Application are Consumer, the applicant of multiple applicants of the mortgage, and its associated table Income. The latter includes the feature GrossIncome, the income each applicant receives. The height of this income partly determines the maximum height of the mortgage amount this applicant can receive and can thus be a relevant feature for the model. It is also important to note that one application can have multiple consumers associated to it (in this data set only 1 or 2 consumers per application), and the consumer can have multiple incomes. For the purposes of this case study the gross incomes will be aggregated by taking the sum of all associated values, as this denotes the total income that is covering the mortgage payments.

Lastly, other noteworthy associated tables to Application are HandlingParty and RealEstate. The latter denotes the data from the piece of real estate this mortgage application is about. It includes many features describing the real estate, such as the year of construction, the purchase amount, building type, the address, and more. The addresses are, however, anonymized in this data set. Next, the Handling Party denotes either an internal or external 'mortgage advisor', which can act as an intermediary between the consumer and the bank.

The ConstructionAccount tables are not included in this case study, as these tables store data from the 'Adminstration’ processes, not the main 'Acceptation' processes, which is out of scope for this study.

Next to official documentation provided by the company, knowledge discovery with several domain experts of the STO application at Topicus has been conducted. This knowledge discovery took place iteratively during the data understanding process, and has also been applied at a later stage during data cleaning and modeling activities. The purpose being to understand data features better, but also the data itself. For example, domain knowledge has been of utmost importance during the data cleaning stage, as outliers could not be interpreted without this knowledge. One example on this was the high occurrence of the year 1900 in the feature of year of construction, which apparently is a default value if the real year of construction is not yet known in the application process.

All in all, the STO data model is vast, well documented, and includes a lot of possible features that can be included in a Machine Learning model to predict the lead time.

Lead Time definition

In the business understanding phase it has been determined to predict the lead time of the mortgage application process. However, defining what the lead time actually constitutes to is not a trivial question. The mortgage application process goes through several process statuses during its lifetime. A simplified general overall process overview is shown in Figure 7.2.1. It should be noted that this is a stark simplification of the real mortgage application process, only the most practically relevant process statuses are included in this simplified model. There, all states a process can go through and the flow of possible state changes is shown. The table in the domain model ProcessStatus keeps track of these state changes by storing, among others, the process status code and a timestamp the change occurred.

All mortgage application processes start with 'Start New Application' ('StartNieuweAanvraag'), and has three possible end points, based on whether the mortgage was accepted and granted or not ('Passed' / 'Gepasseerd', 'Application Declined'/'Aanvraag Afgewezen', and 'Application Canceled'/'Aanvraag Geannuleerd'). To scope this case study all applications that were denied or canceled are excluded.

The lead time could now be defined as the time it takes to reach 'Passed' from the points of 'Start New Application'. However, the phases after 'Binding Offer Sent' show a large degree of uncontrollable variability. These phases represent the time it takes for the mortgage contract to actually go into service. For example, a consumer might already want a mortgage contract set, while the house (s)he wants to buy is still under construction. The mortgage will only be legally activated and set to 'Passed' after the actual legal transfer of the deed and the transfer of funds have taken place, which can be months later. Meanwhile, the bank does have to allocate personnel as no significant work is done in the meantime. As the bank has little influence over this process and the bulk of the administrative work has to be executed in the phases leading up to 'Binding Offer Sent', this status should represent the end of the application process from the bank’s perspective. This also fits the purpose from a business objective perspective; to better allocate resources. As the bank does not have any influence on this, and does not have to allocate personnel in the last process states, it has no added value.

During data exploration it has also been observed that the process can reach 'Binding Offer Sent' multiple times per application: 816 times in Data set A, and 201 times in Data set B. This happens, for example, when the consumer declines this specific offer and wants another one. Then the process is not terminated, but instead, some steps are executed again before it reaches the status 'Binding Offer Sent' again. It has been chosen to only regard the first occurrence of the status 'Binding Offer Sent', not the subsequent ones. In the second iteration of this, the standard process flow is not strictly followed, which will introduce variability.

The ***lead time*** is thus defined as: the time it takes for the process to reach the first occurrence of the process status 'Binding Offer Sent' from the moment the process started, marked by the process status 'Start New Application'.

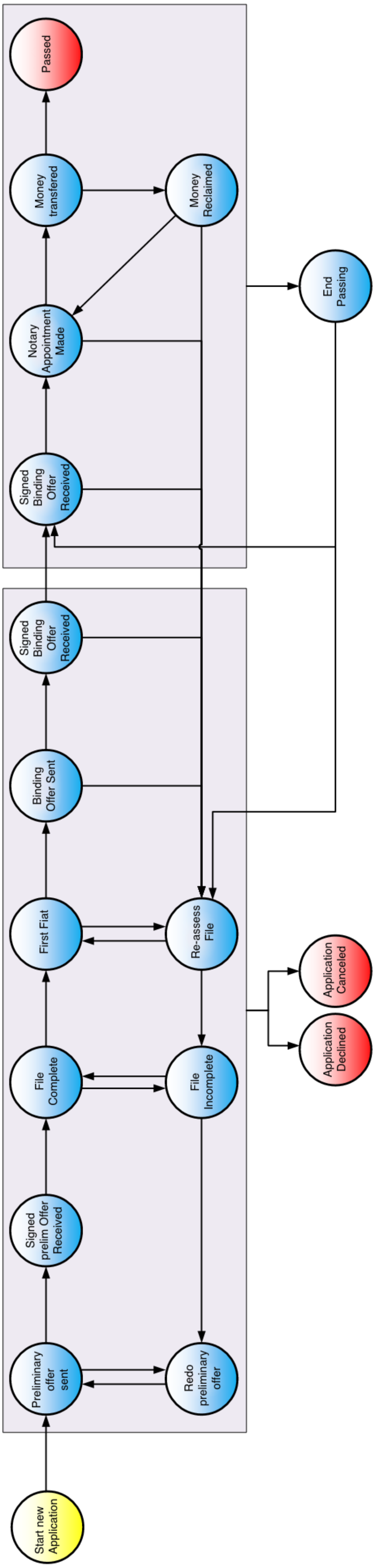


Figure 7.2.1 - Mortgage Application Process State Phases Diagram

7.2.2 Data Quality Assessment

In this section the data quality of the provides data sets is assessed, as part of CRISP-DM.

The data sets provided are originating from the STO database, which is a data warehouse, summarizing data from the operational application Force. As it is a data warehouse, the data are already cleaned to a higher standard than the operational database. The quality of the data, regarding missing values, data formatting, consistency, and outliers, is assessed to be of a high standard. However, some caveats are to be mentioned, discovered during data exploration.

There were some occurrences where the data was filled in, but did not reflect the real-world value. For example, the feature year of construction in the table Real Estate, denoting the year the house is built, has an unusually high occurrence of the value 1900. 15% of the real estate data points had the value 1900 for the attribute YearOfConstruction. After obtaining some domain knowledge, this is explained by the fact that this is a standard value mortgage lenders fill in when the real year of construction of the piece of real estate is not (yet) known. Another instance of the table real estate is initialized when the real construction year is known. As the data keeps track of historic values the newest value is chosen in this study.

There were also some occurrences of missing values in the following fields. First of all, it is assessed how many times the values for YearOfConstruction are 0 or null. This amounts to merely 1 occurrence. This data point is excluded from the data set. Next, the it is assessed how many times the LeadTime is less than 10 hours, which are outliers and looking at the domain knowledge, would be too short, it has: 8 occurrences of this in Data set A, and 4 occurrences in Data set B These missing values are also excluded from the data set. Next, there were some instances of the combined income of consumers being unusually low (below 5000). With domain knowledge at hand, the incomes were too low to obtain a mortgage with any practical value. There were 24 occurrences of this, these data points were excluded.

7.3 Data Preparation

In this phase of CRISP-DM relevant features for the model are selected, data is cleaned, and pre-processed to be suitable for input in the modeling phase.

7.3.1 Feature Selection

Feature selection is a preprocessing step in developing machine learning models which removes irrelevant and redundant data, in order to increase the learning accuracy and improve results (Khalid et al., 2014). Domain knowledge is often used for this step (Kuhn and Johnson, 2013), but there are also more numerical method available. Such as a correlation-based approach, where a good feature set contains features that are highly correlated with the target feature, but uncorrelated with each other (Hall, 1999). Correlated features with each other should therefore not be added. In addition, feature selection is even necessary for training some machine learning models such as Neural Networks. For Neural Networks, the predictive performance can actually decrease when non-predictive features are added. While some other machine learning models are, however, not affected by this (Kuhn and Johnson, 2013).

In this study the following features were selected by means of domain knowledge and an assessment of the correlation between the target feature and the other features. The graphs for the latter can be found in Appendix I. The former is already explained in the data description section. The following features are selected: RequestType, PrimaryHandler, HandlingParty, LoanToValue, Remaining Partial, YearOfConstruction.

Next the the already available features, combined features can also be constructed. This is called feature engineering and can be advantageous to the predictive capacity of a model (Kuhn and Johnson, 2013). For example, a ratio of two other features can be constructed. In this case study, several engineered features are used that were not readily available by looking at the database's features alone. This is done through data exploration in combination with domain knowledge.

In this case study five features were engineered. The first is the target feature Lead Time itself, as mentioned earlier this feature had to be calculated by means of looking at the process states. The

SQL query that calculated this feature can be found in Appendix H.1. Next, a feature called Number of Consumers is constructed. This represents the number of applicants one mortgage application has. The rationale for adding this feature is that an application with more than one consumer will take longer, because each consumer's have to be individually delivered and checked. The next feature is Sum Gross Income, which reflects the combined income of the applicants if there are multiple. Also a feature Sum Principal is added to reflect the mortgage loan height. It is summed because a mortgage can be divided into multiple parts.

The last calculated feature was added during the modeling process and significantly improved the predictive performance of the model. This feature is called Overlap. Overlap represents the number of other simultaneous mortgage application processes being in progress at the start of that specific mortgage application process. The rationale for this feature is that the capacity of a bank is limited, and when a higher number of other applications are in progress, there is less time for the new process. In addition, the feature Overlap shows a stronger correlation than the other selected features, as can be seen in Appendix I. The SQL query extracting this information can be found in Appendix H.2. The full query extracting all features is stated in Appendix H.3.

The resulting list of selected features for input in the model is given in Table 7.3.1.

Table 7.3.1 - Selected Features from the Combined Data Set

Feature	Example value	Type	Min	Max	Mean	St Dev
RequestType	"Consumer"	Categorial	2 categorial values {Consumer, Business}			
PrimaryHandler	"08R56017"	Categorial	96 (55 + 41) categorial values			
HandlingParty	"DC493B-F48D-48FE409"	Categorial	1246 (773 + 473) categorial values			
Overlap	251	Integer	28	7.007	1.466,6	906,1
LoanToValue	0,6521	Float	0,0	1,8	0,7	0,2
RemainingPartial	301.000	Integer	0,0	1.000.000	243.444	144.646,1
YearOfConstruction	1975	Integer	1005	2022	1970	40,6
SumPrincipal	295.000	Integer	0	12.405.800	553.664	542.699,2
NumberOfConsumers	2	Integer	1	4	1,4	0,5
SumGrossIncome	76.000,67	Float	5000	3.554.133,0	50.392,6	63.306,4
LeadTime (target)	750	Integer	10	5749	717,3	465,6

As the improvements in predictive performance by adding more features in the modeling stage plateaued, no more features were selected in this case study. This is due to the diminishing returns on the improvement of the model and the time constraints. For improving the model results in the future it is recommended to consider other (constructed) features.

7.3.2 Data Cleaning

As mentioned in the data quality assessment, the quality of the data is already high. However, there are still some missing values and outliers present in the data sets. These are excluded in the following way.

