**Doctoral thesis**

# PRIVACY PRESERVING VERTICALLY PARTITIONED FEDERATED LEARNING: NEW TECHNIQUES AND CONSIDERATIONS

Florian van Daalen

2025

# PRIVACY PRESERVING VERTICALLY PARTITIONED FEDERATED LEARNING: NEW TECHNIQUES AND CONSIDERATIONS

Dissertation

To obtain the degree of Doctor at Maastricht University,
on the authority of the Rector Magnificus, Prof. Dr. P. Habibović,
in accordance with the decision of the Board of Deans,
to be defended in public
on Friday 28th of March 2025, at 16.00 hours

by

Florian van Daalen

**Supervisor**
Prof. Dr. Andre Dekker, Maastricht University

**Co-supervisor**
Dr. Inigo Bermejo, Maastricht University, University Hasselt

**Assessment Committee**
Prof. Dr. Ir. R.L.M. Peeters (Chair), Maastricht University
Prof. Dr. Ir. L. Peeters, University of Hasselt, Belgium
Prof. Dr. Ir. T. Veugen, University of Twente/TNO
Prof. Dr. S. Wyatt, Maastricht University



Cover Ana Sánchez-Moreno Royer, 2025
Production Florian van Daalen 2025
ISBN 978-94-93406-38-4

*To my wife*

# Contents

# 1 Introduction

Institutions, and society at large, are eager to unlock the knowledge hidden in data. However, it has become apparent that in most scenarios it is simply impossible for an individual party to gather sufficient data on their own[93, 198, 79, 176]. For example, in rare disease research even large hospitals will struggle to collect enough data to perform a solid analysis. This has led to a growing realization that data needs to be shared between parties for the benefit of all involved. However, sharing data is not straightforward. There are both practical and legal concerns that will need to be addressed before data can be shared in a safe manner[14].

The first major concern is that the data may contain sensitive information regarding the data subjects[14]. This information could potentially be used to harm the data subjects. Legal frameworks have been devised to protect the privacy of data subjects and prevent them from being harmed: limiting what data can be shared, how it can be shared[59, 23], and with whom it can be shared. These legal frameworks are focused on protecting the privacy, and general interests, of the individual data subjects.

The second major concern is that the data may contain information that is of vital importance to the data holder[204]. For example, because it may contain company secrets, data may hold a competitive edge, or data may be viewed as an asset due to the effort and cost involved in collecting the data. Consequently data holders will be hesitant to share their data out of fear of losing (competitive) edge. While a suitable reward might convince some data holders to reveal their data[203], there remain datasets which are so valuable that the data holder does not wish to share them regardless of the reward offered. This further limits what can practically be shared.

The third major concern is how difficult data sharing is in practice[93, 198, 79, 176]. It often involves a lot of work as the necessary infrastructure for sharing needs to be created. In order to link two or more datasets, datasets need to be aligned to ensure there are no misunderstandings. Finally, a legal framework needs to be created for the entire project. This is time-intensive and expensive.

These challenges together lead to the rise of the field of federated learning (FL). It has seen many developments the last few years in order to tackle these issues[93, 198, 79, 176]. Within this thesis, we present several novel research works, which we believe will move the field further.

## 1.1 Background

In this section we will briefly introduce the relevant technical topics.

### 1.1.1 Machine learning

The driving motivation behind the wish to share data is the desire to perform statistical analysis on this data for various purposes[65, 97]. Especially the wish to use machine learning to develop models is a major driving factor.

Machine learning is the process of performing statistical analysis on a set of data in order to create a model[90]. This model can then be queried to make predictions, or classify, future samples. For example, a model may be trained on a set of pictures to learn to recognize images of dogs. Once this training phase has been completed the model will then be able to identify if an unknown image contains a dog.

These machine learning algorithms rely on having access to sufficient data to work correctly[109]. If the dataset is too small the resulting model may simply be unusable. Alternatively, the model may work on the sub-population present in the training data. but may not generalize to the general population at large.

The work within this thesis is focused on solving the various technical problems involved with applying machine learning algorithms on a shared dataset that is split across various parties. In other words, the solutions presented in this thesis are adaptation of existing machine learning algorithms for federated learning.

### 1.1.2 Bayesian networks

Bayesian networks are a popular type of machine learning model[29, 137, 186, 27, 117]. Their popularity stems from their ability to incorporate expert knowledge as well as the relative ease with which domain experts without a mathematical background can understand them. Additionally, Bayesian networks have the ability to work with incomplete records.

This makes Bayesian networks a good fit for the project this thesis is a part of. Within this project a model will be built for use within clinical care practice. The medical world already has a rich body of expert knowledge which can be utilized within our models. Additionally, clinicians need to be able to understand, and explain, why a model acts the way it does. This means that using a model that is easy to understand is beneficial. Lastly, medical data is often of limited quality and incomplete. Furthermore, as a major motivation behind the project

is to enable data sharing between parties, we wish to make the barrier to entry to sharing data as low as possible. Utilizing a model that can deal with lower quality data supports this.

### 1.1.3 Ensemble learning

A specific machine learning topic of interest within this thesis is the field of ensemble learning[131, 151]. Ensemble learning is a sub-field within machine learning. In this field one utilizes multiple, different, models at the same time, which work together to jointly produce a new classification. An illustration of this process can be found in figure 1.1 The intuition behind the use of an ensemble is that while each individual model will make mistakes, on average the combination of models will be correct more often than any given individual model thanks to their diversity[98]. Hence, by combining the prediction of each individual model one gets a more accurate final prediction. Furthermore, it is possible to create specialized models within the ensemble which become experts in specific sub-tasks. The opinion of such a specialized expert model can then be given more weight when appropriate.

With respect to the data sharing problem explored in this thesis the ensemble learning approach has a number of possible advantages. The first advantage is that it reduces the need for active data-sharing as the individual models for the ensemble may be trained locally at each party. This significantly reduces the privacy concerns. Additionally, an ensemble based approach could be beneficial in scenarios when there are large imbalances between parties in the data sharing project. For example, imagine a scenario in which two hospitals want to build a joint model, one hospital has $100$ records, the other hospital $10000$. In this case the model is liable to overfit on the larger dataset, largely ignoring the input from the smaller hospital. However, this smaller hospital may represent a very different, but important, subpopulation which can not just be ignored. For example, the smaller hospital represents rural patients while the bigger hospital represents urban patients, two groups which may have very different needs. An ensemble

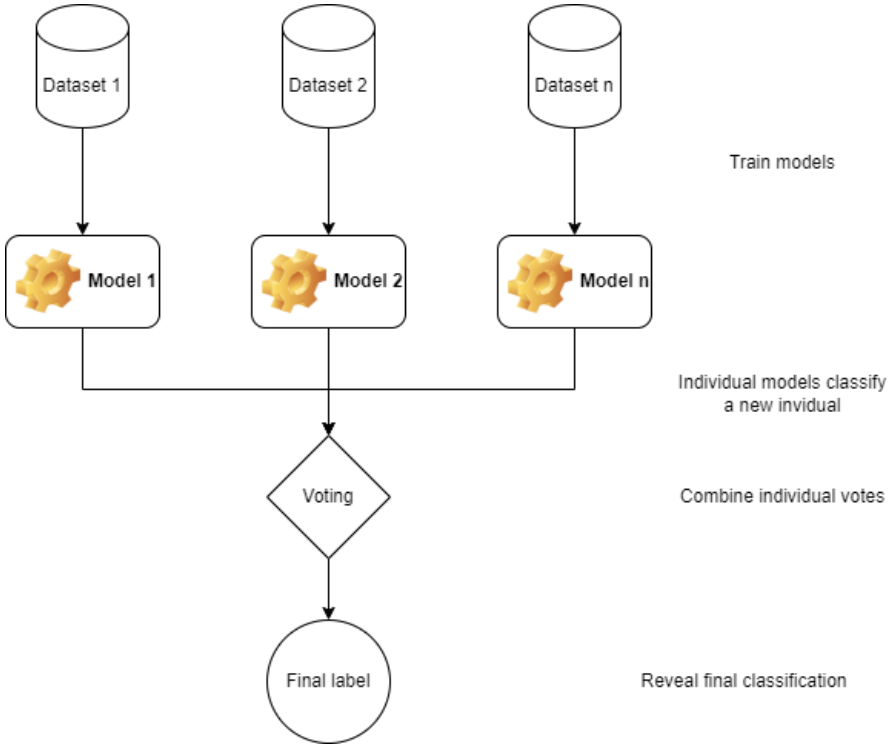


Figure 1.1: An illustration of an ensemble learning setup.

based approach could help ensure the smaller group is not ignored. Lastly, most models rely on the assumption that the data is identically and independently, distributed (IID). Ensembles do not rely on this assumption and have the ability to handle non-IID data[33]. As there is no guarantee the data will split in an IID manner across the different parties this may be a significant advantage.

We will explore this potential in this thesis.

### 1.1.4 Federated learning

Federated learning is a field that recently has seen a lot of development[93, 198, 79, 176]. Within federated learning a central model is trained between several parties without any party sharing its local data with any other party. This training process can roughly be described as follows:

1. Create a global model
2. Share the global model with each party
3. Each local party creates a local update to this model
4. The local updates are aggregated and incorporated into an updated global model
5. Repeat until convergence

The precise mechanics of sharing and incorporating updates into the global model will depend on the type of model that is being trained, as well as on the privacy concerns at play within any given project. An illustration of a federated learning setup can be found in figure 1.2

Federated learning is motivated by the idea that a given model, or any other results from a ”high level” analysis, can be considered less privacy invasive than data belonging to an individual record. The intuition being that ”high level” data, such as a model, an average, or another statistical analysis, aggregates the information present in multiple records. This aggregation obfuscates the sensitive information present in individual records. Additionally, reversing this aggregated result to deduce the contribution of any individual record is difficult, thus contributing to the privacy of the individual records.

At a high level this approach can indeed be considered very effective. However, it is important to note that simply because the individual records have now been aggregated it will not always protect privacy.

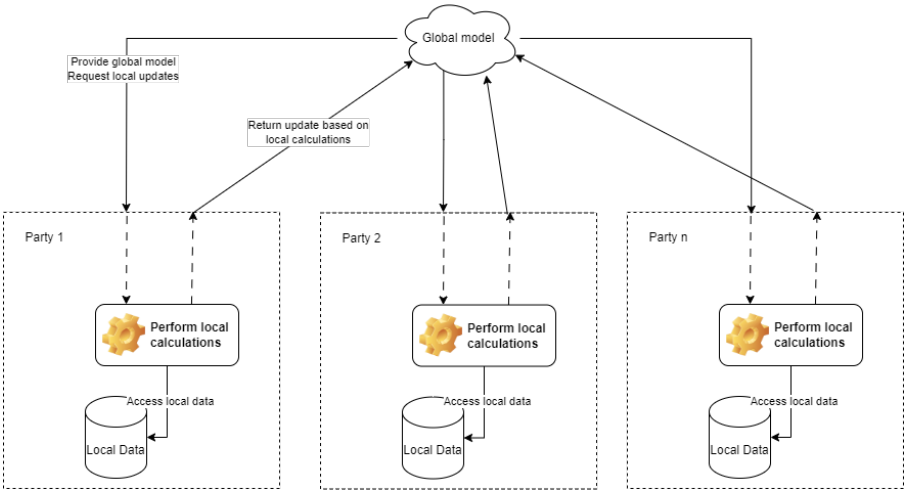


Figure 1.2: An illustration of a federated learning setup.

The aggregated result could still contain artifacts that reveal information about individual records which an attacker could use to break privacy. This can occur when too few records are aggregated, when significant outliers are present in the dataset, or when too many repeat queries are allowed.

Additionally, more specific attacks, such as model inversion attacks[198], gradient leakage[188, 203, 191], as well as attacks in which an attacker has access to external or meta information, should still be considered.

### 1.1.5 Horizontally and vertically partitioned data

Within federated learning data can be split across parties in two ways. The first is a so called horizontal split in the data, the second is a so called vertical split. Data is said to be horizontally split when the different parties involved collect the same attributes regarding a differ-

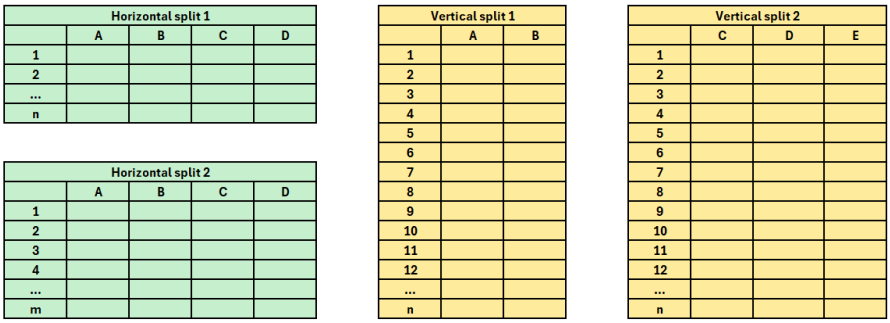


Figure 1.3: An illustration of a horizontally and vertically split data scenarios.

ent population, e.g. two hospitals working together to build a model to treat heart failure. It is said to be vertically split when the different parties collect different attributes regarding the same individuals. For example, a hospital and an insurance company working together. These splits are illustrated in figure 1.3.

### 1.1.6 Secure multiparty computation and secret sharing

When it is necessary to jointly calculate a shared statistic when a vertical split is present in the data, for example when calculating the average income within the dataset (information available at party 1) given the presence of diabetes (information available at party 2), it is necessary to utilize techniques developed within the field of secure multiparty computations (SMPC)[202]. SMPC is a sub-field of cryptography in which protocols are developed to perform joint calculations when no data may be shared. Crucially, a valid SMPC protocol can only be solved by the cooperation of all parties. The protocols cannot be solved using the information available to a single party.

One of the possible ways to achieve this is by utilizing a technique called secret sharing. Secret sharing relies on a secret key, which is split into shares that are distributed across the various parties involved in

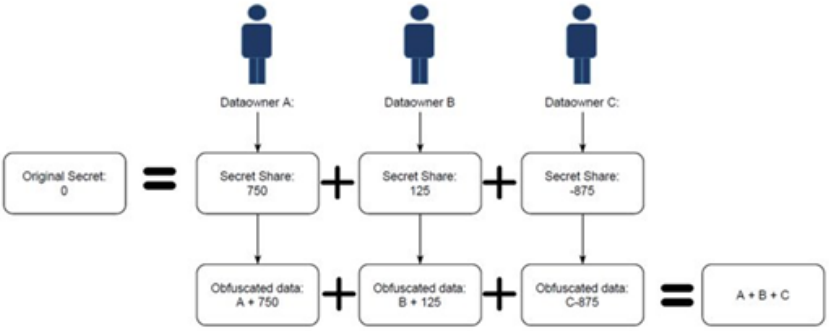


Figure 1.4: An illustration of a basic secret sharing scheme.

the calculation. These secret shares are then utilized to obfuscate the real data during the calculation process. When the whole protocol has finished the final output will reveal the desired result, as if the secret shares were never there, without having revealed any of the inputs.

An example can be found in figure 1.4. The example given here requires a trusted third party to generate the original secret and executes a fairly straightforward calculation. There are also secret sharing schemes that do not require a trusted third party. Additionally, it is important to note that the more complicated the desired calculation, the more complex the individual secret shares and the required mathematics will become.

SMPC provides a large number of techniques to perform various calculations in a secure way which have been mathematically proven to ensure secrecy. However, it is important to note that this secrecy is only guaranteed so long as specific assumptions hold. We have already alluded to some of these assumptions. A technique may require a trusted third party, it may assume that a user cannot execute repeat queries, it may assume attackers do not have access to specific meta-information, as well as many other assumptions. What may be con-

sidered a safe technique in one particular project scenario may not be in another. Additionally, it is important to note that SMPC solutions introduce significant computational overhead. Due to this overhead SMPC is not always a practical solution.

### 1.1.7 Privacy

The work in this thesis revolves around data belonging to different parties with which they wish to perform a joint analysis without revealing data. The wish to keep their data hidden has different motivations. There may be economic, legal, moral, as well as personal motivations depending on the priorities of the parties involved. This naturally brings us to the topic of privacy.

Providing a singular simple definition of privacy is difficult. Researchers with a technical background often focus on the technical questions that need to be solved. Mathematicians may attempt to create a mathematical equation to measure privacy, lawyers may try to carefully define which concepts can be considered private and protect those by law.

These approaches often result in definitions of privacy that are focused on secrecy. While secrecy is relevant, it is important to note that privacy is strongly context dependent. Throughout this thesis we attempt to accentuate this by highlighting in what contexts our technical solutions can be considered appropriate. A deeper discussion on privacy is included in chapter 7.

## 1.2 Challenges

While great strides have been made in federated learning the solutions are not yet perfect. There are a number of pressing issues that still need to be solved.

### 1.2.1 The historic focus on horizontally split data

The field of federated learning has historically focused on so called horizontally split data. There are a number of reasons for this historic focus. Firstly, the institutions who have led research in this area largely operate in horizontally split environments. Secondly, the horizontally split scenario is simpler to deal with. Simply averaging the results from the different sites can already provide reasonable results in such a scenario. However, in a vertically split scenario this is not possible as there is no way to establish correlations between the different attributes owned by different parties without first linking the records.

Due to this historic focus solutions for the vertical scenario are rare, and often underdeveloped. For example, they may be limited to a simple 2 party scenario, but cannot easily be generalized to a scenario with more parties.

This thesis is a part of a larger project (CARRIER, discussed below) in which a vertically split scenario is used and as such this will be a significant focus of the thesis. We will present new solutions, as well as improvements on existing solutions.

### 1.2.2 The cost of privacy

While federated learning algorithms provide certain privacy guarantees it is important to note these guarantees come at a cost when compared to the classical approach of running an algorithm locally.

The first, and most basic problem, is that the various parties involved will need to be convinced FL is an appropriate solution. This involves jumping through various bureaucratic hoops to convince lawyers, IT departments, and managers. Unfortunately this bureaucratic process can take a tremendous amount of time and effort, even if the technical solution is ready to be deployed.

Once everyone has been convinced a second bureaucratic problem arises. The necessary infrastructure needs to be created on which the

technical solution can be deployed. This means that each party needs to configure their own servers. Additionally, secure communication between these servers needs to be established. Lastly, the various parties must align their data; for example, they must ensure they are using the same vocabulary.

These bureaucratic problems can add considerable overhead to a project. While no chapter in this thesis is fully dedicated to these problems, it will be touched upon when discussing the strengths and weaknesses of the proposed methods. A more in depth discussion follows in chapter 7.

Aside from these bureaucratic problems there are also two technical problems that need to be solved. First is the additional computational overhead incurred by the need to communicate, as well as the overhead introduced by the privacy preserving mechanisms. Each round of communication is an extra step compared to running the algorithm fully locally. Obfuscating the data introduces additional overhead. Lastly, SMPC techniques result in more computationally complex algorithms. An example of this increased complexity compared to the original local algorithm can be found in chapter 2. It is important to note that this overhead is not just limited to the time complexity, but the space complexity is affected as well. This means that any FL algorithm has significant overhead compared to their classical counterpart. Throughout the thesis we discuss the computational overhead of each of our proposed algorithms.

Lastly, a FL algorithm may suffer from reduced accuracy when compared to its classical, fully local, counterpart. This can be due to noise introduced by the privacy preserving mechanism, for example when using $\epsilon$-differential privacy[55, 56]. Additionally, the space complexity overhead introduced by the privacy preserving techniques may require that less precise data is used. For example, the overhead may force the user to round the data to a smaller number of decimals, or require images of a lower resolution. We discuss the potential limitations

with respect to model accuracy of our proposed solutions in chapters 2,3,5, and 6.

### 1.2.3 What are we truly protecting?

Most scientific research is conducted by large institutions. These institutions can be both private and public in nature and have varying goals. Their choices of which projects to perform shapes the field. Their priorities determine the type of problem that will receive attention; for example, a large commercial institution which wants to process large volumes of data is likely to perform research that creates new privacy preserving tools as that can help them achieve their goals by allowing them to fulfil the necessary legal requirements. However, this same institution is unlikely to embark on ethical and philosophical research into the concept of privacy, as this is does not directly align with their goals and may even risk exposing some uncomfortable issues for the institute.

This influence does not need to be malicious, or even the result of a conscious effort, but the influence of large institutions can be felt throughout all research done on privacy. While we touch upon this in the technical chapters when we discuss the limitations of the various proposed solutions, we will discuss this phenomena in detail in chapter 7.

## 1.3 The CARRIER project

This thesis is a part of the Coronary ARtery disease: Risk estimations and Interventions for prevention and EaRly detection (CARRIER) project[153]. This project is funded by the Netherlands Scientific Organization under project number 628.011.212.

The ultimate goal of the CARRIER project is to detect risk of cardiovascular disease (CVD) at an early stage in patients. Once detected, patients will receive a personalized intervention, taking into account

their own preferences and abilities, to reduce their risk factors. One of the novel aspects of the CARRIER project is that the screening model will utilize a combination of medical and socio-economic data. This data is captured by different organizations such as general practitioners, the hospitals, and Statistics Netherlands (CBS). We have chosen to deal with this data split by applying federated learning.

This thesis is a part of work package 2 within the CARRIER project. Within this work package, we will create the technical solutions needed to train this screening model in a federated setting. This screening model will then be incorporated into an application to assist with the early detection of patients at risk of CVD. The intended output of this thesis is as follows:

- Develop new, or improve and implement existing, technical solutions to be able to train the models of interests in a vertically partitioned federated learning setting.
- Use these technical solutions to produce a screening model based on the data from GPs, hospitals, and CBS.
- Produce a report of the best practices learned throughout the project so that lessons learned may be brought over to future projects within Statistics Netherlands.

## 1.4 Thesis structure

This thesis can be divided into three parts. The first few chapters describe various privacy preserving algorithms that can be used in a vertically partitioned scenario. The algorithms are presented in the order they build upon each other. The second part consists of chapter 7, in which we discuss how our views of privacy are shaped by big institutions. The final part consists of chapters 8 and 8.4 in which we discuss our findings and present the impact of the research done within this thesis.

In chapter 2 we introduce the privacy preserving n-party scalar product protocol. This is an extension to a 2-party scalar product protocol. As alluded to earlier, much research is focused on relatively simple scenarios with few parties but these approaches do not generalize well to more complex research questions and settings. This extension is a generalization that allows the privacy preserving scalar product protocol to be used in scenarios where multiple parties are involved. The privacy preserving scalar product protocol allows us to answer questions such as "how many individuals in the dataset have $age \geq 50$ and $weight < 75$" in a privacy preserving manner, even when the attributes $age$ and $weight$ are known at different parties. This allows us to use it as a crucial building block within more complex analysis.

With this crucial building block in place we move on to chapter 3. Using the scalar product protocol and synthetic data we build a Bayesian Network in a vertical scenario. This is a commonly used machine learning model, popular because of its explainability, the ease with which it can be understood even without needing a technical background, and its ability to incorporate established expert knowledge.

In chapter 4, we present a literature review in which we explore the idea of using Ensemble Learning in a federated setting. Ensemble learning has a number of advantages which make it a natural fit for a federated setting. However, as it turned out, this is currently an underutilized technique.

Having established that ensemble learning is underutilized, we create our own ensembles of Bayesian networks in chapter 5 using the VertiBayes algorithm introduced in chapter 3. Our experiments indicate that there is indeed considerable promise in this method.

Finally, we end the section on algorithms with chapter 6, in which we describe the Verticox+ algorithm. This is an extension to the existing Verticox algorithm which improves the privacy guarantees provided by solving a flaw within the original algorithm which makes it impractical in a real production setting.

The second part of the thesis consist of chapter 7. In this chapter, we discuss how big institutions, both commercial and public, get to define what privacy is, and the limitations and flaws this introduces within federated learning. We also suggest a number of improvements, which we hope will help improve matters significantly.

Finally, we discuss our findings in chapter 8 and present the impact of our research in chapter 8.4.

# 2

# Privacy Preserving n-Party Scalar Product Protocol

Adapted from: Florian van Daalen et al. "Privacy Preserving n-Party Scalar Product Protocol". In: *IEEE Transactions on Parallel and Distributed Systems* 34.4 (Apr. 2023), pp. 1060–1066. DOI: 10.1109/TPDS.2023.3238768.

## Abstract

Privacy-preserving machine learning enables the training of models on decentralized datasets without the need to reveal the information, both on horizontally and vertically partitioned data. However, it requires specialized techniques and algorithms to perform the necessary computations. The privacy preserving scalar product protocol, which enables the dot product of vectors without revealing them, is one popular example for its versatility. For example it can be used to perform analyses that require counting the number of samples which fulfil certain criteria defined across various sites, such as calculating the information gain at a node in a decision tree. Unfortunately, the solutions currently proposed in the literature focus on two-party scenarios, even though scenarios with a higher number of data parties are becoming more relevant. In this paper, we propose a generalization of the protocol for an arbitrary number of parties, based on an existing two-party method. Our proposed solution relies on a recursive resolution of smaller scalar products. After describing our proposed method, we discuss potential scalability issues. Finally, we describe the privacy guarantees and identify any concerns, as well as comparing the proposed method to the original solution in this aspect. Additionally we provide an online repository containing the code.

## 2.1 Introduction

Federated learning is a field that has recently grown in prominence due to increasing awareness of data privacy issues and data ownership as well as the rising need to combine data originating from different sources[102]. It is a thriving research field that promises to make it possible to apply machine learning algorithms (or any other data analysis) on multiple decentralized datasets in a collaborative manner[93]. This applies to both horizontally and vertically split data. Horizontally partitioned data describes the situation where different organizations collect the same information from different individuals (e.g. the same clinical data collected in multiple hospitals). Vertically partitioned data occurs when different organizations collect different information about the same individuals (e.g. insurance claims and hospital records).

In order to apply machine learning algorithms on decentralized data, various techniques have been proposed to run the necessary analyses in a privacy-preserving manner. The techniques for vertically partitioned data are generally referred to with the umbrella term of secure multiparty computation (SMPC)[202]. SMPC is a research field that focuses on developing methods to calculate functions on decentralized data without revealing the data to other parties.

Examples of the various proposed techniques are machine learning algorithms to train Bayesian networks[26], neural networks[51], or random forests[108]. These algorithms may rely on techniques such as secret sharing[17] and homomorphic encryption[135]. Both secret sharing and homomorphic encryption work at their core by transforming the original values $\alpha$ and $\beta$, owned by different parties, into transformed values $\gamma$ and $\delta$ such that $f(\alpha, \beta) = g(\gamma, \delta)$, thus making it possible to calculate the result of function $f(\alpha, \beta)$ by calculating a different

*The views expressed in this paper are those of the authors and do not necessarily reflect the policy of Statistics Netherlands.*

function $g(\gamma, \delta)$ without ever needing to reveal $\alpha$ or $\beta$. In the case of homomorphic encryption, this is achieved by using encryption schemes that are 'homomorphic' with respect to specific functions, allowing the user to calculate these functions using encrypted data[135]. In the case of secret sharing, the core concept relies on obfuscating the raw data with a secret share (e.g., a random number), and then applying calculations to the obfuscated data in such a way that the secret shares will cancel out in the end[17].

Other techniques focus on specific calculations that can be used as building blocks for machine learning algorithms, such as the scalar product (or dot product) of vectors. The scalar product is an integral part of various machine learning algorithms, such as neural network training[194]. Therefore, secure scalar product protocols have been widely studied in federated learning[50]. In addition, it can be used in combination with clever data representations to calculate various statistical measures in a privacy preserving manner, such as the information gain of an attribute, as well as to classify an individual using a decision tree in a federated setting[50]. More generally speaking the scalar product protocol can be employed to determine the size of a subset of the population that fulfils a set of criteria in a privacy preserving manner, even if the relevant attributes are spread across multiple data owners.

Because of its importance, multiple scalar product variants have been proposed. Du and Atallah proposed several methods for the scalar product[48, 13]. Du et al. also proposed a similar method for secure matrix multiplication to be used in multivariate statistical analysis[49]. Vaidya and Clifton[177] proposed a new method to alleviate the scalability issues of existing methods and used this method to determine globally valid association rules. Du and Zhan[50] proposed yet another alternative, with better time complexity than the method proposed by Vaidya and Clifton[177], and better communication cost than the methods proposed by Du and Atallah[48, 13]. Du and Zhan[50] then used it to train a decision tree in a federated setting. Goethals et al.[72] discovered certain privacy flaws in some of

the earlier mentioned protocols, and suggested an alternative with improved privacy guarantees. Shmueli and Tassa utilize a scalar product protocol to solve a problem with $n$ parties[163], however, it should be noted that they solely use the scalar product protocol to solve multiple independent 2-party sub-problems.

However, all these solutions focus on two party scenarios where the scalar product is concerned. Translating them to scenarios involving more than two parties is not straightforward, if at all possible. This is a significant drawback since in practice often three, or even more parties, can be involved.

In this study, we look at the method proposed by[50] and determine if, and how, it can be scaled to an arbitrary number of parties. This has applications for the various calculations which can (partially) be transformed into a scalar product problem mentioned before, such as calculating information gain or anything else that can be represented as a set-inclusion problem.

## 2.2 Method

In this section, we first introduce the notation used, then we describe the original solution proposed[50]. We will then try to naïvely translate the original solution to an $n$-party situation. This naïve translation will result in several left-over terms in the equations which need to be solved. We will then discuss how these left-over terms can be solved. We will illustrate the steps in this translation with a three-party scenario. Finally, we will give a formal definition for the $n$-party scenario.

In this paper, we use lowercase letters to denote scalars (e.g., '$s$'), uppercase for vectors (e.g., $V$) and uppercase with a bold face for matrices (e.g., '$\mathbf{M}$').

### 2.2.1 Original protocol

The original protocol[50] works as follows. Alice and Bob have different features on the same individuals and want to calculate the scalar product of their private vectors $A$ and $B$, both of size $m$ where $m$ is semi-honest commodity server we have named Merlin. The protocol consists of the following steps.

1. Merlin generates two random vectors $R_a$, $R_b$ of size $m$ and two scalars $r_a$ and $r_b$ such that $r_a + r_b = R_a \cdot R_b$, where either $r_a$ or $r_b$ is randomly generated. Merlin then sends $\{R_a, r_a\}$ to Alice and $\{R_b, r_b\}$ to Bob.
2. Alice sends $\hat{A} = A + R_a$ to Bob, and Bob sends $\hat{B} = B + R_b$ to Alice.
3. Bob generates a random number $v_2$ and computes $u = \hat{A} \cdot B + r_b - v_2$, then sends the result to Alice.
4. Alice computes $u - (R_a \cdot \hat{B}) + r_a = A \cdot B - v_2 = v1$ and sends the result to Bob.
5. Bob then calculates the final result $v1 + v_2 = A \cdot B$.

It should be noted that this protocol utilizes a secret sharing approach. Because of this, the extended $n$-party protocol will utilize the same secret sharing approach.

### 2.2.2 Naïve translation to a three-party scenario

For our three-party scenario we now have Alice, Bob and Claire who want to calculate the scalar product of their three vectors $A$, $B$, and $C$ of size $m$ as well as Merlin who will aid them in the calculation by fulfiling the role of commodity server. The first problem we encounter here is that $A \cdot B \cdot C$ does not result in a scalar, it results in another vector. This means it is impossible to simply chain the scalar product protocol. Hence, we must first translate our scalar product problem into a different form so it can be solved for multiple parties.

To do this we create three diagonal matrices, matrices where only the diagonal has non-zero values, $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ of size $m \times m$, using the original vectors to fill the diagonals. This allows us to calculate $\mathbf{A}\cdot\mathbf{B}\cdot\mathbf{C}$, the result of which is a matrix. To turn this back into a scalar we define a function $\varphi$ which allows us to calculate the sum of the diagonal of a matrix. This means we have translated our 2-party scalar product problem into a 3-party matrix product problem where we calculate $\varphi(\mathbf{A} \cdot \mathbf{B} \cdot \mathbf{C})$. This naïve translation has a similar form as the matrix multiplication method proposed by Du et al.[49] mentioned earlier in this article, however, it includes more than two parties and all of our matrices are diagonal matrices.

It should be noted that this matrix multiplication method cannot simply be used to replace the scalar product protocol, as this would result in individual level data being shared across parties. For example, when using the scalar product protocol to build a decision tree[50], we have diagonal matrices, and the diagonal only contains 0 and 1 values. It would be trivial to deduce which positions only contained a value of 1 at all parties based on the final result using the matrix multiplication approach, which would be a major breach of privacy, as this would allow one to know which individuals were selected.

Having successfully translated our problem into a form where we can work with three parties, we will now attempt to naively translate the protocol. First, it should be noted that Merlin should generate random diagonal matrices instead of vectors. Second, he needs to generate an extra matrix $\mathbf{R_c}$ and scalar $r_c$ to send to Claire. Third, we need to introduce an extra step into our protocol for Claire that is equivalent to step 4 in the two-party protocol. And last, wherever vectors owned by Alice and Bob are multiplied we must now multiply matrices owned by Alice, Bob and Claire. It should also be noted that whenever we are now multiplying matrices, we need to apply the $\varphi$ function to turn the resulting matrix into a scalar. Consequently, our naively adapted protocol will look as follows:

1. Merlin generates three random diagonal matrices $\mathbf{R_a}$, $\mathbf{R_b}$, $\mathbf{R_c}$

and two random scalars $r_a, r_b$. It then calculates a third scalar $r_c$ such that $r_a + r_b + r_c = \varphi(\mathbf{R_a} \cdot \mathbf{R_b} \cdot \mathbf{R_c})$. Merlin then sends $\{\mathbf{R_a}, r_a\}$ to Alice, $\{\mathbf{R_b}, r_b\}$ to Bob and $\{\mathbf{R_c}, r_c\}$ to Claire.

2. Alice calculates $\hat{\mathbf{A}} = \mathbf{A} + \mathbf{R_a}$ and sends it to Bob and Claire, Bob sends $\hat{\mathbf{B}} = \mathbf{B} + \mathbf{R_b}$ to Alice and Claire, and Claire sends $\hat{\mathbf{C}} = \mathbf{C} + \mathbf{R_c}$ to Alice and Bob.

3. Bob generates a random number $v_2$ and computes $u_1 = \varphi(\hat{\mathbf{A}} \cdot \hat{\mathbf{C}} \cdot \mathbf{B}) + r_b - v_2$, then sends the result to Alice.

4. Alice computes $u_2 = u_1 - \varphi(\mathbf{R_a} \cdot \hat{\mathbf{B}} \cdot \hat{\mathbf{C}}) + r_a$, then sends the result to Claire

5. Claire then computes $u_3 = u_2 - \varphi(\mathbf{R_c} \cdot \hat{\mathbf{A}} \cdot \hat{\mathbf{B}}) + r_c$. Claire then sends $u_3$ to Bob.

6. Bob then calculates the final result $u_3 + v_2 = \varphi(\mathbf{A} \cdot \mathbf{B} \cdot \mathbf{C}) - \varphi(\mathbf{R_a} \cdot \mathbf{R_b} \cdot \mathbf{R_c}) - \varphi(\mathbf{A} \cdot \mathbf{R_b} \cdot \mathbf{R_c}) - \varphi(\mathbf{B} \cdot \mathbf{R_a} \cdot \mathbf{R_c}) - \varphi(\mathbf{C} \cdot \mathbf{R_a} \cdot \mathbf{R_b})$[1]

As we can see our final result is not equal to $\varphi(\mathbf{A} \cdot \mathbf{B} \cdot \mathbf{C})$ because there are several left-over terms (i.e., $\varphi(\mathbf{R_a} \cdot \mathbf{R_b} \cdot \mathbf{R_c})$, $\varphi(\mathbf{A} \cdot \mathbf{R_b} \cdot \mathbf{R_c})$, $\varphi(\mathbf{B} \cdot \mathbf{R_a} \cdot \mathbf{R_c})$, and $\varphi(\mathbf{C} \cdot \mathbf{R_a} \cdot \mathbf{R_b})$).

### 2.2.3 Solving the left-over terms

The first left-over that should be solved is the left-over of the form $\varphi(\mathbf{R_a} \cdot \mathbf{R_b} \cdot \mathbf{R_c})$. The protocol will naturally result in a left-over term of the form $(n-2)\varphi(\mathbf{R_a} \cdot \mathbf{R_b} \cdot \mathbf{R_c})$ because we already add the various $r_x$ for each $x \in \{a, b, c\}$ once in step $3-5$, even in the naïve translation. We can solve this leftover term simply by replacing $r_x$ in step $3-5$ with $(n-1)r_x$, because $(n-1)\varphi(\mathbf{R_a} \cdot \mathbf{R_b} \cdot \mathbf{R_c}) = (n-1)(r_a + r_b + r_c)$. For example in step 4 instead of adding $r_a$ we will add $2r_a$ in the 3-party protocol.

[1] A full elaboration of the equation can be found in appendix 1

The remaining left-over terms are $\varphi(\mathbf{A} \cdot \mathbf{R_b} \cdot \mathbf{R_c})$, $\varphi(\mathbf{B} \cdot \mathbf{R_a} \cdot \mathbf{R_c})$, and $\varphi(\mathbf{C} \cdot \mathbf{R_a} \cdot \mathbf{R_b})$. These left-over terms all have the form of $\varphi(\mathbf{X} \cdot \mathbf{R_y} \cdot \mathbf{R_z})$, where $x, y,$ & $z$ represent the different parties Alice, Bob, & Claire, and each of the multiplicands always belongs to a different party (e.g., they are never of the form $\varphi(\mathbf{X} \cdot \mathbf{R_x} \cdot \mathbf{R_y})$). Furthermore, the combined term $\mathbf{R_y} \cdot \mathbf{R_z}$ is known by Merlin, hence this can be rewritten as $\varphi(\mathbf{X} \cdot \mathbf{M})$, where $\mathbf{M} = \mathbf{R_y} \cdot \mathbf{R_z}$ and is owned by Merlin. This means that this left-over problem can be simplified into a 2-party scalar product problem, where Merlin is one of the parties. More generally these left-over terms within an $n$-scalar product protocol are themselves $n - 1$, or smaller, scalar product problems. These smaller scalar product protocols need to be solved with additional commodity servers (i.e., Merlin cannot play that role because he is involved as a party). In section 2.3.2 we will discuss how many commodity servers are needed for a given $n$-party protocol.

With the left-over terms solved we can now create a fully translated protocol to our three-party scenario.

### 2.2.4 Correct adaptation to a three-party scenario

To allow Alice, Bob, and Claire to calculate $\varphi(\mathbf{A} \cdot \mathbf{B} \cdot \mathbf{C})$ the following protocol should be followed.

1. Merlin generates three random diagonal matrices $\mathbf{R_a}$, $\mathbf{R_b}$, $\mathbf{R_c}$ and two random scalars $r_a, r_b$. It then calculates a third scalar $r_c$ such that $r_a + r_b + r_c = \varphi(\mathbf{R_a} \cdot \mathbf{R_b} \cdot \mathbf{R_c})$. Merlin then sends $\{\mathbf{R_a}, r_a\}$ to Alice, $\{\mathbf{R_b}, r_b\}$ to Bob and $\{\mathbf{R_c}, r_c\}$ to Claire.

2. Alice sends $\hat{\mathbf{A}} = \mathbf{A} + \mathbf{R_a}$ to Bob and Claire, Bob sends $\hat{\mathbf{B}} = \mathbf{B} + \mathbf{R_b}$ to Alice and Claire, and Claire sends $\hat{\mathbf{C}} = \mathbf{C} + \mathbf{R_c}$ to Alice and Bob.

3. Bob generates a random number $v_2$ and computes $u_1 = \varphi(\hat{\mathbf{A}} \cdot \hat{\mathbf{C}} \cdot \mathbf{B}) + 2r_b - v_2$, then sends the result to Alice.

4. Alice computes $u_2 = u_1 - \varphi(\mathbf{R_a} \cdot \hat{\mathbf{B}} \cdot \hat{\mathbf{C}}) + 2r_a$, then sends the result to Claire
5. Claire then computes $u_3 = u_2 - \varphi(\mathbf{R_c} \cdot \hat{\mathbf{A}} \cdot \hat{\mathbf{B}}) + 2r_c = \varphi(\mathbf{A} \cdot \mathbf{B} \cdot \mathbf{C}) - \varphi(\mathbf{A} \cdot \mathbf{R_b} \cdot \mathbf{R_c}) - \varphi(\mathbf{B} \cdot \mathbf{R_a} \cdot \mathbf{R_c}) - \varphi(\mathbf{C} \cdot \mathbf{R_a} \cdot \mathbf{R_b}) - v_2$
6. The left-over terms $\varphi(\mathbf{A} \cdot \mathbf{R_b} \cdot \mathbf{R_c})$, $\varphi(\mathbf{B} \cdot \mathbf{R_a} \cdot \mathbf{R_c})$, and $\varphi(\mathbf{C} \cdot \mathbf{R_a} \cdot \mathbf{R_b})$ are solved by separate two-party scalar product protocols. The results are given to Claire and she computes $\varphi(\mathbf{A} \cdot \mathbf{B} \cdot \mathbf{C}) - \varphi(\mathbf{A} \cdot \mathbf{R_b} \cdot \mathbf{R_c}) - \varphi(\mathbf{B} \cdot \mathbf{R_a} \cdot \mathbf{R_c}) - \varphi(\mathbf{C} \cdot \mathbf{R_a} \cdot \mathbf{R_b}) + \varphi(\mathbf{A} \cdot \mathbf{R_b} \cdot \mathbf{R_c}) + \varphi(\mathbf{B} \cdot \mathbf{R_a} \cdot \mathbf{R_c}) + \varphi(\mathbf{C} \cdot \mathbf{R_a} \cdot \mathbf{R_b}) - v_2 = \varphi(\mathbf{A} \cdot \mathbf{B} \cdot \mathbf{C}) - v_2 = u_3$. Claire then sends $u_3$ to Bob.
7. Bob then calculates the final result: $v_2 + u_3 = \varphi(\mathbf{A} \cdot \mathbf{B} \cdot \mathbf{C})$

We have now successfully translated the two-party scalar product protocol into a three-party protocol.[2]

### 2.2.5 Full translation to an $n$-party scenario

The $n$-party protocol can be formalized as follows:

1. If $n = 2$, use the two-party protocol[50], else go to next step.
2. Let $\mathbf{D_1}, \mathbf{D_2}, \ldots, \mathbf{D_n}$ be the diagonal matrices containing the vectors owned by the $n$ parties.
3. Let $\varphi$ be a function that calculates the sum of the diagonal of a matrix.
4. $\mathbf{R_1}, \mathbf{R_2}, .., \mathbf{R_n}$ are random diagonal matrices generated by a commodity server Merlin.
5. Let $\varphi(\mathbf{R_1} \cdot \mathbf{R_2} \cdot \ldots \cdot \mathbf{R_n}) = r_1 + r_2 + \ldots + r_n$ where all but one of the $r_i$ terms are randomly generated.

[2] A practical example of a 3-party scalar product protocol can be found in appendix 2

6. Merlin shares the pairs $\{\mathbf{R_i}, r_i\}$ with the $i$'th party for each $i \in [1, n]$
7. All parties calculate $\hat{\mathbf{D}}_\mathbf{i} = \mathbf{D_i} + \mathbf{R_i}$ and share the result
8. Party 1 generates $v_2$.
9. Party 1 then calculates $u_1 = \varphi(\prod_{i=2}^{n} \hat{\mathbf{D}}_\mathbf{i} \cdot \mathbf{D_1}) + (n-1) \cdot r_1 - v_2$
10. For each other party $i$ calculate $u_i = u_{i-1} - \varphi((\prod_{x=1}^{n} \hat{\mathbf{D}}_\mathbf{x} | x \neq i) \cdot \mathbf{R_i}) + (n-1) \cdot r_i$
11. This results in $\varphi(\mathbf{D_1} \cdot \mathbf{D_2} \cdot .. \cdot \mathbf{D_n}) - \mathbf{L_1} - \mathbf{L_2} - .. \mathbf{L_n} - v_2$ Where $\mathbf{L_i}$ corresponds to leftover terms of the form $\varphi(\prod_{i=1}^{m} \mathbf{D_i} \prod_{j=m}^{n} \mathbf{R_j}$—$i \neq j)$, where all parties are involved, either as $\mathbf{D_i}$, providing their raw data, or as $\mathbf{R_j}$, using their random matrix, but never as both.
12. These leftover terms represent a scalar product problem of at most $n-1$ parties. Thus these sub problems can be solved separately using a smaller $n$-party scalar product protocol.
13. Solving these leftover terms allows party $n$ to calculate $\varphi(\mathbf{D_1} \cdot \mathbf{D_2} \cdot .. \cdot \mathbf{D_n}) - v_2 = u_n$
14. Party 1 can then calculate the final result $u_n + v_2 = \varphi(\mathbf{D_1} \cdot \mathbf{D_2} \cdot .. \cdot \mathbf{D_n})$

This allows us to calculate the scalar product for an arbitrary amount of parties. Pseudocode of the protocol can be found in algorithm 1. Now that we have shown that the protocol can be translated to a scenario with arbitrary $n$ we will discuss how the protocol scales as well as potential security issues in the next section.

### 2.2.6 Commodity server

The $n$-party scalar product protocol contains multiple sub-protocols of at most $n-1$ sized all of which involve data owned by the commodity server in the $n$-party scalar protocol. These sub protocols will need to use a commodity server as well. However, the original commodity

**Algorithm 1:** The n-party scalar product protocol

nPartyScalarProduct($\mathcal{D}$)
**Input :** The set $\mathcal{D}$ of diagonal matrices $\mathbf{D_1}..\mathbf{D_n}$ containing the original vectors owned by the $n$ parties
**Output:** $\varphi(\mathbf{D_1} \cdot \mathbf{D_2} \cdot .. \cdot \mathbf{D_n})$
**if** $|\mathcal{D}| = 2$ **then**
  return 2-party scalar product protocol($\mathcal{D}$);
**else**
  **for** $i \leftarrow 0$ **to** $|\mathcal{D}|$ **by** $1$ **do**
    $\mathbf{R_i} \leftarrow generateRandomDiagonalMatrix()$
  **end**
  Let $\varphi(\mathbf{R_1} \cdot \mathbf{R_2} \cdot .. \cdot \mathbf{R_n}) = r_1 + r_2 + \ldots + r_n$
  Share $\{\mathbf{R_i}, r_i\}$ with the $i$'th party for each $i \in [1, n]$
  $v_2 \leftarrow randomInt()$
  $u_1 \leftarrow \varphi(\prod_{i=2}^{n} \hat{\mathbf{D}}_\mathbf{i} \cdot \mathbf{D_1}) + (n-1) \cdot r_1 - v_2$
  **for** $i \leftarrow 2$ **to** $|\mathcal{D}|$ **by** $1$ **do**
    $u_i = u_{i-1} - \varphi((\prod_{x=1}^{n} \hat{\mathbf{D}}_\mathbf{x} | x \neq i) \cdot \mathbf{R_i}) + (n-1) \cdot r_i$
  **end**
  $y \leftarrow u_n$
  **for** *subprotocol* $\in$ *determineSubprotocols*($\mathcal{D}, \mathcal{R}$) **do**
    $y \leftarrow y-$ nPartyScalarProduct($subprotocol$)
  **end**
  return $y + v_2$
**end**
determineSubprotocols($\mathcal{D}, \mathcal{R}$)
**Input :** The set $\mathcal{D}$ of diagonal matrices $\mathbf{D_1}..\mathbf{D_n}$ of the original protocol. The set $\mathcal{R}$ of random diagonal matrices used in the original protocol
**Output:** The sets $\mathcal{D}_{subprotocol}$ for each subprotocol
**for** $k \leftarrow 2$ **to** $|\mathcal{D}|-1$ **by** $1$ **do**
  $uniqueCombinations \leftarrow selectKSizedCombosFromSet(k, \mathcal{D})$
  **for** $selected \in uniqueCombinations$ **do**
    $subprotocol \leftarrow \mathbf{D_i} | i \in selected + \mathbf{R_j} | j \notin selected$
    $\mathcal{D}_{subprotocols} \leftarrow \mathcal{D}_{subprotocols} + subprotocol$
  **end**
**end**
return $\mathcal{D}_{subprotocols}$

server Merlin cannot be reused as Merlin fulfils the role of data-owner in these sub protocols. In section 2.3.2 we will discuss what influence this will have as $n$ grows and how potential issues can be minimized.

## 2.3 Discussion

In this paper, we have translated an existing 2-party scalar product[50] protocol to an $n$-party protocol. We have shown that a naïve translation is insufficient. However, by using a more sophisticated approach, it is possible to adapt the protocol to work with an arbitrary number of parties. In appendix 2, a fully worked out example of the three-party protocol can be found. Appendix 3 provides references to a repository containing java and python implementations of the $n$-party protocol.

We will now discuss the security and privacy guarantees this $n$-party protocol provides as well as how the complexity scales as the number of parties grows and how practical it is to use this protocol.

### 2.3.1 Security

The proposed method requires a commodity server, which is a semi-honest trusted third party within the calculation. A semi-honest party is a party which executes its part in the protocol accurately, but may try to learn as much as it can from the messages it receives in the process[46]. In this section we will discuss the exact risks involved with this.

As a method that relies on secret shares generated by a semi-trusted third party, this protocol utilizes an approach similar to asymmetric encryption[155], with the individual secret shares performing the role of private keys. This limits the risks involved. However, the trusted third party does introduce a risk in itself.

The risk posed by requiring a semi-honest trusted third party to be the commodity server would be that several semi-honest parties could potentially cooperate with the commodity server in order to jointly learn private data of the other parties. It should be noted that this risk is higher in an Internet of Things (IoT) setting than in a formalized joint research setting. An IoT setting consists of many unverified devices and parties. A formal joint research setting allows all parties involved to verify, and enforce, for example by requiring audits and adding other legal agreements, the integrity of the other parties to a certain extent. This will minimize the risk in practice in this setting. While it would be preferable if privacy could be protected by design with technical solutions, there will always be a need for a certain degree of trust in the various parties involved and legal means are a perfectly acceptable way of achieving the required trust[93].

However, this does not remove the technical possibility of a joint attack when all parties are semi-honest. The local calculations done at a given node $i$ are always of the form: $u_i = u_{i-1} - \varphi((\prod_{x=1}^{n} \hat{\mathbf{D}}_\mathbf{x} | x \neq i) \cdot \mathbf{R_i}) + (n-1) \cdot r_i$. Where $\hat{\mathbf{D}}_\mathbf{x}$ is locally known by every data-owner participating in this protocol. However, $\hat{\mathbf{D}}_\mathbf{x}$ is unknown to the commodity server in this protocol. Assuming the node cooperates with the commodity server, they could then separate $\hat{\mathbf{D}}_\mathbf{x}$ into its components $\mathbf{D_x}$ and $\mathbf{R_x}$. Where $\mathbf{D_x}$ is private data belonging to a different party and $\mathbf{R_x}$ is the random diagonal matrix generated by the commodity server, thus learning $\mathbf{D_x}$. This is a serious concern. This issue is especially relevant in an IoT setting where the trustworthiness of the commodity servers and individual parties is very difficult to verify and enforce.

However, in a formal joint research setting, a sufficient level of trust can be achieved to minimize the risk of this attack by enforcing the commodity server to act as an honest party, not just semi-honest[**kairouz˙advances˙2019**][176] First, it is possible to simply enforce this using legal means and mandate it is honest, however this may not be accepted in practice. Second, it is possible to give all parties involved joint custody over the commodity servers,

thus allowing each party to individually verify the commodity server is completely honest.

Joint custody over the commodity servers could, for example, be achieved by allowing any party to execute independent audits of the commodity server and giving them a veto over the hardware and software setup used on the servers. Such a setup allows each party to individually verify that the commodity server is honest, which works because each party has a vested interested in ensuring the honesty of the commodity server to protect their own data. This should allow the parties to jointly guarantee the commodity server are honest, even if the individual parties themselves are semi-honest.

It is important to note that these security concerns, and the possible solutions, are the same regardless of the size of $n$. That is to say, our proposed $n$-party protocol is equally as secure as the original 2-party protocol proposed by Du and Zhan because the original protocol also uses a trusted third party as commodity server which as we just discussed is the vulnerability exploited in a collusion attack.

### 2.3.2 Scalability

The number of subprotocols will grow with a factorial order of growth with respect to $n$. The reason it scales in this manner is because the subprotocols have the form of $\varphi(\prod_{i=1}^{m} \mathbf{D_i} \prod_{j=m}^{n} \mathbf{R_j} — i \neq j)$. Where all parties are involved, either as $\mathbf{D_i}$, providing their raw data, or as $\mathbf{R_j}$, using their random matrix, but never as both. There will be $\frac{n!}{x!(n-x)!}$ such subprotocols for each $2 \leq x < n$.

These subprotocols will have $x$ $\mathbf{R_j}$ factors and $n - x$ $\mathbf{D_i}$ factors. For example, a three-party protocol will have the following 3 subprotocols involving 2 $\mathbf{R_j}$ factors: $\varphi(\mathbf{A} \cdot \mathbf{R_b} \cdot \mathbf{R_c})$, $\varphi(\mathbf{B} \cdot \mathbf{R_a} \cdot \mathbf{R_c})$ and $\varphi(\mathbf{C} \cdot \mathbf{R_a} \cdot \mathbf{R_b})$. A 4 party protocol will have 4 subprotocols involving 3 $\mathbf{R_j}$ factors: $2 \cdot \varphi(\mathbf{A} \cdot \mathbf{R_b} \cdot \mathbf{R_c} \cdot \mathbf{R_d})$, $2 \cdot \varphi(\mathbf{B} \cdot \mathbf{R_a} \cdot \mathbf{R_c} \cdot \mathbf{R_d})$, $2 \cdot \varphi(\mathbf{C} \cdot \mathbf{R_a} \cdot \mathbf{R_b} \cdot \mathbf{R_d})$, and $2 \cdot \varphi(\mathbf{D} \cdot \mathbf{R_a} \cdot \mathbf{R_b} \cdot \mathbf{R_c})$. As well as 6 subprotocols involving 2 $R_j$ factors:

$\varphi(\mathbf{A}\cdot\mathbf{B}\cdot\mathbf{R_c}\cdot\mathbf{R_d})$, $\varphi(\mathbf{A}\cdot\mathbf{C}\cdot\mathbf{R_b}\cdot\mathbf{R_d})$, $\varphi(\mathbf{A}\cdot\mathbf{D}\cdot\mathbf{R_b}\cdot\mathbf{R_c})$, $\varphi(\mathbf{B}\cdot\mathbf{C}\cdot\mathbf{R_a}\cdot\mathbf{R_d})$, $\varphi(\mathbf{B}\cdot\mathbf{D}\cdot\mathbf{R_a}\cdot\mathbf{R_c})$ and $\varphi(\mathbf{C}\cdot\mathbf{D}\cdot\mathbf{R_a}\cdot\mathbf{R_b})$.

This growth in subprotocols will have an effect on the scalability. We will discuss the two aspects in which this matters in the following two sections.

### Time and Space Complexity

The first aspect affected by the factorial order of growth is the time complexity of the protocol. The amount of direct subprotocols for an $n$-party protocol will be equal to $\frac{n!}{x!(n-x)!}$ for each $x \in [2; n]$. These subprotocols may also have further subprotocols themselves. Furthermore, the amount of messages that need to be send for a given protocol are as follows; 1 message needs to be send from the commodity server to each of the $n$ dataowners to share the relevant pair of $\{\mathbf{R_i}, r_i\}$. Each party then shares its matrix $\hat{\mathbf{D}}_\mathbf{i}$ with each other party, resulting in $n \cdot (n-1)$ messages. Finally each party has to share its subresult once, resulting in a further $n$ messages. This means a total of $n+n^2$ messages for a given protocol.

In order to put this into perspective we show the number of protocols as a function of $n$ in figure 2.1. In addition to this, the results of a small experiment measuring the runtime performance, where the $n$-party protocol was used to calculate the number of individuals fullfilling certain attribute requirements, can be found in figure 2.2. This experiment was run on a windows laptop using an Intel(R) Core(TM) i7-10750H processor with 16GB of memory and 6 cores. All parties had a local datastation on this laptop, no significant optimization was implemented.

As can be seen in figure 2.1 the required number of protocols and messages grow quickly as $n$ grows. This is a significant downside of this protocol. The results of the small runtime experiment further supports this, as the runtime does grow rapidly as the number of parties grows.

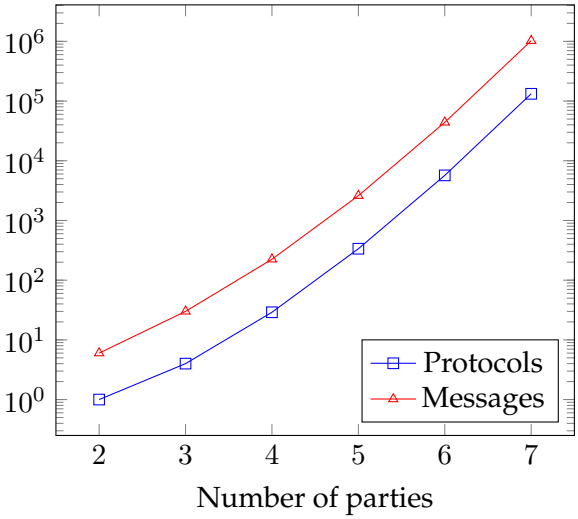


Figure 2.1: Rate at which the number of protocols and messages grow as functions of $n$. (y-axis in log-scale)

However, it also shows that the protocol can easily deal with larger datasets as dataset size barely influences the runtime. It should also be noted that there is considerable room for parallelization within the protocol, allowing the protocol to still be usable in practice. The following steps can be parallelized: first, every subprotocol can naturally be calculated in parallel as these are independent problems. Secondly every calculation in substep 11 detailed in section 2.2.5 can be calculated in parallel as well. Both options will reduce the running time of the protocol, considerably, allowing it to still be a practical solution in many settings. In addition to this, the actual use of the protocol within model training can be optimized, for example by running multiple $n$-party product protocols in parallel.

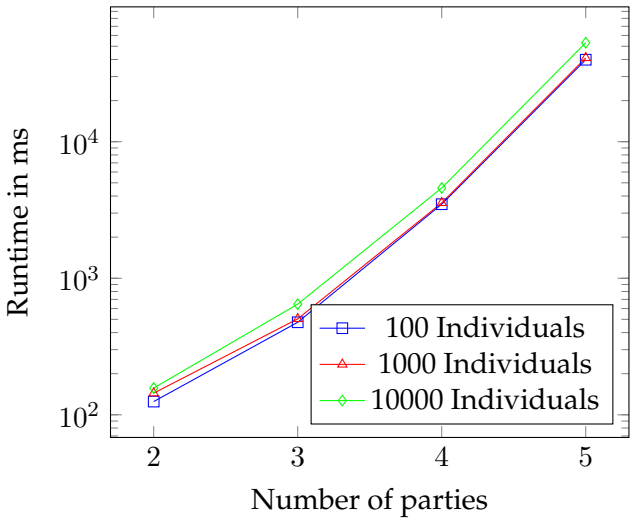


Figure 2.2: Average time in ms necessary to calculate the number of individuals fullfilling the requirements of 2-5 attributes divided over 2-5 parties for different population sizes. (y-axis in log-scale)

### Commodity Servers

It should be noted that these subprotocols need their own commodity server because no party may be both data owner and commodity server in a given protocol. Hence, we cannot reuse the original commodity server Merlin as it fulfils the role of a data owner in the subprotocols.

A naive solution to the problem posed by this need would be to set up sufficient commodity servers to deal with every sub-protocol. However, the amount of commodity servers needed will scale linearly with $n$, since a commodity server can be shared across all subprotocols of the same size. As the largest subprotocol in an $n$-party protocol will be an $(n-1)$-party subprotocol, and a two-party protocol will have no subprotocol, we will need $n-1$ commodity servers to solve an $n$-party

problem. While this might be manageable for small $n$ this eventually becomes untenable.

An alternative to this naïve solution would be to have the various parties double as commodity servers whenever they are not involved in a calculation themselves. To show that this is a viable, and safe solution, we will first divide the subprotocols into two categories. All subprotocols have the form $\varphi(\prod_{i=1}^{m} \mathbf{D_i} \prod_{j=m}^{n} \mathbf{R_j} | i \neq j)$, this can be further subdivided into subprotocols which contain only $1$ $\mathbf{D_i}$ term, which will have the form $\varphi(\mathbf{D_i} \cdot \mathbf{R_j} \cdot ... \cdot \mathbf{R_m})$, and subprotocols with multiple $\mathbf{D_i}$ terms.

The first category of subprotocols, which only contain one $\mathbf{D_i}$ term, can be solved by simply sharing the result of random matrices $\mathbf{R_j} \cdot \mathbf{R_j} \cdot \cdot \ldots \cdot \cdot \mathbf{R_m}$ with the owner of $\mathbf{D_i}$. $\mathbf{R_j} \cdot \mathbf{R_j} \cdot \cdot \ldots \cdot \cdot \mathbf{R_m}$ is itself a random matrix, provided there are at least two $\mathbf{R_j}$ factors involved, which cannot be used to leak any information. For example, the sub-protocols in the three-party protocol can be solved this way without requiring extra commodity servers. Doing this will also be faster than using the two-party scalar product protocol as it only requires a straightforward multiplication instead of the entire scalar product protocol. It should however be noted that the solution to this subprotocol may never be revealed to the commodity server that owns the $\mathbf{R_j}$ terms, as this would allow the commodity server to calculate $\mathbf{D_i}$. For example, if we are calculating $\mathbf{A} \cdot \mathbf{R_b} \cdot \mathbf{R_c}$ the result should never be revealed to Merlin, as revealing this would allow Merlin to learn Alice's data. This is of course also true in the original 2-party protocol.

The second category of subprotocol, which contains multiple $\mathbf{D_i}$ terms, can reuse one of the parties which is not currently providing data (i.e. a $\mathbf{D_i}$ term) as the new commodity server. This is secure as there is no need to reveal anything to the commodity server during the calculation. All it needs to do is generate and share the new $\{\mathbf{R_i}, r_i\}$ pairs for this subprotocol. As such, it never needs to see any (sub)results, and thus cannot reverse engineer anything. Additionally, the same party

should never be used twice as a commodity server in any set of subprotocols. That is to say, if Alice handles a 3-party subprotocol then she should not handle any child protocols that arise as a consequence of this specific 3-party subprotocol. Fortunately, it is easy to avoid this as there will always be at least one new party available to fulfil the role of commodity server for the new subprotocols.

While this is a practical solution to the need for multiple commodity servers, it does come with the major caveat that one must be certain no parties will attempt to cooperate to jointly learn private data of the other parties. As pointed out in section 2.3.1, the protocol is vulnerable to this type of attack.

## 2.4 Conclusion

In this paper, we have explained how the two-party scalar product protocol by Du and Zhan[50] can be scaled to an $n$-party scalar product protocol. We have illustrated how it works using a three-party scenario, after which we have given the formal definition of the protocol for any number of parties. This protocol can be used to calculate a number of metrics, such as the information gain of an attribute[50], in a scenario with an arbitrary number of parties. The benefit of being able to calculate such metrics is that it opens up the door for other more complex analysis. For example, using the information gain one can build a decision tree or apply feature selection.

Similarly, by using an innovative data representation the $n$-party protocol can be used to classify an individual in a privacy preserving manner using a decision tree[50]. By using other innovative data representations this $n$-party protocol could potentially be used for a wide variety of analysis and calculations. Aside from these benefits, which require the problem at hand to be rephrased into a scalar product problem, there is also the obvious benefit that it allows the use of the scalar product itself in an $n$-party scenario. This allows the use of any calculation that would normally rely on the scalar product in a classical

machine learning setting but which cannot be executed easily in a federated setting without a private $n$-party scalar product protocol.

While not appropriate in every scenario (scalability and the need for more commodity servers or semi-honest servers as the number of parties grows are a practical concern), we believe this is still a valuable tool in the federated learning toolbox.

### 2.4.1 Future work

For future work we would like to devise $n$-party protocols with better time complexity, as well as find a way to remove the vulnerability to joint-attacks introduced by the need for a commodity server.

In addition to this it would be valuable to investigate to which extend our extension to $n$ parties can be applied to the secure matrix multiplication proposed by Du et al.[49]. The protocol used for matrix multiplication is very similar to the 2-party scalar product protocol we extended, as such our extension should be of use when extending this matrix multiplication protocol.

Lastly, we are planning to utilize the $n$-party scalar product protocol to implement various federated algorithms so we can test the practical viability of this protocol in a real life setting.

# 3

# VertiBayes: Learning Bayesian network parameters from vertically partitioned data with missing values

Adapted from: Florian van Daalen et al. "VertiBayes: learning Bayesian network parameters from vertically partitioned data with missing values". en. In: *Complex & Intelligent Systems* (Apr. 2024). DOI: 10.1007/s40747-024-01424-0.

## Abstract

Federated learning makes it possible to train a machine learning model on decentralized data. Bayesian networks are widely used probabilistic graphical models. While some research has been published on the federated learning of Bayesian networks, publications on Bayesian networks in a vertically partitioned data setting are limited, with important omissions, such as handling missing data. We propose a novel method called VertiBayes to train Bayesian networks (structure and parameters) on vertically partitioned data, which can handle missing values as well as an arbitrary number of parties. For structure learning we adapted the K2 algorithm with a privacy-preserving scalar product protocol. For parameter learning, we use a two-step approach: first, we learn an intermediate model using maximum likelihood, treating missing values as a special value, then we train a model on synthetic data generated by the intermediate model using the EM algorithm. The privacy guarantees of VertiBayes are equivalent to those provided by the privacy preserving scalar product protocol used. We experimentally show VertiBayes produces models comparable to those learnt using traditional algorithms. Finally, we propose two alternative approaches to estimate the performance of the model using vertically partitioned data and we show in experiments that these give accurate estimates.

## 3.1 Introduction

Federated learning is a field that recently rose to prominence due to the increased focus on data-hungry techniques, privacy concerns and protection of the data[102, 93]. Using federated learning, it is possible to train a machine learning model without needing to collect the data centrally[102]. Since it rose to prominence, various techniques for training a model on centrally collected data have been adapted to be used on data that is either horizontally or vertically partitioned[93]. Data is said to be horizontally partitioned if multiple parties collect

the same variables though from different individuals, e.g., two hospitals who want to build a model to predict heart failure. It is said to be vertically partitioned when multiple parties collect different variables about the same individuals, for example, data from a hospital and from a health insurance company where both parties have unique variables about the same patients.

A type of model that can benefit from federated learning is Bayesian networks. Bayesian networks are probabilistic graphical models that have been widely used in artificial intelligence[137, 186, 27, 117]. They are popular because they can be built, verified, or improved, by combining data with existing expert knowledge. For example, medical doctors can manually create the network structure, ensuring it models already known dependencies correctly, while the conditional probability distributions are estimated from data. Thanks to its graphical representation and probabilistic reasoning, it is also a relatively intuitive model for non-technical personnel. This makes it very useful in scenarios where non-technical personnel needs to make decisions based on the model, for example when used as a tool to inform healthcare policies.

### 3.1.1 Existing literature on Bayesian networks in a federated setting

While research has been published on federated learning of Bayesian networks[201, 195, 206, 126], publications on Bayesian networks trained on vertically partitioned data (also referred to as heterogeneous data in the literature) are limited. One proposed method[126] only deals with horizontally partitioned data, and the other approaches[201, 195, 206] are all only capable of handling two-party scenarios. In addition to this none of the proposed methods can handle missing values in the dataset.

Unfortunately, these two aspects are important in practical applications. Missing data is a common problem in real world scenarios this

is especially true in federated scenarios where the different parties involved may have different data collection protocols and quality standards. In order to still have as large, and representative, a dataset as possible, records with missing data cannot be excluded.

The limitation to two-party scenarios is also a major downside in a federated setting. At its core federated learning attempts to combine data from as many data-sources as possible. Limiting algorithms to two-party scenarios runs directly counter to this goal.