All instances where the Lead Time was less than 10 hours were excluded. This amounted to the exclusion of 9 data rows. All data rows where the Sum Gross Income was below 5000 euros were removed, 368 instances. Also all instances were no consumers were linked to were excluded, 2 instances. Some instances of YearOfConstruction were set in the future. These data points were kept, as they represent houses still under construction, but the mortgage application process was already finished.

7.3.3 Data Pre-Processing

After the right features are selected and the data are cleaned, the data is pre-processed. Here, the data is transformed to be in the right format for input in a machine learning model. This includes label encoding, one hot encoding, normalization, and formatting the data.

One Hot Encoding

The features RequestType, PrimaryHandler, and HandlingParty are categorial features. They have a limited number of textual values. As Machine Learning models require their input to be solely numerical, this information needs to be encoded. All features are of nominal value, not ordinal, because they do not have an order embedded within the possible values. Therefore, these features will be encoded by the One Hot Encoding technique. To quote Harrag and Gueliani (2020), One Hot Encoding is: "a process by which categorical variables are converted into a form that could be provided to machine learning and deep learning algorithms to do a better job in prediction. It is a group of bits among which the legal combinations of values are only those with a single high (1) bit and all the others low (0)". Here each categorial value is represented in their own newly constructed feature, with either a value of 1 or 0, denoting whether this data point belongs to this class or not. For the feature RequestType one feature is satisfactory, as it only represents two categorial values. For the latter two features, only the top 25 most occurring categories are chosen (and one 'other' feature), to not include too many features to the model.

Normalization

Kurt and Johnson (2013) state that for Neural Networks (and Support Vector Machines) the data need to be scaled, normalized, and centered to obtain better results for the model. This is called normalization or feature scaling. It basically transforms all value ranges to the same or a similar scale, for example 0,0 to 1,0. The advantages of feature normalization on Neural Networks is also confirmed by Shanker et al. (1996), who compared several normalization techniques. The normalization is done feature wise, looking at the minimum and maximum values per feature. A common formula for normalization is: y = (x – min) / (max – min), according to Witten et al. (2016). Another technique is to apply standardization with y = (x – mean) / standard deviation (Witten et al., 2016). Which is a more statistical approach.

Looking at the data in this case study, there are large differences in the ranges. While the LoanToValue feature ranges from 0,0 to 1,8, SumPrincipal ranges from 0 to over 12 million. This is almost a difference of 7 orders of magnitude. Therefore, this data set will be scaled to represent only values between 0 and 1. In this case study the MinMaxScaler function from the sklearn Python library is used. It applies the normalization formula of y = (x – min) / (max – min), and transforms all values to a scale of -1 to 1 (or 0 to 1 if there are no negative values).

7.4 Modeling

In this part the modeling phase in CRISP-DM is described.

7.4.1 Choice of Model

There are a vast number of machine learning models available. Each belonging to a certain category of problem. Bishop (2006) names several machine learning models, such as: Linear Regression models, Bayesian Linear Regression, Neural Networks, Kernel methods, Support Vector Machines, and other classification and kernel methods. Looking at the type of problem, which is prediction of the lead time, a model supporting a regression problem would suffice. As already mentioned the scope of this case study is to develop a Neural Network. A local Neural Network model is trained for both banks, and a combined Federated Learning model is trained on both data sets.

7.4.2 Loss Function & Optimizer

How does a Neural Network model learn? It uses two main functions: a loss function, which calculates how far the preliminary training predictions of the model are from the truth, and an optimizer, which updates the parameters of the model based on a gradient to steer it towards a lower loss.

The task at hand in this case study is a regression task which can produce arbitrary values (i.e. not merely between 0 and 1). For this type of task Chollet (2018) suggests to use the *MSE* (Mean Squared Error) or *MAE* (Mean Absolute Error) as the loss function. The MAE loss function represents the total summed difference over all training examples: the absolute difference between the predicted value and the truth value. This loss function will calculate the performance of the model while training, and the objective is to minimize this loss function. The used loss function for this case study will be the MAE.

The minimization task of this loss function is done via an optimizer. This optimizer determines how the network will be updated based on the loss function (Chollet, 2018). The most popular optimizer used in Neural Networks is gradient descent (Ruder, 2016). Gradient descent minimizes the objective function J(θ), θ representing the model's parameters. It updates the parameters of the model in the opposite direction of the gradient of the objective function with respect to the parameters. The optimization task stops when a (local) minimum is reached (Ruder, 2016).

However, gradient descent is computationally expensive. Therefore, many models use Stogastic Gradient Descent (SGD) which performs a parameter update for each training example, instead of for the whole training data set (Ruder, 2016). Next to SGD, Ruder (2016) names several derived alternatives to gradient descend:

- SGD, stogastic gradient descent;
- Momentum;
- Nesterov accelerated gradient;
- Adagrad;
- Adadelta;
- RMSprop;
- Adam;
- AdaMax;
- Nadam.

Of these algorithms RMSprop, Adadelta, and Adam are very similar and do well in similar circumstances (Ruder, 2016). Kingma and Ba (2014) show that the bias-correction in Adam slightly outperforms RMSprop towards the end of optimization as gradients become sparser. Adam can therefore be considered as the better choice. Adam will be used as the optimizer for this case study. Adam, Adaptive Moment Estimation, is a method that computes adaptive learning rates for each parameter. In addition, it store an exponentially decaying average of past squared gradients (Ruder, 2017).

In addition, each of the listed optimization functions are a form of stogastic gradient descent. Therefore, they only use a subset of the training data, instead of the whole training data set. For this, the model has a tunable variable called batch size, which sets the number of stogastic training examples per optimization step. Goodfellow et al. (2016) state that larger batches provide a more accurate estimate of the gradient, but with less than linear returns. They suggest a batch size range from 32 to 256, based on the size of the model and the computing power available. In this case study, the used batch size is set to 32.

Both the chosen loss function MAE and the optimizer Adam are built into the TensorFlow library used in this case study.

7.4.3 Parameter Initialization

For a Neural Network model to start learning it's initial parameters, or weights, need to be set as it needs an starting point. This initialization is, however, not trivial, as most algorithms are strongly affected by the initialization technique (Goodfellow et al., 2016). Setting the weights to 0 is also not a good choice, as this involves a symmetry problem. To quote Goodfellow et al. (2016): "perhaps the only property known with complete certainty is that the initial parameters need to 'break symmetry' between different units. If two hidden units with the same activation function are connected to the same inputs, then these units must have different initial parameters. If they have the same initial parameters, then a deterministic learning algorithm applied to a deterministic cost and model will constantly update both of these units in the same way." (Goodfellow et al., 2016). Therefore, it is preferred to randomly initialize the parameters of the model; in this way the symmetry can be broken. In used the TensorFlow in this case study, the parameters are initialized randomly.

7.4.4 Neural Network Architecture Design

In this section, the architecture design of the neural network is described: its number of nodes, input, output, and hidden layers, and the activation function.

The choice for a neural network design seems, however, semi-arbitrary, and heavily tied to the problem at hand. To quote Chollet (2018): "picking the right network architecture is more an art than a science; and although there are some best practices and principles you can rely on, only practice can help you become a proper neural-network architect." Therefore, an iterative approach will be used in this case study, based on trial and error.

There are, however, some constants in the design: the input and output layers. The input layer has a node for each input feature. In this case study amounting to 8 nodes and however many one hot encoded features are encoded from the 2 other categorial features. The output layer constitutes 1 node, it outputs a real value; the prediction for the lead time. In this case study, after many iterations, the neural network design has 3 hidden layers, with respectively 20, 10, and 10 nodes.

One last design question for Neural Networks is the activation function. Without an activation function the model would only be able to learn linear transformations, which is too restrictive for many tasks (Chollet, 2018). There are many activation functions available. The most popular is the ReLU activation function (Ramachandran et al., 2017), the rectified linear unit. This activation function will be used in this case study and is included in the TensorFlow library.

7.4.5 Convergence

A neural network is trained by means of iterations, also called epochs. During each iterations several steps are conducted. First, the input training data is propagated forwards, which results to an output, the lead time prediction in this case. Next, an algorithm called backpropagation is used to compute the gradient of the loss function with respect to the current weights of the model (Chollet, 2018). In this way the model learns. Training a neural network may require many epochs to reach an optimum. A heuristic that can be used is to use this many epochs where the learning rate (the decrease of the loss function) plateaus (Chollet, 2018; Goodfellow et al., 2016).

To mitigate the possibility of reaching a sub-optimal local minimum the model is trained several times on different train-test splits. The fact that both data sets show similar (loss) results and a similar number of epochs to reach convergence can be seen as an indication that a proper (local) minimum is reached.

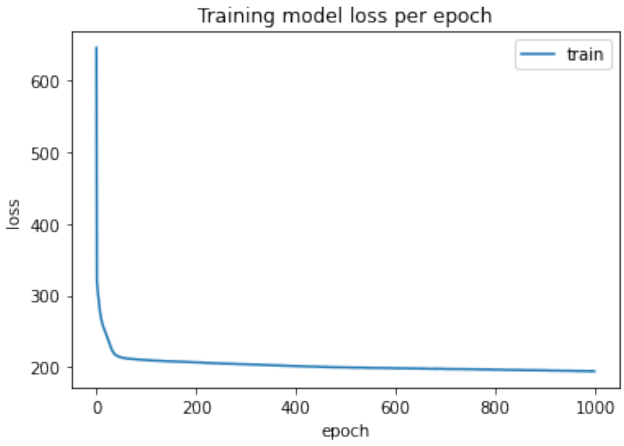


Figure 7.4.3 - Training (MAE) Loss per Epoch on Centralized Combined Data Set A & B

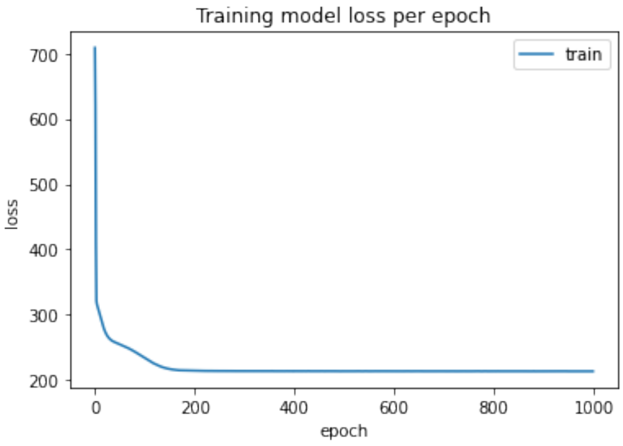


Figure 7.4.4 - Training (MAE) Loss per Epoch for the Federated Learning Model

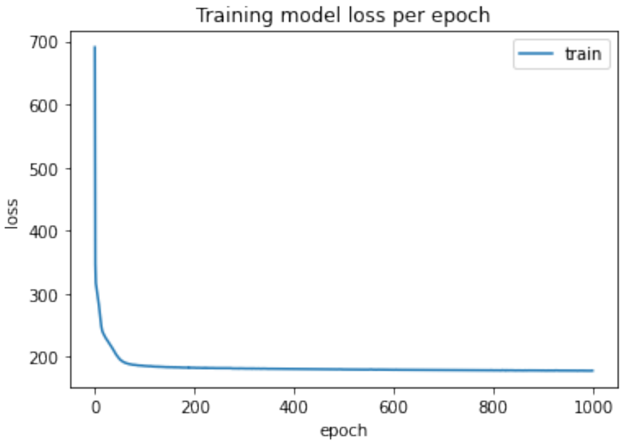


Figure 7.4.1 - Training (MAE) Loss per Epoch on Data Set A

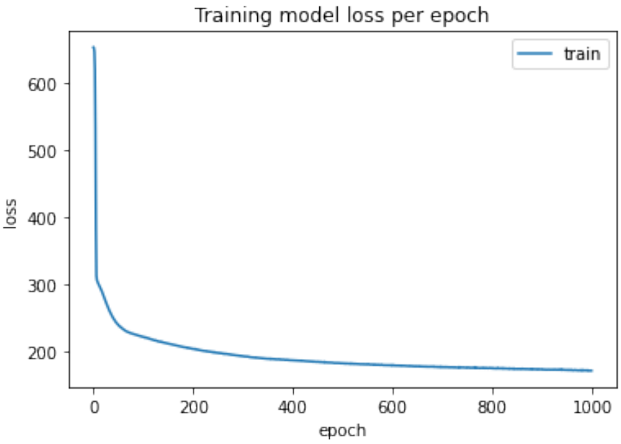


Figure 7.4.2 - Training (MAE) Loss per Epoch on Data Set B

In figures 7.4.1, 7.4.2, 7.4.3, 7.4.4, graphs are presented that show the convergence of each local Neural Network model for each data set, central approach on the combined data set, and the Federated model. The number of epochs set is 1000. Here it clearly shows that there is a steep decline in the MAE on the training data at the beginning, and it starts to plateau after about 40 to 200 epochs, only providing diminishing returns afterwards. At 1000 epochs the decrease in the MAE on the training data is minimal or has even totally flattened. Also, no significant decrease in the MAE on the test set is observed after 300 epochs.