### 3.1.2 Our contribution

In this article, we propose a novel method called VertiBayes to train Bayesian networks on vertically partitioned data, which can handle missing values as well as an arbitrary number of parties. In doing so we overcome the drawbacks the existing solutions have. This will allow us to train Bayesian networks in a vertically split federated setting, with an arbitrary number of parties, that are comparable to networks trained in a classical centrally trained setting.

The rest of the article is laid out as follows. First we will give some background information about Bayesian networks in general, and explain how these are trained in a classic scenario were all data is available centrally. Then we will describe our proposed method. After this we will describe the experimental setup we used to verify the federated model is similar to the centrally trained model. Followed by a discussion where we will go over aspects such as scalability and privacy concerns.

### 3.1.3 Bayesian networks

In this section, we will shortly explain how a Bayesian network is generally trained in a central setting.

Training a Bayesian network consists of two phases: structure learning and parameter learning. The first phase, structure learning, consists

of determining the structure of the graph (i.e., the set of links between variables) and can be done either manually, using expert knowledge, or automatically, using algorithms such as K2[29]. The second phase is the so-called parameter learning. In this phase, the conditional probability distributions (CPDs) for each node in the network are determined. Throughout this paper, we will focus on CPDs in the form of conditional probability tables (CPTs) as these are the most common form of CPD. In the next subsections, we will discuss how this is done in a centrally trained scenario. After which we will discuss how these methods need to be adapted for the federated scenario.

### Structure learning

The structure of a Bayesian network can be either determined manually or learnt using an algorithm. Here, we focus on the latter, since the former does not involve data analysis. One of the most popular structure learning algorithms is K2, which performs a heuristic search for a viable structure by scoring potential parent nodes for a given node and step-wise adding the highest scoring parent[29]. The scoring function used in K2 is described in equation 3.1 below.

$$f(i, \pi_i) = \prod_{j=1}^{q_i} \frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \prod_{k=1}^{r_i} a_{ijk}! \tag{3.1}$$

where $pi_i$ is the set of parents of node $x_i$. $q_i$ is the number of possible instantiations of the parents of $x_i$ present in the data. $r_i$ is the number of possible values the attribute $x_i$ can take. $a_{ijk}$ is the number of cases in the dataset where $x_i$ has it's $k^{th}$ value and the parents are initiated with their $j^{th}$ combination. $N_{ij} = \sum_{k=1}^{r_i} a_{ijk}$, is the number of instances where the parents of $x_i$ are initiated with their $j^{th}$ combination.

It is important to note that the resulting structure depends on the order in which nodes are introduced into the K2 algorithm. As such, it is possible to construct different structures for the same data.

### Parameter learning

There are two relevant scenarios to consider when performing parameter learning: with and without missing data. When there is no missing data, CPDs can be learned using the maximum likelihood[96]. To calculate the maximum likelihood for an attribute $X$ with a set of parents $Y$ we simply have to calculate: $P(X = x_i|Y_i = y_i) = N_(x_i, y_i)/N_(y_i)$ , where $N_(x_i, y_i)$ is the number of records where $X = x_i$ and $Y_i = y_i$ and $N_(y_i)$ is the number of records where $Y_i = y_i$. In the presence of missing data, the maximum likelihood for training a Bayesian network is commonly estimated using algorithms such as Expectation Maximization (EM)[44, 101]. The EM algorithm consists of the following two steps repeated iteratively until convergence is reached:

1. Estimate the likelihood of your data using your current estimates of the probabilities.

2. Update your estimates.

To estimate the likelihood of the current estimates in the E-step the following equation needs to be solved:

$$E = \prod_{i=1}^{n} P(d_i) = \prod_{i=1}^{n} \prod_{j=1}^{p} P(x_{ij}|y_{ij}) \tag{3.2}$$

where $n$ is the number of samples in the dataset, $p$ is the number of nodes in the network, $P(d_i)$ is the likelihood of the $i-th$ sample, $x_{ij}$ is the value of the $j-th$ node in the $i-th$ sample, and $y_{ij}$ is the set of values of the parents of the $j-th$ node in the $i-th$ sample. The appropriate values $P(x_{ij}|y_{ij})$ need to be selected from the current estimate of the CPDs based on the attribute values of this particular sample.

It is important to note that since this algorithm is a hill climbing-type algorithm, it can get stuck in local optima. Therefore, it is good practice to run the algorithm several times with different random initializations and use the best result[96].

## 3.2 Method

### 3.2.1 VertiBayes

In this section we present our novel method VertiBayes and explain how it handles the various additional hurdles and concerns that arise in a vertically partitioned federated setting. First, we will discuss how to perform structure learning. In the second subsection, we will discuss parameter learning. After this we will discuss the time complexity of VertiBayes. Finally, we will discuss the impact a vertically split scenario has on classification and model validation for Bayesian networks, as well as provide several solutions to deal with the problems that arise.

#### Structure learning

As mentioned previously, structure learning can be done using the K2 algorithm. In this subsection, we will discuss how to adapt the K2 algorithm to a vertically partitioned scenario. To solve this equation, the following information needs to be collected:

1. The number of possible values for the attribute $X$.
2. The number of instances that fulfil $X = x_i$ and $Y = y_j$, where $X$ is the child attribute, $x_i$ is a given value for $X$, $Y$ is the set of parent attributes, and $y_j$ is a given set of assigned values to $Y$. This needs to be calculated for every possible set $j$.

The number of possible values of attribute $X$ can be calculated trivially without revealing any important information to an external party in a vertically split federated setting as all relevant information is available locally at one party.

To calculate the number of instances that fulfil $X = x_i$ and $Y = y_i$ (the number of instances for each possible value of $X$ for each possible configuration of $X$'s parents) we have to calculate the number of

instances that fulfil certain conditions across different datasets. There are different approaches we could utilize to solve this problem.

For example, we could leverage $\epsilon$-differently privacy[55] to create a solution. This approach is relatively simple, however, it introduces noise, which can be problematic for smaller probabilities or for nodes with many parents, where a small amount of noise from each parent will eventually add up.

Alternatively we could attempt to solve it using homomorphic encryption[135]. Homomorphic encryption avoids adding any noise, but it is computationally expensive, especially as the K2 algorithm would require a fully homomorphic encryption scheme.

Finally a secret-sharing approach based in secure MultiParty Computation (MPC)[202] is an option. It is less computationally expensive than (full) homomorphic encryption, and does not introduce any noise. As such we propose to use this approach.

We propose to use the privacy preserving scalar product protocol to calculate the scalar product of vectors, one for each site, where each individual is represented as $1$ or $0$ depending on whether they fulfil the local conditions (in this case whether the child and parent nodes have the appropriate values). Earlier research has used this approach to calculate the information-gain when training a decision tree[50], which at its root, poses the same problem we face here. Additionally, this protocol also works in a hybrid setting, which allows our proposed method to be as versatile as possible.

Various variants of the privacy preserving scalar product protocol have been published[50, 48, 13, 72, 177]. Most of these focus on $2$-party scenarios but variants do exist for $N$ parties[37]. These methods have different advantages and disadvantages, such as different privacy guarantees and risks, different runtime complexities, and different communication cost overheads. Because of this, the preferred method will differ per scenario. A K2 implementation using one of these protocols

will have the same privacy guarantees and risks but will pose no additional privacy concerns beyond those posed by the chosen protocol.

### Parameter learning

During parameter learning, the actual CPDs will be calculated. There are two scenarios that need to be considered: with and without missing values. EM works under the assumption that data is missing at random or missing completely at random.

**Without missing values** As discussed earlier, parameter learning without missing values can be done by calculating the maximum likelihood for various attribute values. This means calculating for each node $i$, $N_{ij}$, the number of samples for each possible configuration of the parents of node $i$ and $N_{ijk}$ the number of samples for each possible configuration of the parents where the value is $k$, for each possible value of the node. $N_{ijk}$ can be calculated by simply summing the various $N_{ij}$ values. For the sake of performance it is advised to do this. These can be calculated using the scalar product protocol as explained earlier when describing the solution for K2. As such, performing parameter learning in a vertically split federated scenario with no missing values is not a problem and can be done without any significant additional privacy risks compared to the central variant beyond the risks involved in the scalar product protocol implementation used.

**With missing values** As mentioned in section 3.1.3 Expectation maximization requires that the appropriate values $P(x_{ij}|y_{ij})$ are selected from the current estimate of the CPDs based on the attribute values of this particular sample.

However, selecting the appropriate values $P(x_{ij}|y_{ij})$ can only be done when all child and parent node values are known. This is not possible in a privacy preserving setting if the child and parent nodes are spread over multiple parties. Conversely, even if it was possible to somehow select the appropriate values $P(x_{ij}|y_{ij})$, they may also never be revealed to anyone as it would be trivial to look up the parent and child node values in the CPD as the $P(x_{ij}|y_{ij})$ values will likely be unique.

It should be noted that a theoretical solution would be a layered approach combining homomorphic encryption with the privacy preserving scalar product protocol. However, due to the time complexity of each privacy preserving technique involved, the need to repeatedly execute the expectation step, and the fact this will need to be done for every single individual present in the training set, this is not practically viable.

Therefore, we conclude that the EM algorithm cannot be easily applied in a vertically split federated scenario without severe limitations. Instead, we propose the following three-step solution, which we have dubbed VertiBayes.

1. Treat"missing" as a valid value and train an intermediate Bayesian network using maximum likelihood on the training data.
2. Generate synthetic data (including "missing" values) using this intermediate Bayesian network
3. Train the final model on this synthetic data using the EM algorithm

As discussed earlier, parameter learning in a vertically split federated setting without missing values is possible with the privacy guarantees provided by the privacy preserving scalar product protocol used. Generating synthetic data by using this intermediate model also does not add any additional privacy concerns compared to a centrally trained

model, as this is a basic functionality of any Bayesian network. On the contrary, the final model has a reduced risk of data leak because it is trained on synthetic data[2, 93].

The proposed process, which is illustrated in Figure 3.1, allows us to train a Bayesian network in a vertically split federated setting with missing values without any additional privacy concerns compared to a centrally trained model. However, it should be noted that it is possible that a loss of signal may occur due to the three-step approach. In the experiment section, we will test if our proposed method avoids this potential pitfall.

### Time complexity when training a model in a federated setting

A major downside to federated learning is that the time complexity is usually considerably worse compared to the centralized setting. This is unavoidable due to the extra overhead created by communication as well as the increased complexity introduced by the privacy preserving mechanisms. In this subsection, we will discuss the time complexity of VertiBayes.

In our implementation, there are two important factors to take into account. The first is the number of parties $n$. This is a major bottleneck as the $n$-party scalar product protocol implementation we have used scales combinatorically in the number of parties.

The second important factor is the size of the CPDs that need to be calculated, as each unique probability that needs to be calculated requires a separate $n$-party scalar product protocol to be solved. As such, our implementation scales linearly in the number of probabilities that need to be calculated. The time complexity of various aspects for our implementation can be found in table 3.1.

Important to note is that the population size is not a main driver of the runtime. A relatively simple network trained on a small dataset,

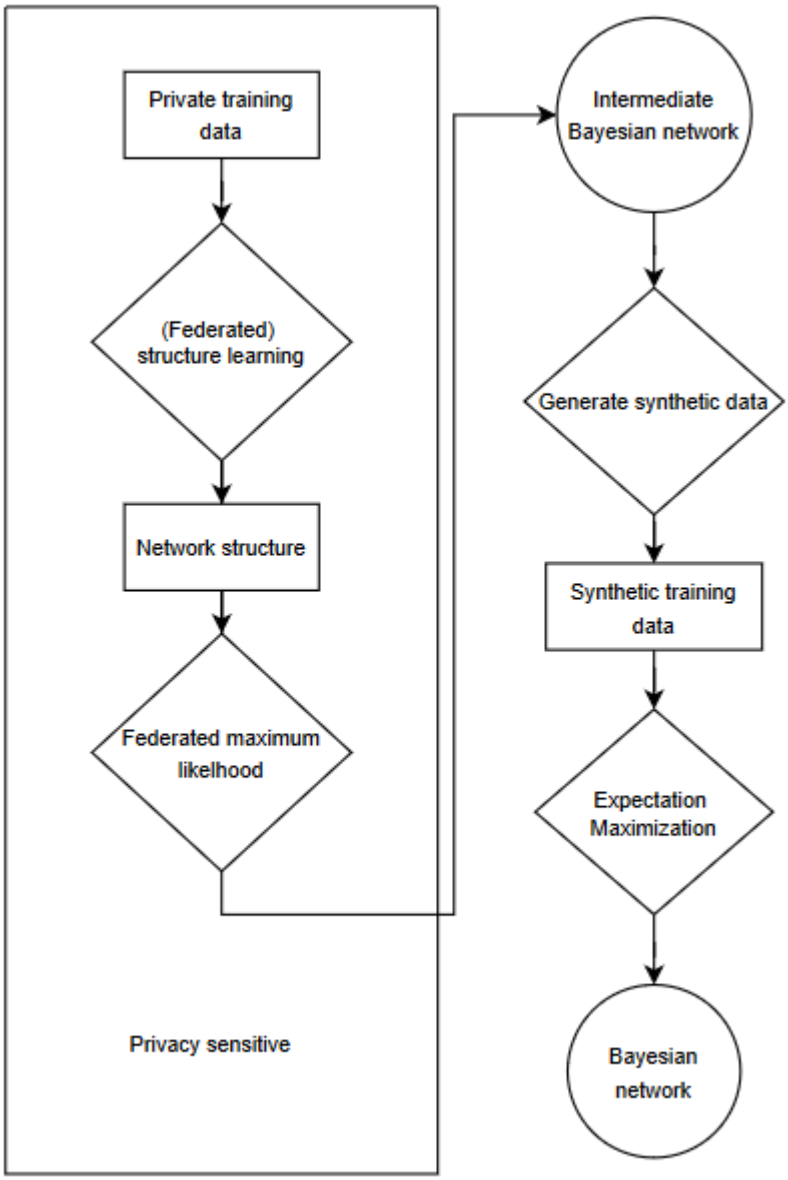


Figure 3.1: Training process for VertiBayes.

Table 3.1: Time complexity

| | |
|---|---|
| Number of Scalar product protocols | $O(m)$, where $m$ is the number of unique parent-child value combinations for which a probability needs to be calculate |
| Number of scalar product subprotocols per protocol | $\frac{n!}{(x!(n-x)!)}$, for each $x, 2 <= x <= n$, where $n$ is the number of parties involved in the protocol |
| Number of multiplications per subprotocol | $O(p * n * (n-1))$, where $p$ is the population size, and $n$ is the number of parties involved in the protocol |

but with a high number of unique attribute values will have a significantly longer runtime than a more complex network with few unique attribute values. This is because the overall time complexity is dominated by the number of scalar product protocols and subprotocols, which is independent of the population size, but dependent on the number of probabilities that need to be calculated.

Finally, it is important to note that there is ample room for parallelization to improve the running time as each scalar product protocol that is needed for VertiBayes is fully independent and can easily be run in parallel.

### Federated classification and model validation

The process of using the model to classify new instances in a federated setting is itself a complex problem that depends strongly on the type of model used. In this subsection, we will discuss the methods that are available to classify an individual in a vertically partitioned setting using a Bayesian Network and the implication this has for the validation of the model.

Classification of new samples using a Bayesian Network in a vertically partitioned federated setting suffers from the same issues as the expectation step in the EM algorithm. To classify an instance from vertically partitioned data, we need to select the appropriate probabilities from

the CPD. As discussed before, this is not viable while preserving privacy when parent and child nodes are split over multiple parties. This has major consequences for the validation of a new model in a federated setting.

As such, whenever possible the validation should be done using a publicly available dataset which avoids the need for privacy preserving measure during validation. If such a dataset is not available, we propose two different approaches,"Synthetic Cross-fold Validation" (SCV) and "Synthetic Validation Data Generation" (SVDG), to validate the model in a privacy preserving manner.

SCV uses the synthetic data generated by the intermediate Bayesian network as both training and validation data by executing the EM training using k-fold cross validation. However, it is possible that this results in overfitting on the synthetic data and therefore the performance estimate may be biased by the intermediate Bayesian network.

SVDG splits the private dataset into training and validation sets. It will then train a Bayesian network on the training set in a federated manner as normal. On the validation set, it will train a federated network using only the federated maximum likelihood approach. We can then use the Bayesian network trained on the validation set to generate a synthetic validation dataset. This approach reduces the risk of overfitting the previous approach suffered from but may lead to biased estimates if the synthetic validation set is not representative of the original validation set, for example because the test-fold was too small.

These approaches avoid leaking real data, but as mentioned, they may not be viable in practice. An illustration of the two new approaches can be seen in Figure 3.2

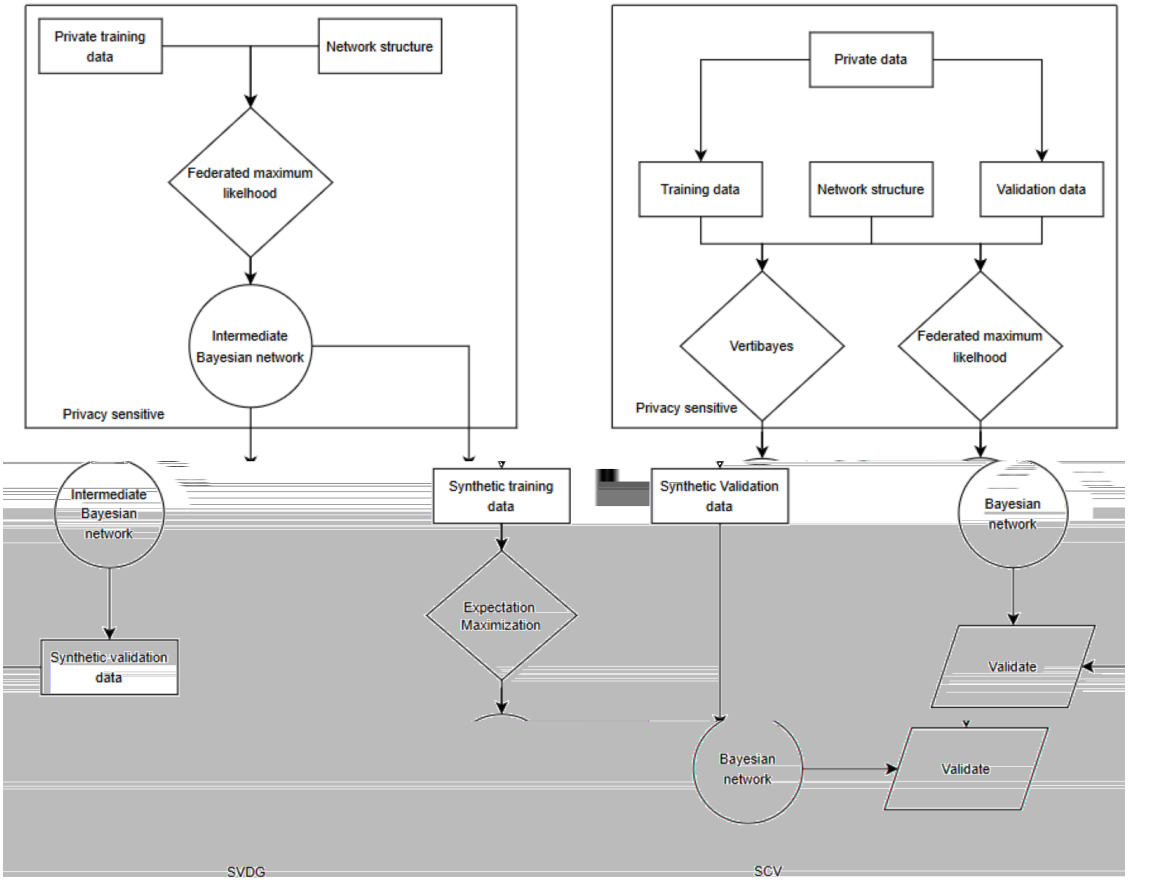


Figure 3.2: Flow diagrams for proposed validation procedures SCV (left) and SVDG (right).
SCV utilizes the first step of VertiBayes as normal, performing (federated) structure learning and federated maximum likelihood learning. The synthetic dataset generated during step 2 is split into a training and validation set. The training set is used as normal in step 3, which is then validated using the validation set.
SVDG splits the data before performing any training into a training and validation set. The training set performs VertiBayes as normal. While we only run step 1 and 2 on the validation set. The synthetic data generated is then used to validate the model that was trained on the training set.

#### Hyperparameters

There are a number of choices which need to be made when initializing a new run of VertiBayes. These choices represent the various hyperparameters that can be set. The choices are as follows:

- Will structure learning be done using a predefined structure based on expert knowledge, or by utilizing the K2 algorithm.
- If K2 is used for the structure learning, what are the maximum amount of parents a node may have.
- Is discretization of continuous variables done using predefined bins based on expert knowledge, or utilizing an automatic approach. There are different strategies possible for automatic discretization which may have their own hyperparameters.
- What validation strategy is chosen from among the options in subsection 3.2.1

The chosen structure learning approach can have a major impact on the performance of the resulting model. Similarly, the discretization approach can have a big impact as we will show in our experiments.

In the next section, we will perform experiments to validate that VertiBayes results in networks with similar performance as a centrally trained model. We will also verify if the two proposed approaches to validation give the same results as the validation on the public data, and if there are scenarios where they are inappropriate due to the aforementioned risks.

### 3.2.2 Experimental setup

In order to validate our proposed approach, we have implemented it in a combination of Java and Python[1] using Vantage6[124] and ran

[1]Our code can be found in the following two git repositories

a number of experiments. Vantage6 is an open-source infrastructure for privacy preserving federated learning which utilizes Docker. We compare the performance of our algorithm with a centrally trained Bayesian network using WEKA[69], a machine learning library written in Java.

The goal of the experiments is to show that the networks created by VertiBayes and the models created in a classic centralized scenario are the same. We did not compare results against other federated methods because 1) they cannot cope with missing data and 2) we use centralized learning as a baseline and want to show that VertiBayes performs equally well as a centralized approach

### Structure learning

To validate our federated implementation of the K2 algorithm we ran experiments using the Iris[41], Asia[100], Alarm[18], and Diabetes[167] datasets. As K2 is deterministic and dependent on the order in which the nodes are put into the algorithm we ensured this was the same for both the federated learning and centralized learning model and then compared the resulting structures.

### Parameter learning

In our experiments regarding parameter learning, we have used the Iris[41], Asia[100], Alarm[18], and Diabetes[167] datasets. In the case of the Iris dataset, we predict the ”label” attribute, for Asia we predict ”lung”, for Alarm we will be predicting ”BP” and for Diabetes we

- Main algorithm code: `https://github.com/MaastrichtU-CDS/vertibayes`
- Vantage6 wrapper code: `https://github.com/MaastrichtU-CDS/vertibayes_vantage6`

will predict "outcome". The Iris dataset contains $150$ samples, the Diabetes dataset contains $768$ samples, while the other two datasets contain data of $10.000$ samples. The Asia and Alarm datasets come with a predefined structure. The Iris dataset uses a naive Bayes structure. The Diabetes dataset also uses a predefined structure.

Both the Iris and Diabetes dataset contain continuous variables. For the sake of a fair comparison between the central and federated models, these were discretized into bins before starting our experiments, where each bin contains at least $10\%$ of the total population as well as a minimum of $10$ individuals. If the last bin cannot be made large enough to fulfil these criteria it is simply added to the previous bin. In the case of a dataset of less than $10$ individuals, the bin will simply contain all possible values. For our experiments the bins were predefined alongside the predefined network structure, but the bins can also be generated on the fly during the training of the federated model using the same discretization strategy. The simplicity of this strategy allows it to be executed without needing additional privacy preserving mechanics. However, it should be noted that this is not the best possible discretizing strategy. For example, a model using the Minimum Description Length method (MDL)[63] for discretization, or utilizing expert knowledge, might outperform this setup.

Slight variations of this discretization strategy were used in preliminary experiments. However, we will not list the results of those variants here as they produced similar results.

To test the effect of missing values we have done experiments where we randomly set $0\%$, $5\%$, $10\%$ and $30\%$ of the values to missing. The performance of the models is measured by calculating the area under receiver operating characteristics curve (AUC). The centrally trained model is internally validated using $10$-fold cross validation. The federated model is validated using the two different validation approaches described in the last section; it is also validated against a "public" central dataset, which is simply the left-out fold from the original dataset. All of these approaches use $10$-fold cross validation. We also compare

the Akaike information criterion (AIC)[6] values of the network for their original (private) training data. This is done to determine if there is any difference in the CPDs used by the two models. As mentioned before, the goal of the experiments is to get similar networks, as such we would expect the AUC and AIC of the networks produced by VertiBayes and the central approach to be similar. The AUC was chosen as a relevant metric because it is a powerful performance metric which naturally corrects for certain biases. For example, it does not have the same biases towards the majority class that accuracy has. AIC was chosen because it is a standard measure within Bayesian Network learning used to compare the complexity of the networks. This is important since we are not just interested in achieving similar performance, but wish to create similar networks.

For all of these experiments the data is randomly partitioned over two parties by dividing the original dataset into two equally large sets of attributes.

#### Scaling in the number of data-parties

To illustrate VertiBayes works when dealing with multiple parties the aforementioned parameter learning experiment was repeated for 3-8 parties using the Asia dataset. As this dataset contains 8 attributes this means the number of attributes varied from 4 to 1 attribute per party.

## 3.3 Results

The results of our experiments can be found in this section. First we will discuss the results of the structure learning experiments. After that we will cover the experiments regarding parameter learning.

### 3.3.1 Structure learning

The federated implementation of the K2 algorithm consistently resulted in the same structure as the centrally trained model for all datasets.

### 3.3.2 Parameter learning

Tables 3.2 and 3.3 show the results of our parameter learning experiments. It lists the AUC for the centrally trained model, as well as the AUC's for the various (federated) validation schemes. The AUC's of the federated model are listed in bold if the difference with the central model is larger than $0.05$. The AIC is shown for the centrally trained model, this is compared to the AIC for the federated model trained. The difference between the central and federated AIC is listed in bold if the difference is at least $5\%$. An AIC closer to $0$ is better. A negative percentage in the AIC difference column indicates the federated model performed better. The results are shown per dataset for differing levels of missing data; no missing data, $5\%$ missing data, $10\%$ missing data, and $30\%$ missing data. VertiBayes showed similar performance to the centrally trained model in all scenarios.

### 3.3.3 Time complexity

To illustrate the time complexity we kept track of the runtime of the parameter learning experiments. The results can be found in Figure 3.3. The relative runtimes are plotted versus population size, number of attributes, and total size of the CPDs that need to be calculated during the Maximum Likelihood stage of VertiBayes. As can be clearly seen in these graphs the performance depends mostly on the size of the CPDs, and is virtually independent from population size. Number of attributes does correlates with a longer runtime because more attributes often implies more probabilities will need to be calculated to fill all CPDs However, since the size of the CPDs also depend on how

Table 3.2: Results of the experiments.
The experiments are grouped per dataset. The AUC is represented for the centrally trained model, as well as the two federated validation schemes; "Synthetic Cross-fold Validation" (SCV) and "Synthetic Validation Data Generation" (SVDG). AUC values are listed in bold if they differ more than $0.05$ from the central model.

| | | AUC | | | |
|---|---|---|---|---|---|
| | | Training method | | | |
| | | Centralized learning | Federated learning | | |
| Dataset | Missing data % | Public validation | Public validation | SCV validation | SVDG validation |
| Alarm population size: 10000 | 0 | 0,91 | 0,91 | 0,91 | 0,91 |
| | 5 | 0,88 | 0,89 | 0,88 | 0,89 |
| | 10 | 0,85 | 0,86 | 0,86 | 0,86 |
| | 30 | 0,76 | 0,76 | 0,76 | 0,76 |
| Asia population size: 10000 | 0 | 1,00 | 1,00 | 1,00 | 1,00 |
| | 5 | 0,76 | 0,76 | 0,76 | 0,76 |
| | 10 | 0,69 | 0,7 | 0,7 | 0,7 |
| | 30 | 0,58 | 0,59 | 0,58 | 0,59 |
| Diabetes population size: 768 | 0 | 0,8 | 0,77 | **0,95** | 0,79 |
| | 5 | 0,79 | **0,74** | **0,92** | 0,76 |
| | 10 | 0,75 | 0,72 | **0,90** | 0,73 |
| | 30 | 0,61 | 0,57 | **0,80** | 0,6 |
| Iris population size: 150 | 0 | 0,99 | 0,98 | 1,00 | 1,00 |
| | 5 | 0,97 | 0,96 | 0,99 | 0,97 |
| | 10 | 0,9 | 0,89 | 0,95 | 0,92 |
| | 30 | 0,98 | 0,99 | 1,00 | 0,99 |

connected the network is and how many values each attribute can take it does not correlate perfectly.

### Scaling in the number of data-parties

The results of our experiments with the Asia dataset using multiple parties can be found in table 3.4. The number of parties has no meaningful impact on the performance of VertiBayes. The minor differences are due to the inherent variation introduced by several random factors within VertiBayes, such as the random nature of the synthetic data that is generated during the second step of VertiBayes.

Table 3.3: Results of the experiments.
The experiments are grouped per dataset. An AIC closer to 0 is better, a negative percentage in the AIC difference collumn indicates the federated model was better. This value is listed in bold if the difference between the federated AIC and the central AIC is at least 5%

| | | AIC | | |
|---|---|---|---|---|
| Dataset | Missing data % | Centralized learning | Federated learning | AIC difference |
| Alarm population size: 10000 | 0 | -340571 | -318469 | **-6,49%** |
| | 5 | -315856 | -313612 | -0,71% |
| | 10 | -340823 | -350576 | 2,86% |
| | 30 | -444297 | -444866 | 0,13% |
| Asia population size: 10000 | 0 | -22555 | -22559 | 0,02% |
| | 5 | -23430 | -23395 | -0,15% |
| | 10 | -24105 | -24090 | -0,06% |
| | 30 | -25517 | -25613 | 0,37% |
| Diabetes population size: 768 | 0 | -13593 | -14407 | **5,99%** |
| | 5 | -13699 | -14556 | **6,25%** |
| | 10 | -13761 | -14408 | 4,70% |
| | 30 | -14736 | -15136 | 2,71% |
| Iris population size: 150 | 0 | -1036 | -1022 | -1,33% |
| | 5 | -1243 | -1176 | **-5,37%** |
| | 10 | -1491 | -1022 | **-31,49%** |
| | 30 | -1099 | -1381 | **25,67%** |

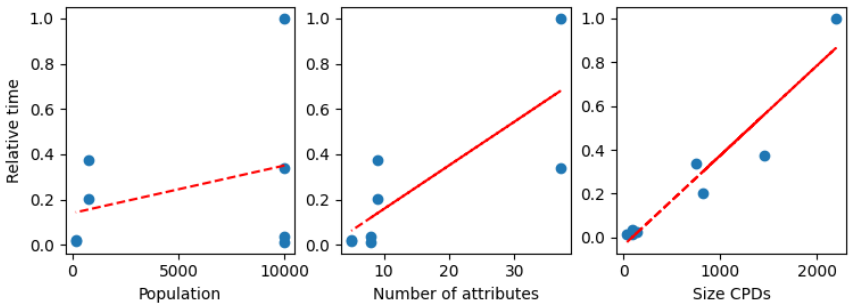


Figure 3.3: Relative runtime of VertiBayes using various datasets plotted versus population size, number of attributes, and total number of probabilities that need to be calculated during the Maximum Likelihood stage.

Table 3.4: Results of the repeated experiments with multiple parties for the Asia dataset.

| | | AUC | | | | |
|---|---|---|---|---|---|---|
| | Number of parties | Public validation | SCV validation | SVDG validation | AIC score | Running time MS |
| Asia Missing data: 0% | 2 | 0,97 | 1,00 | 1,00 | -22575 | 142292 |
| | 3 | 0,98 | 1,00 | 1,00 | -22564 | 144124 |
| | 4 | 0,98 | 1,00 | 1,00 | -22642 | 145343 |
| | 5 | 0,98 | 1,00 | 1,00 | -22600 | 144756 |
| | 6 | 0,97 | 1,00 | 1,00 | -22570 | 142854 |
| | 7 | 0,99 | 1,00 | 1,00 | -22530 | 144551 |
| | 8 | 0,98 | 1,00 | 1,00 | -22488 | 145143 |
| Asia Missing data: 10% | 2 | 0,70 | 0,71 | 0,70 | -23888 | 426849 |
| | 3 | 0,70 | 0,70 | 0,70 | -23837 | 426379 |
| | 4 | 0,70 | 0,69 | 0,71 | -23837 | 425364 |
| | 5 | 0,70 | 0,70 | 0,71 | -23918 | 427819 |
| | 6 | 0,70 | 0,71 | 0,71 | -24042 | 427391 |
| | 7 | 0,69 | 0,69 | 0,71 | -23871 | 432755 |
| | 8 | 0,70 | 0,70 | 0,70 | -24073 | 412467 |

## 3.4 Discussion

In this paper, we have proposed a novel method to train Bayesian networks in a federated setting using vertically partitioned data with missing values. The results of our experiments have shown that it is possible to perform both structure and parameter learning of Bayesian networks in such a setting with reasonable accuracy. Structure learning can be performed by adapting any of the existing structure learning algorithms to use a secure multiparty computation algorithm. In this study, we used a protocol within the K2 algorithm, but the same approach could be applied to other score-based algorithms or even constraint-based algorithms such as the PC algorithm[169]. Parameter learning requires one of two approaches. When there is no missing data present, the scalar product protocol can be used directly to compute the maximum likelihood, or when missing data is present the three-step solution described in section 3.2.1 using the EM algorithm can be applied. We will now discuss the performance of VertiBayes compared to a centrally trained model as well as the limitations in terms of scalability. Lastly, we will discuss the sensitive information that may be leaked by any Bayesian network and the limitations this brings in a federated setting.

### 3.4.1 Model performance and validation

Our experiments show that the resulting models produced by VertiBayes are comparable to the centrally trained models. As such, there is generally no meaningful difference in terms of AUC or AIC. The added privacy guarantees make it possible to train a model in a vertically partitioned setting. This takes away certain barriers with respect to data-sharing, allowing models to be trained on larger sets of data, which contains data that would have been inaccessible in a centralized setting. Utilizing this normally inaccessible data should lead to improved models in real life scenarios.

Furthermore, the experiments show that it is possible to validate the

model in a privacy-preserving manner despite it being impossible to efficiently classify an individual in a privacy-preserving manner.

It is however important to note that in certain edge-cases some validation approaches can show unreliable results. For example, the SVDG approach can cause problems when the test-folds are too small and the bins are generated on the fly while training, as opposed to working with pre-defined bins. If there are not enough individuals in the test-set to create multiple accurate bins on the fly this strategy will result in a loss of information. This was most notable when running the preliminary experiments with the Iris dataset, which is quite small.

Similarly, the model ran into problems using the SCV approach whenever the CPDs become too large because a node has multiple parents with a significant amount of bins. This is notable in the results of the Diabetes model. Certain nodes with multiple parents would end up with CPDs that contained more cells than there are individuals in the dataset. This led to an overestimation of the AUC if the SCV approach was used, as it overfits on the training data. However, this was not an issue when utilizing the SVDG approach, due to the stronger separation between training and validation data. Using larger bins can alleviate this problem to some extent. However, the bins cannot be made arbitrarily large as this will eventually cause a loss of information. Using expert-knowledge based discretization strategies tailored to each dataset, or a better automatic discretization strategy such as the MDL method mentioned earlier, would help avoid these problems.

These problems of overfitting and loss of information, show that it is extremely important to have an appropriate discretization strategy. So long as the potential pitfalls surrounding discretization are addressed, VertiBayes can be used to train and validate a Bayesian network in a vertical federated setting.

### 3.4.2 Scalability

Any algorithm that is adapted to a federated setting will be slower than the central counterpart due to the overhead caused by the protocols used to protect privacy. Our experiments confirmed the potential issues we brought up in section 3.2.1 when we discussed the theoretical time complexity.

The effects of population-size proved to be negligible. The effects of the complexity of the network structure, that is to say the number of nodes and links within the network, is only relevant in so much that it creates more probabilities to calculate. As expected, the size of the CPDs that had to be calculated had the greatest effect on the total runtime.

The number of parties also proved to not be very impactful. This intuitively makes sense as the bottleneck for VertiBayes lies in the calculation of the CPDs. When calculating the CPD for a particular attribute we can easily deduce the maximum parties involved in this calculation. For example $P(X|Y)$ will involve at most 2 parties in a vertical partitioned setting as it only involves 2 attributes. Similarly, calculating $P(X|Y,Z)$ will involve at most 3 parties. This means that the effect of the number of parties is naturally limited depending on the maximum amount of parents a node has in the network structure.

This does mean that there are practical limitations to using VertiBayes, as it may take too long to train a large or complex network. However, it should be noted that in certain settings a longer runtime might still be acceptable. For example, it is perfectly acceptable that training a model for use in a clinical setting takes an extended amount of time.

### 3.4.3 Sensitive information in published Bayesian networks

Publishing a Bayesian network, or any machine learning model, will reveal certain information about the training data, regardless of how the network is trained. When publishing a Bayesian network two important aspects will be revealed: the network structure and the CPDs.

The network structure will only reveal which conditional dependencies exist amongst attributes, which is not sensitive data in most scenarios. The CPDs on the other hand, can potentially be used to reconstruct individual level data from the training-set, when the probabilities in the CPD's are based on one, or a few individuals. An effective countermeasure is using k-anonymity[171] to ensure that each probability in the CPDs represents a minimum amount of samples and that no probabilities of $0$ or $1$ are present in the CPDs. Such probabilities make it considerably easier to deduce individual level data, and they can also lead to artifacts when using the network.

Lastly, a public Bayesian network can be used by one of the parties that participated in the training to guess (although not reconstruct) the data of the other parties based on their own data. Similarly, any third party with partial data can use the final model to estimate the missing values in his dataset. This is unavoidable and it should be taken into consideration when decisions are made about which models are to be made public.

These concerns imply that there are practical limits to what privacy preserving techniques should aim for. Trying to prevent any and all privacy issues using privacy preserving techniques during the training phase is futile when models are made public as the models themselves will always reveal some information.

### 3.4.4 Adapting the structure learning approach

In this article we choose to utilize the K2 algorithm to learn the network structure. Other approaches exist as well, these can be score-based[143] or constraint based[75]. Extending VertiBayes to utilize one of these alternatives is possible. At its core VertiBayes uses the privacy preserving n-scalar product protocol to calculate simple statistics such as the maximum likelihood. Any score based on constraint based structure learning algorithm that can is based on similar simple

statistics can be calculated in a privacy preserving manner in a similar way.

## 3.5 Conclusion

In this paper, we have proposed a novel approach to train Bayesian network parameters from a vertically partitioned data with and without missing values. This method can deal with an arbitrary number of parties, only limited by the runtime. We have shown that there are no additional privacy risks compared to a centrally trained model beyond the ones presented by the specific privacy preserving scalar product protocol implementation used. Our experiments show there are no meaningful differences in performance between models trained with VertiBayes and models trained centrally, provided continuous variables are adequately discretized. They also show it is possible to estimate the performance of a model with vertically partitioned data with a reasonable accuracy. As such, VertiBayes is a useful tool for training Bayesian networks in a vertically partitioned setting. Utilizing Bayesian networks in a vertically partitioned setting, will unlock normally inaccessible data, which will lead to improved models in real life scenarios.

## 3.6 Future Work

When using the model in a federated setting with a vertical split, it is currently not possible to efficiently classify or predict a new instance in a privacy preserving manner using a Bayesian network. It would be beneficial if a solution for this was found and implemented. In addition to this, VertiBayes could be improved by implementing more advanced discretization methods, such as MDL, in a vertically partitioned setting as our current implementation relies either on a very basic automatic discretization approach or the use of experts, which may

not be the best discretization approach possible. Additionally, the impact of different missing data mechanisms on our proposed approach should be investigated. In this article we used data that missed completely at random, however, we did not look at other missing mechanisms, such as missing at random. It would be worthwhile to investigate whether this significantly influences the performance of the resulting model. Lastly, it would be beneficial to run experiments in different real life scenarios to verify how VertiBayes scales in practice, especially with regards to the number of parties participating. As mentioned in subsection 3.4.2 we do not expect major in realistic scenarios, but this should be verified.

# 4

# Federated Ensembles: a literature review

Adapted from: Florian van Daalen et al. *Federated Ensembles: a literature review*. en. Dec. 2022. DOI: 10.21203/rs.3.rs-2350540/v1.

## Abstract

Federated learning (FL) allows machine learning algorithms to be applied to decentralized data when data sharing is not an option due to privacy concerns. Ensemble-based learning works by training multiple (weak) classifiers whose output is aggregated. Federated ensembles are ensembles applied to a federated setting, where each classifier in the ensemble is trained on one data location. The aim of this review is to provide an overview of the published literature on federated ensembles, their applications, the methods used, the challenges faced, the proposed solutions and their comparative performance. We searched for publications on federated ensembles on five databases (ACM Digital Library, IEEE, arXiv, Google scholar and Scopus) published after 2016.

We found 26 articles describing studies either proposing federated ensemble applications or comparing federated ensembles to other federated learning approaches. Federated ensembles were used for a wide variety of applications beyond classification. Advocates of federated ensemble mentioned their ability to handle local biases in data. In comparison to federated learning approaches, federated ensembles underperformed in small sample sizes and highly class imbalanced settings. Only 10 articles discussed privacy guarantees or additional privacy preserving techniques.

Federated ensembles represent an interesting alternative to federated averaging algorithms that is inherently privacy preserving. They have proven their versatility but remain underutilized.

## 4.1 Introduction

Federated learning (FL) brings machine learning to a setting where data is divided across various data-owners who wish to perform an analysis on their combined data while also keeping their data private[102]. In order to perform these analyses[55], FL relies on techniques such as $\epsilon$-differential privacy, homomorphic encryption[135], and multiparty computation (MPC)[202]. A commonly used approach in FL is federated averaging[118], where models are iteratively trained on local data and averaged into a global model. While this approach does perform adequately, it still has certain drawbacks. For example, it is possible to reverse engineer the input data based on the local updates that are communicated[191]. Furthermore, this approach generally does not explicitly take into account heterogeneity across different data sites, in other words, that the data may not be independent and identically distributed (IID) over the various parties. While federated models may be able to deal with certain biases, simply aggregating the updates may result in a model that performs well on the population at one party, but poorly on the population of a different party. Even more sophisticated aggregation schemes may not be able to avoid this.

A possible alternative to this approach is to use ensemble-based learning[131, 151]. Ensemble based learning works by training multiple (weak) classifiers. These classifiers work together using an aggregation scheme, such as majority voting, to jointly classify individuals. Ensembles rely on having a diverse set of classifiers[98], the intuition being that if one classifier misclassifies an individual, the other classifiers will cover for this mistake. This allows the ensemble as a whole to perform at a high level even when the individual classifiers are relatively weak.

*The views expressed in this paper are those of the authors and do not necessarily reflect the policy of Statistics Netherlands.*

Federated ensembles[182], ensembles of classifiers derived from local data from multiple parties in a federated setting, are an intuitive fit for a federated setting, as they naturally fulfil the various requirements this setting imposes. The first requirement fulfiled by ensembles is the need for privacy, since ensembles avoid the need to share any information beyond the local model. This removes the need to share any raw data and provides a baseline level of privacy.

A second major requirement in federated learning is the handling of heterogeneous data (not IID) across the various parties. Local biases in the population of each party might exist and it might be important for them to be reflected by the model. Ensembles are capable of dealing with non-IID data[33] in such a setting. Furthermore, an ensemble can still take advantage of local quirks and avoid overfitting to the data owned by a dominant data-provider, which is something "traditional" federated learning models may find difficult.

A third requirement is to keep running time complexity, as well as the communication overhead incurred by coordination of multiple parties, within manageable bounds. The need for communication in an ensemble is kept to a minimum, as only the local classifiers need to be shared. On top of that, the local training can be done in parallel, keeping the running time to a minimum.

Ensembles also suffer from limitations. Traditionally mentioned downsides are their reduced interpretability[68], increased complexity[157], and the fact that finding the right combination of models in the ensemble is more of an art than a science[8]. Ensembles rely on combining the output of multiple weak, but diverse, classifiers to be accurate. If the classifiers are not diverse enough, the ensemble will not work well.

In this literature review, we will provide an overview of the published literature on federated learning using ensembles, their applications, the methods used, the challenges faced, the proposed solutions and their comparative performance. While previous literature exists that

compares FL with ensembles, these publications are one-off comparisons in specific scenarios and, to the best of our knowledge, no systematic review on the subject has been published yet.

### 4.1.1 A short introduction to federated learning

As briefly mentioned before, FL is a machine learning approach that can be used in settings where data is spread over multiple parties. This data may not be collected centrally due to privacy concerns. Since the data cannot be centrally collected to run an analysis on, users need to bring the analysis to the individual parties and run calculations locally. Results of these local calculations can then be shared in a privacy preserving manner to execute a complex analysis[43]. An illustration of this approach can be found in figure 4.1. This is often done as part of an iterative process.

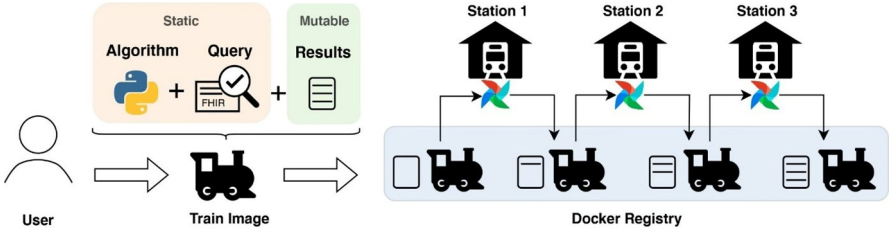


Figure 4.1: The personal health train approach[43]; instead of bringing the data to the researcher the researcher brings an algorithm to the data. Results are combined in a privacy preserving manner, and the researcher only sees the end result.

Broadly speaking, the data can be split up in three different ways: horizontally, vertically, or in a hybrid manner. Data is said to be horizontally split if different parties have access to the same set of attributes, but for different subsets of the global population. It is said to be vertically split if the different parties have data concerning the same population, but each party has a different set of attributes. A hybrid scenario

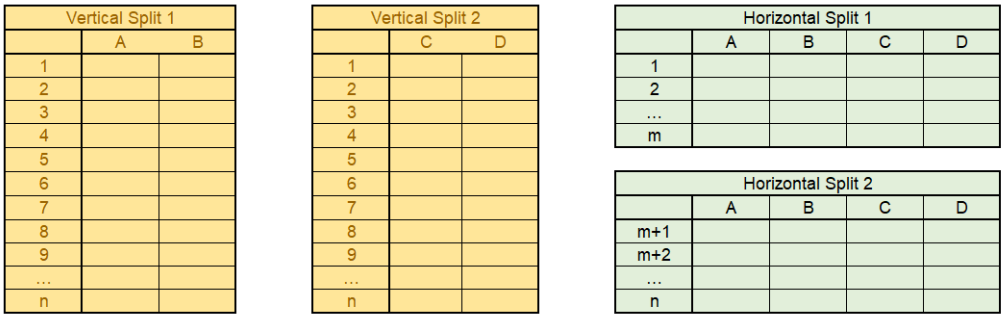


Figure 4.2: Illustrative examples of different data splits.

includes both types of splits at once. A horizontal and vertical scenario are illustrated in figure 4.2.

Horizontally split scenarios are often considered easier as averaging the local results from each party is a straightforward solution that can be used to apply several machine learning algorithms in these split settings[118]. However, averaging local results is not a viable solution for vertical splits. As a result algorithms for vertically split scenarios often need to be more complex to properly preserve privacy.

### 4.1.2 A simple taxonomy of federated ensembles

When creating a taxonomy for ensemble learning one traditionally focuses on the following questions[149, 74].

- What (mix of) base-classifier(s) is used?
- Which voting scheme is used to combine the individual votes into the final classification?
- How large is the ensemble?
- How is diversity among classifiers within the ensemble ensured?

A taxonomy of federated ensembles will largely focus on the same questions. Additionally, since federated ensembles specifically take advantage of the inherent data split present in a federated setting the question of how to ensure diversity is simplified.

The most basic version of a federated ensemble will simply create a classifier per party present in the federated scenario. This will naturally ensure a certain level of diversity among the classifiers. However, it is possible to improve on this diversity if the parties can be divided into distinct groups. In this case it may be interesting to implement an ensemble with a classifier for each such group. An illustrative example can be found in figure 4.3. This usage of the natural split present in a federated setting and the potential grouping of parties is the main aspect on which a taxonomy of federated ensembles will differ from a taxonomy of classical ensembles.

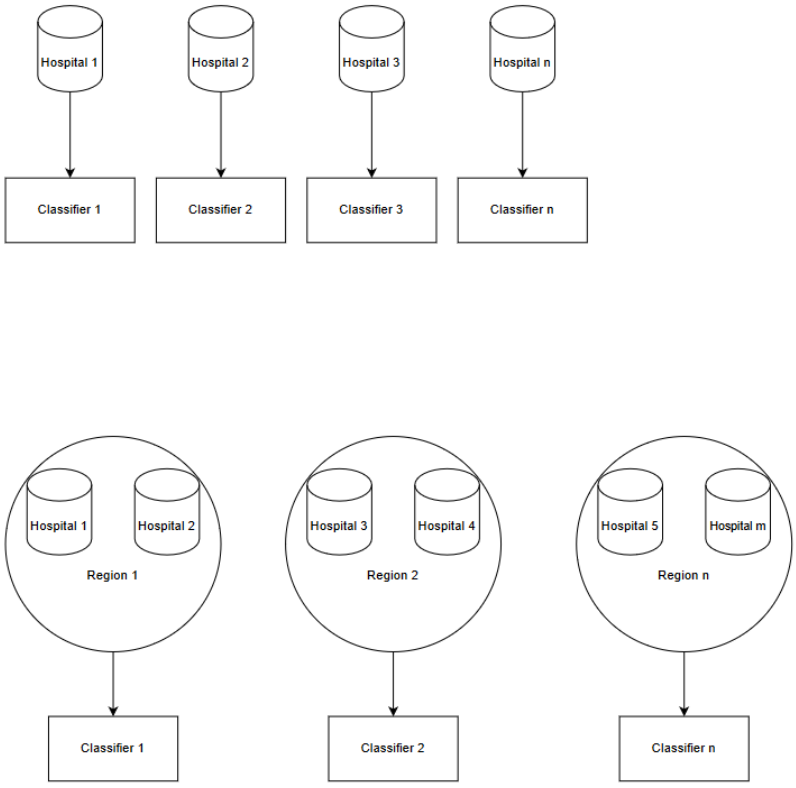


Figure 4.3: Illustrative example of two possible sets of base classifiers that can be used to create a federated ensemble. One set uses a classifier per party, the other set groups the parties based on their geographical region and then creates a classifier per region.

## 4.2 Methods

### 4.2.1 Sources

The following five databases are searched: ACM Digital Library, IEEE, arXiv, Google scholar and Scopus. We include arXiv because we wanted to include relevant preprints that have not been peer-reviewed yet. The search was executed on 19/7/2021.

### 4.2.2 Query

The query used combines the terms "federated learning" "ensemble*" with the AND operator. These terms are searched in the full text, references and meta-data of the articles.

### 4.2.3 Exclusion and inclusion criteria

Articles are included in the literature review if they discus the use of federated ensembles, defined as ensembles formed by classifiers learnt from local data. These articles either position federated ensembles as a solution to a particular problem, or use federated ensembles as a comparator. We use the following exclusion criteria to screen the records found.

- Anything published before 2015 is excluded, because the term "federated learning" was introduced in 2016 by Google.
- Everything that is not a scientific journal or conference proceedings, such as a book, was excluded.
- Articles that do not discuss federated ensembles, but were included in the search results for the following reasons:
    1. The single appearance of any keyword or any synonym, is in the references, or one-time mentions to these references (i.e. X et al. used ensembles to solve this).

2. The keywords are used as a polysemy; i.e. the word ensemble in a musical context can be used to refer to a band or orchestra[16].
3. The keywords are used in two independent contexts; i.e., a publication on cryptography mentioning that an ensemble-based attack can be effective against a federated system; however, the attack itself is not federated.
4. Only mention federated ensembles as potential future work.
5. Publication is a literature review.
6. The publication discusses an ensemble based technique where models are trained. locally, but these are then aggregated into one model[26]. These can be considered "Ensemble inspired federated techniques".
7. Federated implementations of Ensemble classifiers, such as random forest[108], that do not utilizes the federated nature of the data to attempt to gain an advantage. This can be considered "Non-federated Ensembles".

In order to determine if a publication was eligible, we first screened the title and abstract. Due to the limited information available in the abstract and title, it was often required to also quickly scan the full text for the relevant keywords. The PRISMA diagram of our search results can be seen in figure 4.4.

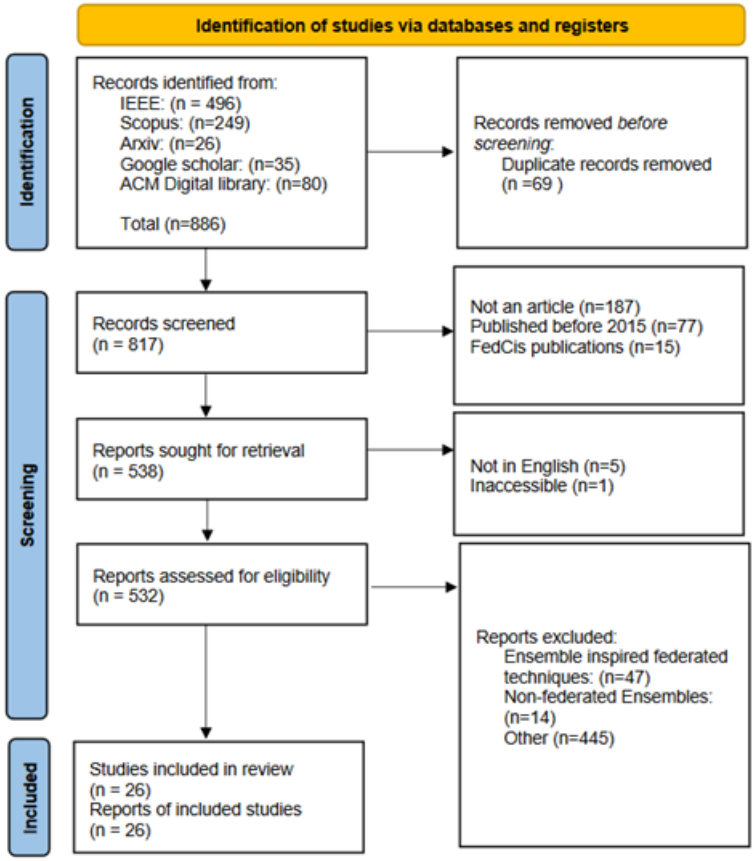


Figure 4.4: PRISMA diagram

### 4.2.4 Data items

First, we are interested if the publication focuses on horizontal or vertical partitioned data. Second, we want to know in which scenarios the proposed methods are appropriate, as well as what their goals are. Furthermore, we want to know what privacy guarantees are given, if any. Lastly, we are interested in the experimental setup; what type of data was used, how many parties were involved, the population size of the test setup, the performance of the models, etc. In addition to these more generic aspects, we also want to know the specific details of the ensembles. We investigate if they require specific base-classifiers and what voting schemes are used.

## 4.3 Results

We found 886 articles in total during our search, 860 of which were excluded, resulting in 26 articles that were fully reviewed. We will discuss the results of the reviews in the following sections.

### 4.3.1 Thematic grouping

The initial screening allows us to group the publications resulting in the following two relevant categories:

1. Federated ensembles: each party maintains its own model, these are combined into an ensemble (e.g., via voting). This category we are most interested in as it represents publications that propose new federated ensemble solutions.

2. Federated learning is compared against federated ensembles. This category is interesting because it directly compares federated ensembles with other federated learning techniques, and as such may give an insight as to when using ensembles in a federated setting is a good idea.

The papers in these two groups were considered for inclusion in this review. These publications were read in detail.

In the following subsections, we will present our findings about the final set of selected papers. First, we will broadly discuss noticeable trends in the two categories we have determined to be of interest. Then we will go more in depth about how they handled privacy and their experimental validation. Lastly, we will discuss the notion of combining "standard" federated learning with federated ensembles.

### Federated ensembles

We will first discuss the publications which propose federated ensembles, that is, specifically take advantage of the naturally occurring split in data to build the ensembles by training one or more local models per data-owning party. This group of publications contains 20 out of the 26 publications included in this review.

A summary of the relevant aspects of these publications can be found in table 1.

There is a varied set of goals in the publications describing federated ensembles. These do not only discuss straightforward classification tasks, but also attempt to solve other problems. Other subjects included learning or sharing various forms of information in a privacy preserving manner, such as; generating synthetic data which can be used for further analyses without fear of leaking privacy sensitive data[66], sharing labels[115], learning hyperparameters[160], and fulfiling a segmentation task[62]. Furthermore, there are publications focused on providing protection against various attacks[24]. Some publications utilize ensembles to deal with unreliable up-time of devices[162]. In addition to this, ensembles are utilized to create dedicated classifiers for subpopulations or even entirely separate complex subtasks[127]. Lastly, there are publications which simply recommended certain approaches to learning classifiers[127, 182, 132].

Another noteworthy aspect is that the ensembles in these publications are often specifically employed to handle heterogeneity in the data. This can be as simple as capturing small, but significant, local biases in the population. There are also attempts to deal with more complicated forms of heterogeneity such as open set problems, where each node can have a different set of locally known class labels. Lastly, there are ensembles that were created out of dedicated "expert" classifiers that focus on specific class labels.

Several papers combine both federated learning and ensemble learning. Instead of training a model for each party, they propose dividing the number of parties into multiple groups, train a model per group (using a federated approach such as federated averaging) and then combine these models in an ensemble.[24] did this to protect against byzantine attacks. The logic behind their method being that this way only one classifier in the ensemble would ever be poisoned if a node got corrupted. Thus, minimizing the risk of poisoning attacks.[127] did this to create several layers of (ensembles of) classifiers which handle more and more specific tasks. For example, a system that is being trained to recognize handwriting could consist of a classifier trained to recognize different types of scripts (Latin vs. Korean), combined with a classifier trained to recognize a specific individual's handwriting. This approach results in a more robust model that can deal with complex tasks while providing added security as a corrupted, or broken, party will only ever affect one model in the ensemble, thus limiting the potential damage it can do.

### Comparisons between federated ensembles & other types of federated learning

Next we will discuss the papers which explicitly compare federated ensembles against other types of federated learning. These articles use a basic ensemble setup (e.g. without feature selection or hyperparameter tuning) and compare it to either an established federated setup (e.g.

fedAVG) or their own federated proposal. A summary of the relevant aspects of these publications can be found in table 2.

All six articles claim federated learning outperforms federated ensembles. Two papers acknowledge multiple reasons why their experimental setup may not be optimal for ensembles, such as a small local population size and potential biases in the local populations and admit this may be a potential explanation for their relatively poor performance[187, 52]. The remaining four papers do not give any such explanations and argue that ensembles are a poor choice due to the inherent heterogeneous nature of the data in a federated setting. This is in direct contrast to the papers discussed in the previous subsection, which explicitly claim federated ensembles work especially well with heterogeneous data.

### 4.3.2 Privacy

Surprisingly, 16 out of the 26 papers included do not discuss privacy at all or mention it only in passing, despite privacy being a major concern within federated learning. Those that mention it, utilize a number of different ways to attempt to protect privacy.

First, the papers that do discuss privacy agree that the use of federated ensembles naturally limits the amount of data to be shared, which is a significant advantage. Most only share the final local models, but some papers also share some additional statistics.[196] shares a minimal amount of raw data, which is covered by a privacy budget. On the other hand,[10] shares some statistics that are used to determine if a given classifier would be relevant for a given individual e.g., "this classifier was trained on male patients". The authors acknowledge that this is a potential leak, but they claim that these statistics are safe to share.

In addition to this, three of the methods proposed do not even share the final model. They only share the predictions made for a new individual that needs to be classified. In these cases, privacy is also fur-

ther amplified by utilizing differential privacy (for example, adding noise to the predictions under the assumption the noise will average out within the ensemble), as well as homomorphic encryption to obfuscate the partial predictions of the local classifiers.

An example of this is[28], which attempts to hide as much as possible, reveals only the final classification, and keeps even the models secret. This secrecy is justified because expert knowledge about specific attributes could be used to deduce information. For example, if a node "suspected fraud" is present in a decision tree it stands to reason that the child with fewer individuals corresponds to true, as only a small subset of the population is ever suspected of fraud. However, it does note that it should still be possible to inspect the models locally, for example to ensure legal compliance. For example, one should be able to validate a model does not discriminate on the grounds of race.

Finally, some of the methods require an extra step to align the classifiers in the ensemble, which may have privacy implications. For example,[146] acknowledges that the need to align its experts can result in a potential leak. However, the authors claim that this is a sufficiently difficult task that in practice it will not be a concern.

### 4.3.3 Experimental validation

The experimental validation in the various papers shows the versatility of ensembles on a wide range of classification problems on different types of data as the majority show successful experiments where the ensembles are compared to other approaches. The reviewed papers work with both tabular (17 articles) and image data (12 articles). Horizontal (20 articles) and vertically split data (1 article), as well as a combination of the two (3 articles) can be dealt with by the ensembles. The number of parties involved in the experiments varies greatly, ranging from two to 250. Most articles focus on 2-10 parties and the articles with 20+ parties are all focused on horizontal partitioning. There are also multiple experiments that test the performance of ensembles

when the population is heavily skewed across parties. Both in the sense of one class-label being over/underrepresented at a certain party as well as a given party having a much greater population.

The publications comparing ensembles with federated learning do show it is possible to create scenarios where the individual classifiers are so weak the ensemble as a whole still cannot compete with the larger federated classifier[78, 103, 52, 187]. It should however be noted that in these scenarios the local population was either extremely small, very imbalanced resulting in overfitting on the majority class[78, 52], or it was acknowledged that the ensemble would have benefited from a different experimental setup[187]. Several of the publications acknowledge these issues, while the others do not discuss these aspects of their experimental setup. Regardless of this acknowledgment, these articles do show a weakness that needs to be considered. If the local population is extremely small or imbalanced, a small ensemble consisting of only a few classifiers is not going to perform adequately.

| Year & Author | Base model | Voting scheme | Data split | Appropriate scenarios/Goals |
|---|---|---|---|---|
| [66] | Differentially private decision trees | Majority voting, Average voting | Horizontal | Data-sharing |
| [183] | Any | Dynamic policy-based | Both | Classification |
| [139] | Neural networks with similar architecture | Highest confidence Average voting | Horizontal with missing data | Open-set classification |
| [196] | Any | Max model prediction | Horizontal | Classification with locally biased data |
| [182] | Any | Dynamic policy based | both | Classification with heterogeneous data |
| [5] | Any | Weighted average | Horizontal | Classification with specialized sites |
| [62] | Neural networks | Arithmetic mean | Horizontal | Semantic segmentation |
| [10] | Any parametric model | Average voting | Horizontal | Classification with real-time updates & heterogeneous data |
| [113] | Any | Majority vote | Horizontal | Classification without sharing models nor local classifications |
| [132] | Any | Average voting majority voting | Horizontal | Classification |
| [87] | Gradient-based models | Weighted average | Horizontal | Classification with locally biased data |
| [107] | Neural networks | Weighted average | Horizontal | Classification with heterogeneous data |
| [162] | Neural networks (applicable to others) | Average voting | Both | Classification with unreliable devices and heterogeneous data |
| [28] | Random forests | Majority vote | Vertical | Classification |
| [24] | Neural networks | Majority vote | Horizontal | Classification while protecting against byzantine attacks |
| [172] | Tree-based (applicable to others) | Average voting | Horizontal | Classification with heterogeneous data |
| [115] | Random forest | Not discussed | Horizontal | Classification with partially overlapping incomplete labels |
| [127] | Not discussed | Not discussed. | Horizontal | Classification with tasks that can be divided into subtasks |
| [146] | Neural network | Weighted average | Horizontal | Classification with heterogeneous cluster data |
| [160] | Any | Weighted average | Horizontal | Classification and hyperparameters learning |

| | Privacy guarantees | Dataset type | Number of parties | Population | Comments |
|---|---|---|---|---|---|
| | Differential privacy to share the trees in the ensemble. | Census of income<br>Default on credit cards<br>Diabetes<br>Bike Sharing<br>Online news popularity | 10 -100 | 1,151-45,211 | Generates synthetic data to train models thus preserving privacy. |
| | Not applicable | Toy example<br>bird-song data<br>terrorism data | 2 | Not reported | |
| | Only final model is shared | Simulated medical data | 4 | 320 | |
| | Privacy budget – minimize exchange of data. Local models shared as black boxes. | Toy example<br>Fashion-MNIST<br>Handwriting data | 3-7 | 1,000- 128,300 | Takes advantage of local biases and retraining the local models |
| | Not discussed | NSL-KDD | 3 | Millions | Dynamic ensembles for optimal classification. Considerable improvements for specific domains, especially the underrepresented, compared to majority voting. |
| | Not discussed | MNIST<br>CIFAR-10 | feb-45 | 60 | Ensemble of classifiers with known different expertise |
| | Differential privacy based on noisy voting / PATE | BraTS 2019 | 8 | 310 | PATE first trains local models. Local models then form an ensemble to label public set. Public set can then be used for training. |
| | Only local models and aggregated data shared. | Intel Lab Gas Sensor Array Drift<br>Unmanned Surface Vehicles | apr-25 | 1,672-2,300,000 | Creates dynamic ensembles based on the individual to be classified. |
| | Homomorphic encryption during aggregation. Differential privacy for final prediction. | MNIST<br>NSL-KDD<br>breast cancer data set | 20-250 | 569- 25,000 | |
| | Not discussed | MNIST | Not reported | Training: 60,000<br>Test: 10,000 | |
| | Not discussed | Synthetic temperature data<br>Bird behavior data<br>Synthetic dataset | 5-150 | 50-1,500 | Classifiers in the model take into account how far they diverge from neighbors while training. |
| | Not discussed | PeMS | 20 | Not reported. | Combines fedAVG-style learning with ensembles: divide parties into K-clusters. Train a NN on these clusters using fedAVG. |
| | Not discussed | CIFAR-10 | 16 | 60 | |
| | Homomorphic encryption | Credit1<br>Credit2<br>JDT | 2 | 30,000-1,500.000 | Focus on not revealing attributes |
| | Not discussed | MNIST HAR | 30-1000 | 10,299- 70.000 | Federated learning with an ensemble of classifiers trained on K out of N nodes |
| | Not discussed | eICU-CRD | 20 | 7,022 | Custom ensembles (in terms of site & population similarity) to deal with heterogeneous data |
| | Not discussed | Assays from companies | 5 | 11,791 | |
| | Not discussed | Not discussed | Not applicable | Not applicable | Ensemble of (an ensemble of) models with each specializing in a certain aspect of the overarching task |
| | Privacy weaknesses assessed . | MINST<br>CiIFAR10<br>CiIFAR100 | 20-100 | Not reported | Ensemble of federated classifiers where each classifier is specialized in a certain population. |
| | Not discussed | MINST<br>FASHION-MINST<br>Covertype<br>HAR | 4 | 10,299-60,000 | Learning hyperparameters, either as ensemble with different parameters per party or as one shared set of hyperparameters. |

Table 4.1: Federated Ensembles

## 4.4 Discussion

In this article, we have reviewed the existing literature on federated ensembles. We found two sets of relevant articles. The reviewed articles show that federated ensembles are not just useful for classification, but for a wide range of tasks such as synthetic data generation, label sharing, security and many more, further accentuating the potential of ensembles in a federated setting. Federated ensembles are applied to settings with both vertically and horizontally split data. The articles mostly discuss IoT settings with a high number of parties as well as smaller cross-silo settings with only a handful of parties.