7.5 Evaluation

This section presents the evaluation of the model and also the last practical phase of CRISP-DM.

7.5.1 Evaluation Method

In order to evaluate the models, the original objective is revisited: to predict mortgage application lead times. In order to achieve this, the data need not only be predictive of the already available data, but even more so on future data. This cannot be accomplished by merely training the model on the heuristic of attaining a low loss on the training data. This would make the model too specific, i.e. an overfit (Shmueli and Koppius, 2010). To make the model more robust, the model is evaluated on different data than the training data.

Chollet (2018) names two ways this can be accomplished:
(i) by a simple hold-out split;
(ii) by k-fold cross-validation.

In the first technique, the simple hold-out split, the data is split into a training set and a test set (usually an 80:20 split). Here, the model is trained each iteration on the training set, but evaluated on the test set. To show the predictive performance results that are more likely to be generalizable, as the model has not seen this data before (Chollet, 2018). The second technique works in the same manner, but randomizes and repeats this process multiple times. Here, the data set is split in k subsets, where one of these subsets is used as the test set and the remainder as the training set. This is performed k times such that every k-th subset is used as a test set once. After these k iterations, the mean of the results are taken (Friedman et al., 2001). In this study, due to the to the fragmented nature of the data sets in the Federated approach, instead of a k-fold cross validation, a simple hold-out split is used with an 80:20 split.

The metric used to evaluate the model on will be the MAE of the difference between the truth value, the actual lead time in the test set, and the predicted lead time by the model. This is done for every data point in the test set and then averaged to get the MAE for this model. This evaluation is done for every model: the local Neural Network model on both data sites, and on the Federated Neural Network model on the data of both data sites. In addition, a centralized model is trained where the data of both data sets are combined, and this result is compared to the Federated model.

7.5.2 Evaluation Results

In Table 7.5.1 the predictive results on both the test set as the training set are presented for each of the four models: the local models on data set A and B, the centralized model on the combination of data sets A and B, and the Federated model. Here the MAE for the lead time (in hours) is given.

Table 7.5.1 - Predictive Performance Results

	MAE Lead Time (in hours) on Test set	**MAE by guessing with average lead time**	**Difference**	**(MAE loss on Training set)**
Local Model Data Set A	**193**	337	*144*	178
Local Model Data Set B	**213**	316	*103*	171
Centralized Model	**205**	335	*130*	181
Federated Model	**209**	335	*126*	188

For data set A, the MAE, mean absolute error, of the predictions of the model on the test set is 193 hours (8 days). Comparing this to just guessing the average lead time for this data set (a MAE of 337), it performs 144 hours better. The results for the smaller data set B are slightly worse with a MAE of 213 hours on the test set, which only performed 103 hours better than guessing the average lead time.

The centralized model and the federated model perform very similarly. This is expected, as in the literature review it was concluded that Federated Learning algorithms do not have a significantly worse predictive performance than a centralized model, but instead are very similar. This is another empirical confirmation to this insight from the literature study.

Next, both models perform worse than data set A, but better than data set B. Data set B is about four times smaller than data set A. Therefore, it could be that due the extra data available, a better model can be built. The worse performance than the local model on data set A can be explained by the fact that data set A already had plenty of data, and the added data makes the model less specific for this data set. This could prove to be advantageous for the generalizability of the model for future mortgage application. Also, the differences between the models are rather small, thus a strong conclusion cannot be made about this.

Next, the practical implications are discussed. The models have a MAE on the test set of between 193 to 209. Although it is much better than just guessing the average, of between 103 and 144 hours better, the error margin is still high for its purposes. Practically, it means that the average prediction error is still around 8 days, on an average lead time of a bit more than a month, making it not that suitable for real-world usage. Therefore, there are still improvement to be made to make the model suitable for practical usage. Looking at the scope of this case study, it is not surprising, as the main objective was to show the applicability of the chosen Federated Learning algorithm in the real world, not primality building the most predictive model possible.

8. Conclusion

This chapter first presents conclusions by reflecting on the earlier posed research questions and their results. Then a discussion on the results is given. After that the study's main theoretical and practical contributions are stated, alongside its limitations. Lastly, possible directions for future research are suggested.

8.1 Conclusions

This research set out to answer the following main research question:

What is an appropriate methodology to help organizations choose the most suitable Federated Learning method given their situation regarding data-related characteristics and privacy requirements?

To answer this main research question, the problem is broken up into several pieces, which together aim at answering the main research question. Conclusions on the research on a per sub research question basis are discussed next.

RQ1: What is the definition of Federated Learning according to the literature?

Federated Learning is still a novel research area, the formal definition was only set in 2017 and most papers are just published last year (2019). Federated Learning started as a means of attaining more data input for machine learning models at different distinct data sites, while still preserving privacy. The main advantage: more data input is being available, and more data, under the general assumption that the combined data set is iid, means that a better machine learning model can be developed. To tackle the privacy constraints, almost all Federated Learning methods do not share the local data with another data site; the data does not leave its origin. Instead, partial local models are trained and merely the model (or parameter) updates are shared with a central server which aggregates these results.

This study synthesized the information present in 10 studies about Federated Learning and deduced common characteristics found. From this, a clear and usable definition of Federated Learning is formed: *Federated Learning is a form of distributed machine learning where a global model is trained on a central server utilizing multiple separate heterogenous edge devices, while still preserving privacy by not permitting the data to leave their origin devices.*

RQ2: What Federated Learning methods exist in the literature?

Federated Learning has a myriad of distinct methods present in the literature. These methods can be categorized in two ways. The first are proto-federated learning methods, which are in a research line started before Federated Learning had a formal definition. Most of these methods are standard machine learning algorithms adapted for distributed usage with some privacy mechanism present. Looking at these studies, each developed their own custom method and tackled challenges like networking, consolidation, and communication strain on their own, effectively reinventing the wheel each time again. Next are the second category of Federated Learning methods, which all started with the Federated Averaging (FedAvg) method of McMahan et al (2017). FedAvg is also the most commonly mentioned method. Most papers after this develop their new Federated Learning method based on FedAvg or compare their method to FedAvg, which is seen as a baseline. A comprehensive table of all methods identified in the literature can be found in Chapter 3.2.

RQ3: What are the main differentiating characteristics of the Federated Learning methods found in the literature?

To purposefully show relevant differences between the Federated Learning methods identified in the previous research question, 5 differentiating characteristics are devised for Federated Learning. *The differentiating characteristics* of Federated Learning methods are defined as*: characteristics of Federated Learning methods (i) which may limit options or impact the desired outcome regarding a organization's data-related characteristics and privacy considerations, i.e. are* *relevant to the to-be-designed method, and (ii) have variation in implementation among the Federated Learning methods, i.e. not all Federated Learning methods have the same implementation regarding this characteristic.*

There are 5 differentiating characteristics devised for Federated Learning. They are deduced from the characteristics and information about the various Federated Learning methods identified earlier. These 5 differentiating characteristics are:
1. Data partitioning;
2. Underlying machine learning models;
3. Privacy guarantees;
4. Performance (accuracy, predictive performance);
5. Non-iid data support.

These 5 differentiating characteristics are used as the basis for the designed method, as these characteristics serve to make an informed decision among the various Federated Learning methods. In this way, a structured comparison can be made based on the scope and goal of the study.

The first three differentiating characteristics are investigated in this research question and their values for each Federated Learning method are deduced. These results can be found in table format in ***Appendix F.2***. The remaining two differentiating characteristics are not static; they show some interrelationship where a trade-off can be made. Therefore these are investigated separately, in research question 4 and 5.

RQ4: What are the differences in predictive performance among Federated Learning methods?

This research question's aim is to investigate the differences in predictive performance among the found Federated Learning methods. It is the fourth differentiating characteristic.

An overarching view in Federated Learning research is that most methods (custom ones and FedAvg) achieve similar predictive performance results compared to a centralized approach. Meaning that the privacy-preserving mechanisms of Federated Learning do not significantly impede model accuracy. Also, multiple researchers claim that their Federated Learning method outperforms a local-only approach. Lastly, among a few Federated Learning methods directly compared to each other, the best known results are that of the FedAvg method, when compared in an external study.

It is important that a clear distinction be made between non-iid and iid data sets, as many standard Federated Learning methods do not perform well on non-iid data. Adapted Federated Learning methods who work better in non-iid settings are developed. Of these methods Zhao et al's (2018) method shows the greatest improvement in accuracy of up to 55%, but does, however, require data sharing. The Astraea method does not require data sharing, but only shows a modest increase in accuracy of about 6%. A more detailed investigation of the effect of non-iid data on Federated Learning is included in the next research question.

These conclusion, however, should be stated alongside some considerations. First of all, it became apparent that many studies in this SLR both introduce both their newly-developed method and also a comparison of this method to another. This could lead to potential bias, as the researcher who created the method, which is fine-tuned for a particular data set, is now also comparing these results to another method. For a proper comparison an external study should be conducted, where the researcher compares several Federated Learning methods from other researchers on the same data set(s) and with the same objective. This study, therefore, considers these external studies (such as that of Nilsson et al, 2018) to be of higher quality and weights these higher in the results. In this way potential bias can be reduced.

Second of all, there is a discussion within the Federated Learning domain whether traditional methods such as FedAvg work well on non-iid data. The original authors claim that working well on non-iid data is one of the pillars of their Federated Learning method. However, many studies published later on demonstrated both theoretically and empirically that predictive performance is indeed negatively impacted in non-iid settings. The reason behind this will be investigated further in the next research question below.

The best performing Federated Learning methods identified in the literature based on both non-iid support and privacy are summarized in table format in Appendix F.1.

RQ5: What is the effect of Federated Learning's consolidation technique of utilizing multiple data sites on predictive performance?

To better understand the relation between the consolidation technique Federated Learning uses and the effect it has on predictive performance this research question was formulated.

First, the mechanism of consolidation in Federated Learning is investigated. Consolidation here means the mechanism how Federated Learning combines and weights the information from the locally trained models. This mechanism is found to be a relatively simple (or naive) approach: it merely aggregates all data points from each data site with the same weight, or it aggregates each data site with the same weight in some cases. The implication of this is that some data sites may have more influence over the trained model than others (or in the second case that some data points may have more influence). This can also be represented in formula form. The goal of Federated Learning is to minimize the following objective function:

$$\min_{w} F(w)\,, \text{ where } F(w) := \sum_{k=1}^{m} p_k F_k(w)$$

where m is the total number of devices, w is the input parameter (i.e. input training data), F_k is the local objective function, and p_k specifies the relative impact of each device. This relative impact is usually set as: $p_k = 1/n$ or $p_k = n_k/n$, where n is the total number of training examples, and n_k is the number of training examples of a particular local device k (Li and Smith, 2019).

It has been shown in several empirical studies that Federated Learning (in an iid context) achieves very similar performance to a centralized approach. A centralized approach is when the data of each data site is transferred to a central server and a traditional Machine Learning model is trained. This means that Federated Learning is advantageous to use regarding its privacy preservation characteristics, but does so by not significantly impeding the predictive performance of the overall model.

The most interesting part is when this finding is linked with the problems seen with non-iid data. Although early research claims Federated Learning works well with non-iid data, many empirically-backed studies proved otherwise. The effects of non-iid data show an accuracy loss of up to 55% for some data sets. The way studies which tackled this problem is by changing the way the consolidation of multiple data sources is conducted. Many Federated Learning methods created for non-iid usage try to solve this accuracy loss by balancing the data between data sites, others share data among data sites, and others change the weights attributed to some of the local models. Concluding, a link has been made between the way consolidation is done in Federated Learning (a simple and naive approach as of right now), and the negative impact on model performance in non-iid contexts.

RQ6: What is an appropriate method for identifying non-iid data sets in the context of Federated Learning?

This research question sets out to identify and devise a supplemental method to identify non-iid data sets in a Federated Learning context. As the distinction between a non-iid and iid data set is of high impact for the choice among Federated Learning methods, a theoretically-backed sub-method in identifying non-iidness is of added value to the method.

Non-iidness is traditionally defined as a property of one data set, where the data draws are not independent and/or data is not identically distributed. Therefore, a different approach needs to be devised for a Federated Learning setting. As in Federated Learning there is not one data set, but multiple separated data sets which by principle cannot be combined. Thus non-iidness cannot be assessed by means of an evaluation on one data set. Instead, it should be assessed by comparing each potential new data set incrementally to the existing data sets. For this, the methodology of Rabanser et al. (2018) is used and altered. The proposed method consists of three steps:

1. Dimensionality reduction, i.e. feature selection, to reduce the number of features to a manageable number with only relevant features;
2. A three-criteria test on non-iidness; and lastly,
3. A final assessment, where conclusions are drawn based on the findings in the previous step.

For the second step the three criteria of causes of non-iidness in Federated Learning by Duan (2019) is used:

1. Size Imbalance, where the data size on each device (or client) is uneven;
2. Local Imbalance, where each device does not follow a common data distribution;
3. Global Imbalance, means that the collection of data in all devices is class imbalanced.

In this way, an assessment on the non-iidness of a newly added data set to a Federated Learning system can be made.

In addition, an interesting link between the first criterium, size imbalance, and the consolidation technique discussed in the research question 5 can be made. As the consolidation mechanism follows a naive approach, attributing the same weight to each data point, the importance of the criterium size imbalance can be explained. In the case where one data site provides an overwhelming majority of the data points, it also has, by definition, the most influence on the resulting global model. This is only a problem when there is also a local of global imbalance. The size imbalance will have a leveraging effect on these imbalances.

RQ7: How to design a methodology that fits the goal of the main research question?

This research question seeks a method for designing a method, i.e. a meta-methodology. In this way the design process is structured and theoretically-backed. Such a meta-methodology is provided by Harmsen's (1997) Situational Method Engineering. Here, the situation is characterized by the goal, scope, and objectives of this study. These serve as criteria for the selection of method fragments. In this study, the selected method fragments coincide largely with the Federated Learning differentiating characteristics deduced from the literature review. Later, the method is assembled by using the created method fragments and by utilizing some set assembly rules, as stated by the meta-methodology. Concluding, the Situational Method Engineering of Harmsen (1997) provides a structured way to design the target methodology of this study.

RQ8: How to evaluate the designed methodology?

The designed method was founded on literature insights and its design process was scientifically-backed by a meta-methodology, but this is no substitute to validating and evaluating. Therefore, this method is evaluated in two subsequent steps coming from the research methodology DSRM: validation by executing the method in a real-world case study, and evaluating this case study by demonstration to experts and measuring their responses. As described by Wieringa (2014), an artifact, in this case a method, should be evaluated by utility. This will be used as the main metric. In addition, the resulting Federated Learning algorithm is implemented in a separate case study.

First, the validation by doing a real-world case study. The case study was conducted at the software company Topicus in the financial department, with data from financial institutions. The problem context was drawn from a real-world need and fit the scope of the designed method. By executing the case study, it showed that the method is feasible for usage in a real-world problem context. Also, by executing this case study an example of the method is given, further enhancing the understanding of the working of the method and showing the utility of the method.

Second, the method is demonstrated to 5 experts in the field of data science by means of describing and explaining the method, and showing its execution by means of the case study. Then, experts' are asked to fill in a survey. This survey contains standard questionnaire items from the UTAUT model, which is a model to predict future usage behavior by means of 5 determinants: performance expectancy, effort expectancy, social influence, facilitating conditions, and behavioral intent. This model fits the goal of measuring utility, as the concepts are closely related.

The survey results are positive, the response averages on performance expectancy, effort expectancy, and behavioral intent where the highest, showing that the respondents are likely to use the method and see productive value in it. Some slightly lower scores, but still positive nonetheless, on social influence and effort expectancy show that the method operates in an environment that is not familiar with Federated Learning. This is not surprising, as Federated Learning is a new research area, thus organizations have not developed the necessary infrastructure, knowledge base, and support for facilitating projects in this area.

Third, the case study's resulting Federated Learning method, FedAvg, is implemented in a subsequent case study at the same company. The purpose of this is to show that the result of the method is also suitable for practical usage, next to only having a theoretically sound backing.

Concluding, by means of the literature background and the three-way evaluation, the method not only has a theoretical backing, but also positive validation for usage in the real world.

Concluding Remarks

This research has succeeded in its main goal of designing a method to support organizations make an informed choice among Federated Learning methods available, based on their data-related characteristics, privacy requirements, and business objectives. It first set out to define Federated Learning and, by means of this definition, make inventory of all Federated Learning methods presented in the Systematic Literature Review. After the various Federated Learning methods are identified, their main differences are investigated. This is done by investigating 5 differentiating characteristics of these Federated Learning methods. These differentiating characteristics, then, served as the basis for designing a method to make an informed choice. This method was designed by an academically backed meta-methodology. Then it was validated for utility in three steps. First, through a real-world case study, which showed the applicability of this method in a real-world setting. Second, by means of a demonstration of the method and case study to experts, which showed overall positive results. Third, the resulting FedAvg method was implemented in a subsequent case study at the company, to show its practical relevancy.

All in all, the main artifact of this study - the method - is based on insights from the literature, designed by means of an academic meta-methodology, and positively validated by means of a real-world case study and a demonstration to experts.

8.2 Discussion

This section presents a discussion on the research methodology and results. This is an additional discussion next to the partial discussions found at the end of each of the research question result sections in Chapters 3, 4, 5, and 6.

In research question 4 the predictive performance differences among Federated Learning methods are investigated. However, in a machine learning, and therefore Federated Learning, context, this is difficult to achieve. This is because the predictive performance results are not only tied to the algorithm used, but also based upon the data set used and the objective set. In this way, it is difficult to directly compare results from a literature study alone. This has been done regardless. There were some (external) studies which either directly compared multiple Federated Learning methods at once on the same data set and with the same objective, or separate studies which directly compared their newly developed method to a previous publication on the same data set. (Such as that of Nilsson et al, 2018.) Only studies which conformed to these requirements were used as input for the answering of this research question. For this reason, the results are rather limited to a few Federated Learning methods. Therefore, in the designed method, the Federated Learning methods present in the comparable study set take precedence over the other methods. It could therefore be the case that in practice other found Federated Learning method would produce better results, but trustworthy comparable information is not available, and are therefore excluded.

This study uses Situational Method Engineering as the meta-methodology for designing a method. Although this meta-methodology's scope includes generic methods, it is tailored to IT projects in its guidelines. This manifests itself by the situation characterization phase. Here all proposed metrics and guidelines are tailored to IT project characteristics. To strengthen the structure of this meta-methodology, the results of the first two phases and their guidelines of the DSRM, Problem identification & motivation and Defining solution objectives, are used instead. The phases of both methodologies show similarities and have the same overall goal.

Next, the assembly of the designed method is tied to the results drawn from the SLR. This was done to satisfy one of the situational independent criteria in the method assembly process: efficiency. The efficiency criterium states that method should fulfill its duty at minimal cost and effort. For example, the conditional split operator in the method serves to minimize the time it takes to come to an optimal choice among the Federated Learning methods. It does so by skipping method steps whose goal it is to gather unnecessary information, i.e. information that in that particular situation does not contribute to a better result anymore. This on the one hand makes the method more efficient for users, but on the other hand more time consuming to design. This efficiency gain is tied to the results of the SLR. Therefore, if these insights change or new Federated Learning methods with other properties emerge - and they are likely to change as the research area is still new and under development - then the assembly process will result in a different method. This tight coupling between data and method is practical and useful for the users of the method, but can, however, be seen as too specific. If the literature study is repeated often, then a more general method would be more suitable.

In the same manner, the 5 identified differentiating characteristics are tied to the Federated Learning methods available. Although the definition remains the same, the identification of these differentiating characteristics is based on the differences between the characteristics of the available Federated Learning methods. Therefore, if more Federated Learning methods are developed with significantly other characteristics, this process of devising the differentiating characteristics should be repeated. However, as the research on Federated Learning is standardizing since the FedAvg method of McMahan et al, it is not a likely scenario that the main properties of Federated Learning will undergo radical differences in the near future.

The next point is about the evaluation. By choosing the UTAUT model as a tool for evaluation an assumption was made. This assumption states that utility, the overarching measure of evaluation as stated by Design Science, is measured by the UTAUT model. However, the UTAUT model states that it predicts the usage behavior. This study makes the argument that utility and willingness to use the artifact are closely related, as stakeholders, users will not use the method if they do not see utility in it. Also, the utility is validated by executing the case study in a real-world scenario.

In the evaluation part of this study, the case study and the evaluation by demonstration are conducted at the same company. This way of measuring is as close to a real-world implementation as possible - users are directly involved in the case study situation - and arguably the preferred setup. It could, however, create bias in future comparisons. In possible future evaluations of this method where the case study company differs from the workshop evaluation company, the results should be compared with this information in mind, as it would not be a fair comparison. It is expected that in such a case the results will likely be lower, as the participants of the evaluation are not as closely linked to the presented situation as in the former case.

The evaluation by demonstration and measurement by the UTAUT model shows positive results. Especially in the determinant performance expectancy and effort expectancy. The high score in performance expectancy show that the users see potential productivity gains in using the method, i.e. utility. Effort expectancy's high score indicates that the method is set up to be efficient enough, conforming to the efficiency situational dependent criterium in the method assembly process. The determinants facilitating conditions and social influence also show positive results, but lower than the preceding determinants (up to 0,6 points on a 5 point scale lower). It could be that these determinants are scored lower due to the fact that Federated Learning is a new research area, and knowledge, support in organizations is not yet materialized. The method can, therefore, still be improved on those points. Looking at those determinants, a suggestion for improvement is to provide better additional guidelines and additional resources for other supporting stakeholders such as supervisors and management.

Next, the survey's descriptive statistics show that the respondents are predominantly male and young. As these properties are moderating factors in the UTAUT model, this further enhances the positive results found on the determinants performance expectancy and effort expectancy, while diminishing the slightly lower results on facilitating conditions and social influence (the latter having a higher effect among females and older users, and the inverse is true for the former). However, this leveraging effect cannot be stated in general, as the participants in the evaluation were not selected randomly, and the group was not meant to make any claim on representativity.

Next, although the method was founded upon insights from the literature, validated by means of a case study and evaluated by experts, the implementation of the chosen Federated Learning method is also implemented in a cast study to show its practicality. This is to validate the practicality of the result of the method, not merely relying on a literature based backing.