We found a collection of articles that compare federated ensembles with other federated learning techniques, which consistently claim that federated ensembles are outperformed by federated learning. This is not surprising as the aim of these papers is largely to promote their own proposed method when comparing it with federated ensembles. These papers often suffer from suboptimal experimental setups for the ensemble approaches. Local classifiers are left with populations that are too small, resulting in underfitting, or with extremely biased populations, where certain groups are significantly under/overrepresented, resulting in overfitting on the dominant groups, or in certain cases, both of these problems. Consequently, the ensembles perform poorly. Two of the six articles in this set acknowledge how their setup could potentially explain the poor performance of the ensembles. The remaining four simply claim ensembles are a poor fit due to the heterogeneity present in federated learning. However, it should be noted that both Ensemble Learning as a field, and many of the publications promoting the use of Federated Ensembles, claim the exact opposite, that is, that ensembles are especially suited to deal with heterogeneous data as it results in diverse classifiers, which ensembles need[98].

It is, however, still important to note that these flawed experimental setups do capture one of the major drawbacks ensembles face. An ensemble relies on the mistakes of an individual classifier being over-

ruled by the majority of the ensemble; hence, an ensemble needs a diverse set of classifiers to function. It is important to keep in mind that what exactly is required to ensure sufficient diversity within the ensemble will depend on the classifiers used within the ensemble. An ensemble of neural networks will have different requirements regarding minimum local population size and its capabilities to deal with imbalanced data than an ensemble of decision trees. If this diversity cannot be achieved, for example because locally available data is extremely limited, such as in these experiments, then federated ensembles will perform poorly.

The publications proposing the various federated ensembles consistently show that ensembles work well in a federated setting and provide strong privacy guarantees while being relatively easy to implement. This is however not their greatest strength. Their greatest strength lies in their ability to work with heterogeneous non-IID data without losing information. This makes it easier to personalize, or specialize, an ensemble-based approach. Especially when the ensemble can be created dynamically based on the input of the to-be-classified individual. It also allows ensembles to capture outliers and underrepresented groups. In a regular federated setting, it is possible for a subset of the population to become dominant, for example because one data source is much larger than the others. This can result in a skewed final model. An ensemble approach avoids this potential problem. A realistic scenario where this is relevant is for example when a small rural hospital cooperates with a big urban hospital. The bigger urban population might drown out the rural population when using a traditional federated approach, but the smaller rural population can still be represented properly when using an (weighted) ensemble. In extremis, an ensemble can even be used to create a varied set of experts, not just capturing outliers but also specializing in various subtasks[127].

Not only can the ensembles in this set of articles deal with generally heterogeneous data, but they can even deal with open set problems: it is not necessary for each party to know every possible label present in

| Year & Author | Data split | Appropriate scenarios/Goal | Privacy guarantees | Dataset type |
|---|---|---|---|---|
| [185] | Horizontal | Classification with local biases and incomplete class set | Not discussed | ImageNet, LSUN, Places36 |
| [78] | Horizontal | Classification | Privacy preserving oversampling | MNIST1, KDD99 6c2, SDD3, Statlog Data Set 4, HAR5, STL-10 dataset |
| [103] | Horizontal | Classification | Noise and $\epsilon$-differential privacy | ABIDE I preprocessed dataset |
| [25] | Horizontal | Classification with heterogeneous data | Not discussed | NSL-KDD, DS2OS Traffic Traces, SCADA Traffic and Payload Datasets, gas & water |
| [52] | Horizontal | Classification | Differential privacy | Image dataset of crops tabular data about state of the land. |
| [187] | Horizontal | Learning a side-channel attack | Not discussed | Energy consumption of a chip |

the ensemble to be able to deal with all labels. A classifier can even be given the option to abstain from voting in the ensemble, or simply vote "unknown" when a certain label is locally unknown. This makes it possible for accurate ensembles to be created even when local datasets contain extremely biased populations[139].

A drawback of ensembles, especially in a vertically split federated settings, is that they may struggle to find interactions between variables owned by different parties and thus, in different local classifiers. There are some workarounds for this, for example, certain forms of feature selection can still be applied to the ensemble[33] or one can implement a federated method to calculate statistical measures such as the covariance for such variables[111, 104], but it remains a practical limitation. Another drawback ensembles may potentially face in a federated setting is an explosion in the size of the ensemble. While in a cross-silo federated learning environment, the number of classifiers would be affordable, in the Internet of Things (IoT), where it is possible to have hundreds, thousands, or even more devices in a network, it may not be practical to create an ensemble.

An unexpected trend we have observed in the literature is ensemble

| | Number of parties | Population | Comments |
|---|---|---|---|
| | 3-20 | 10,000+ | |
| | Not reported | 2,000-18,000 | Extreme imbalance (1:100 minority / majority class ratio). Impact on ensembles not discussed. |
| | 4 | Patients: 370 Total images: 63,550 | Different validation scheme for federated learning and ensembles. Validation scheme unfavorable for ensembles. Possible effects biases in dataset not discussed. |
| | 4 | Millions of instances | |
| | 9-11 | Images: 30 per party Tabular data: 9 US states / 1 entry per county. | Acknowledges low training population influencing ensemble performance. |
| | Training: 3 Validation:6 | Training: 300,000 per party Validation: 1,000 per party | Test-setup is acknowledged to be not optimal for ensembles |

Table 4.2: Publications comparing federated learning to ensemble learning

based learning and federated learning being frequently presented as completely separate, competing choices[52, 78]. We did not expect this due to the natural fit ensembles have for this environment. We suspect this is due to how federated learning first originated. Federated learning was first used in, and is still primarily focused on, IoT settings with edge devices. These settings pose two major issues for ensembles. First, the need to run computations on relatively weak edge devices means it is not always possible to train, or even just run, a sophisticated model locally. Secondly, these settings are normally populated by many devices. While some of the papers shown here do utilize large ensembles consisting of hundreds of classifiers, this is still comparatively small to a realistic IoT setting, which can easily include thousands of devices. It may simply not be practical to use an ensemble in such a setting. However, as mentioned before, outside of the IoT setting, these issues are not relevant. As such, we would have expected ensembles to be viewed as a common federated learning solution in cross silo settings.

Our search strategy might have some limitations. We use a broad search query because an initial exploratory search indicated that limiting the search of key terms to titles, keywords and abstracts would have resulted in the exclusion relevant papers. The downside of such a broad query is that we encounter many false positives (e.g. articles from a conference unrelated to federated learning with the word "federated" in its name or articles referring to ensemble orchestras). Another issue is that using synonyms for "federated" results in a high proportion of false positives, due, among others, to the existence of the so-called "distributed ensembles", which are unrelated to federated learning. For this reason, the search is limited to articles published in 2016 or later, because the term "federated learning" was introduced by Google researchers in 2016. Therefore, there may have been older articles focused on federated ensembles, but using different terminology, that we may have missed due to our search strategy. However, based on the references from the articles reviewed, we are confident that almost all relevant publications were covered in this review.

The articles we found indicate there are still at least three important open questions regarding federated ensembles. The first question is how to deal with the outcome (or class-labels) only being available at one party in a vertical setting. This makes building a model fully locally impossible and building a federated model where only one attribute is locally missing may pose an increased risk of information leakage for certain models. However, it could also prove beneficial for certain models as less information is shared that needs to be protected. None of the articles discussed this question.

The second question concerns detecting correlations between attributes that are spread across the various data owners in a federated setting, and thus across the various models. This is a general downside of ensembles, as ensembles are more complex to interpret than a single model. It is possible to apply certain feature selection methods to ensembles even when the data that is split is not IID[92]. The more accurate methods rely on wrapper based feature selection and may detect interactions between the variables used by

different models by maximizing the performance of the ensemble as a whole[105, 33, 92]. However, these may not be viable in a federated setting due to privacy and time complexity constraints. On the other hand, existing federated methods for correlation detection, which often implement filter-based feature selection, might not result in the best possible ensembles, as they cannot detect interactions across the models. There is also a minor risk that filter-based feature selection will end up dropping variables that contain correlations, which only become relevant when looking at the ensemble as a whole, while they appear irrelevant when considered in isolation.

The third open question is how to best exploit the advantages of federated ensembles in a vertically split setting. The articles do repeatedly mention that ensembles are very good at dealing with biases in the data as well as non-IID data. However, no one discusses how to detect and purposefully take advantage of subtle biases in a vertical setting. For example, there will be a dependency when one party is a General Practitioner (GP) and the other is the specialist the patient was referred to by the GP based on what the GP saw. Detecting these dependencies and taking advantage of them via federated ensembles would be beneficial.

## 4.5 Conclusion

In this literature review, we have provided an overview of the current state of federated ensembles. The existing literature shows that this is a promising field; as federated ensembles have been shown to be applicable to a wide range of different tasks in a federated setting and be able to deal with both horizontally and vertically split data. In addition, they are a good fit for federated learning due to their privacy guarantees and their ability deal with the inherent heterogeneity and non-IID nature of distributed data in the real world. However, some studies have also shown that federated ensembles are not the best choice in every situation and provide a cautionary tale about the

use of ensembles when local data is too biased or the local population too small. In conclusion, we believe federated ensembles will play a more important role within the federated learning community in the near future, especially outside the IoT setting. Lastly, there are three important open questions regarding federated ensembles that have not been discussed in the literature: how to share class labels in a vertically partitioned setting, how to determine correlations between variables in different parties, and how to best detect and exploit the subtle biases and dependencies in a vertically split scenario. These three questions pose interesting avenues for future work.

# 5

# Federated Bayesian Network Ensembles

Adapted from: Florian Van Daalen et al. "Federated Bayesian Network Ensembles". In: *2023 Eighth International Conference on Fog and Mobile Edge Computing (FMEC)*. IEEE. 2023, pp. 22–33.

## Abstract

Federated learning allows us to run machine learning algorithms on decentralized data when data sharing is not permitted due to privacy concerns. Ensemble-based learning works by training multiple (weak) classifiers whose output is aggregated. Federated ensembles are ensembles applied to a federated setting, where each classifier in the ensemble is trained on one data location. In this article, we explore the use of federated Bayesian network ensembles (FBNE) in a range of experiments and compare their performance with both locally trained models and models trained with VertiBayes, a federated learning algorithm to train Bayesian networks from decentralized data. Our results show that FBNE outperform local models and provides, among other advantages, a significant increase in training speed compared with VertiBayes while maintaining a similar performance in most settings. We show that FBNE are a potentially useful tool within the federated learning toolbox, especially when local populations are heavily biased, or there is a strong imbalance in population size across parties. We discuss the advantages and disadvantages of this approach in terms of time complexity, model accuracy, privacy protection, and model interpretability.

## 5.1 Introduction

Federated learning allows machine learning algorithms to be applied to decentralized data when data sharing is not an option due to privacy concerns[93]. Traditionally federated learning approaches train a model iteratively on local data[118]. The local results are then averaged back into a single global model. Privacy is preserved using epsilon-differential privacy[55], homomorphic encryption[135], and multiparty computation (MPC)[202] during this process. The specific techniques used depend on the way the data is split across the various parties. If the data is split horizontally, i.e. each party has data belonging to a different population but the attributes are the same, simpler techniques can be used. While this approach yields good results in many cases, it suffers from several limitations.

A major downside is that it does not explicitly consider heterogeneity across different data sites. It assumes that the data is independent and identically distributed (IID) over the various parties. However, in practice, federated environments will often be subject to local biases. A common scenario in which federated learning is implemented occurs when multiple hospitals combine their data to build a joint model[43]. These hospitals may have very different population sizes, which may cause the final model to overfit on the biggest hospital. Additionally, the hospitals might have biases in their populations: i.e., an urban and a rural hospital will have different patient populations. Furthermore, the hospitals may even be on opposite sides of the globe, adding further biases into the distribution due to cultural, socio-economic, and many other factors. Simply averaging over these diverse populations may result in models that fit neither population, or it may result in the

*The views expressed in this paper are those of the authors and do not necessarily reflect the policy of Statistics Netherlands.*

*This research received funding from the Netherlands Organization for Scientific Research (NWO): Coronary ARtery disease: Risk estimations and Interventions for prevention and EaRly detection (CARRIER): project nr. 628.011.212.*

model overfitting on one particular population, while ignoring others.

If the data is split vertically, i.e. when different parties have different variables about the same population, it may be possible that there are dependencies between the parties. For example, if one party is a general practitioner (GP) and the other party is the specialist clinician, the GP might have referred the patient to the specialist and there will be a dependency between the two datasets. The GP might, for example, have started treatment based on the data he has, which will influence the data the specialist receives.

These diverse types of bias may create problems when using the traditional federated learning approach. An alternative is the use of federated ensembles; ensembles of classifiers each of which has been trained on the local data of each party in a federated setting[36]. Ensemble based learning works by combining multiple (weak) classifiers which work together to jointly produce classifications using various voting schemes[131]. It relies on a diverse set of classifiers, under the assumption that if one classifier makes a mistake the other classifiers will correct it. This allows ensembles to achieve a high performance, even when the individual classifiers are weak. A major advantage of ensemble learning is that it can deal with non-IID data[32]. It can even take advantage of the dependencies by using (dynamically) weighted voting schemes. For example, an ensemble of experts can weigh the votes of classifiers trained on a similar population, or trained on specific sub-tasks, more strongly[146, 127].

Current research into federated ensembles is limited[36]. There is only a small body of current work specifically looking into federated ensembles. There are still several general open questions, mainly:

1. How to share class labels in a privacy preserving manner in a vertically distributed setting.
2. How to detect and exploit the subtle biases and dependencies that exist across parties.

3. How to determine correlations between the various attributes split across parties.

In addition to these general questions, there is also room for exploring different types of ensembles, using different base-classifiers. In this article, we will explore the use of federated ensembles consisting of Bayesian networks (BN). We will compare these ensembles with VertiBayes[38], a federated implementation of BNs, as well as with a BN that was centrally trained. We will compare the various options in terms of technical complexity, training time required, privacy protection, and model performance.

## 5.2 Methods

### 5.2.1 Bayesian networks

Bayesian networks (BN) are widely used probabilistic graphical models consisting of a directed acyclic graph (the structure) where each node represents a variable and arcs represent conditional dependencies. Each node has a set of conditional probability distributions (the parameters)[137]. Their ability to combine existing expert knowledge with data has given them great utility and popularity. In addition, their graphical representation and probabilistic reasoning makes them relatively intuitive to understand models for non-technical personnel. This makes them especially useful in scenarios where non-technical personnel need to be able to understand the models, for example when models are used to inform clinical decisions.

### 5.2.2 VertiBayes

VertiBayes[38] is an implementation of BN learning algorithms in a federated environment. It works both in vertical and horizontally distributed scenarios, as well as in hybrid scenarios. In addition to this, it can deal with missing data[44, 101]. Furthermore, it includes a federated implementation of the K2 algorithm[29], allowing it to learn a

network structure on the fly. Lastly, VertiBayes includes several validation methods, of various computational complexity, that can be used to validate the model in a privacy preserving manner. This makes it an appropriate tool in a federated setting where data quality across parties may not be guaranteed.

It has a similar performance compared to a centrally trained model. In addition, it provides the same privacy guarantees as the n-party scalar product protocol[37] used. However, it is considerably more time consuming to train a model using VertiBayes than it is to train a model centrally. The time complexity mostly depends on the number of probabilities that need to be calculated during parameter learning.

### 5.2.3 Federated Bayesian Network Ensembles

Federated Bayesian Network Ensembles (FBNE) are an ensemble learning approach where the base classifier consists of Bayesian networks. Each data owner within the federated setting makes their own Bayesian network based on locally available data. This local data is only enriched with the class label (should this not be available locally) in a privacy preserving manner using VertiBayes. The local classifiers can then be used in an ensemble to classify a new individual.

It is possible to use FBNE in horizontally split, vertically split, and hybrid settings. In the case of a hybrid split, one may decide to build the models purely based on local data, or to allow the hybrid variables to also utilize the data available at other data parties. For example, if party 1 contains attribute A & B, and party 2 contains attributes B & C, we can choose to either build a model using only the data available locally at party 1, or to build a model which also includes the data party 2 has regarding attribute B. In addition, it is possible with the use of predefined structures to mix and match variables from various parties to create the optimal ensemble.

### Privacy risks

The privacy risks posed by FBNE are the same as those posed by VertiBayes. That is to say, there are no major risks during training. However, the BNs themselves do still contain information. The structure and CPDs included in the BNs will be revealed if they are published. The published networks can be used to predict missing values in a dataset by a third party. This is inherent to how BNs work and as such is unavoidable. An ensemble of BNs poses a similar risk.

### Runtime advantages

FBNE are significantly faster than VertiBayes as the majority of the calculations can be done locally. This minimizes the use of complex MPC needed to preserve privacy.

### Performance advantages and disadvantages

FBNE may outperform a single model. The ensembles may catch local biases that are lost when only a single model is built. Furthermore, by using weighted voting it becomes possible to create a mixture of experts. This is especially advantageous if it is known that the various data parties have biases in their data. For example, in a scenario where hospitals work together to build an ensemble it may be useful to weigh the votes if one hospital specializes in a certain type of patient.

### Interpretability

Ensembles are generally less interpretable than a single classifier. However, Bayesian networks are highly interpretable thanks to their graphical representation. While it is not possible to directly detect interactions across the various individual classifiers within

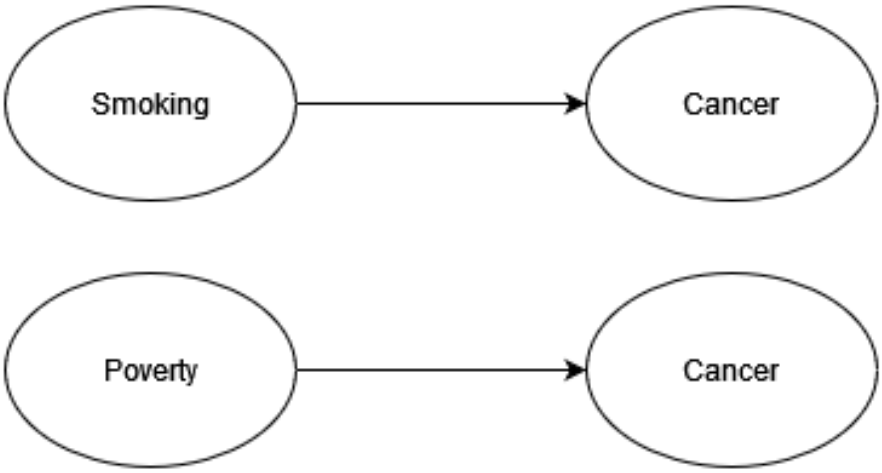


Figure 5.1: Two local networks based on locally available data

the ensemble, it is possible to use the individual classifiers to guide research into variable interactions in a smart manner. For example, by comparing the sparsity of the local networks to determine if certain variables are actually of interest to the outcome variable. In addition to this, it can be possible to use expert knowledge to deduce possible interactions. Take the following toy examples shown in figure 5.1.

In addition to these two models, expert knowledge indicates that poverty increases the likelihood of smoking. Based on this expert knowledge and the local models that were created it can be deduced that the global structure might be similar to the network shown in figure 5.2

Furthermore, it should be noted that it is possible to apply feature selection across ensembles using wrapper-based approaches. By utilizing this type of feature selection, it may be possible to detect mediating effects between attributes.

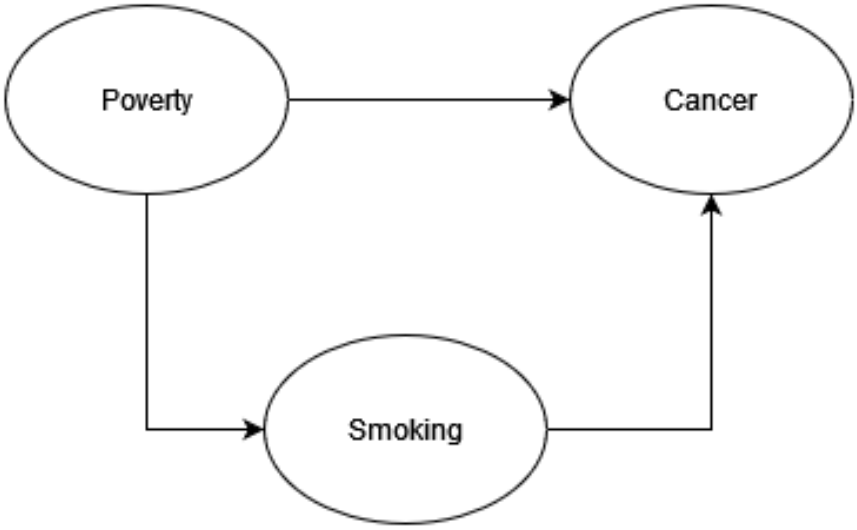


Figure 5.2: Global network created by combining the two local models from figure 5.1 while utilizing expert knowledge.

### Robustness

FBNE bring two further advantages inherent to ensemble learning. First, ensemble learning can be used to deal with non-IID data[32]. As federated learning scenarios inherently deal with scenarios where one combines data from a diverse set of data sources, this can be beneficial. Additionally, an ensemble is inherently robust against certain types of attacks, such as poisoning attacks[175], that may occur in a federated setting by adversarial data-sources. If one model in the ensemble is poisoned, the other models within the ensemble may still be able to correct for this, safeguarding the performance of the ensemble as a whole.

Table 5.1: A general description of experimental datasets used

| Name | Number of attributes | Number of individuals | Missing value in original dataset |
|---|---|---|---|
| Iris | 5 | 150 | No missing values |
| Autism | 20 | 704 | Missing values present |
| Asia | 8 | 10.000 | No missing values |
| Alarm | 37 | 10.000 | No missing values |
| Diabetes | 9 | 768 | No missing values present |
| Mushroom | 23 | 8124 | Missing values present |

## 5.3 Experiments

We conducted a diverse set of experiments to evaluate the performance of FBNEs[1]on datasets with different sample sizes and variable numbers and different missingness levels. We used the following datasets: Iris[41], Autism[174], Asia[100], Diabetes[167], Alarm[18] and Mushroom[154]. A general description of the datasets used can be found in Table 5.1

Each dataset was tested with varying levels of added missing values (no missing values, 5%, 10% &, 30% missing at random). We tested the following federated split scenarios:

- Vertical split with the attributes split randomly between parties. Each party had at least 2 local attributes. This was done for a setting with 2 and a setting with 3 parties.
- Horizontal split with the samples split randomly between parties. Each party had at least 50 records. Again this was done for a setting with 2 and a setting with 3 parties.
  - Additionally the horizontal splits were tested at varying levels of bias. To induce this bias we would make it more likely for individuals of a certain label to be put in a specific data station, i.e. individuals with the class label "true" are more likely to be put in party 1, individuals with class label

[1]An implementation of FBNE can be found in the following git repository: `https://github.com/MaastrichtU-CDS/bayesianEnsemble`

"false" are more likely to be put in party 2. In the case of non-binary labels, such as for the Iris dataset, the bias would be created by piting label 1 against the rest. The following levels of bias were introduced: no bias, 75%, 85%, and 95% bias.

- Hybrid split. The attributes were randomly split in two, the samples in one of these splits were then randomly split in two again. This results in 3 parties in total, each with at least 2 local attributes and 50 local records.

Each of these scenarios was run 10 times for each dataset. It should be noted that not every dataset could be used in every scenario. For example, the Iris dataset only contains 5 attributes and thus cannot be used in a 3-party vertically split scenario.

In addition to these random splits, an experiment was also run where data was manually split in a "realistic" manner in a vertical setting based on expert knowledge. These experiments attempted to represent a realistic split where different parties collected different types of data. For example, the Autism data set contained attributes representing answers to a questionnaire, and some general personal attributes such as age, sex, and country of residence. This data was split in such a way that one party had the questionnaire answers, and the other party had the remaining attributes. As mentioned above, horizontal bias was addressed separately in the horizontal experiments.

We compared the performance of FBNE with VertiBayes as well as a centrally trained Bayesian network. In addition, the performance of the local models on their local data is used as a baseline, if the local model already creates a sufficiently strong classifier there is no need for a federated model. The performance was measured by calculating the AUC using a 10-fold cross validation.

In all cases both structure and parameters were learnt. Continuous variables were discretized into intervals which contained at least 10% of the total population.

The experiments were run on a Windows laptop using an Intel(R) Core(TM) i7-10750H processor with 16GB of memory and 6 cores. All parties had a virtual server on this laptop.

## 5.4 Results

A selection of the results can be seen in tables tables 5.2 and 5.3 and section 5.4. The selected results illustrate the general trends seen across all experiments. The remaining experimental results can be found in the appendix 4.

In all tables the highest AUC is indicated with '*', the second highest with '†'.

### 5.4.1 Runtime

The training process for FBNE was consistently faster than VertiBayes in every experimental setting and for every dataset used. However, it should be noted that the differences vary widely and depend on the dataset. These differences were mainly driven by the different network structures. FBNE have the advantage that most calculations can be done locally. During our experiments FBNE were faster than VertiBayes by a factor that ranged from twice as fast to fifty times as fast. It should be noted that it is difficult to predict per scenario how large the processing speed gains will be as it is difficult to predict the resulting network structure.

### 5.4.2 Performance

FBNE largely showed the same performance as VertiBayes, scoring similar AUC's. However, it should be noted that for specific datasets, and for specific data-splits, the ensembles can perform significantly better. The largest difference being nearly a 0.1 difference in AUC.

Table 5.2: Experimental results vertically split 2-party scenarios where attributes were randomly split across parties. '*' Indicates the best performing model, '†' indicates the second best performing model.

| | | AUC | | | | |
|---|---|---|---|---|---|---|
| Name | Missing Data Level | FBNE | Party 1 | Party 2 | Central | VertiBayes |
| Alarm population size: 10000 | 0 | 0,888* | 0,793† | 0,675 | 0,789 | 0,789 |
| Asia population size: 10000 | 0 | 0,996* | 0,918 | 0,886 | 0,995† | 0,991 |
| | 0.05 | 0,742† | 0,694 | 0,696 | 0,735 | 0,776* |
| | 0.1 | 0,622† | 0,594 | 0,591 | 0,615 | 0,677* |
| | 0.3 | 0,418 | 0,401 | 0,415 | 0,42† | 0,581* |
| Autism population size: 704 | 0 | 0,909† | 0,803 | 0,824 | 0,824 | 0,93* |
| | 0.05 | 0,8† | 0,71 | 0,727 | 0,735 | 0,853* |
| | 0.1 | 0,737† | 0,67 | 0,672 | 0,675 | 0,834* |
| | 0.3 | 0,531† | 0,484 | 0,497 | 0,498 | 0,716* |
| Diabetes population size: 768 | 0 | 0,811* | 0,744 | 0,69 | 0,78 | 0,808† |
| | 0.05 | 0,757† | 0,658 | 0,684 | 0,723 | 0,772* |
| | 0.1 | 0,693† | 0,597 | 0,624 | 0,668 | 0,767* |
| | 0.3 | 0,452† | 0,418 | 0,406 | 0,44 | 0,667* |
| Iris population size: 150 | 0 | 0,94* | 0,882 | 0,908† | 0,885 | 0,75 |
| | 0.05 | 0,892* | 0,833 | 0,789 | 0,877† | 0,787 |
| | 0.1 | 0,782* | 0,702 | 0,705 | 0,706† | 0,701 |
| | 0.3 | 0,658† | 0,6 | 0,606 | 0,673* | 0,607 |
| Mushroom population size: 8124 | 0 | 0,999* | 0,987† | 0,898 | 0,999* | 0,986 |

In addition to outperforming VertiBayes by a relevant margin in specific scenarios, the following general trends were visible in our experiments. First, FBNE performed very well in scenarios with no missing data. FBNE achieved the highest AUC in 80% of the scenarios with no missing data. However, VertiBayes performed better in scenarios with missing values. VertiBayes achieved the highest AUC in roughly 70% of the scenarios.

Another interesting trend that was visible in our experiments is that in horizontally split scenarios the local models can perform well if the local data quality is high, occasionally performing similarly to the federated and centralized models. However, it should be noted that this is rare.

Table 5.3: Experimental results vertically split 2-party scenarios where attributes were manually split across parties to simulate a realistic biased split. '*' Indicates the best performing model, '†' indicates the second best performing model.

| | | AUC | | | | |
|---|---|---|---|---|---|---|
| Name | Missing Data Level | FBNE | Party 1 | Party 2 | Central | VertiBayes |
| Autism population size: 704 | 0 | 0,889* | 0,832 | 0,720 | 0,851† | 0,834 |
| | 0.05 | 0,789* | 0,726 | 0,669 | 0,738 | 0,780† |
| | 0.1 | 0,728† | 0,678 | 0,599 | 0,689 | 0,743* |
| | 0.3 | 0,497† | 0,491 | 0,379 | 0,497† | 0,625* |
| Iris population size: 150 | 0 | 0,940* | 0,845 | 0,888 | 0,912† | 0,748 |
| | 0.05 | 0,887* | 0,635 | 0,872 | 0,878† | 0,736 |
| | 0.1 | 0,774* | 0,640 | 0,696 | 0,699† | 0,671 |
| | 0.3 | 0,654† | 0,477 | 0,664 | 0,667* | 0,622 |
| Mushroom population size: 8124 | 0 | 0,992* | 0,880 | 0,987 | 0,987† | 0,987† |

It is important to note that neither approach was optimized, and better models can potentially be created. For example, a network structure generated using expert knowledge, as opposed to using an automatic approach like the K2 algorithm, may result in better Bayesian networks. Both VertiBayes and FBNE could benefit from such expert knowledge. In addition, weighted voting, and especially dynamically weighted voting, can improve the performance of FBNE.

Table 5.4: Experimental results hybrid split 3-party scenarios where only locally available data was used in the local model. '*' Indicates the best performing model, '†' indicates the second best performing model.

| | | AUC | | | | | |
|---|---|---|---|---|---|---|---|
| Name | Missing Data Level | FBNE | Party 1 | Party 2 | Party 3 | Central | VertiBayes |
| Asia population size: 10000 | 0 | 0,996* | 0,928 | 0,940 | 0,938 | 0,995† | 0,987 |
| | 0.05 | 0,742 | 0,689 | 0,670 | 0,670 | 0,746† | 0,789* |
| | 0.1 | 0,627† | 0,611 | 0,596 | 0,595 | 0,624 | 0,668* |
| | 0.3 | 0,418† | 0,400 | 0,406 | 0,407 | 0,418† | 0,568* |
| Autism population size: 704 | 0 | 0,911* | 0,805 | 0,821 | 0,820 | 0,833† | 0,826 |
| | 0.05 | 0,803† | 0,707 | 0,718 | 0,719 | 0,741 | 0,932* |
| | 0.1 | 0,739† | 0,671 | 0,667 | 0,665 | 0,698 | 0,891* |
| | 0.3 | 0,537† | 0,473 | 0,493 | 0,493 | 0,491 | 0,783* |
| Diabetes population size: 768 | 0 | 0,814* | 0,700 | 0,734 | 0,735 | 0,776† | 0,781 |
| | 0.05 | 0,749† | 0,685 | 0,657 | 0,656 | 0,726 | 0,753* |
| | 0.1 | 0,694† | 0,633 | 0,596 | 0,607 | 0,676 | 0,740* |
| | 0.3 | 0,452† | 0,420 | 0,407 | 0,409 | 0,440 | 0,691* |
| Iris population size: 150 | 0 | 0,952* | 0,912† | 0,888 | 0,894 | 0,885 | 0,700 |
| | 0.05 | 0,886* | 0,784 | 0,849 | 0,853 | 0,874† | 0,805 |
| | 0.1 | 0,798* | 0,725 | 0,730 | 0,725 | 0,688 | 0,642 |
| | 0.3 | 0,633† | 0,603 | 0,575 | 0,582 | 0,669* | 0,610 |

Table 5.5: Experimental results horizontally split 2-party scenarios where records are randomly split across parties. '*' Indicates the best performing model, '†' indicates the second best performing model.

| | | AUC | | | | |
|---|---|---|---|---|---|---|
| Name | Missing Data Level | FBNE | Party 1 | Party 2 | Central | VertiBayes |
| Asia population size: 10000 | 0 | 0,996* | 0,995† | 0,995† | 0,995† | 0,988 |
| | 0.05 | 0,740 | 0,741† | 0,741† | 0,740 | 0,765* |
| | 0.1 | 0,624† | 0,623 | 0,623 | 0,624† | 0,670* |
| | 0.3 | 0,416 | 0,418† | 0,418† | 0,415 | 0,568* |
| Autism population size: 704 | 0 | 0,868* | 0,777 | 0,774 | 0,847† | 0,834 |
| | 0.05 | 0,787* | 0,464 | 0,464 | 0,737 | 0,781† |
| | 0.1 | 0,691† | 0,466 | 0,463 | 0,691† | 0,746* |
| | 0.3 | 0,534† | 0,392 | 0,413 | 0,492 | 0,628* |
| Diabetes population size: 768 | 0 | 0,781† | 0,500 | 0,500 | 0,781† | 0,782* |
| | 0.05 | 0,725 | 0,480 | 0,480 | 0,729† | 0,752* |
| | 0.1 | 0,671 | 0,447 | 0,447 | 0,680† | 0,736* |
| | 0.3 | 0,441† | 0,431 | 0,437 | 0,437 | 0,648* |
| Iris population size: 150 | 0 | 0,957* | 0,903 | 0,898 | 0,925† | 0,782 |
| | 0.05 | 0,877† | 0,829 | 0,837 | 0,879* | 0,759 |
| | 0.1 | 0,790* | 0,749 | 0,774† | 0,7049566 | 0,675 |
| | 0.3 | 0,611† | 0,494 | 0,533 | 0,677* | 0,609 |
| Mushroom population size: 8124 | 0 | 0,988* | 0,988* | 0,988* | 0,987† | 0,987† |

## 5.5 Discussion

Our experiments show that FBNE can be a suitable solution in certain scenarios. In this section we will take a deeper dive into the differences between FBNE and VertiBayes.

### 5.5.1 Runtime

The reduced training time which was observed during the experiments is a strong advantage in favor of the federated ensembles, especially in time critical applications or in cases when MPC solutions such as VertiBayes are too time consuming. FBNE have this advantage due to the fact that the vast majority of calculations can be done locally and thus requires far fewer computationally difficult operations than VertiBayes does. An overview of the differences in time complexity between VertiBayes and FBNE can be found in table 5.6.

Table 5.6: Time complexity

| | VertiBayes | FBNE |
|---|---|---|
| Number of Scalar product protocols | $O(m)$, where $m$ is the number of unique parent-child value combinations for which a probability needs to be calculate | $O(l)$, where $l$ is the number of unique parent-child value combinations involving the label attribute for which a probability needs to be calculated |
| Number of scalar product subprotocols per protocol | $\frac{n!}{(x!(n-x)!))}$, for each $x, 2 <= x <= n$, where $n$ is the number of parties involved in the protocol | No subprotocols unless FBNE are working with a hybrid split and incorporates all available data into local models. |

In addition to the improved runtime already observed here, it should be noted that our implementation of the ensembles has not been fully optimized. There is room for further improvements, especially with respect to parallelization, which will further increase the gap in terms of runtime. However, since the goal of this study was to simply explore the potential of FBNE, not to provide an optimized implementation, this will not be a part of this study.

### 5.5.2 Performance

FBNE largely showed the same performance as VertiBayes, however, in specific scenarios it significantly outperformed VertiBayes. The largest difference being nearly a 0.1 difference in AUC. This does indicate that FBNE are potentially very useful in the right situation. However, it is very difficult to determine when this is the case without simply training the FBNE.

Two trends were visible with respect to the performance differences. First, FBNE perform very well in scenarios with no missing data. However, VertiBayes performs better in scenarios with missing values. There are two plausible reasons that explain why VertiBayes performs better in these scenarios. The first explanation is that FBNE did not have a large, or diverse, enough ensemble in the experimental scenarios to work properly. Ensemble learning relies on a diverse set of classifiers which can correct each other's mistakes, with only 2 or 3 models in our scenarios the ensembles may not be able to do this consistently. The second possible explanation is that the synthetic data generation step within VertiBayes allows it to bootstrap itself for an improved performance.

Another interesting trend that is visible in our experiments is that in horizontally split scenarios the local models occasionally perform well when local data is of a high quality, especially when the local population is not biased in any way. This reminds us that it is always important to ask if a federated model is truly necessary. Building a federated model is only worthwhile if the data added from other parties adds extra information. But if local data is already sufficiently large, and representative of the true population, then a federated model may not be needed. However, if the local data is small, or biased in some way, then a federated approach is needed.

It is important to note that neither approach was optimized and better models can potentially be created. For example, both VertiBayes and FBNE can benefit from improvements, such as using expert knowledge

to build the optimal structures. In addition to improvements that could be applied to both approaches, weighted voting, and especially dynamically weighted voting, can improve the performance of FBNE.

### 5.5.3 Privacy concerns & disclosure control

Our implementation of FBNE uses VertiBayes at its core. As such, the privacy guarantees are largely the same. However, there are two aspects in which they potentially differ from VertiBayes:

1. The classification of individuals and evaluation of the model.
2. The consequences of having multiple networks.

With FBNE the individual classifiers can create their classifications fully locally at the party they belong to given that required attributes for each local model should be available locally. These individual classifications can be combined using homomorphic encryption, resulting in a final classification which can be shared. For example, estimated probabilities can be weighted according to the voting power of a particular classifier, then encrypted using an additive homomorphic encryption scheme, and all encrypted weighted probabilities are summed. The sum is then decrypted and divided by the total weight to get the weighted average probabilities, which determine the final classification. Combining the votes in this way prevents any local data from being shared and allows FBNE to be evaluated without the need of the homomorphic encryption and a privacy preserving n-scalar product protocol[37] or other more complex evaluation methods VertiBayes needs[38]. This is a strong advantage when new samples need to be classified or predicted in a federated manner.

The other aspect in which FBNE differ from VertiBayes is that the end-result consists of multiple networks instead of one. Ensembles might provide a minor advantage with respect to privacy in this case. In both

cases, it is possible to learn dependencies between attributes and certain statistics about the training set, based on the CPDs and network structures. Similarly, based on local, incomplete data, the network can be used to predict missing values. These are unavoidable consequences of using Bayesian networks. However, because FBNE have the information split up over multiple networks, it will be difficult to do this for every attribute. As discussed in section 5.2.3, it is very difficult to determine any relation between two attributes when those two attributes are split over two Bayesian networks. This provides some additional privacy protection compared to a single network.

### 5.5.4 When should FBNE be preferred over VertiBayes

Due to the significant advantage in terms of runtime it may be beneficial to use FBNE as an exploratory first step before deciding if using VertiBayes is worth it. This naive approach will already provide reasonable results.

The general advantages of each approach can be found in table 5.7.

In addition to these advantages which hold in general, one of the two may perform better depending on how the data happens to be split. However, there is currently no good way to predict which approach will achieve the highest accuracy.

Table 5.7: Comparison of main features of FBNE and VertiBayes

| FBNE | VertiBayes |
|---|---|
| Faster | More complete view of dependencies between attributes |
| Slightly better privacy guarantees | Easier to understand & interpret than an ensemble |
| Possible to capture biases in local population | May outperform FBNE when the ensemble is too small or not diverse enough |
| Can easily classify new samples in a privacy preserving manner | |
| Can handle non-IID data more easily | |
| Inherently robust against certain types of attacks | |

## 5.6 Conclusion

In this article, we have proposed the use of Federated Bayesian Network Ensembles (FBNE) and assessed their usefulness in a battery of experiments. We have shown the approach performs well in a range of situations and datasets, often achieving similar results when compared to VertiBayes, an alternative federated method. FBNE are significantly faster than VertiBayes, provide slightly better privacy guarantees, and are easier to use in a scenario where future classifications will also be done in a federated setting. On the other hand, VertiBayes results in more interpretable models and makes it easier to determine the dependencies between variables split over multiple sites.

The notable advantage in terms of runtime does mean that it is easily possible to use FBNE as an initial exploratory option. Since it is currently not possible to preemptively determine which approach will

result in the highest accuracy, using this naive approach and simply training both models might be the best course of action currently available. Additionally, this means it can be very useful when exploring new federated datasets.

### 5.6.1 Future work

We would like to explore ways to determine which approach is more effective as it would be highly beneficial to be able to know beforehand if an ensemble-based approach will outperform a single model. Additionally, we would like to run experiments to discover if (dynamically) weighted voting could be used to significantly improve the performance of the ensembles. Lastly, it would be extremely valuable if these experiments could be run on real use cases. This would allow the experiments to work with realistic biases and remove the need to artificially create these biases in our experimental setup, resulting in much more realistic experimental scenarios.

# 6

# Verticox+

## Abstract

Federated learning allows us to run machine learning algorithms on decentralized data when data sharing is not permitted due to privacy concerns. Various models have been adapted to use in a federated setting. Among these models is Verticox, a federated implementation of Cox proportional hazards models, which can be used in a vertically partitioned setting. However, Verticox assumes that the class label is known locally by all parties involved in the federated setting. Realistically speaking this will not always be the case and thus would require the label to be shared. However, sharing the label would in many cases be a breach of privacy which federated learning aims to prevent. Our extension to Verticox, dubbed Verticox+, solves this problem by incorporating a privacy preserving n-party scalar product protocol at several stages. This allows it to be used in scenarios where the label is not locally known at each party. In this article, we demonstrate that our algorithm achieves equivalent performance to the original Verticox implementation. We discuss the changes to the computational complexity and communication cost caused by our additions.

## 6.1 Introduction

Federated learning is a field that recently rose in prominence because of an increased focus on privacy by the general public as well as from legal bodies[93, 102]. In order to fulfil the stricter privacy requirements that were demanded by new laws such as the European General Data protection Regulation (GDPR) existing models were adapted and improved. Verticox is one such adaptation.

Verticox[39] aims to provide a privacy preserving implementation of a Cox proportional hazards (CPH) model[30] in a vertically partitioned federated learning setting. Data is said to be vertically partitioned when the attributes are split between multiple parties. In contrast it is said to be horizontally partitioned when the records are split between

multiple parties. It utilizes an Alternating Direction Method of Multipliers (ADMM) framework[20] to preserve privacy. It can be used both for the training of a new Cox model as well as to classify a new individual.

However, Verticox relies on the assumption that the class label is known locally at every party. This assumption is unfortunately not realistic as in vertically partitioned scenarios each attribute will normally only be locally known at one party, this includes the class label.

Alternatives exist, such as the method proposed by Miao et al.[122] to compute CPH using cyclical coordinate descent, but it is still the case that outcome data needs to be shared with other parties. Kamphorst et al.[94] train a CPH model that uses secure multiparty computation[202] to compute log-partial likelihood at every iteration without revealing patient level data to other parties. However, the cryptographic protocols add significantly to the computational complexity and communication overhead. As such neither alternative is practical.

In this article, we propose a new extension to Verticox, which we have dubbed Verticox+. By utilizing the privacy preserving n-party scalar product protocol[37], we avoid the assumption made in the original Verticox implementation. We will also experimentally show that the added computational complexity of using this protocol is negligible in practice.

The rest of the article is built up as follows; first, we will discuss how the original Verticox protocol works, followed by an explanation of the privacy preserving $n$-party scalar product protocol. Once both protocols have been explained we will describe the improved protocol Verticox+. We will then describe our experimental validation followed by a short discussion.

The implementation of Verticox+ is available on GitHub[1] and

[1] `https://github.com/CARRIER-project/verticox`

has been designed to work with the Vantage6 federated learning framework[124].

## 6.2 Background

In the following subsections we will discuss the background of our solution. First, we will introduce Verticox, and then we will introduce the n-party scalar product protocol.

### 6.2.1 Verticox

Verticox is a decentralized version of the Cox proportional hazards regression model where covariates can be distributed over multiple data sources. The parameters are computed without sharing raw data between the parties and the resulting model is equivalent to a centralized version of a Cox model. The original algorithm achieves this by decomposing the original optimization problem for Cox proportional hazards into subproblems that can be solved separately. The Verticox algorithm first estimates the parameters at the client-side based on the covariates that are available locally to each party. Next, an aggregation of these results is sent to a central server, which combines the results of the various parties, and further optimizes the parameters. The updated values are then passed back to the parties at the start of a new iteration.

### 6.2.2 $n$-party scalar product protocol

The $n$-party scalar product protocol is a protocol that can be used to calculate a scalar product in a privacy preserving manner in a federated setting. The pseudocode can be found in algorithm 2. It can be used in both a horizontally and vertically partitioned data setting. Furthermore, it can deal with an arbitrary number of parties. The scalar protocol is an important part in certain traditional

machine learning approaches[194], which makes a privacy preserving method inherently valuable. In addition to this, it can be used to execute complex calculations in a privacy preserving manner when combined with a well-chosen representation of the sensitive data. It has been used to train decision trees[50], Bayesian networks[38], and ensembles[179], in a privacy preserving manner. We will utilize the same approach of well-chosen data representation in combination with the protocol to create Verticox+.

**Algorithm 2:** The n-party scalar product protocol

nPartyScalarProduct($\mathcal{D}$)
**Input :** The set $\mathcal{D}$ of diagonal matrices $\mathbf{D_1}..\mathbf{D_n}$ containing the original vectors owned by the $n$ parties
**Output:** $\varphi(\mathbf{D_1} \cdot \mathbf{D_2} \cdot .. \cdot \mathbf{D_n})$
**if** $|\mathcal{D}|= 2$ **then**
  return 2-party scalar product protocol($\mathcal{D}$);
**else**
  **for** $i \leftarrow 0$ **to** $|\mathcal{D}|$ **by** $1$ **do**
    $\mathbf{R_i} \leftarrow generateRandomDiagonalMatrix()$
  **end**
  Let $\varphi(\mathbf{R_1} \cdot \mathbf{R_2} \cdot .. \cdot \mathbf{R_n}) = r_1 + r_2 + \ldots + r_n$
  Share $\{\mathbf{R_i}, r_i\}$ with the $i$'th party for each $i \in [1, n]$
  $v_2 \leftarrow randomInt()$
  $u_1 \leftarrow \varphi(\prod_{i=2}^{n} \hat{\mathbf{D}}_\mathbf{i} \cdot \mathbf{D_1}) + (n-1) \cdot r_1 - v_2$
  **for** $i \leftarrow 2$ **to** $|\mathcal{D}|$ **by** $1$ **do**
    $u_i = u_{i-1} - \varphi((\prod_{x=1}^{n} \hat{\mathbf{D}}_\mathbf{x} | x \neq i) \cdot \mathbf{R_i}) + (n-1) \cdot r_i$
  **end**
  $y \leftarrow u_n$
  **for** *subprotocol* $\in$ *determineSubprotocols($\mathcal{D}, \mathcal{R}$)* **do**
    $y \leftarrow y-$ nPartyScalarProduct($subprotocol$)
  **end**
  return $y + v_2$
**end**
determineSubprotocols($\mathcal{D}, \mathcal{R}$)
**Input :** The set $\mathcal{D}$ of diagonal matrices $\mathbf{D_1} \ldots \mathbf{D_n}$ of the original protocol. The set $\mathcal{R}$ of random diagonal matrices used in the original protocol
**Output:** The sets $\mathcal{D}_{subprotocol}$ for each subprotocol
**for** $k \leftarrow 2$ **to** $|\mathcal{D}|-1$ **by** $1$ **do**
  $uniqueCombinations \leftarrow selectKSizedCombosFromSet(k, \mathcal{D})$
  **for** $selected \in uniqueCombinations$ **do**
    $subprotocol \leftarrow \mathbf{D_i} | i \in selected + \mathbf{R_j} | j \notin selected$
    $\mathcal{D}_{subprotocols} \leftarrow \mathcal{D}_{subprotocols} + subprotocol$
  **end**
**end**
return $\mathcal{D}_{subprotocols}$

## 6.3 Verticox+

Verticox+ is an extension of Verticox where there is no longer a requirement to share the outcome data with all parties involved. By making use of the $n$-party-scalar-product-protocol, we have been able to isolate the outcome data to a single server. We make a slight modification to the original algorithm to incorporate the $n$-party scalar product protocol. Table 6.1 explains the notations that will be used throughout the reminder of the article.

Table 6.1: Notations

| Notation Description $K$ | Total number of parties |
|---|---|
| $N$ | Total number of records |
| $\beta_k$ | Coefficients at party $k$ |
| $T$ | Number of distinct event times |
| $t_n$ | Distinct event time of patient $n$ |
| $p$ | Index of iteration |
| $\rho$ | Penalty parameter of ADMM method |
| $z$ | Auxiliary variable |
| $D_t$ | The index set of records with observed events |

The pseudocode for the original Verticox algorithm can be found in algorithm 3.

The main privacy issue lies within solving $\beta_k^p$. This is done using equation 6.1

$$\begin{aligned} \beta_k^p = \left[\rho \sum_{n=1}^{N} x_{tnk}^T x_{tnk}\right]^{-1} \cdot \\ \left[\sum_{n=1}^{N} (\rho z_{nk}^{p-1} - \gamma_{nk}^{p-1}) x_{nk}^T + \sum_{t=1}^{T} \sum_{n \epsilon D_t} x_{nk}\right] \end{aligned} \tag{6.1}$$

The problem lies in the last part of the equation: $\sum_{t=1}^{T} \sum_{n \epsilon D_t} x_{nk}$. This part has a reference to $D_t$, which is the index set of samples with an

**Algorithm 3:** Original Verticox algorithm

**Data:** Local data at each party
**Result:** Converged Cox proportional hazard model
initialization;
**while** *Stopping criterion has not been reached* **do**
  **for** *Each party* **do**
    Solve $\beta_k^p$
    Return the aggregated result to the central server
  **end**
  Sever aggregates subresults
  Server calculates auxiliary value $\overline{z^p}$
  Server updates $z_{nk}^p$
  Server sends $z_{nk}^p$ and aggregation to parties
  Local parameters are updated
**end**

observed event at time $t$ . Therefore, for every time $t$ we need to select the samples with an observed event. This requires the availability of outcome data at every party. In real-world use cases, this is not always possible.

Verticox+ will solve this problem by making use of the $n$-party-scalar-product-protocol. To do that, we translate the inner sum $\sum_{n \epsilon D_t} x_{nk}$ to a scalar product: $u_{kt} = x_k \cdot \overrightarrow{(D_t)}$

In this case, $\overrightarrow{(D_t)}$ is the Boolean vector of length $N$ that indicates for each sample whether it had an event at time $t$ (indicated as 1) or not (indicated as 0). $\beta_k^p$ will now be solved according to equation 6.2.

$$\begin{aligned} \beta_k^p &= \left[\rho \sum_{n=1}^{N} x_{tnk}^T x_{tnk}\right]^{-1} \\ &= \left[\sum_{n=1}^{N} (\rho z_{nk}^{p-1} - \gamma_{nk}^{p-1}) x_{nk}^T + \sum_{t=1}^{T} u_{kt}\right] \end{aligned} \tag{6.2}$$

Since $u_{kt}$ per time $t$ stays constant over iterations, we will only need to compute this once at the initialization step. The rest of the algorithm will remain the same.

A summary of the updated Verticox+ algorithm can found in algorithm 4:

**Algorithm 4:** The Verticox+ algorithm

**Data:** Local data at each party
**Result:** Converged Cox proportional hazard model
At every party $k$ compute: $u_{kt} = x_k \cdot \overrightarrow{(D_t)}$
**while** *Stopping criterion has not been reached* **do**
    **for** *Each party* **do**
        Solve $\beta_k^p$ using precomputed $u_{kt}$
        Return the aggregated result to the central server
    **end**
    Sever aggregates subresults
    Server calculates auxiliary value $\overline{z^p}$
    Server updates $z_{nk}^p$
    Server sends $z_{nk}^p$ and aggregation to parties
    Local parameters are updated
**end**

## 6.4 Time complexity & communication overhead

In this section we will discuss the time complexity and communication overhead of Verticox+. We will start by discussing these aspects of Verticox, to provide a baseline. Afterwards we will discuss the time complexity of the $n$-party scalar product protocol. Finally, we will discuss the consequences of combining these two protocols.

### 6.4.1 Time complexity

Let us consider how the addition of the $n$-party scalar product protocol affects the time complexity. Since the $n$-party scalar product protocol has only been used at the client side, the time complexity at the server side will remain $O(N^3)$, which is the complexity of the Newton-Rhapson optimization.

At the client side, the original computational complexity was determined by generating and inverting matrix $\sum_{n=1}^{N} x_{nk}x_{nk}^T$ are $O(NM_k^2)$ and $O(M_k^3)$ respectively.

### 6.4.2 Time complexity $n$-party scalar product protocol

The time complexity of the $n$-party scalar product protocol is based on two factors, the size of the vectors, and the number of parties involved. The number of parties involved is the major driving force behind the time complexity, causing the runtime to scale factorially as the number of parties increases. The time complexity can be found in table 6.2.

Table 6.2: Time complexity $n$-party Scalar Product Protocol

| | |
|---|---|
| Number of scalar product subprotocols per protocol | $\frac{K!}{x!(K-x)!}$ for each $x$ , $2 \leq x \leq K$, where $K$ is the number of parties involved in the protocol |
| Number of multiplications per subprotocol | $O(N^2K)$ where $N$ is the population size, $K$ is the number of parties involved in the protocol |
| Number of multiplications for 2-party protocol | $O(N^2)$ |

The rate at which the number of sub protocols grows is an unfortunate drawback of the $n$-party scalar product protocol. However, in practice this problem is limited as the $n$-party product protocol can often be kept to a minimum size by only including the parties actually involved in the calculation. In this case the $n$-party protocol is only used to combine the covariates at a single party $k$, the one holding the outcome

labels. As such the number of parties involved will always be 2. By limiting the number of parties in this manner, the protocol remains practical in realistic settings.

### 6.4.3 Communication cost

The original Verticox sends intermediate values $z_{nk}$, $\overline{\gamma}_n$, and $\overline{\sigma}_n$ from the central server to the clients at every iteration. In turn, the clients send back $\overline{\sigma}_{nk}$. This results in a communication cost of $4NK$.

The communication cost of the $n$-party scalar product protocol scales similarly to its time complexity as it is dependent on the number of (sub)protocols. Each (sub)protocol requires $K + K^2$ messages of size $N$ to be sent to the group of parties involved, where $K$ is the number of parties involved in that protocol. As mentioned before, we only use the $n$-party protocol to aggregate data of 2 parties, which means that there will only one (sub) protocol. The total communication cost will be $6N$.

### 6.4.4 Fixed precision

The $n$-party scalar product protocol is designed to work using integer values. However, within Verticox+ it will be used to calculate results that depend on floating point values. In order to make these values useable within the $n$-party scalar product protocol we will make use of fixed-point precision. The values will be scaled by a fixed factor; this factor corresponds to the required precision (e.g. the value will be scaled by a factor $10000$ when working with a fixed precision of $5$ decimals). Once the $n$-party scalar product protocol has finished the final result will be scaled back to the desired precision.

This fixed precision approach makes it viable to use the $n$-party scalar product protocol even when it is necessary to work with floating point values. In principle any level of precision can be chosen, however there

will be a trade-off; a greater precision will result in larger numbers being used in the $n$-party scalar product protocol. This can create technical problems when it results in a number overflow error. Additionally, numbers with more digits will take longer to multiply. As such, a high precision will eventually affect the runtime performance of Verticox+. However, we experimentally determined that a fixed precision of $5$ decimals is sufficient for most purposes. Furthermore, we expect the effect on the total runtime of Verticox+ to be minimal as the bottleneck is outside of the part that utilizes the $n$-party scalar product protocol.

## 6.5 Experimental validation

We ran several experiments to validate our method. We implemented the algorithm in Python and Java, and ran the parties in separate Docker containers. We used a single virtual machine with Ubuntu $22.04$, $8$ cores with a clock speed of $1996.250$ MHz and $32$ GB RAM running in SURF research cloud, which is part of the Dutch national research infrastructure. As data we used part of the SEER dataset that is available on zenodo[156]. The parameters of the algorithm that we kept fixed can be found in table 6.3.

Table 6.3: Experimental parameters

| Parameter | Fixed value |
|---|---|
| Penalty parameter $\rho$ | $0.25$ |
| Fixed precision of $n$-party protocol | $5$ |
| Newton-Raphson precision | $0.00001$ |

We ran $3$ different experiments. In the first experiment, we fixed the number of records to $100$ and varied the number of parties and iterations to see how that will affect runtime and accuracy. Accuracy has

been measured in $4$ different ways. In theory, adding the $n$-party protocol to the original Verticox algorithm will introduce inaccuracy into the model because the values need to be expressed in fixed-point precision. To test whether this is true in practice we ran our implementation of the original Verticox algorithm with the same parameters. We use c-index[81] to compare the predictions of the model to the ground truth. Additionally, we used $3$ metrics to compare the resulting coefficients against ones that have been computed by a central Cox proportional hazards model. For this, we compute mean squared error (MSE), summation of the absolute difference (SAD), and maximum absolute difference (MAD). As can be seen in table 6.4 the accuracy of the central model is identical to the accuracy of Verticox+. This is because the variables in the SEER dataset require limited precision, since they consist of values with no more than 2 digits. Looking at MSE, SAD and MAD (figure 6.1), we can see that the difference between Verticox+ and a Cox proportional hazards learned on centralized data diminishes after a few hundred iterations.

Table 6.4: Performance of original Verticox algorithm

| parties | iterations | mse | sad | mad | c-index Verticox+ | c index central |
|---|---|---|---|---|---|---|
| 2 | 100 | 1.3971e-07 | 1.5828e-03 | 7.7770e-04 | 0.5556 | 0.5556 |
| | 500 | 3.1573e-12 | 5.0113e-06 | 4.3208e-06 | 0.5556 | 0.5556 |
| | 1000 | 1.1810e-12 | 3.2347e-06 | 2.6231e-06 | 0.5556 | 0.5556 |
| | 1500 | 1.6147e-12 | 3.4639e-06 | 3.1000e-06 | 0.5556 | 0.5556 |
| | 2000 | 3.3313e-12 | 5.1218e-06 | 4.4400e-06 | 0.5556 | 0.5556 |
| 3 | 100 | 8.6498e-07 | 2.7485e-03 | 2.2286e-03 | 0.5556 | 0.5556 |
| | 500 | 1.2316e-12 | 2.9918e-06 | 2.7126e-06 | 0.5556 | 0.5556 |
| | 1000 | 1.5045e-12 | 3.8314e-06 | 2.8914e-06 | 0.5556 | 0.5556 |
| | 1500 | 2.1889e-12 | 4.7820e-06 | 3.4278e-06 | 0.5556 | 0.5556 |
| | 2000 | 1.1238e-13 | 1.0021e-06 | 8.1104e-07 | 0.5556 | 0.5556 |
| 4 | 100 | 3.3872e-06 | 5.6448e-03 | 4.4484e-03 | 0.5556 | 0.5556 |
| | 500 | 4.9900e-13 | 1.8713e-06 | 1.7280e-06 | 0.5556 | 0.5556 |
| | 1000 | 5.5778e-13 | 2.4865e-06 | 1.6397e-06 | 0.5556 | 0.5556 |
| | 1500 | 5.0120e-13 | 1.9261e-06 | 1.7280e-06 | 0.5556 | 0.5556 |
| | 2000 | 8.4410e-13 | 2.8067e-06 | 2.1761e-06 | 0.5556 | 0.5556 |
| 5 | 100 | 9.4398e-06 | 9.0487e-03 | 7.4777e-03 | 0.5556 | 0.5556 |
| | 500 | 3.8154e-13 | 1.8684e-06 | 1.4609e-06 | 0.5556 | 0.5556 |
| | 1000 | 2.3788e-13 | 1.3077e-06 | 1.1916e-06 | 0.5556 | 0.5556 |
| | 1500 | 5.5411e-13 | 2.0127e-06 | 1.8174e-06 | 0.5556 | 0.5556 |
| | 2000 | 5.4146e-15 | 2.2211e-07 | 1.7827e-07 | 0.5556 | 0.555 |

The second experiment evaluates how runtime scales with increasing number of covariates (features) in the model. Again, we fixed the number of records to $100$ and the number of iterations to $500$. The number

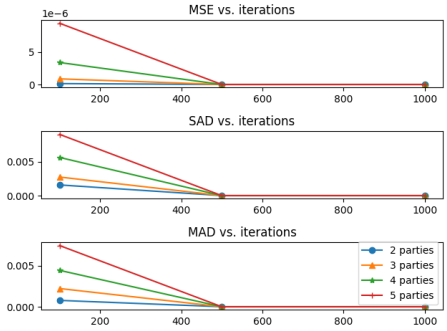


Figure 6.1: The MSE, SAD, & MAD scores of Verticox+

of parties has been fixed to 3. We evaluated the algorithm runtime from 2 and up to 10 features.

As can be seen in figure 6.2, 6.3, & 6.4, our addition of the $n$-party protocol does not negatively affect the runtime. In fact, Verticox+ even has a shorter runtime for preparation as the number of parties increases. This is unexpected and likely relates to the implementation details of the $n$-party protocol. While we reimplemented the original Verticox algorithm in Python, the $n$-party protocol was actually implemented in Java. Since Java is a compiled language, it generally performs faster than the interpreter language Python. In the end, the bottleneck will not be in the preparation time, but rather in runtime of the main part of the algorithm where the model converges. In this part, Verticox+ performs the same as its predecessor.

In the third experiment we fixed the number of iterations to 500 and the number of parties to 3. We set the number or records to 50, 100 or 500 and timed the runtime. As can be seen in figures 6.5 and 6.6, the number of records does not affect the runtime significantly during the preparation phase. Convergence runtime is affected by the number of records, but not more than the original Verticox.

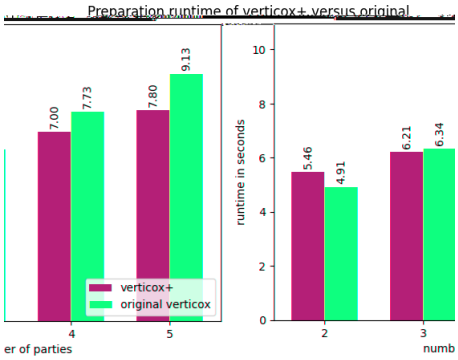


Figure 6.2: Comparison between Verticox+ and Verticox of the runtime duration of the preparation phase

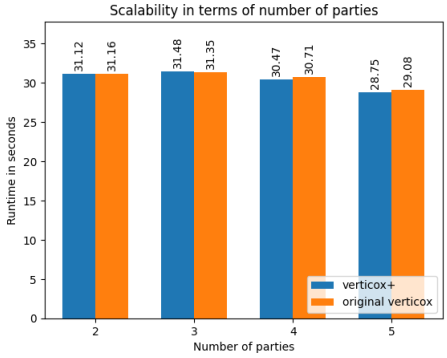


Figure 6.3: Runtime duration of Verticox+ with various numbers of parties

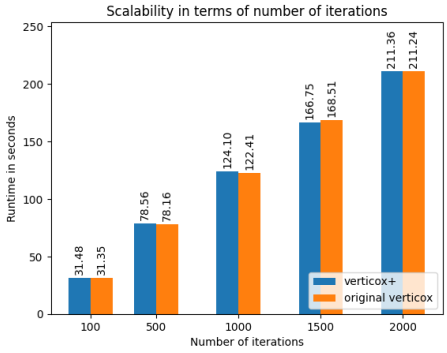


Figure 6.4: Runtime duration of Verticox+ with various numbers of maximum iterations

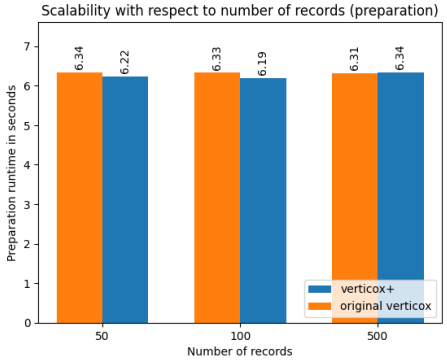


Figure 6.5: Runtime duration of the preparation phase of Verticox+ using various numbers of records

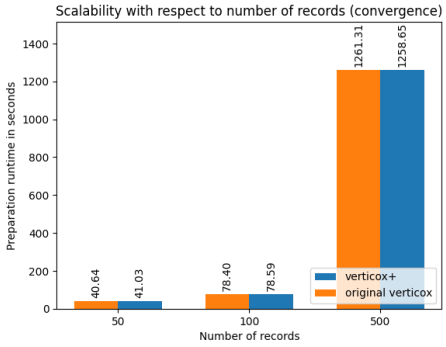


Figure 6.6: Runtime duration of the convergence phase of Verticox+ using various numbers of records

## 6.6 Discussion

With the addition of the $n$-party protocol to the Verticox algorithm, all data used for the analysis can stay at the source, including outcome data. Moreover, the addition of the $n$-party protocol, which is potentially heavy in terms of computation, did not add significantly to the total runtime in our experiments. Neither does the additional overhead introduced by the fixed-point precision. This is because the bottleneck of the computation lies within the Newton-Raphson optimization from the original algorithm. This indicates that Verticox+ is a viable extension of the original algorithm.

There are still a couple of security issues to consider though. The Verticox+ algorithm shares record-level aggregations with the central server. That is, in every iteration the parties share their risk estimates for every record with the central server. Although this is not raw data, it is still patient level information. Additionally, it can be viewed as relatively sensitive data as it represents the risk of a given disease for a specific patient. Direct access to this information could be problematic if it falls in the wrong hands.

However, by placing the server in the care of the party that already owns the class label data the practical risk is limited. Providing this party with the risk scores minimizes the privacy concerns as this party already knows the true labels this risk score represents, and thus would not learn anything new. This limits the risk of direct access to the risk score.

Access to the risk estimates in each iteration also opens an additional possible attack[188, 89, 191]. The aggregating party could attempt to reverse engineer the training data belonging to each other party based on the intermediate values revealed between the iterations. However, this does require the aggregating party to know which attributes are present at each other party. Additionally, reversing this information becomes more complex as the number of attributes at the other party grows.

This privacy concern could be mitigated by moving the central aggregation away from the outcome datasource and performing the central aggregation on a "neutral" server provided by a trusted third party. The outcome data would have to be queried by the central server with the $n$-party protocol. Unfortunately, this means that the $n$-party protocol would have to be run in every iteration, instead of only during the preparation phase. The concern is that this will add a significant increase to the total runtime. Adding the 2-party protocol with complexity $O(N^2)$to the Newton-Rhapson optimization $(O(N^3))$ will turn it into a complexity of $O(N^4)$. Additionally, the $n$-party protocol will add a constant communication overhead to this part of the computation $(O(6))$. Although this overhead is constant, in practice the communication overhead is the bigger bottleneck when compared to the computational cost, and will add significantly to the total duration of the algorithm.

A more practical solution may be to mandate the use of a framework like Vantage6, which provides an infrastructure that explicitly limits what the aggregating party is able to do by only allowing pre-approved Docker images with vetted code to be executed. By explicitly creating this limitation, the various parties involved can establish a sufficient level of trust that no data will be leaked.

This risk, and the limitations imposed by the time complexity of the technical solutions, highlight the need for a comprehensive legal and infrastructure solutions to augment the technical privacy preserving solutions in any real world project. This also means that Verticox+ is best used in a setting where such things can viably be implemented. Implementing such solutions, and establishing the required level of trust, is difficult in an open internet of things setting, where any party is free to join. However, in a formal research setting this is indeed viable.

The $n$-party scalar product protocol brings one additional privacy concern compared to the broader Verticox+ protocol. It requires a trusted

third party, which can generate secret shares and aggregate the intermediate results of the protocol. Similar to the previous concerns, the use of a framework such as Vantage6 is an excellent solution to set up the necessary infrastructure to ensure the reliability of the trusted third party.

## 6.7 Conclusion

In this paper, we have provided an extension to the original Verticox protocol that we dub Verticox+. The original protocol allows the user to train a Cox Proportional Hazard model in a vertically partitioned federated setting. However, the original algorithm relies on the assumption that every party involved has access to the class label for each record. This is unrealistic in a vertical scenario and would most likely require this class label to be shared, which represents a serious privacy concern as the class label used to train a Cox proportional hazard model represents a sensitive attribute, such as a hospitalization event or death due to a certain disease. Verticox+ removes the need for this assumption by using the $n$-party scalar product protocol to perform the relevant calculations in a privacy preserving manner.