Also, in this case study, two local Machine Learning models and a centralized model were built, next to the Federated model. This showed that the centralized model shows similar results as the Federated model, confirming the literature insights that Federated Learning's privacy preserving mechanisms do not significantly impede predictive performance. In addition, it showed that some locally learned models perform better on that respective data site than the Federated model. It should be noted that the differences in predictive performance between all - local, centralized, and federated - models are quite similar.

8.3 Generalizability

Although the designed method has only been evaluated at one company in one industry, the artifact of this study has been designed with generalizability in mind. The method was designed not specifically for this company, but for a set general problem context. Within this problem context, stated in the introduction, a systematic literature review was executed. The insights from the literature set the basis for the method. For example, the main components of the method, the list of Federated Learning algorithms and the 5 differentiating characteristics, are general to Federated Learning, not a specific industry. Therefore, the method is suitable for all organizations identifying with the stated problem context in the introduction.

The method has been evaluated at a software company who provides software for the financial sector. It is recommended that the method is evaluated at other industries as well. A good next step would be in the healthcare sector, as many Federated Learning studies found are already conducted in the context of healthcare, i.e. the interest is already there. Here, a more ad-hoc approach may be needed as in this case study there was an overarching software development company which provided a central point and has standardized software with similar domain models. In other situations this might not be this clear, for example, in healthcare there may not be an overarching software company, making it more decentralized. It is therefore advised to form a central steering unit with all participating units, incorporating both central planning and increasing the willingness for everyone to participate and oversee the process.

8.4 Contributions

In this section both the theoretical and practical contributions of this study are stated.

8.4.1 Theoretical Contributions

First of all, this study synthesizes a clear view on what Federated Learning is according to the literature. As of right now, due to the research area being relatively new, information about Federated Learning is fragmented, not completely overlapping, and sometimes even contradictory. This is problematic, especially for organizations and researchers trying to get familiar with this topic. Before this study they would have to read many papers in order to gain a clear understanding of what Federated Learning is. This study synthesizes one clear agreed-upon definition of Federated Learning, based on the most common characteristics in the literature. This is useful as some studies make the definition too broad for its purpose. E.g. some studies also include a decentralized topology for Federated Learning, however, this only makes the definition too broad for its function, as all Federated Learning methods reviewed only work with a central aggregation server.

Next, this study composes a comprehensive list of the most prevalent Federated Learning methods. This has not been done before in previous literature publications to this extend. The methods are usually introduced on a one-per-paper basis, making the information fragmented. This study makes inventory of the Federated Learning methods found in the literature and presents them in a clear manner. It lists 17 Federated Learning methods found in the SLR; the most in any study as of to date. This comprehensive list provides the foundation to make comparisons between the Federated Learning methods.

To purposefully compare the Federated Learning methods found, this study introduces the definition of differentiating characteristics. These differentiating characteristics provide a way to merely look at meaningful differences between Federated Learning methods, not trivial ones. The study devised 5 differentiating characteristics in Federated Learning, and along these created a clear overview of the main differences between the available Federated Learning methods.

Another theoretical contribution this study makes is the investigation of the link between the consolidation mechanism of Federated Learning and the lower predictive performance results of Federated Learning on non-iid data. This study provides a thorough review on the effect of the way consolidation is done in Federated Learning. The technique with which Federated Learning consolidates information per data source has been investigated. It has been found to be simple/ naive averaging, where every data point (or data site) is attributed an equal weight. This study is the first to question this basic assumption present in Federated Learning research on such a thorough basis. It questions the relatively simple, naive, way consolidation of data among data sites is done. This study then links this assumption to problems seen in Federated Learning regarding non-iid data. Other studies did mention this problem implicitly by adapting their Federated Learning methods to work better on non-iid data, but did not make this direct link with the way consolidation is done. The relevance of this discovered link is that it shows a possible avenue to do future research on. This link could be used to investigate whether this naive method of combining data could be modified in future federated learning methods, in order to achieve better predictive results.

In addition, this study confirms the earlier found literature insight that Federated Learning models do not perform significantly worse than a centralized approach. The results of the case study implementing FedAvg and a centralized approach show very similar predictive performance results, with only a slight edge for the centralized approach over the Federated model.

8.4.2 Practical Contributions

This study designed a method that supports organizations who want to choose a Federated Learning method, which supports their organization's specific data-related characteristics and business goal. The designed method is founded upon insights from a systematic literature review, successfully validated by a real-world case study, and positively evaluated by experts. The method, and the theoretical insights that it is founded upon, provides organizations new to Federated Learning a clear and comprehensive way to better understand what Federated Learning is, and what the best solution-fit would be for their specific situation. This study provides the most comprehensive review of existing Federated Learning methods as of yet, before this, organizations would have to do a systematic literature review themselves to attain this level of information overview.

Next, this study provides a way for organizations to detect non-iid data sets in a Federated Learning context. Non-iidness has traditionally been defined, and its tools adapted to, working with one general data set. This is not applicable to Federated Learning, due to the distributed and privacy-preserving nature. Instead, an iterative method is presented, which compares a new data set to existing ones, and tests it via three main causes of non-iidness in Federated Learning. In this way, a more practical method is introduced to work in a Federated Learning context.

8.5 Limitations

This study, as all studies, is bound by limitations. These limitations are discussed in this section.

First, the research area is relatively new. The earliest paper used in the SLR is from 2013 and the first formal definition of Federated Learning is from 2017. Also, as the meta-analysis of the SLR showed in chapter 2, most research has just been published last year (2019). It could therefore be that most research is yet to be published, as the area is still gaining momentum. This new research could change current findings or strengthen currently weaker ones.

Second, the SLR was conducted by performing one broad search query for all research questions. The reasoning is currently valid, because the newness of the research area still makes manually going through a smaller number of papers possible, and even preferable, to not leave out any papers based on a specific search query, as a few papers missing has a large relative impact right now. However, this may not hold for future research as the number of studies published will expand, and multiple more specific search queries will have to be constructed.

Third, this study is only focused on Federated Learning as a solution for the problem context, as is inherent to the scope of the research. It could be that other methods are useful too and these are now excluded. For example, during the SLR it became apparent that ensemble methods and transfer learning are also methods that could provide a solution to the problem context.

Fourth, only a few studies actually made comparisons between multiple Federated Learning methods in an external and independent way; i.e. on the same data sets and by another author than the creator of the method. Therefore, the number of well-founded and credible comparisons in terms of predictive performance were limited. It is advised to conduct this research again if more of these types of comparisons are performed in the future.

Fifth, is regarding the designed method. Although the method has positively been validated and evaluated on (practical) utility by means of a case study and expert evaluation, it has not been validated on its subsequent implementation results. In other words, the resulting implementation afterwards is not validated for practicality.

Also, the scope of the method is limited to finding a good solution-fit regarding an organizations data-related characteristics, like privacy and data partitioning, and its business goal. It does, however, not include more practical elements like technical implementation requirements, infrastructure, implementation costs, and implementation time. This has been due to the limited practical information available in the literature.

8.6 Future Work

In this section possible avenues for future research are suggested.

First of all, it is suggested that designed method should be validated and evaluated at multiple different industries. Not only at the financial sector as is the case right now. Although the method was designed for a general purpose, i.e. all organizations which fit the stated problem context, it is of added value to the generalizability of the method to be validated at multiple organizations in different industries. As much research on Federated Learning has been conducted in healthcare, this would be an ideal industry to start, as the interest in Federated Learning in that area has already been shown. Also regarding the evaluation, the method has not been evaluated with a representative respondents group in mind. To make the results of the evaluation more generalizable, it is suggested that a representative group is to be drawn in future research.

It would also be useful to provide more validation case studies on the resulting Federated Learning algorithms of the method. As of right now only one Federated Learning algorithm, that of FedAvg, is implemented. It is useful to also implement the other Federated Learning algorithms, show their differences in predictive performance, and confirm that the method's resulting Federated Learning algorithm is indeed the best solution-fit. This can serve as feedback to improve the method based on empirical results.

The next point is also regarding the method. It is suggested that the method should be updated when new research on Federated Learning is available. Because the research area is relatively new and under increased development. For this purpose, the method assembly step in the method engineering phase will need to be updated too, as the assembly process is tied to the data.

Regarding the conducted literature study, some gaps in research were identified. These are discussed next.

Firstly, there is no comparison made in the research between local machine learning models and a centralized approach in non-iid data settings. As could give relevant insight into the answering of research questions 3 and 4, performing this comparison is useful as future work. It could provide insight into the practicality and usefulness of global models in non-iid contexts. As Federated Learning performs sub-optimally in these context, it is interesting to investigate whether this is due to the techniques used in Federated Learning or if its due to the nature of the underlying data.

Secondly, in addition to the previous point, there are only few external (non-primary) papers which test and compare these different Federated Learning methods to each other on the same dataset and with the same objective. More of these types of comparisons between Federated Learning methods should be performed to be able to make a better comparison among the myriad of Federated Learning methods available. This would also make the method more comprehensive and practically relevant.

Thirdly, there is no comprehensive comparison made in the research between local machine learning models and a federated approach in non-iid data settings. As the results of research question 5 showed, Federated Learning methods do not perform optimally when the combined data sets are non-iid, even when using specialized Federated Learning methods for non-iid contexts. It could, therefore, be the case that individual parties are better off in terms of predictive performance results by simple training a local model instead of applying a Federated Learning approach. These comparisons are not yet made in the literature and could provide relevant insight into comparing a larger number of situations.

Lastly, the discovered link between the naive way of consolidating data from multiple data sources in Federated Learning and the worse performance on non-iid data sets it causes could be investigated further. This could provide important insights to improve upon the current naive method of combining data and create a new type of Federated Learning methods which perform consolidation in a more intelligent way. As seen in the SLR, Zao et al. (2018) show that there is a up to a 55% reduction in accuracy in traditional Federated Learning methods, while the Astraea method, specially adapted for non-iid data without the need for data sharing, only shows improvement of 5,59%. This shows that there are still improvements to be made.

References

Allende-Cid, H., Allende, H., Monge, R., & Moraga, C. (2013). Wind Speed Forecast under a Distributed Learning Approach. In 2013 32nd International Conference of the Chilean Computer Science Society (SCCC) (pp. 44-48). IEEE.

Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer. 98, 146

Boyd, S., Parikh, N., & Chu, E. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. Now Publishers Inc.

Brisimi, T. S., Chen, R., Mela, T., Olshevsky, A., Paschalidis, I. C., & Shi, W. (2018). Federated learning of predictive models from federated electronic health records. International journal of medical informatics, 112, 59-67.

Cao, L. (2014). Non-iidness learning in behavioral and social data. The Computer Journal, 57(9), 1358-1370.

Chandrashekar, G., & Sahin, F. (2014). A survey on feature selection methods. Computers & Electrical Engineering, 40(1), 16-28.

Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., & Wirth, R. (2000). CRISP-DM 1.0: Step-by-step data mining guide. SPSS inc.

Chollet, F. (2018). Deep Learning with Python. New York: Manning Publications Co.

Clauset, Aaron (2011). "A brief primer on probability distributions". Santa Fe Institute.

Darrell, T., Kloft, M., Pontil, M., Rätsch, G., & Rodner, E [Dagstuhl]. (2015). Machine learning with interdependent and non-identically distributed data (dagstuhl seminar 15152). In Dagstuhl Reports (Vol. 5, No. 4). Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

Deist, T. M., Jochems, A., van Soest, J., Nalbantov, G., Oberije, C., Walsh, S., ... & Dekker, A. (2017). Infrastructure and distributed learning methodology for privacy-preserving multi-centric rapid learning health care: euroCAT. Clinical and translational radiation oncology, 4, 24-31.

Deist, T. M., Dankers, F. J., Ojha, P., Marshall, M. S., Janssen, T., Faivre-Finn, C., ... & Zhang, Z. (2020). Distributed learning on 20 000+ lung cancer patients–The Personal Health Train. Radiotherapy and Oncology, 144, 189-200.