Our experiments show that Verticox+ achieves comparable performance to both Verticox and a centrally trained model. This indicates Verticox+ works as intended. Additionally, our experiments show that the added overhead introduced by using the $n$-party scalar product protocol is manageable as the optimization step forms a much more significant bottleneck. As such, the runtime duration is comparable to the original Verticox algorithm as well.

While Verticox+ improves the privacy guarantees, a number of practical concerns remain. The $n$-party scalar product protocol relies on a trusted third party. Additionally there is a theoretical possibility of a malicious party reconstructing an approximation of the data, akin to a

gradient leak attack in deep learning settings. These risks can be mitigated by applying multiple layers of security measures, such as offering access to only a small number of trusted researchers. Additionally the relevant legal frameworks also need to be established. The need for such frameworks also serves as a reminder that purely technical privacy preserving solutions are not sufficient to establish the necessary trust needed for any federated learning project.

The need for such frameworks, as well as the time complexity of Verticox+, does limit Verticox+ to certain scenarios. Scalability concerns, as well as the need for trust third parties and a complexity of creating the necessary legal and infrastructure frameworks, means that Verticox+ is not a great fit for an internet of things scenario with many parties, all of which have an extremely low level of trust. However, in formal settings, where it is easier to vet the parties involved, and where parties have access to the technical infrastructure necessary to deal with the scalability issues, it is a great tool in the federated learning toolbox.

## 6.8 Future work

There are currently three major limitations that we would like to improve upon. The current implementation of Verticox+ has not been made to deal with a hybrid split in the data, that is to say a split that is partially horizontal and partially vertical. While certain parts, such as the $n$-party scalar product protocol, do not need any additional work to fit in a hybrid setting, we need to determine if it is possible to use the algorithm as a whole in a hybrid setting.

Secondly, the role of aggregator currently befalls to the party that owns the outcome data. If the role of aggregator could be moved to a neutral party without data, it would not know which records the intermediate values are linked to. This lowers the risk of data leaking. Lastly, we wish to improve the runtime complexity of the optimization step, for example by using a different faster optimization algorithm. This step is currently a considerable bottleneck in the algorithm, and improving

it would lead to significant gains in terms of the running time of the algorithm.

# 7

# A Critique of Current Approaches to Privacy in Machine Learning

## Abstract

Access to large datasets, the rise of the Internet of Things (IoT) and the ease of collecting personal data, have led to significant breakthroughs in machine learning. However, they have also raised new concerns about privacy and proprietary data protection. Controversies like the Facebook-Cambridge Analytica scandal highlight unethical practices in today's digital landscape. Historical privacy incidents have led to the development of technical and legal solutions to protect data subjects' right to privacy. However, within machine learning, these problems have largely been approached from a mathematical point of view, ignoring the larger context in which privacy is relevant. This technical approach has benefited data-controllers and failed to protect individuals adequately. Moreover, it has aligned with Big Tech organizations' interests and allowed them to further push the discussion in a direction that is favorable to their interests. This paper critiques current privacy approaches in machine learning and explores how various big organizations guide the public discourse, and how this harms data subjects. It also critiques the current data protection regulations, as they allow superficial compliance without addressing deeper ethical issues. Finally, it argues that redefining privacy to focus on harm to data subjects rather than on data breaches would benefit data subjects as well as society at large.

## 7.1 Background

The promise to deliver innovation in fields as diverse as healthcare, transportation and education has made it difficult to ignore the appeal of collecting and processing vast amounts of personal data. Access to large datasets, the rise of the Internet of Things (IoT), and the ease of collecting personal data, have led to significant breakthroughs in machine learning[93, 198, 79, 176]. However, they have also raised concerns about privacy and proprietary data protection[14]. Awareness of privacy issues in the era of Big Data is growing, fueled by recent controversies such as the Facebook-Cambridge Analytica scandal[85, 145] and reports from privacy watchdogs like the Mozilla Foundation[1], which highlighted the unethical practices that have become commonplace in today's digital landscape. Personal data is extremely valuable[133] and often harvested without the knowledge or consent of individuals, leading to potentially negative consequences, not only for them, but also for society as a whole.

Partially in response to these concerns, European lawmakers created the General Data Protection Regulation (GDPR) in 2016[59]. In the US, the State of California soon followed suit, implementing the Californian Consumer Privacy Act (CCPA) in[23], soon amended by the California Privacy Right Act (CPRA). These two acts however only apply to Californians, and there is no federal-level data protection regulation in the United States outside of the Health Insurance Portability and Accountability Act (HIPAA), which only applies to health data. Both the GDPR and CCPA provide enforceable rights to data subjects and clearly define the notion of lawful data processing, with real repercussions in case of non-compliance[57, 70, 144, 59, 60, 31, 60] [1]. This formed the perfect context to encourage the further development of

*This research received funding from the Netherlands Organization for Scientific Research (NWO): Coronary ARtery disease: Risk estimations and Interventions for prevention and EaRly detection (CARRIER): project nr. 628.011.212.*

[1] See GDPR, Chapter 8. Specifically, GDPR, Article 83 (4-6), and Article 84.

so-called "privacy-preserving" data analysis solutions enabling machine learning models to be trained without compromising privacy and therefore avoiding data protection related fines. Various metrics, such as k-anonymity[171], sensitivity and $\epsilon$-differential privacy[55, 56], have been established, advertised as ways to measure such privacy-preservation in an objective and generalizable manner.

This paper aims to critically reflect on the current approaches to privacy in machine learning. First, we will briefly introduce the concept of privacy as understood within social science and law. We will argue that privacy has increasingly been approached as a mathematical concept, explaining how this technical approach, while beneficial for data-controllers, fails to protect the interests of data subjects. Next, we will consider the role of Big Tech in defining what should or should not be considered private, and how their influence significantly impacts the social understanding of privacy. Finally, we will discuss how the current situation might be improved to benefit individuals and society as large, by arguing for a shift from privacy-preserving machine learning towards an approach focused on risk assessment and harm mitigation.

## 7.2 What even is privacy?

To assess the question of privacy preservation, we first must understand the concept of privacy itself. In this section, we will present how it is explained in social science and law. These views will then be contrasted to the understanding of privacy within data science and machine learning. It goes beyond the scope of this paper to define privacy in detail. This section merely serves to illustrate the topic. For more details we refer the reader to the broader literature. Warren and Brandeis[190] gave one of its first definitions, presenting privacy as "the right to be left alone" in 1890[2], as a response to the increasingly intrusive behavior of the newspapers and paparazzi of the time. Their

[2]To see the development of the concept over time, see also Alan F. Westin[193].

article became a catalyst to the debate on individuals' right to control other people's access to them. A more recent understanding of privacy was developed by Nissenbaum[128, 129], who argues that the level of protection required is dependent on context. Indeed, sharing personal information with one's doctor is perfectly acceptable, yet that same information ending up in the hands of financial institutions is much less so. A universal definition is unlikely to emerge, yet some ideas are clearly associated with the concept of privacy, such as: autonomy, control and self-determination[148, 42, 22, 168].

In European law, a clear definition of privacy is lacking. Rather, the concept of "personal data"[59][3], namely data that can be linked back to an individual, is considered data that requires protection, and, by extension, should be considered "private". The right to privacy, despite not being mentioned directly or clearly defined, can thus be understood as the motivation behind much of the legislation surrounding data protection.

## 7.3 Privacy as a mathematical concept

A different understanding of "privacy" is its interpretation within the realm of machine learning and data science, and even within this realm, definitions of privacy vary. Repercussions for non-compliance being severe, the GDPR has sometimes had the unintended consequence of hindering data-sharing across institutions and EU member states, even for research purposes[138]. Yet the appeal of conducting research based off large amounts of data processing has not diminished. As a response, technological solutions have been developed to reduce privacy leaks and thus data processing becomes more compliant with existing legislation. A few of these solutions include Multiparty Computation (MPC)[202], Federated Learning (FL)[55], homomorphic encryption[135], and synthetic data[38, 76, 45, 58] to replace real data when training models. Finally, attempts

[3] See GDPR, Article 4(1)

to measure privacy in a concrete mathematical manner have been developed. In this context, privacy is approached mainly using the following three nonconflicting methods: (a) by setting and utilizing privacy thresholds, (b) by focusing on limiting data breaches, which we will discuss in section 4, and (c) through the use of so-called "privacy-preserving" technologies.

### 7.3.1 Setting and utilizing privacy thresholds

Whether it is hypothesis testing with p-values[9], creating a sufficient level of privacy with a privacy budget using schemes like k-anonymity or $\epsilon$-differential privacy, or deciding if a model's predictions are accurate enough, statistical measures use specific thresholds as cut-off points to determine if the scenario passes a test. However, these thresholds are often set based on historic precedent rather than any truly objective reasoning. While these thresholds can be informative to a certain extent, the focus on these historic precedents causes researchers to be mostly concerned with simply passing this threshold, which has resulted in several important problems.

First, researchers are often not aware of how and why these thresholds were set[123]. This is especially true for researchers who are not statisticians themselves. For example, most researchers working with quantitative data know about p-values, but probably would not be able to explain why the common threshold used to indicate statistical significance was set to 0.05. Yet, they will still accept or dismiss research based on this threshold. Second, this can also tempt said researchers to tweak their experiments in various ways to pass this test, which might mislead research findings[82, 19, 91]. In the context of privacy this may mean that a researcher may mindlessly accept a privacy solution because a statistical test shows that with $p < 0.05$ no data is leaked. Lastly, while these measures are often effective at ranking different scenarios, it can be extremely difficult to meaningfully explain the practical differences between a "good" and a "bad" score. Combined with the arbitrary nature of the threshold this makes it very dif-

ficult to explain why a solution is dismissed as "bad", other than a simple "computer says no". While there is occasionally pushback against this blind reliance on arbitrary thresholds, but it is still a problem that can commonly be observed.

### 7.3.2 Privacy-preserving technologies

Approaching privacy as a technical problem has inevitably led to attempts to solve it technologically. In recent years, privacy-preserving or enhancing technologies have been developed to minimize the risk of data leakage and data reidentification. These solutions enable organizations to undertake multi-institution research projects[153]. They have notably been used to develop various commercial products, such as personalized advertisements, predictive text models for mobile phones, and recommender systems based on users' profiles and purchase history, but also to improve public services. Hospitals have used Federated Learning to combine patient data in a privacy-preserving manner to train machine learning models for disease diagnosis, which in turn improves healthcare offerings[93, 102, 200, 199, 180, 176, 106, 15].

The progress made in developing these privacy enhancing tools is undeniable. However, current literature is primarily focused on the technical aspects of privacy and ignores other important issues. Additionally, while mathematical measures of privacy may allow solutions to be ranked easily, this ranking is largely a theoretical exercise, and it may be difficult to determine the exact practical differences between two competing solutions. Lastly, reducing privacy to a purely mathematical problem gives it an "objective" veneer, which can be used to whitewash a project. Additionally, large tech organizations may push their preferred metric in an effort to shape the discussion on privacy in ways that benefit their business model. In the following sections we will elaborate these topics.

## 7.4 The misplaced focus on preventing data leaks and its consequences

Privacy is important in 4 aspects of the development and implementation of data-driven projects. These aspects are: (1) the training of the model, (2) the use of the model, (3) the technique or technology deployed, (4) aim and application of the project. It is only when privacy is accounted for in all 4 of these aspects that such a project can be considered "privacy-preserving". These aspects tend to compete for researchers' attention. For instance, problems that arise during training (1) might include technical challenges such as data leaks or poisoning attacks by malicious parties. Similarly, issues falling under aspect (2) are primarily technical, such as concerns about model inversion. Additionally, there may be practical problems that may need to be solved regarding the model use, for example where is the model hosted and how it accesses new data. However, the use of the resulting analyses (4) introduces more social and ethical considerations. For example, could the resulting model lead to discrimination, or reinforce existing biases[184, 53]? Answers to these concerns are often less straightforward.

Likewise, ensuring the proper technique is applied to a specific problem (3) is relatively straightforward to establish and control. For example, if a project requires zero trust then it is trivial to establish that techniques that rely on a trusted third party are inappropriate. However, determining whether these techniques are implemented in an ethically responsible manner, and will not be abused in the future, for example after a change of leadership, is considerably more difficult. Ethically assessing an algorithm is challenging; but the manner in which it is used deserves attention. As a result, it is common to focus purely on the technical aspects, ignoring the ethical, legal and societal aspects (ELSA), of which privacy is part. Technical data leaks usually result in damage to the data-controller, revealing industry secrets and/or causing organizations to lose their commercial advantage, as well as lead to significant reputational damage and/or fines to the data-controller.

This has led to zero-trust policies and complicates large cooperative projects, as sharing data is often deemed too risky or complex to execute safely.

However, these technical leaks do not necessarily lead to real harm for the data subjects. For example, the data leaked might not be directly identifiable without the use of additional information that is only available to the data-controller. While it may be technically possible to combine and cross-reference various external data sources to identify individual data subjects, this is unlikely to be feasible and the risk should be weighed against the effect and probability of successful attack. Additionally, the more sensitive the data, the harder cross-referencing becomes; sensitive data is not only better protected, but also harder to acquire[59][4]. This greatly limits the real harm done to the data-subjects. Finally, the step from a data leak to personal harm or damages of an individual often requires an active and conscious act of someone, which may not be the case.

To illustrate this, let us look at one of the most famous examples of data reidentification, the Netflix competition of 2006, in which researchers used a freely available public IMDB dataset to re-identify the records contained in the Netflix dataset[125]. This was despite the fact that the Netflix dataset was considered not to contain any sensitive identifiable data. Additionally, had the IMDB dataset not existed, it would have been possible to create a reference database via phishing attempts, such as using seemingly harmless quizzes on the internet[136], or by

[4]See GDPR, Article 9. Processing of special categories of personal data. See GDPR, recital 51. further clarifying the protection of sensitive personal data, lifting the restriction on processing in cases were explicit consent is provided by the data subject, or: '[...] for the performance of a task carried out in the public interest or in the exercise of official authority vested in the controller.'. See GDPR, recital 52. which further derogates the processing prohibition of special data for the public interest: 'Such a derogation may be made for health purposes, including public health and the management of health-care services, especially in order to ensure the quality and cost-effectiveness of the procedures used for settling claims for benefits and services in the health insurance system, or for archiving purposes in the public interest, scientific or historical research purposes or statistical purposes.'

abusing data-leaks. In this instance, the risk of reidentification was very high, thanks to publicly available reference datasets, as is clear in hindsight. However, it is important to note that the real harm to data-subjects from this leak was minimal or non-existent as it contained little to no new information compared to the already public IMDB database, and knowing who makes which comments on which movie has very limited risk of damage to that reviewer anyway.

In contrast, a medical dataset does not have an easily accessible public database that could be used to re-identify individuals. It's also much less likely one would be able to successfully employ phishing websites to gain access to a dataset to cross-reference. No matter how personal the medical data may be, they are, in practice, extremely unlikely to lead to patient reidentification and subsequent harm and damages.

Another example of how minor leaks are often presented as major problems can be seen in the following paper. Slokom et al. claim synthetic data is not privacy preserving because they devised an attack which revealed sensitive data[166]. However, they overlook key contextual aspects. Most notably, that the leaked sensitive information is limited and often does not meaningfully improve upon baseline prediction of sensitive attributes; in some scenarios, it even performs worse. Even when successful, the attack only slightly exceeds random performance, achieving about $60\%$ accuracy on a sensitive binary attribute. This high level of uncertainty means that such an attack cannot be deemed a serious privacy threat in this scenario. If attacks with such high levels of uncertainty are deemed major breaches of privacy, publishing any analysis would be impossible, as even basic analyses reveal information[54, 164]. While the attack vector is relevant and a potential concern in specific scenarios, Slokom et al. do not address its practical limitations.

These examples are illustrative of the broader problem within privacy-preserving literature caused by the focus on technical research questions and failure to consider contextual practical implications and limitations. These studies usually only consider the worst-case scenario

where the attacker has practically unlimited resources, or it is assumed that a reference dataset exists to identify individuals. Each record is presupposed to always be unique enough to be identifiable, even if best practices show that such outliers are uninformative and should be removed from your dataset during preprocessing, thus providing a minimum level of privacy via k-anonymity[171]. Additionally, the impact of an attack is based on the amount of data revealed, not on the contextual harm it can do to the data subject. For example, if an attacker attacks two image recognition models, one trained on faces, one trained on MRI images, and in both cases retrieves an image, and nothing more, this is treated as an equivalent leak with equivalent damage in both scenarios. This ignores the fact that one image may be easier to identify but contains relatively limited sensitive information, while the other image contains a lot of sensitive information but may be more difficult to identify. The potential for harm towards the data subject is vastly different in the two scenarios.

In summary, the real impact on the data subjects in the given context is rarely considered, and neither are their preferences. This leads to a focus on secrecy over privacy. Which in turn also leads to researchers ignoring other important, and often connected, ethical aspects such as the risk of biases harming the data subjects. It also leads to researchers overlooking alternative solutions, such as legal solutions. Lastly, because of this focus on secrecy and the relatively short-term goal of protecting the data-controlers' interests, researchers and engineers often ignore the long-term implications of a project for the data-subjects.

The focus should instead lay on how the different stakeholders involved are affected, with a strong focus on the data subjects, by potential leaks, as well as how "normal" use of the data would affect them. Additionally, researchers and policy makers should rely less on generic definitions of the risks involved, instead the risks should be estimated on a project-by-project basis. Lastly researchers should acknowledge and actively push for alternative solutions. They should not be allowing privacy preserving technologies to be used to whitewash questionable projects. We will further discuss this practice of

whitewashing in the next section.

## 7.5 The role of Big Tech in defining what is, should or should not be private

In order to create and maintain a situation where they benefit, Big Tech companies have successfully pushed their own agenda, by influencing our understanding of privacy. This paper has introduced 4 aspects of AI privacy in section 4 above: (1) the training of the model, (2) the use of the model, (3) the technique or technology deployed, and (4) the aim and application of the project. Big Tech companies, however, almost entirely focus their privacy preserving efforts on the first 3 aspects. Indeed, those are the phases that allow them to focus on "objective" technical mathematical problems, for which easily demonstrable solutions can be found. Aspect 4, however, is entirely neglected, as it is more likely to raise questions of a more ethical nature that cannot be addressed straightforwardly. This attitude is in alignment with the general practices highlighted in section 4. This section goes further with that observation, arguing that Big Tech is directly involved in maintaining a reductive understanding of privacy as an issue that can be fixed technologically. This allows them to circumvent deeper questions about their business model. Rather than to rethink the way that they collect and process data, they instead (a) created services advertised as "free" but that users in fact pay for by giving away personal data, (b) hide under the promise of "privacy-preserving" techniques, (c) use these techniques to justify targeted advertising and (d) eliminate dissent and competition through brain draining and lobbying.

### 7.5.1 The expectation of "free" online services

Meta, Amazon, Alphabet: all offer services – such as online shopping apps, search engines, and social media apps - that appear free and are so convenient that their use has become the norm. That those services

are not 'free' for use but paid by the trade-in of personal information is not often clear to users, although awareness has been rising. In the EU, lawmakers have deployed efforts to protect the rights of individuals to their personal data with the General Data Protection Regulation. While the GDPR has, at times, constituted a minor setback or annoyance for these companies, it did not result in them rethinking their incredibly profitable business model. Instead, they opted for privacy-washing and complying in ways that could be qualified as questionable. For example, back in October 2023, Meta announced that it would give its European users the choice to use their platform without being shown relevant ads[61, 150], if they agreed to pay a monthly premium of 9,99€ for the web versions, and 12,99€ for the apps. This was a direct response to European legislators warning Meta that they could not force its users to consent to their data being extracted by making them leave the platform if they wished to preserve their data[71]. Yet, the effect was the same: Meta users were greeted with a long wall of text and the choice to tick either one box or the other, deciding whether they wanted to keep using the platform for free, or pay a subscription. This tactic, qualified "pay or okay"[173], was harshly criticized by the European Data Protection Board. Indeed, the hefty price tag, together with growing apathy amongst social media users[112, 80, 189], is unlikely to lead to them paying such a sum in the name of privacy.

### 7.5.2 Technological privacy preservation

Aside from offering convenient, attractive and "free" services to their users, Big Tech companies would also suggest that their processing of your personal data is entirely safe and private. In order to sell this point of view, they are heavily involved in the development and promotion of privacy-preserving technologies. Given that their business model relies on the use of vast amounts of personal data, they have a clear stake in the development of such technologies, as well as their perception by legislative authorities and the public. One of such technologies is federated learning, a term coined by Google, which heav-

ily relied on this technique to develop personalized text prediction in the Gboard, the virtual keyboard with auto-correct and text prediction functionality[15]. Many promotional materials can be found to sing the praises of this learning method, all the while obscuring the fact that the company, although indirectly so, is accessing the contents of our emails, messages, and all other text input processed using Google's software. Is personalized text prediction worth such an invasion of privacy? The default on our machines would suggest that the answer is yes.

Given that personal data have become highly commodified and profitable goods, companies naturally try to accumulate as much of it as possible and allow their data scientists to run numerous analyses on it. Hiding behind the promise of privacy-preservation enables this data behaviour: after all, if the data used is anonymous[59][5], why should consumers or legislators be worried[6]? However, this anonymity claim is flawed: while privacy preserving techniques (PPTs) can guarantee a certain layer of security for a single analysis, one could run multiple queries concurrently in order to reveal private information. Just because a technology makes data processing safer, does not mean safety is guaranteed. On the contrary, being truly concerned with privacy would mean implementing queries that are predefined and limited in scope for specific high-level functionalities: for instance, building a model. Such an approach would align more closely with EU legislation and its guidelines on ethical & trustworthy AI[141], which promote data minimization, human oversight, and prevention of harm. Big tech companies' tendency to greedily accumulate data directly contradicts these principles.

[5]See GDPR, Article 2(1), respecting Article 4(1), and recital 26. Truly anonymous data, as explained in recital 26, does not fall within the material scope of the GDPR (Art.2(1)).

[6]Personal data protection legislation largely does not bite on anonymous - truly de-identified - data; research ethics has always seen consent and anonymisation as the gold standards of protection of the individual. This leaves open other dignity breaches in relation to de-identified data.

Rather than fundamentally rethink their unethical business practices, Big Tech has instead focused on advertising technological fixes to the issue of privacy, using objective numbers to solve an issue that is, in fact, societal. As such, the mathematical conception of privacy entirely benefits their agenda. It is much easier to develop new technology to remain under a set privacy threshold rather than to think more deeply about the ethical and safety concerns associated with their processing of personal data. Furthermore, this focus on secrecy and preventing data leaks presents a clear commercial advantage. To ensure that their vision is widely disseminated, Big Tech has funneled a significant amount of energy and funds into research that aligns with their agenda and promotes the use and effectiveness of PPTs[192, 134, 205, 130]. Consequently, views commonly held in privacy-preserving literature tend to align with the interests of these companies and organizations, a point this paper will elaborate on further in section 7.5.4.

### 7.5.3 Privacy preservation and audience targeting

While PPTs afford a higher level of safety in gathering and processing personal data, their use (aspect 4) can result in models which would constitute an intrusion in one's private life. A blatant example of this is targeted advertising[170, 11, 142, 47], which is usually activated by default, and is not easily disabled. Furthermore, allowing your data to be collected is often the condition to access or use most services provided online. Companies claim that they are able to perform such a service while preserving user privacy[147], by using state of the art PPTs. However, it can easily be argued that targeted ads themselves present a breach of privacy. They reflect a user's browsing history, past purchases, etc. Who hasn't had the experience of searching something up, only for ads for that very item being plastered all over the next website we visited? Other examples below are empirical evidence of these practices. Expectant mothers are likely to engage with posts and hashtags related to pregnancy, leading to the ads being showed to them being very baby-centric. This could lead to potentially in-

advertently revealing pregnancies if one uses their personal device in view of someone else, but it can also lead to personal harm. In 2018, a woman reported that Facebook would continuously show her ads for baby products while she was grieving the loss of her unborn child[21]. The company had successfully detected her pregnancy, but not that it had resulted in a stillbirth. With Roe v. Wade being overturned by the Supreme Court in the US in 2022, additional concerns are rising regarding the right to privacy about one's pregnancy status[95]. The popularity of period-tracking apps is leading to fears that such data might end in the hands of prosecutors trying to enforce the criminalization of abortion. A growing body of literature discusses the many risks associated with Big Tech's access to information about female reproductive health[119, 161, 40, 120, 83]. The Cambridge-Analytica scandal[85] has also shown that ads can be used to successfully influence democratic elections by targeting the individuals most likely to be swayed[7], invading user privacy and using the information collected in the process to manipulate them.

### 7.5.4 Biased research and lobbying

A few powerful companies hide behind the label "Big Tech": arguably the most important are Meta, Alphabet, Microsoft, Amazon and Apple[181, 4]. Together, they form an oligarchy, dominating the IT market worldwide. But this influence does not stop there. Recent studies have shown that Big Tech financially backs a large proportion of academics in the field of AI[140, 3, 121, 130]. In doing so, they ensure that research aligns with their interest. An example of this is how research is currently being conducted on the topic of "fairly" rewarding data-contributors[113]. This would involve rewarding them proportionally to the value of the data they contributed, in a privacy-preserving manner, to a project. Such a system would entirely benefit these oligarchies, as they are the legal custodians of the largest amount of data, and would be completely irrelevant as far as the individual data subjects are concerned as individual subjects will never provide enough data

to receive meaningful rewards under such a scheme. It would disproportionately hurt marginalized communities, which already benefit the least from improvements in AI, while suffering the most from its side-effects[64, 12, 197]. In funding such research, oligopolies gain credibility and build trust by claiming to implement "fair" and "privacy preserving" AI, even though they are clearly serving their own interest, even at the cost of harming others. Consequently, government bodies, who rely on academia for guidance on how to regulate AI, are likely to be influenced by this agenda too[3].

When they are not actively draining brains from academia or from promising startups[73], Big Tech companies actively challenge what little is left of their competition through lobbying. It is interesting to consider the difference in the western public perception of TikTok and Meta, two companies that offer fundamentally similar services with the same modus operandi. That TikTok is facing being banned in the US whereas Meta is allowed to thrive is therefore puzzling, until one becomes aware of the role that the latter played in that situation. It was indeed reported that Meta had spent millions on lobbying for such a ban[159], insisting that the Chinese-owned TikTok represented not only a threat to the privacy of its users, but also to national security, helped in the process by US lawmakers' long history of sinophobia, which the COVID-19 pandemic only worsened[165]. In the EU, the app was banned from the official devices of government personnel for similar reasons[114], while the use of US-owned social media platforms remains allowed. This raises the suspicion that Meta rid itself of its biggest competitor on the market, by presenting itself as less of a threat than its Chinese counterpart[110].

## 7.6 Discussion

The way the issue of privacy is understood, approached and "solved" within the field of machine learning is currently flawed. Rather than serving the interests of the people whose privacy is at risk, arguably it

aligns with those of Big Tech companies. They are able to profit from constantly exploiting our personal sphere, extracting as much data as possible, and selling this data to third parties that use this information to either sell us goods and services or, more worryingly, influence our beliefs and behaviour. This situation needs to change. This section goes over potential solutions to some of the problems highlighted above.

First, the definitions of privacy need to be improved. Currently, what is considered an invasion of privacy will not necessarily lead to harm to data subjects. A great deal of effort is deployed in protecting data against such invasions, even when they do not harm data subjects in any tangible way. Making possible harm to data subject central to that definition would be a huge improvement. As of now, there is a disproportionate focus on data leaks, which is more likely to benefit data-controlers, their trade secrets, and their position of power on the (AI) market, rather than data subjects themselves. Additionally, acknowledging the fact that privacy itself is a nuanced concept, which cannot entirely be solved through technological means, would be a more honest approach. Acknowledging this limitation would help counter the false sense of security that one's data can be kept $100\%$ private even when harvested by Big Tech. A false sense of security that these companies will happily use.

Under the current definition of privacy within machine learning, a data leak comprising information about a random unknown patient's blood type that could very difficultly be linked back to them would be considered a privacy issue, whereas a specific individual woman being labeled as "pregnant" even when she did not voluntarily share that information about herself, might not be considered a privacy issue. This situation is nonsensical and is not beneficial to data subjects.

Secondly, when it comes to European regulation, while steps have been taken in the right direction, they are insufficient to generate change. For example, while the GDPR introduced the option of consent for

personal data processing, this condition is easily met by implementing pop-up, sometimes pre-ticked consent boxes to websites and apps that collect personal information[88], by implementing "pay or okay" subscription models to give users the illusion of agency, or more recently, using legitimate interest as a justification to unilaterally decide to employ user information to train AI models, as well as by making it the user's responsibility to become aware of this policy and to object by filling out a form that demands that the user justifies their decision[77, 116, 67, 86, 158]. Using clever legal loopholes, companies are able to continue their activity in compliance with European legislation, without meaningfully reducing the possible harm caused to data subjects. The vision of the EU is short-sighted and fails to address the deeper issues caused by the very business model of these companies, which have cleverly avoided making any real change to their unethical practices. Focusing on user consent as a legal basis for data processing does not seem to be the answer when many data subjects have grown accustomed to using free apps and services in exchange for data or feel like they have lost complete control over their personal information. Additional steps in the regulation of AI have been taken, comprising the Digital Market Act, the Digital Services Act and the AI Act, however they are likely to have a similarly limited effect, although perhaps the observed shift[99], in the AI act, to focusing on possible harm and risks to humans, which is one of the 7 guiding principles of trustworthy AI according to EU, could lead to some more promising results. As discussed, data science projects often focus on the training and implementation stages of their models, while neglecting their possible real-life consequences. Becoming more conscious of these long-term implications would make it easier to identify potential risks further down the line. This would not only be an improvement on a project-by-project basis, but also create a healthier culture, where data scientists look beyond their direct responsibilities toward potential future problems. Furthermore, a clearer understanding of the real impact of data leaks onto data subjects, rather than data-controllers, would be beneficial. Current research focuses on theoretical impacts, rather than realistic impact. Identifiable data may be of extremely limited value to

attackers, and reidentification itself might not necessarily lead to harm to the data subjects.

Thirdly, while shifting to a risk or harm-based approach rather than a consent-based or technologically private approach would be an improvement, another step in the right direction would be to give more importance to the purpose of data processing. While exceptions to certain GDPR obligations are in place for data processing that is undertaken for research or common good purposes, it is still frequently the cause of much confusion and frustration for many who perform such work. Research has suffered time and time again from attempting to comply with rules for which they were not the primary target. At the same time, EU regulation has failed to significantly hinder unethical and invasive large scale data collection and manipulation by Big Tech companies. Purposes such as targeted advertising and online profiling should be more strictly regulated, especially when they have proven to be harmful. Without a bolder regulatory approach, EU data subjects will continue to see their data being extracted from them and sold to third parties that may not serve their interests. Furthermore, as this is often done without the subjects' knowledge, existing legal recourse becomes virtually inapplicable: without information about who is accessing or processing one's data, and for what purpose, how would one request that they cease to do so? Lack of transparency about data handling has led to the inability for people to effectively exercise their rights to control their personal information.

## 7.7 Conclusion

The machine-learning field has attempted to reduce the complex notion of "privacy" to a purely mathematical, technically solvable problem. This has led to several issues: the creation of privacy thresholds that hold little meaning, the claim that certain technologies will guarantee that personal data will remain private, and an overall focus on the development of such privacy-preserving techniques

while completely neglecting longer-term effects of large-scale data-intensive projects. Treating privacy like a simple, solvable issue has allowed the Big Tech oligarchy to continue profiting from harvesting data from millions of users without having to pay sufficient attention to the potential harm caused to data subjects by their activities, justifying their behaviour by advertising their technologically robust privacy-preservation techniques. The steps taken in data protection law so far have not had a significant impact and led to new issues for actors that process data for purposes should instead have been facilitated, such as research.

It is our hope that this article will spark new discussions surrounding the role of Big Tech, and researchers themselves, in defining privacy not only within the machine-learning field, but also in policy-making and public discourse. These could in turn help reshape the privacy protection framework so that it focuses on those who truly should be protected: the data subjects.

# 8

# Discussion

The desire to unlock the knowledge hidden in data, and the realization that individual institutions struggle to gather sufficient data has led to an ever increasing wish to use siloed data. This required the sharing of data and brings with it numerous legal and practical concerns which need to be addressed. The field of federated learning arose to address these concerns[93, 198, 79, 176, 102]. Within this thesis, we have presented several novel works which we believe will move the field forward, especially in vertically partitioned scenarios. The solutions presented in this thesis are largely focused on solving the technical challenges associated with such scenarios. However, they are not limited to the pure technical considerations. Furthermore, we are of the opinion that limiting our work to purely technical solutions would be a disservice to the individuals whose privacy we claim to protect. In the remainder of this chapter we will briefly discuss our technical contributions to the field, followed by a discussion of the ethical considerations we think are important. Lastly, we will provide a summary of the direction we think researchers should move into.

## 8.1 Technical results

Throughout this thesis we have presented several technical solutions for federated learning in a vertically partitioned scenario.

In chapter 2, we introduced the privacy preserving $n$-party scalar product protocol. This protocol can be used to answer basic queries about a federated dataset, such as how many individuals fulfil a specific requirement in a privacy preserving manner, even when data is vertically split over multiple parties. In this chapter, we show how the original protocol[50], which could only handle two parties, can be extended to scenarios with more than two parties. This new $n$-party protocol can be used as a building block in more complex analyses, such as the training of a machine learning model[194, 50, 38, 178].

In chapter 3, we used the privacy preserving $n$-party scalar product protocol[37] as a building block to create a new algorithm "VertiBayes". This algorithm can be used train a Bayesian network in a federated setting while preserving privacy. Bayesian networks are a commonly used model, popular for the ease with which they can be interpreted without needing a technical background. Additionally, they can incorporate existing expert knowledge, making them a popular solution in fields where such knowledge is available[137, 186, 27, 117]. VertiBayes is the first vertically partitioned federated implementation of a Bayesian network that can be used with an arbitrary number of parties.