Duan, M., Liu, D., Chen, X., Tan, Y., Ren, J., Qiao, L., & Liang, L. (2019, November). Astraea: Self-balancing federated learning for improving classification accuracy of mobile deep learning applications. In 2019 IEEE 37th International Conference on Computer Design (ICCD) (pp. 246-254). IEEE.

European Parliament and Council of European Union. (2016). GDPR Regulation (EU) 2016/679. Available at https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32016R0679&from=EN (Accessed on 3 September 2020)

Friedman, J., Hastie, T., Tibshirani, R. (2001). The elements of statistical learning (Vol. 1). Springer series in statistics Springer.

Gong, Y., Fang, Y., & Guo, Y. (2016). Private data analytics on biomedical sensing data via distributed computation. IEEE/ACM transactions on computational biology and bioinformatics, 13(3), 431-444.

Goodfellow, I., Bengio, Y., Courville, A. (2016). Deep Learning (Adaptive Computation and Machine Learning series). The MIT Press.

Hard, A., Rao, K., Mathews, R., Ramaswamy, S., Beaufays, F., Augenstein, S., ... & Ramage, D. (2018). Federated learning for mobile keyboard prediction. arXiv preprint arXiv:1811.03604.

Harmsen, A. F., Ernst, M., & Twente, U. (1997). Situational method engineering. Utrecht: Moret Ernst & Young Management Consultants.

Harrag, F., & Gueliani, S. (2020). Event Extraction Based on Deep Learning in Food Hazard Arabic Texts. arXiv preprint arXiv:2008.05014.

Huang, L., Shea, A. L., Qian, H., Masurkar, A., Deng, H., & Liu, D. (2019). Patient clustering improves efficiency of federated machine learning to predict mortality and hospital stay time using distributed electronic medical records. Journal of biomedical informatics, 99, 103291.

Ilias, C., & Georgios, S. (2019). Machine Learning for All: A More Robust Federated Learning Framework.

Jalalirad, A., Scavuzzo, M., Capota, C., & Sprague, M. (2019, December). A Simple and Efficient Federated Recommender System. In Proceedings of the 6th IEEE/ACM International Conference on Big Data Computing, Applications and Technologies (pp. 53-58)

Japkowicz, N., & Stephen, S. (2002). The class imbalance problem: A systematic study. Intelligent data analysis, 6(5), 429-449.

Jochems, A., Deist, T. M., Van Soest, J., Eble, M., Bulens, P., Coucke, P., ... & Dekker, A. (2016). Distributed learning: developing a predictive model based on data from multiple hospitals without data leaving the hospital–a real life proof of concept. Radiotherapy and Oncology, 121(3), 459-467.

Khalid, S., Khalil, T., & Nasreen, S. (2014). A survey of feature selection and feature extraction techniques in machine learning. In 2014 Science and Information Conference (pp. 372-378). IEEE.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Kitchenham, B., & Charters, S. (2007). Guidelines for performing systematic literature reviews in software engineering.

Konečný, J., McMahan, H. B., Yu, F. X., Richtárik, P., Suresh, A. T., & Bacon, D. (2015). Federated learning: Strategies for improving communication efficiency. arXiv preprint arXiv:1610.05492.

Kuhn, M., & Johnson, K. (2013). Applied predictive modeling. Springer.

Kurgan, L. A., & Musilek, P. (2006). A survey of Knowledge Discovery and Data Mining process models. The Knowledge Engineering Review, 21.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. nature, 521(7553), 436-444.

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

Liu, Y., Kang, Y., Zhang, X., Li, L., Cheng, Y., Chen, T., ... & Yang, Q. (2019). A communication efficient vertical federated learning framework. arXiv preprint arXiv:1912.11187.

Malle, B., Giuliani, N., Kieseberg, P., & Holzinger, A. (2017, August). The more the merrier-federated learning from local sphere recommendations. In International Cross-Domain Conference for Machine Learning and Knowledge Extraction (pp. 367-373). Springer, Cham.

McMahan, B., Konečný, J., & Ramage, D. (2015). Federated optimization: Distributed optimization beyond the datacenter. arXiv preprint arXiv:1511.03575.

McMahan, B., Moore, E., Ramage, D., Hampson, S., & y Arcas, B. A. (2017, April). Communication-efficient learning of deep networks from decentralized data. In Artificial Intelligence and Statistics (pp. 1273-1282).

Meka, R., & Patil, A. (2015). Performing Predictive Data Analytics in Data Mining Using Various Tools. IJITR, 3, 2229–2233.

Nilsson, A., Smith, S., Ulm, G., Gustavsson, E., & Jirstrand, M. (2018, December). A performance evaluation of federated learning algorithms. In Proceedings of the Second Workshop on Distributed Infrastructures for Deep Learning (pp. 1-8).

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design science research methodology for information systems research. Journal of management information systems, 24(3), 45-77.

Peteiro-Barral, D., & Guijarro-Berdiñas, B. (2013). A survey of methods for distributed machine learning. Progress in Artificial Intelligence, 2(1), 1-11.

Rabanser, S., Günnemann, S., Lipton, Z.C. (2018). Failing loudly: An empirical study of methods for detecting dataset shift. arXiv preprint arXiv:1810.11953

Ramachandran, P., Zoph, B., & Le, Q. V. (2017). Searching for activation functions. arXiv preprint arXiv:1710.05941.

Rouhani, B. D., Mahrin, M. N. R., Nikpay, F., Ahmad, R. B., & Nikfard, P. (2015). A systematic literature review on Enterprise Architecture Implementation Methodologies. *information and Software Technology*, *62*, 1-20.

Ruder, S. (2016). An overview of gradient descent optimization algorithms. arXiv preprint arXiv:1609.04747.

Rumbold, J.M.M., Pierscionek, B.K. (2017). A critique of the regulation of data science in healthcare research in the European Union. BMC Med Ethics 18, 27. https://doi.org/10.1186/s12910-017-0184-y

Sattler, F., Wiedemann, S., Müller, K. R., & Samek, W. (2019). Robust and communication-efficient federated learning from non-iid data. IEEE transactions on neural networks and learning systems.

Schmid, P. (2019). Technological and commercial design parameters. (De)centralized AI Solutions. In Workshops and Work-InProgress Contributions at S-BPM ONE (p. 49).

Shanker, M., Hu, M. Y., & Hung, M. S. (1996). Effect of data standardization on neural network training. Omega, 24(4), 385–397. doi:10.1016/0305-0483(96)00010-2

Shaoxiong, J., Pan, S., Long, G., Li, X., Jiang, J., & Huang, Z. (2019, July). Learning private neural language modeling with attentive aggregation. In 2019 International Joint Conference on Neural Networks (IJCNN) (pp. 1-8). IEEE.

Shmueli, G., Koppius, O. (2010). Predictive analytics in information systems research. Robert H. Smith School Research Paper No. RHS.

Sun, X., Bommert, A., Pfisterer, F., Rähenfürher, J., Lang, M., & Bischl, B. (2019, September). High dimensional restrictive federated model selection with multi-objective bayesian optimization over shifted distributions. In Proceedings of SAI Intelligent Systems Conference (pp. 629-647). Springer, Cham.

Teunissen, G. (2020). Research Topics: A Systematic Literature Study on Federated Learning. University of Twente

The Open Group Architecture Framework [TOGAF]. (2011). TOGAF® Version 9.1: Evaluation Copy. Retrieved July 2020 from The Open Group: https://www2.opengroup.org/ogsys/jsp/publications/PublicationDetails.jsp?catalogno=g116

Venkatesh, V., Morris, M. G., Davis, G. B., & Davis, F. D. (2003). User acceptance of information technology: Toward a unified view. MIS quarterly, 425-478.

Verma, D. C., White, G., Julier, S., Pasteris, S., Chakraborty, S., & Cirincione, G. (2019, May). Approaches to address the data skew problem in federated learning. In Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications (Vol. 11006, p. 110061I). International Society for Optics and Photonics.

Wang, S., Tuor, T., Salonidis, T., Leung, K. K., Makaya, C., He, T., & Chan, K. (2019). Adaptive federated learning in resource constrained edge computing systems. IEEE Journal on Selected Areas in Communications, 37(6), 1205-1221.

Webster, J., & Watson, R. T. (2002). Analyzing the past to prepare for the future: Writing a literature review. MIS quarterly, xiii-xxiii.

Wieringa, R. J. (2014). Design science methodology for information systems and software engineering. Springer.

Witten, I. H., Frank, E., Hall, M. A., Pal, C. J. (2016). Data mining: practical machine learning tools and techniques with Java implementations. Acm Sigmod Record.

Wolfswinkel, J. F., Furtmueller, E., & Wilderom, C. P. (2013). Using grounded theory as a method for rigorously reviewing literature. European journal of information systems, 22(1), 45-55.

Yang, Q., Liu, Y., Chen, T., & Tong, Y. (2019). Federated machine learning: Concept and applications. ACM Transactions on Intelligent Systems and Technology (TIST), 10(2), 1-19.

Zarsky, Tal. (2017). Incompatible: The GDPR in the Age of Big Data. Seton Hall Law Review, Vol. 47, No. 4(2). Available at SSRN: https://ssrn.com/abstract=3022646

Zhao, Y., Li, M., Lai, L., Suda, N., Civin, D., & Chandra, V. (2018). Federated learning with non-iid data. arXiv preprint arXiv:1806.00582.

Appendix

Appendix A - Extraction Form 1

paper name	*e.g. Brisimi 2018 - Federated learning of predictive models from federated Electronic Health Records*
Freeform	*e.g. This paper discusses the data distribution imbalance and decomposes it into multiple classes of imbalance that occurs when FL is used. Mentions that this is rarely tackled in research.*
Kind of Study	*Journal paper / Conference Proceeding / Book section*
Purpose/Goal of the study	*e.g. Apply a distributed learning approach to improve performance of wind speed forecast…*
Research method used.	*Case study, survey, experiment, interview to obtain data, observation*
Main finding/conclusion	*e.g. Data imbalances lead to worse prediction accuracy in traditional FL methods like FedAvg…*
Quality assessment	
Relevance of study to RQ	*Yes / partial / little / none*
How well are the practices or factors defined?	*Yes / partial / little / none*
How clearly is the research process established?	*Yes / partial / little / none*
How clearly are the work limitations documented?	*Yes / partial / little / none*
RQs extraction	
FL Definition	*In terms of: (i) formal definition, (ii) network/communication strain, (iii) privacy, (iv) system heterogeneity, (v) statistical heterogeneity, (vi) processing power*
What (FL) method for using multiple data sources is used?	*Extract information relevant to each RQ*
How does this method consolidate features from multiple data sources?	*Extract information relevant to each RQ*
How does this method compare to other and to local-only methods in terms of predictive performance?	*Extract information relevant to each RQ*
What type(s) of data partitioning does the FL method support? (Differentiating Characteristic)	*FL method X supports: horizontally partitioned data only / vertically partitioned data only / both*
What Machine Learning model and problem does this FL method support? (Differentiating Characteristic)	*FL method supports* *Underlying ML model: Linear model / Bayesian network model / Decision Tree / Clustering model / rule-engines / Gaussian mixture models / Support Vector Machine / Neural Network* *Problem type: Regression problem / Classification problem / Rule-learning problem / Clustering problem (unsupervised)*
What privacy protection does this FL method guarantee and support? (Differentiating Characteristic)	*no privacy (data sharing) / privacy by aggregation (no data sharing) / No data sharing + Differential Privacy / No data sharing + cryptographic method. (list synthesized by categorizing all option found in this SLR)*

Potential References	*e.g. [4] Jochems A, Deist TM, van Soest J, Eble M, Bulens P, Coucke P, et al. Distributed learning: Developing a predictive model based on data from multiple hospitals without data leaving the hospital – A real life proof of concept. Radiother Oncol 2016;121:459–67*

Appendix B - Extraction Form 2

paper name	*e.g. Zhao (2018) - Federated Learning with non-iid data*
Freeform	*e.g. This paper discusses the data distribution imbalance and decomposes it into multiple classes of imbalance that occurs when FL is used. Mentions that this is rarely tackled in research.*
Kind of Study	*Journal paper / Conference Proceeding / Book section*
Purpose/Goal of the study	*e.g. Apply a distributed learning approach to improve performance of wind speed forecast…*
Research method used.	*Case study, survey, experiment, interview to obtain data, observation*
Main finding/conclusion	*e.g. Data imbalances lead to worse prediction accuracy in traditional FL methods like FedAvg…*
Quality assessment	
Relevance of study to RQ	*Yes / partial / little / none*
How well are the practices or factors defined?	*Yes / partial / little / none*
How clearly is the research process established?	*Yes / partial / little / none*
How clearly are the work limitations documented?	*Yes / partial / little / none*
RQs extraction	
Non-iid/iid data Definition	*In terms of: formal definition, challenges.*
Method to identify non-iid data in the context of Federated Learning.	*Yes / Reference to other study / No*
Potential References	*e.g. [4] Dagstuhl (2015) - Machine Learning with Interdependent and Non-identically Distributed Data*

Appendix C - UTAUT Survey

Introductory description:
Thank you for participating in this study. Please complete the workshop and read through the workshop documents first. Your answers will be collected anonymously. Please always select the most appropriate answer. Always take the presented method and the case study in mind. The survey will take at most approximately 15 minutes.

Moderating factors:
Gender
What is your gender?
<male/female/prefer not to say>

Age
What is your age?
<options:>
20 years old or younger
21-25 years old
26-30 years old
31-35 years old
36-40 years old
40-45 years old
45-50 years old
51-55 years old
55-60 years old
61 years old or older

Experience
What job do you occupy at the company?
<open>

What is the highest level of formal education you have completed?
<primary education, secondary education (middelbare school), college (HBO), university bachelor degree (WO bachelor), university master degree (WO master), PhD>

What was your study program?
<open>

How many years of working experience do you have?
<0-2, 3-5, 6-10, 10+ years>

How familiar are you in with the following concepts?
Machine Learning <no familiarity, beginner, medium, senior, expert>
Data Science <no familiarity, beginner, medium, senior, expert>

With what frequency do you work with machine learning or similar concepts?
Never, seldom, sometimes, regularly, very often

<For all below. Likert scale labels: strongly disagree, disagree, neutral, agree, strongly agree>
Performance Expectancy:

- **I would find the method (or: way of working) useful in implementing a Federated Learning system. (U6)**
- **Using the method enables me to implement a Federated Learning system more quickly. (RA1)**
- **Using the method increases my productivity when implementing a Federated Learning system. (RA5)**

Effort Expectancy:

- **My interaction with the method would be clear and understandable. (EOU3)**
- **It would be easy for me to become skillful at using the method. (EOU5)**
- **I would find the method easy to use (EOU6)**
- **Learning to use the method will be easy for me. (EU4)**

Attitude Towards Using Technology

- **Using the method is a good idea. (A1)**
- **The method makes my work more interesting. (AF1)**
- **Working with the method is fun. (AF2)**
- **I would like working with the method. (Affect1)**

Social Influence

- **People who influence my behavior think that I should use the method. (SN1)**
- **People who are important to me think that I should use the method. (SN2)**
- **I expect my seniors/management at Topicus to be helpful in the use of the method. (SF2)**
- **In general, I expect the organization to support the use of the method. (SF4)**

Facilitating Conditions

- **Topicus will provide the resources necessary to use the method. (PBC2)**
- **I have the knowledge necessary to use the method. (PBC3)**
- **I have the resources necessary to use the method. (FC3)**
- **The method is compatible with other systems or ways of working I use. (PBC5)**
- **Support from an individual/a group, or a service is available when problems are encountered using this method. (FC3)**

Behavioral Intention of the System

- **I intend to use the system to help me complete my job. (BI1)**
- **I predict I would use the system in the future to help me complete my job. (BI2)**
- **I plan to use the method in the future when implementing a Federated Learning system. (BI3)**

Additional comments:

If you have any additional comments on the proposed method, please state them here: (optional)
<open field>

Appendix D - List of Abbreviations

ADMM: Alternating Direction Method of Multipliers

AUC: Area under the Curve. A measure to assess predictive performance in Machine Learning.

BPMN: Business Process Model and Notation (http://www.bpmn.org)

CRISP-DM: Data mining research methodology by Chapman et al. (2000)

DSRM: Design Science Research Methodology

HPD: Horizontally partitioned data

VPD: Vertically partitioned data

NN: Neural Network

FedAvg: The Federated Averaging algorithm, by McMahan et al. (2017)

UTAUT: The Unified Theory of Acceptance and Use of Technology model by Venkatesh et al. (2003)

SLR: Systematic Literature Review

MAE: Mean Absolute Error
MSE: Mean Squared Error

Appendix E - List of Definitions

Definition 1
Federated Learning:
Federated Learning is a form of distributed machine learning where a global model is trained on a central server utilizing multiple separate heterogenous edge devices, while still preserving privacy by not permitting the data to leave their origin devices.

Definition 2
Data Site:
Data of a specific domain, clinical research for example, could be located in different places and it is expensive to carry data from one site to another due to technical or privacy concerns. We denote one of such a integrated data unity as a data site. There is a need to train a specific machine learning model for the [whole] domain, which requires collaboration across data sites (Sun et al., 2019).

Definition 3
Differentiating Characteristic:
The differentiating characteristics of Federated Learning methods are defined as: characteristics of Federated Learning methods which both (i) limit options or impact the desired outcome regarding a organization's data-related characteristics and privacy considerations, i.e. those that are relevant to the to-be-designed method, and (ii) have variation in implementation among the Federated Learning methods, i.e. not all Federated Learning methods have the same implementation regarding this characteristic.

Definition 4.1
Underlying Machine Learning model list:
Linear model, Bayesian network model, Decision Tree, Clustering model, Rule-engines, Gaussian mixture models, Support Vector Machine (SVM), Neural Network (NN).

Definition 4.2
Machine learning problem type list:
Linear problem, Regression problem, Classification problem, Rule-learning problem, Clustering problem (unsupervised), Language modeling problem.

Definition 5.1
Horizontally partitioned data (HPD):
Horizontally partitioned data means that the data at each data site have the same features, i.e. attributes or columns in traditional data base terms, but include different subjects, i.e. rows. For example, imagine a software company who sells software to small businesses to conduct their administration digitally. Each subject (i.e. small business) uses exactly the same type of software and generated the same types of data, i.e. the columns of the domain model are the same.

Definition 5.2
Vertically partitioned data (VPD):
Vertically partitioned data constitutes the opposite: the data of one subject, i.e. row, is present at multiple data sites. Each data site, therefore, has a different set of features, i.e. columns or attributes. For example, in the case of hospitals, where a patient's health records are scattered across many hospitals. One hospital has data about his blood work, while another specialized hospital only stores the results of a lung scan. The different hospitals store data about the same subject (the patient), but have different data features or columns of that patient.

Definition 6
Privacy Guarantee Levels in Federated Learning:

1. *Violates the no data sharing principle*
   In this category the Federated Learning methods give no guarantee whatsoever about the no data sharing pillar of Federated Learning. Although Federated Learning is founded on the idea

of no data sharing, in this category, each local data site does not have control over their data. The data of each local data site can be shared with other data sites or the central server.

2. *Privacy by no data sharing*
   This category can be seen as a standard of Federated Learning. Federated Learning is founded by the principle of local data sites having ownership over their own data. In this category privacy is upheld by not allowing the data itself leave each local data site. Instead, only aggregates in the form of partial model updates (i.e. parameter updates during model training) are shared with a central server.
3. *Additional privacy mechanism*
   In addition to privacy by no data-sharing, some studies indicate that this privacy guarantee level is still not enough. In fact, they claim that even the aggregate data, i.e. the parameter updates that are communicated with the central server, are private information. In addition, when the central server of other local data sites cannot be trusted, additional privacy mechanisms are also of importance. Both claimed by Gong et al (2016) and Jochems et al (2016). The additional privacy mechanisms mentioned in the studies used in this systematic literature review are: Anonymization, Differential Privacy, Secure Multi-Party Computation, Homomorphic Encryption.

Definition 7
ADMM:
"A simple but powerful algorithm that is well suited to distributed convex optimization, and in particular to problems arising in applied statistics and machine learning. It takes the form of a decomposition-coordination procedure, in which the solutions to small local subproblems are coordinated to find a solution to a large global problem" (Boyd et al., 2011).

Appendix F - Federated Learning Lookup Tables

Table F.1 - Best Performing Federated Learning Methods per Situation (Given the other characteristics are: HPD, Supported by NN, Privacy Level is 2 or lower)

Method	Situation
FedAvg, McMahan (2017)	Best-performing method in an iid data context.
Zhao et al's (2018) method	Best-performing method in a non-iid data context (55% increase in accuracy compared to baseline). Requires data sharing between data sites.
Astraea method, Duan (2019)	Adapted method for non-idd data context. Shows a 6% increase in accuracy compared to baseline. Does not require data sharing.

Table F.2 - Differentiating Characteristics (1-3) of Federated Learning Methods

FL Method	Data partitioning	Type of ML model & problem	Privacy Guarantee
Allende-Cid et al's (2013) method	HPD	Linear model. Linear/regression problem. (Regression)	Privacy by no data sharing
Gong et al's (2016) method	HPD & VPD	Linear model. Classification problem. (Logistic regression)	Additional privacy mechanism (homomorphic encryption)
Deist et al's (2017) method	HPD	SVM. Classification problem.	Privacy by no data sharing
Deist et al's (2020) method	HPD	Linear model. Classification problem. (Logistic regression)	Privacy by no data sharing
Jochems et al's (2016) method	HPD	Bayesian network model. Linear/regression problem.	Privacy by no data sharing
Brisimi et al's (2018) method	HPD	SVM. (Binary) classification problem.	Privacy by no data sharing
Federated Averaging (FedAvg) - McMahan et al (2017)	HPD	Neural Network. Classification & linear/regression problems.	Privacy by no data sharing
Federated Stochastic Block Coordinate Descent (FedBCD) - Liu et al (2019)	VPD	Neural Network. Classification & linear/prediction problems.	Privacy by no data sharing
Federated Stochastic Variance Reduced Gradient (FSVRG) - Nilsson et al (2018)	HPD	Based on FedAvg. So NN. Classification & linear/regression problems.	Privacy by no data sharing
CO-OP - Nilsson et al (2018)	HPD	Based on FedAvg. So NN. Classification & linear/regression problems.	Privacy by no data sharing
Restrictive Federated Model Selection (RFMS) - Sun et al (2019)	HPD	Bayesian model. (Binary) classification problem.	Privacy by no data sharing
Federated recommender system - Jalalirad et al (2019)	HPD	Neural Network. Classification & linear/prediction problems. (Recommender system)	Privacy by no data sharing
Zhao et al's (2018) method	HPD	Neural Network. classification & linear/regression	Violates no data sharing principle
Astraea method - Duan (2019)	HPD	Neural Network. Classification & linear/regression	Privacy by no data sharing
Attentive Federated Aggregation algorithm (FedAtt) - Shaoxiong et al (2019)	Not mentioned. Estimated guess: HPD*	Neural Network. Language modeling problem.	Additional privacy mechanism (differential privacy)
CBFL - Huang et al (2019)	HPD	Clustering model. Classification problem. (Unsupervised learning)	Privacy by no data sharing
Verma et al's (2019) method	HPD	Neural Network. Classification problem.	Privacy by no data sharing

Appendix G - UTAUT Evaluation Survey Results

Table G - UTAUT Evaluation Survey Results

			Respondents				
Question	**Item Code**	**Avg**	**1**	**2**	**3**	**4**	**5**
What is your gender?			Female	Male	Male	Male	Male
What is your age?			21-25 years old	31-35 years old	26-30 years old	21-25 years old	31-35 years old
What job do you occupy at the company?			Data analyst	Agile Coach	Data Scientist	Data Analist	Senior product owner
What is the highest level of formal education you have completed?			university master degree (WO master)	university master degree (WO master)	university master degree (WO master)	college (HBO)	university master degree (WO master)
What was your study program?			Business informatics (with data science track)	Industrial Engineering & Management, Business & IT	Business Analytics	Business IT and management	Information science
How many years of working experience do you have?			0-2 years	3-5 years	3-5 years	0-2 years	6-10 years
How familiar are you with the following concepts? [Machine Learning]		2,8	2	2	4	3	3
How familiar are you with the following concepts? [Data Science]		3,4	2	3	4	3	5
With what frequency do you work with Machine Learning or similar concepts? [1-5]		3	4	2	3	2	4
PE1: U6 [I would find the method (or: way of working) useful in implementing a Federated Learning system]	U6	*4,8*	5	5	5	5	4
PE2: RA1 [Using the method enables me to accomplish tasks more quickly]	RA1	*4,2*	4	4	4	5	4
PE3: RA5 [Using the method increases my productivity when implementing a Federated Learning system]	RA5	*4*	4	4	4	4	4
EE1: EOU3 [My interaction with the method would be clear and understandable.]	EOU3	*4*	3	3	5	4	5

EE2: EOU5 [It would be easy for me to become skillful at using the method]	EOU5	*4,6*	4	5	5	5	4
EE3: EOU6 [I would find the method easy to use]	EOU6	*4,6*	4	4	5	5	5
EE4: EU4 [Learning to use the method will be easy for me]	EU4	*4,6*	4	4	5	5	5
A1: A1 [Using the method is a good idea]	A1	*4,6*	5	5	5	4	4
A2: AF1 [The method makes my work more interesting]	AF1	*3,6*	4	5	3	2	4
A3: AF2 [Working with the method is fun]	AF2	*4,2*	5	5	4	3	4
A4: Affect1 [I would like working with the method]	A1	*4,4*	5	5	4	4	4
SI1: SN1 [People who influence my behavior think that I should use the method]	SN1	*3,8*	4	4	4	4	3
SI2: SN2 [People who are important to me think that I should use the method]	SN2	*3,6*	4	4	4	3	3
SI3: SF2 [I expect my seniors/management at Topicus to be helpful in the use of the method]	SF2	*4,2*	5	3	5	4	4
SI4: SF4 [In general, I expect the organization to support the use of the method]	SF4	*4*	5	3	4	4	4
FC1: PBC2 [Topicus will provide the resources necessary to use the method]	PBC2	*4,4*	5	4	4	4	5
FC2: PBC3 [I have the knowledge necessary to use the method (given the guidelines and input documents provided)]	PBC3	*4,4*	4	4	5	5	4
FC3: PBC4 [I have the resources necessary to use the method]	PBC4	*3,4*	3	3	3	3	5
FC4: PBC5 [The method is compatible with other systems or ways of working I use]	PBC5	*3,4*	3	2	4	4	4
FC5: FC3 [Support from an individual/a group, or a service is available when problems are encountered using this method]	PC3	*3,4*	4	3	4	3	3

BI1: BI1 [I intend to use the system to help me complete my job]	BI1	*4,2*	4	5	4	4	4
BI2: BI2 [I predict I would use the system in the future to help me complete my job]	BI2	*4*	3	5	4	4	4
BI3: BI3 [I plan to use the method in the future when implementing a Federated Learning system]	BI3	*4,4*	4	4	5	5	4

Appendix H - Data Extraction SQL Queries

H.1 - Calculating the Lead Times for each process

```
SELECT P.ProcessNumber AS ProcessNumber, MIN(PS.DateTime) AS StartDate,
MAX(PS.DateTime) AS EndDate, DATEDIFF(HOUR, MIN(PS.DateTime),
MAX(PS.DateTime)) AS LeadTime
FROM processinfo.Process P
RIGHT JOIN processinfo.ProcessStatus PS ON P.ProcessNumber =
PS.Process_ProcessNumber
LEFT JOIN processinfo.ProcessType PT ON P.ProcessType_Id = PT.Id
WHERE P.Whitelabel_WhitelabelNumber = 20
AND PT.ProcessTypeOfProcess = 'Acceptatie'
AND ( PS.Code = 'StartNieuweAanvraag'
           OR
           PS.Code = 'BindendAanbodVerstuurd'
      )
GROUP BY P.ProcessNumber
HAVING COUNT(P.ProcessNumber) = 2;
```

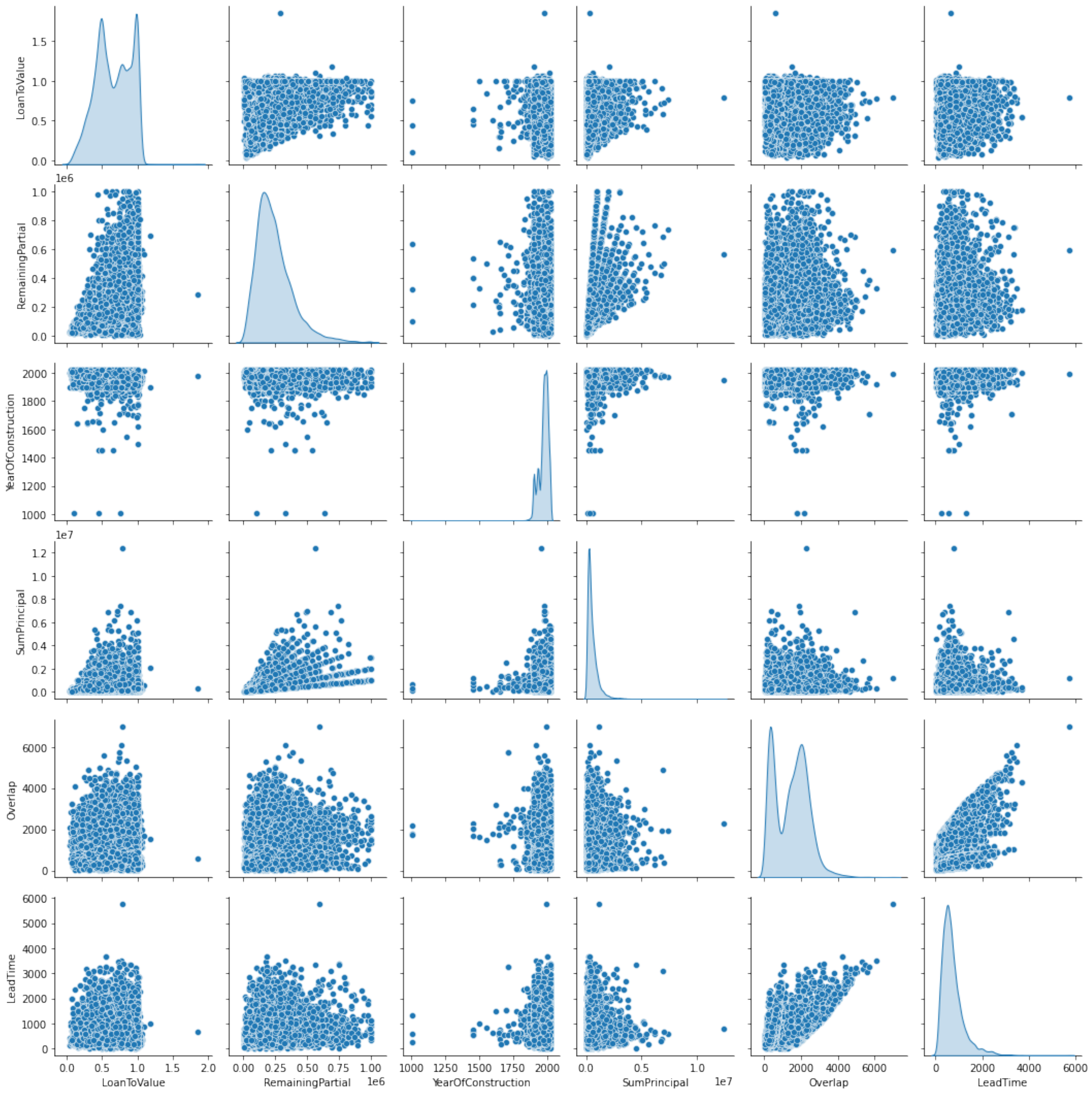
H.2 - Calculating the Overlap (number of overlapping processes)

```
SELECT *, (SELECT COUNT(*) FROM processinfo.ProcessLeadTimeFive test
WHERE NOT (v.StartDate > EndDate or v.EndDate < StartDate) ) as Overlap
FROM processinfo.ProcessLeadTimeFive v;
```

H.3 - Full Query Extracting all Features

```
SELECT STable.RequestType, STable.PrimaryHandler,
STable.HandlingParty_Id, STable.Overlap, STable.LoanToValue,
STable.RemainingPartial, STable.YearOfConstruction, STable.SumPrincipal,
COUNT(STable.ConsumerNumber) AS NumberOfConsumers,
SUM(STable.GrossIncome) AS SumGrossIncome, STable.LeadTime

FROM (SELECT FPTable.ProcessNumber, P.PrimaryHandler, FPTable.LeadTime,
FPTable.Overlap,
           A.RequestType, A.LoanToValue, A.RemainingPartial,
           A.HandlingParty_Id, HP.HandlingPartyNumber,
           MAX(RE.YearOfConstruction) AS YearOfConstruction,
           SUM(L.Principal) AS SumPrincipal,
           C.ConsumerNumber, MAX(I.GrossIncome) AS GrossIncome
FROM ( SELECT * FROM processinfo.ProcessLeadTimeOverlapTwenty ) AS
FPTable

LEFT JOIN processinfo.Process P ON FPTable.ProcessNumber =
P.ProcessNumber
LEFT JOIN processinfo.Application A ON FPTable.ProcessNumber =
A.Process_ProcessNumber
LEFT JOIN processinfo.HandlingParty HP ON HP.Id = A.HandlingParty_Id
LEFT JOIN processinfo.RealEstate RE ON RE.Application_Id = A.Id
LEFT JOIN processinfo.Loan L ON L.Application_Id = A.Id
LEFT JOIN processinfo.Consumer C ON C.Application_Id = A.Id
LEFT JOIN processinfo.Income I ON I.ProcessInfoConsumer_Id = C.Id

GROUP BY FPTable.ProcessNumber, P.PrimaryHandler, FPTable.LeadTime,
FPTable.Overlap,
           A.RequestType, A.LoanToValue, A.RemainingPartial,
A.HandlingParty_Id, HP.HandlingPartyNumber,
           C.ConsumerNumber
) AS STable

WHERE RequestType IS NOT null AND LoanToValue IS NOT null AND
RemainingPartial >= 0 AND YearOfConstruction IS NOT null AND LeadTime >
10
GROUP BY STable.PrimaryHandler, STable.LeadTime, STable.Overlap,
STable.RequestType, STable.LoanToValue, STable.RemainingPartial,
STable.YearOfConstruction, STable.SumPrincipal, STable.HandlingParty_Id,
STable.HandlingPartyNumber
HAVING COUNT(STable.ConsumerNumber) > 0 AND SUM(STable.GrossIncome) >
5000
;
```

Appendix I - Correlation Based Feature Selection Graphs


LoanToValue
1e6
RemainingPartial
YearOfConstruction
1e7
SumPrincipal
Overlap
LeadTime
LoanToValue
1e6
RemainingPartial
YearOfConstruction
1e7
SumPrincipal
Overlap
LeadTime