In chapter 4, we present a literature review on the use of ensemble learning[131, 151] within federated learning. Our initial hypothesis was that ensemble learning makes is natural fit to handle the split nature of federated learning[182]. Additionally, the use of ensembles would provide a natural level of privacy protection, as it reduces the need to communicate between parties. However, not only is the current use of ensemble learning in a federated setting limited, but ensemble learning was sometimes viewed as a completely different, competing solution[52, 78]. After this surprising finding we attempt to build

our own ensembles in chapter 5. In this chapter, we introduce the federated Bayesian network ensemble (FBNE); an ensemble of Bayesian networks which takes advantage of the federated nature of the data. We show that FBNE is a viable alternative to VertiBayes[38], providing a number of advantages; a reduced communication overhead noticeably reduces the training time, it is simpler to use FBNE in a federated nature to classify new records, and lastly, ensembles can potentially help deal with biases in the training data[32]. Biases are likely to be present in a federated scenario as the different parties involved may serve different populations and may follow different protocols, both of which can easily introduce biases into the dataset. It achieves these advantages while retaining its performance, and potentially even outperforming VertiBayes in the right circumstances. However, it should be noted that the inherent disadvantages of ensembles remain present. Ensembles are more difficult to interpret than a single model. Additionally, the base classifiers used in the ensemble need to achieve a minimum quality. For example, should individual parties have too few records to build local models with sufficient quality, an ensemble of these models will perform poorly.

Moving on to chapter 6, we utilize the privacy preserving $n$-party scalar product protocol to improve the original Verticox algorithm[39], creating "Verticox+". The original algorithm can be used to train a Cox proportional hazard model in a vertically partitioned scenario. However, it relies on the assumption that the event time is known locally at each party. This assumption is impractical in a realistic scenario as a vertically partitioned scenario implies each attribute, including the event time attribute, is only known at one party. This would imply that the original Verticox algorithm requires the sharing of this event time attribute, which presents a potential privacy concern. Verticox+ removes this assumption, thus removing the privacy concern and broadening the potential use-cases of Verticox.

These technical chapters show that it is feasible to create technical solutions to perform a given analysis in a privacy preserving manner.

## 8.2 Ethical and cultural considerations

While the bulk of this thesis is focused on providing technical solutions privacy cannot be viewed from a purely technical point of view. Throughout this thesis we occasionally touch upon the broader context of privacy, and in chapter 7 we take a deep dive into the subject. Broadly this has led to the following results.

### 8.2.1 Arbitrary definitions

As mentioned in section 8.1, we delved into the use of ensemble learning within a federated setting in chapter 4. Our original hypothesis was that ensemble learning formed a natural fit for the split data scenario federated learning presents. However, not only did it turn out that the current use of ensemble learning in a federated setting was limited, but the established literature cannot even agree on what exactly counts as ”federated learning”. Ensemble learning was sometimes viewed as a completely different, competing solution[52, 78]. This trend is also observable in broader literature, with certain parts of the community deeming even SMPC and secret sharing as separate solutions.

This tendency to present different technical solutions as competing solutions, as opposed to complementary solutions, results in the premature dismissal of potential solutions. To avoid limiting our potential toolbox we need a cultural shift and stop defining what is and is not ”true” federated learning based on arbitrary definitions.

### 8.2.2 The limitations of technical solutions

The solutions proposed in the various technical chapters of this thesis show that it is feasible to create technical solutions to perform a given analysis in a privacy preserving manner. However, in each of these chapters similar technical limitations need to be acknowledged. These limitations force us to acknowledge that there are scenarios in which

these solutions may not be appropriate. The solutions presented in this thesis are tailored towards scenarios with a relatively low number of parties, with access to certain expertise, hardware, and software. Additionally, the proposed solutions require a certain level of trust to have been established between the various parties.

This is not an inherent problem, the proposed solution works within the limitations of research projects such as the CARRIER project, the driving force behind this thesis. Additionally, by honestly acknowledging these limitations further projects can easily determine if the solutions presented here are appropriate for the problem they wish to solve themselves.

However, much of federated learning literature aims to provide universal solutions that are appropriate regardless of context. As a result, contextual limitations are rarely acknowledged. Furthermore, when they are acknowledged they are often deemed a major weakness. Especially the open admittance that certain problems cannot be solved with technical means, but require legal, or other solutions, is considered deeply unfavorable. Refusing to acknowledge these limitations, as well as refusing to discuss potential alternatives, or even refusing to question the need for a federated project, creates a false sense of security. It leads to projects being presented as privacy preserving despite major concerns.

### 8.2.3 The ethics & culture of privacy preserving research

We fully dive into the ethics, general viewpoints, and broader culture of the federated learning and privacy preserving community in chapter 7. As we discuss in this chapter, the community is shaped by the interests of big public and commercial institutions[140, 3, 121, 130]. The priorities and interests of these institutions does not always align with those of the data-subjects whose privacy they claim to protect.

These institutions shape how privacy is viewed. This can be a consequence of lobbying, internal policies, or their scientific output[73, 159,

110, 114]. For example, Google was the first to introduce the term federated learning in 2016, as briefly touched upon in section 8.2.1, certain parts of the community still hold on to the original definition given by Google as the one true definition of federated learning, viewing other solutions as direct competition. This is only a small example of the how such institutions have the power to shape the field, both on accident and on purpose, and guide the discussion through this.

The wish for universal solutions mentioned in section 8.2.2 is another example that originates from these institutions. Approaching privacy as a technical problem allows these institutions to create a veneer of objectivity. By focusing purely on the technical problems institutions can whitewash questionable projects; hiding behind the claim that a project is privacy preserving while ignoring more fundamental questions about the ethical implications of their projects.

Even when there is no malicious intent, the field may still be harmed as a consequence of the influence of these institutions. For example, the focus on technical solutions has led to the community at large viewing privacy and secrecy as equivalent concepts. Secrecy being broken does not mean the data-subjects are harmed, however this focus on secrecy does guarantee that leaks are treated as a major scandal which results in reputational harm to the data processor. Consequently this has led to institutions prioritizing the wrong thing as they try to minimize the damage of a breach of secrecy without considering if it actually leads to a breach of privacy. Additionally, it pushes the community towards universal solutions, which can capture the concept of privacy in a mathematical function. Ignoring the fact that privacy as a concept cannot be captured by equations, as it is dependent on context and culture. Ultimately this results in sub-optimal outcomes for the data-subjects whose privacy we claim to protect.

Lastly, the community rarely acknowledges that privacy is relevant throughout the entire lifecycle of a project. Privacy preserving techniques are often created to provide protection during a specific phase of a project, for example during the training of a model.

However, what happens to the privacy of individuals during the remaining phases of the project lifecycle is rarely considered. There is little value in using privacy preserving techniques to train a model, if the goal of a project is to invade the privacy of future subjects using that model. This need to look at the entire lifecycle, and not just focus on one small step, is rarely acknowledged within broader literature.

## 8.3 Future directions

Much progress has been made to create various technical solutions to improve privacy guarantees. These techniques show that it is feasible for parties to jointly perform certain analyses on their private data without having to reveal the data to the other parties when a certain level of trust is established. This has resulted in many tools that can provide extra layers of security for projects where privacy is a concern. However, the influence of large institutions, as well as the natural inclination of technical researchers to focus on technical solutions, means that the current solutions do not always align with what data-subjects truly need. As a result there are two important aspects to consider for future research.

### 8.3.1 Ethical considerations & a shift in culture

The influence of large institutions, both public and commercial, has resulted in the community developing a tunnel vision on technical solutions. The community is pushed to create techniques that work regardless of context, to prioritize secrecy above all else, and to eliminate the need for any supplementary solutions such as legal agreements. We prioritize the interests of these institutions over the interests of the data-subjects whose privacy we claim to protect. Privacy preserving techniques are used to whitewash ethically questionable projects, while at the same time important projects for the public good are undermined by a fear of scandals and restrictive rulings.

The privacy preserving community needs to reevaluate its priorities. It needs to acknowledge that not all problems can be solved with a technical solution. The influence of these institutions needs to be curbed, and a greater focus on the interests of the data-subjects is needed. The definition of privacy needs to shift away from secrecy, and focus more on the realistic harm done to data-subjects. This includes a greater focus on the goal of any given project; even the best privacy preserving techniques will not help if the goal of the project is itself an invasion of privacy.

Lastly, the naive pretense that a universal solution can be created, which works in any context, should be dropped. Both for the benefit of the community, as for the benefit of the data-subjects we wish to protect.

This will require a higher level of cooperation between legal, technical, and ethical experts. Additionally, it will require the development of better legal frameworks to help push back against the influence of the big institutions that currently get to define the topic.

### 8.3.2 Improving technical solutions

While a cultural shift, and redefinition of what it means to protect privacy, is needed, it should be acknowledged that the privacy preserving solutions that have been developed remain useful tools, provided their limitations are acknowledged. Further development of these solutions remains valuable and is needed.

Current solutions are heavily focused on horizontally partitioned scenarios. Consequently, vertical solutions are underdeveloped. However, vertical scenarios represent some incredibly interesting use-cases. Combining data from different parties who historically do not work together could lead to insights in various fields. It is especially relevant in fields where it is well established that data from different sources is relevant, but where this data is not utilized as it is collected by different parties. The link between socio-economic data and healthcare data,

collected by different parties, in the CARRIER project that is driving this thesis is but one example of such a use-case. Creating better federated learning solutions to support such projects would be beneficial.

Many algorithms simply do not have a vertical implementation. Such implementations are needed to unlock the information hidden across vertically split data. If a vertical implementation exists, it often suffers from performance issues. Vertical scenarios often require more complex solutions to maintain privacy(or rather secrecy), which comes with an associated overhead. Improving the time and space complexity of existing solutions is necessary. However, it should be noted that vertical scenarios often involve fewer parties, when compared to horizontal parties. Additionally, vertical scenarios currently rarely involve edge-devices and generally utilize more powerful hardware. Lastly, vertical scenarios are often research projects between different institutions, who can afford to wait for a model to finish training. This means that what constitutes an acceptable performance is wildly different compared to horizontal scenarios. As such, while optimization remains important, it is unlikely to be the top priority in a vertical partitioned scenario, or at the very least, the focus of optimization may be on different aspects compared to horizontal federated learning solutions.

Lastly, the existing implementations need to be made production ready. While many federated learning libraries and tools are currently being developed, many projects do not deliver products that are ready to be deployed in the real world. The product may be a simple proof of concept, it may be written to work in a specific environment and not translate to other environments, it may rely on assumptions which are unrealistic, or it may lack basic functionality, such as user management or logging, which is uninteresting for a research project, but very important in a practical setting. Research projects into federated learning should have a greater focus on ensuring their output is production ready so it does not devolve into a purely technical exercise that never sees use in the real world.

### 8.3.3 Preserving privacy in ways that matter

The work done to preserve privacy is promising. While large institutions hold a disproportionate amount of influence, and technical solutions may not be able to solve everything there have been clear improvements over the years. If we continue to develop technical solutions to tackle the various challenges, while honestly acknowledging their limitations, and focusing on what truly matters while pushing back against the institutions trying to whitewash their projects, we will be able to protect the privacy of the data subjects.

## 8.4 Concluding remarks

Much work has been done in the field of federated learning. This has resulted in many technical solutions for pressing problems. Within this thesis we have presented a number of novel solutions to further expand the toolbox available when working with vertically partitioned data. We hope that these solutions will see use in many future projects and will be further improved by those who follow us.

Additionally, we hope that our critique and advice regarding the ethics and culture of privacy research is taken to heart. Working with siloed data is becoming ever more important; making sure this is done in a proper manner, with care for the data subjects, is of the utmost importance.

We are hopeful for the future and look forward to see what will be achieved using the solutions created with federated learning.

# Bibliography


[1] *'Privacy Nightmare on Wheels': Every Car Brand Reviewed By Mozilla — Including Ford, Volkswagen and Toyota — Flunks Privacy Test*. en. Sept. 2023.

[2] Nazmiye Ceren Abay et al. "Privacy Preserving Synthetic Data Release Using Deep Learning". In: *Machine Learning and Knowledge Discovery in Databases*. Ed. by Michele Berlingerio et al. Vol. 11051. Series Title: Lecture Notes in Computer Science. Cham: Springer International Publishing, 2019, pp. 510–526. DOI: 10.1007/978-3-030-10925-7_31.

[3] Mohamed Abdalla and Moustafa Abdalla. "The Grey Hoodie Project: Big Tobacco, Big Tech, and the Threat on Academic Integrity". In: *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*. AIES '21. New York, NY, USA: Association for Computing Machinery, July 2021, pp. 287–297. DOI: 10.1145/3461702.3462563.

[4] Pauline Affeldt and Reinhold Kesler. "Big Tech Acquisitions — Towards Empirical Evidence†". In: *Journal of European Competition Law & Practice* 12.6 (June 2021), pp. 471–478. DOI: 10.1093/jeclap/lpab025.

[5] S. Ahn, A. Özgür, and M. Pilanci. "Global Multiclass Classification and Dataset Construction via Heterogeneous Local Experts". In: *IEEE Journal on Selected Areas in Information Theory* 1.3 (Nov. 2020), pp. 870–883. DOI: 10.1109/JSAIT.2020.3041804.

[6] H. Akaike. "A new look at the statistical model identification". In: *IEEE Transactions on Automatic Control* 19.6 (Dec. 1974), pp. 716–723. DOI: 10.1109/TAC.1974.1100705.

[7] Jonathan Albright. *How Trump's campaign used the new data-industrial complex to win the election*. "en-US". Nov. 2016.

[8] Shahid Ali, Sreenivas Sremath Tirumala, and Abdolhossein Sarrafzadeh. "Ensemble learning methods for decision making: Status and future prospects". In: *2015 International Conference on Machine Learning and Cybernetics (ICMLC)*. 2015 International Conference on Machine Learning and Cybernetics (ICMLC). Vol. 1. July 2015, pp. 211–216. DOI: 10.1109/ICMLC.2015.7340924.

[9] Valentin Amrhein, Sander Greenland, and Blake McShane. "Scientists rise up against statistical significance". en. In: *Nature* 567.7748 (Mar. 2019), pp. 305–307. DOI: 10.1038/d41586-019-00857-9.

[10] C. Anagnostopoulos. "Edge-centric inferential modeling & analytics". In: *Journal of Network and Computer Applications* 164 (Aug. 2020). DOI: 10.1016/j.jnca.2020.102696.

[11] Bea Andrea Antonio et al. "Invasion or Personalization: An Overview on User Attitudes towards the Privacy Issues in Targeted Advertising in NCR and Its Effect in Consumer Purchase Behavior". en. In: *Journal of Business and Management Studies* 4.2 (Mar. 2022). Number: 2, pp. 38–47. DOI: 10.32996/jbms.2022.4.2.4.

[12] A. Arora et al. "Risk and the future of AI: Algorithmic bias, data colonialism, and marginalization". In: *Information and Organization* 33.3 (Sept. 2023), p. 100478. DOI: 10.1016/j.infoandorg.2023.100478.

[13] Mikhail J. Atallah and Wenliang Du. "Secure Multi-party Computational Geometry". In: *Algorithms and Data Structures*. Ed. by Gerhard Goos et al. Vol. 2125. Series Title: Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 165–179. DOI: 10.1007/3-540-44634-6_16.

[14] Marieke Bak et al. "Federated learning is not a cure-all for data ethics". en. In: *Nature Machine Intelligence* 6.4 (Apr. 2024). Publisher: Nature Publishing Group, pp. 370–372. DOI: 10.1038/s42256-024-00813-x.

[15] Syreen Banabilah et al. “Federated learning review: Fundamentals, enabling technologies, and future applications”. en. In: *Information Processing & Management* 59.6 (Nov. 2022), p. 103061. DOI: 10.1016/j.ipm.2022.103061.

[16] Adriano Baratè, Goffredo Haus, and Luca Andrea Ludovico. “Learning, Teaching, and Making Music Together in the COVID-19 Era Through IEEE 1599”. In: *2020 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*. 2020 International Conference on Software, Telecommunications and Computer Networks (SoftCOM). ISSN: 1847-358X. Sept. 2020, pp. 1–5. DOI: 10.23919/SoftCOM50211.2020.9238238.

[17] Amos Beimel. “Secret-Sharing Schemes: A Survey”. In: May 2011, pp. 11–46. DOI: 10.1007/978-3-642-20901-7_2.

[18] Ingo A. Beinlich et al. “The ALARM Monitoring System: A Case Study with two Probabilistic Inference Techniques for Belief Networks”. en. In: *AIME 89* (1989). Publisher: Springer, Berlin, Heidelberg, pp. 247–256. DOI: 10.1007/978-3-642-93437-7_28.

[19] Lex M. Bouter et al. “Ranking major and minor research misbehaviors: results from a survey among participants of four World Conferences on Research Integrity”. en. In: *Research Integrity and Peer Review* 1.1 (Dec. 2016), p. 17. DOI: 10.1186/s41073-016-0024-5.

[20] Stephen Boyd et al. “Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers”. English. In: *Foundations and Trends® in Machine Learning* 3.1 (July 2011). Publisher: Now Publishers, Inc., pp. 1–122. DOI: 10.1561/2200000016.

[21] Gillian Brockell. “Perspective — Dear tech companies, I don’t want to see pregnancy ads after my child was stillborn”. en-US. In: *Washington Post* (Dec. 2018).

[22] 1 Senat Bundesverfassungsgericht. *Bundesverfassungsgericht - Decisions - Decision on the constitutionality of the 1983 Census Act*. en. Gerichtsentscheidung. Archive Location: de Publisher: Bundesverfassungsgericht. Dec. 1983.

[23] *California Code Civil Code - CIV DIVISION 3 - OBLIGATIONS PART 4 - OBLIGATIONS ARISING FROM PARTICULAR TRANSACTIONS TITLE 1.81.5 - California Consumer Privacy Act of 2018 Section 1798.192. Universal Citation: CA Civ Code § 1798.192*. 2023.

[24] Xiaoyu Cao, Jinyuan Jia, and Neil Zhenqiang Gong. "Provably Secure Federated Learning against Malicious Clients". en. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 35.8 (May 2021). Number: 8, pp. 6885–6893. DOI: 10.1609/aaai.v35i8.16849.

[25] Sayan Chatterjee and Manjesh Kumar Hanawal. "Federated learning for intrusion detection in IoT security: a hybrid ensemble approach". In: *International Journal of Internet of Things and Cyber-Assurance* 2.1 (Jan. 2022). Publisher: Inderscience Publishers, pp. 62–86. DOI: 10.1504/IJITCA.2022.124372.

[26] Hong-You Chen and Wei-Lun Chao. "FedBE: Making Bayesian Model Ensemble Applicable to Federated Learning". In: *arXiv:2009.01974 [cs, stat]* (Jan. 30, 2021).

[27] Ruimin Chen et al. "Ontology-driven learning of bayesian network for causal inference and quality assurance in additive manufacturing". In: *IEEE Robotics and Automation Letters* 6.3 (2021), pp. 6032–6038.

[28] Xiaolin Chen et al. "Fed-EINI: An Efficient and Interpretable Inference Framework for Decision Tree Ensembles in Vertical Federated Learning". In: *2021 IEEE International Conference on Big Data (Big Data)*. Dec. 2021, pp. 1242–1248. DOI: 10.1109/BigData52589.2021.9671749.

[29] Gregory F. Cooper and Edward Herskovits. “A Bayesian method for the induction of probabilistic networks from data”. en. In: *Machine Learning* 9.4 (Oct. 1992), pp. 309–347. DOI: 10.1007/BF00994110.

[30] D. R. Cox. “Regression Models and Life-Tables”. In: *Journal of the Royal Statistical Society: Series B (Methodological)* 34.2 (Jan. 1972), pp. 187–202. DOI: 10.1111/j.2517-6161.1972.tb00899.x.

[31] J. Cromack. *Why Amazon’s GDPR fine really matters: Consent in marketing*. en-gb. Aug. 2021.

[32] F. van Daalen et al. “An Ensemble Approach to Time Dependent Classification”. In: *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. Dec. 2018, pp. 1007–1011. DOI: 10.1109/ICMLA.2018.00164.

[33] F. Van Daalen et al. “An Ensemble Approach to Time Dependent Classification”. In: *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA). Dec. 2018, pp. 1007–1011. DOI: 10.1109/ICMLA.2018.00164.

[34] Florian van Daalen et al. “A Bayesian Network Approach to Lung Cancer Screening: Assessing the Impact of Data Quantity, Quality, and the Combination of Data from Danish Electronic Health Records”. en. In: *Cancers* 16.23 (Jan. 2024), p. 3989. DOI: 10.3390/cancers16233989.

[35] Florian van Daalen et al. *A Response to: A Note on "Privacy Preserving n-Party Scalar Product Protocol"*. 2024.

[36] Florian van Daalen et al. *Federated Ensembles: a literature review*. en. Dec. 2022. DOI: 10.21203/rs.3.rs-2350540/v1.

[37] Florian van Daalen et al. “Privacy Preserving n-Party Scalar Product Protocol”. In: *IEEE Transactions on Parallel and Distributed Systems* 34.4 (Apr. 2023), pp. 1060–1066. DOI: 10.1109/TPDS.2023.3238768.

[38] Florian van Daalen et al. “VertiBayes: learning Bayesian network parameters from vertically partitioned data with missing values”. en. In: *Complex & Intelligent Systems* (Apr. 2024). DOI: 10.1007/s40747-024-01424-0.

[39] Wenrui Dai et al. “VERTICOX: Vertically Distributed Cox Proportional Hazards Model Using the Alternating Direction Method of Multipliers”. en. In: *IEEE Transactions on Knowledge and Data Engineering* (2020), pp. 1–1. DOI: 10.1109/TKDE.2020.2989301.

[40] Sourya Joyee De and Abdessamad Imine. “Consent for targeted advertising: the case of Facebook”. en. In: *AI & SOCIETY* 35.4 (Dec. 2020), pp. 1055–1064. DOI: 10.1007/s00146-020-00981-5.

[41] Maria De Marsico et al. “Mobile Iris Challenge Evaluation (MICHE)-I, biometric iris dataset and protocols”. en. In: *Pattern Recognition Letters*. Mobile Iris CHallenge Evaluation part I (MICHE I) 57 (May 2015), pp. 17–23. DOI: 10.1016/j.patrec.2015.02.009.

[42] Jodi Dean. “Review of Regulating Intimacy: A New Legal Paradigm”. In: *Journal of Law and Society* 30.3 (2003), pp. 453–458.

[43] Timo M. Deist et al. “Distributed learning on 20 000+ lung cancer patients – The Personal Health Train”. en. In: *Radiotherapy and Oncology* 144 (Mar. 2020), pp. 189–200. DOI: 10.1016/j.radonc.2019.11.019.

[44] A. P. Dempster, N. M. Laird, and D. B. Rubin. “Maximum Likelihood from Incomplete Data via the EM Algorithm”. In: *Journal of the Royal Statistical Society. Series B (Methodological)* 39.1 (1977), pp. 1–38.

[45] Emily L. Denton et al. “Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks”. In: *Advances in Neural Information Processing Systems*. Vol. 28. Curran Associates, Inc., 2015.

[46] Quang Do, Ben Martini, and Kim-Kwang Raymond Choo. “The role of the adversary model in applied security research”. en. In: *Computers & Security* 81 (Mar. 2019), pp. 156–181. DOI: 10.1016/j.cose.2018.12.002.

[47] Leyla Dogruel. “Too much information!? Examining the impact of different levels of transparency on consumers’ evaluations of targeted advertising”. In: *Communication Research Reports* 36.5 (Oct. 2019), pp. 383–392. DOI: 10.1080/08824096.2019.1684253.

[48] Wenliang Du and M.J. Atallah. “Privacy-preserving cooperative statistical analysis”. en. In: *Seventeenth Annual Computer Security Applications Conference*. New Orleans, LA, USA: IEEE Comput. Soc, 2001, pp. 102–110. DOI: 10.1109/ACSAC.2001.991526.

[49] Wenliang Du, Yunghsiang S. Han, and Shigang Chen. “Privacy-Preserving Multivariate Statistical Analysis: Linear Regression and Classification”. In: *Proceedings of the 2004 SIAM International Conference on Data Mining (SDM)*. Proceedings. Society for Industrial and Applied Mathematics, Apr. 22, 2004, pp. 222–233. DOI: 10.1137/1.9781611972740.21.

[50] Wenliang Du and Zhijun Zhan. “Building decision tree classifier on private data”. In: *Proceedings of the IEEE international conference on Privacy, security and data mining - Volume 14*. CRPIT ’14. AUS: Australian Computer Society, Inc., Dec. 2002, pp. 1–8.

[51] Moming Duan et al. “Self-Balancing Federated Learning With Global Imbalanced Data in Mobile Systems”. In: *IEEE Transactions on Parallel and Distributed Systems* 32.1 (Jan. 2021). Conference Name: IEEE Transactions on Parallel and Distributed Systems, pp. 59–71. DOI: 10.1109/TPDS.2020.3009406.

[52] Aiden Durrant et al. “The role of cross-silo federated learning in facilitating data sharing in the agri-food sector”. In: *Computers and Electronics in Agriculture* 193 (Feb. 1, 2022), p. 106648. DOI: 10.1016/j.compag.2021.106648.

[53] *Dutch childcare benefit scandal an urgent wake-up call to ban racist algorithms*. en. Oct. 2021.

[54] Cynthia Dwork and Moni Naor. “On the Difficulties of Disclosure Prevention in Statistical Databases or The Case for Differential Privacy”. en. In: *Journal of Privacy and Confidentiality* 2.1 (Sept. 2010). DOI: 10.29012/jpc.v2i1.585.

[55] Cynthia Dwork and Aaron Roth. “The Algorithmic Foundations of Differential Privacy”. en. In: *Foundations and Trends® in Theoretical Computer Science* 9.3-4 (Aug. 2015), pp. 211–407. DOI: 10.1561/0400000042.

[56] Cynthia Dwork et al. “Calibrating Noise to Sensitivity in Private Data Analysis”. en. In: *Theory of Cryptography*. Ed. by Shai Halevi and Tal Rabin. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2006, pp. 265–284. DOI: 10.1007/11681878_14.

[57] EDPB. *1.2 billion euro fine for Facebook as a result of EDPB binding decision — European Data Protection Board*. May 2023.

[58] Peter Eigenschink et al. “Deep Generative Models for Synthetic Data: A Survey”. In: *IEEE Access* 11 (2023), pp. 47304–47320. DOI: 10.1109/ACCESS.2023.3275134.

[59] *EU General Data Protection Regulation (GDPR): Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation), OJ 2016 L 119/1*. en-US. 2016.

[60] Euronews. *Meta hit with €265 million fine by Irish regulators for breaking Europe’s data protection law — Euronews*. Nov. 2022.

[61] *Facebook and Instagram to Offer Subscription for No Ads in Europe*. en-US. Oct. 2023.

[62] Dominik Fay, Jens Sjölund, and Tobias J. Oechtering. "Decentralized Differentially Private Segmentation with PATE". In: *arXiv* (Apr. 1, 2020), arXiv:2004.06567.

[63] Usama M. Fayyad and Keki B. Irani. "Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning". In: *Ijcai.* 1993, pp. 1022–1029.

[64] Stefan Feuerriegel, Mateusz Dolata, and Gerhard Schwabe. "Fair AI". en. In: *Business & Information Systems Engineering* 62.4 (Aug. 2020), pp. 379–384. DOI: 10.1007/s12599-020-00650-3.

[65] Ana Sofia Figueiredo. "Data sharing: convert challenges into opportunities". In: *Frontiers in public health* 5 (2017), p. 327.

[66] Ferdinando Fioretto and Pascal Van Hentenryck. "Privacy-Preserving Federated Data Sharing". In: *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*. AAMAS '19. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2019, pp. 638–646.

[67] Cormac Fitzgerald. *Facebook will soon use your photos, posts and other info to train its AI. You can opt out (but it's complicated)*. en. May 2024.

[68] Raquel Florez-Lopez and Juan Manuel Ramon-Jeronimo. "Enhancing accuracy and interpretability of ensemble strategies in credit risk assessment. A correlated-adjusted decision forest proposal". In: *Expert Systems with Applications* 42.13 (Aug. 1, 2015), pp. 5737–5753. DOI: 10.1016/j.eswa.2015.02.042.

[69] Eibe Frank, Ian H. Witten, and Mark A. Hall. *Data Mining, Fourth Edition:Practical Machine Learning Tools and Techniques — Guide books*. en. 2016.

[70] *GDPR Enforcement Tracker - list of GDPR fines*.

[71] Tom Gerken. “Facebook and Instagram launch ad-free subscription tier in EU”. en-GB. In: *BBC* (Oct. 2023).

[72] Bart Goethals et al. “On Private Scalar Product Computation for Privacy-Preserving Data Mining”. en. In: *Information Security and Cryptology – ICISC 2004*. Ed. by David Hutchison et al. Vol. 3506. Series Title: Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 104–120. DOI: 10.1007/11496618_9.

[73] Sharon Goldman. *The ‘Meta AI mafia’ brain drain continues with 3 more major departures*. en. Apr. 2024.

[74] Heitor Murilo Gomes et al. “A survey on ensemble learning for data stream classification”. In: *ACM Computing Surveys (CSUR)* 50.2 (2017), pp. 1–36.

[75] Christophe Gonzales, Axel Journe, and Ahmed Mabrouk. “Constraint-Based Bayesian Network Structure Learning using Uncertain Experts’ Knowledge”. In: *Thirty-fourth International Florida Artificial Intelligence Research Society Conference*. Vol. 34. 1. 2021.

[76] Karol Gregor et al. “DRAW: A Recurrent Neural Network For Image Generation”. en. In: *Proceedings of the 32nd International Conference on Machine Learning*. ISSN: 1938-7228. PMLR, June 2015, pp. 1462–1471.

[77] Paulius Grinkevičius. *Meta uses your data to train AI, and it doesn’t want you to opt out*. en-US. May 2024.

[78] Hongtao Guan, Xingkong Ma, and Siqi Shen. “DOS-GAN: A Distributed Over-Sampling Method Based on Generative Adversarial Networks for Distributed Class-Imbalance Learning”. en. In: *Algorithms and Architectures for Parallel Processing*. Ed. by Meikang Qiu. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2020, pp. 609–622. DOI: 10.1007/978-3-030-60248-2_42.

[79] Lexie Hagen. *Privacy Preserving Machine Learning: Maintaining confidentiality and preserving trust*. en-US. Nov. 2021.

[80] Eszter Hargittai and Alice Marwick. ""What Can I Really Do?" Explaining the Privacy Paradox with Online Apathy". en. In: *International Journal of Communication* 10.0 (July 2016), p. 21.

[81] Frank E Harrell Jr, Kerry L Lee, and Daniel B Mark. "Multivariable prognostic models: issues in developing models, evaluating assumptions and adequacy, and measuring and reducing errors". In: *Statistics in medicine* 15.4 (1996), pp. 361–387.

[82] Megan L. Head et al. "The Extent and Consequences of P-Hacking in Science". en. In: *PLOS Biology* 13.3 (2015), e1002106. DOI: 10.1371/journal.pbio.1002106.

[83] Rachael Louise Healy. "Zuckerberg, get out of my uterus! An examination of fertility apps, data-sharing and remaking the female body as a digitalized reproductive subject". en. In: *Journal of Gender Studies* 30.4 (May 2021), pp. 406–416. DOI: 10.1080/09589236.2020.1845628.

[84] Margrethe Bang Henriksen et al. "Lung Cancer Detection Using Bayesian Networks: A Retrospective Development and Validation Study on a Danish Population of High-Risk Individuals". In: *Cancer Medicine* 14.3 (Jan. 2025), e70458. DOI: 10.1002/cam4.70458.

[85] Joanne Hinds, Emma J. Williams, and Adam N. Joinson. ""It wouldn't happen to me": Privacy concerns and perspectives following the Cambridge Analytica scandal". In: *International Journal of Human-Computer Studies* 143 (Nov. 2020), p. 102498. DOI: 10.1016/j.ijhcs.2020.102498.

[86] *Hoe Meta gegevens gebruikt voor generatieve AI-modellen*. nl. 2024.

[87] Lingzhou Hong, Alfredo Garcia, and Ceyhun Eksin. "Distributed networked learning with correlated data". en. In: *Automatica* 137 (Mar. 2022), p. 110134. DOI: 10.1016/j.automatica.2021.110134.

[88] Agnieszka Jablonowska and Adrianna Michatowicz. "Planet49: Pre-Ticked Checkboxes Are Not Sufficient to Convey User's Consent to the Storage of Cookies Case Notes". eng. In: *European Data Protection Law Review (EDPL)* 6.1 (2020), pp. 137–142.

[89] Xiao Jin et al. "CAFE: Catastrophic Data Leakage in Vertical Federated Learning". In: *Advances in Neural Information Processing Systems*. Vol. 34. Curran Associates, Inc., 2021, pp. 994–1006.

[90] Taeho Jo. "Machine learning foundations". In: *Machine Learning Foundations. Springer Nature Switzerland AG. https://doi.org/10.1007/978-3-030-65900-4* (2021).

[91] Leslie K. John, George Loewenstein, and Drazen Prelec. "Measuring the Prevalence of Questionable Research Practices With Incentives for Truth Telling". en. In: *Psychological Science* 23.5 (May 2012), pp. 524–532. DOI: 10.1177/0956797611430953.

[92] A. Jović, K. Brkić, and N. Bogunović. "A review of feature selection methods with applications". In: *2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. 2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO). May 2015, pp. 1200–1205. DOI: 10.1109/MIPRO.2015.7160458.

[93] Peter Kairouz et al. "Advances and Open Problems in Federated Learning". English. In: *Foundations and Trends® in Machine Learning* 14.1–2 (June 2021). Publisher: Now Publishers, Inc., pp. 1–210. DOI: 10.1561/2200000083.

[94] Bart Kamphorst et al. "Accurate training of the Cox proportional hazards model on vertically-partitioned data while preserving privacy". eng. In: *BMC medical informatics and decision making* 22.1 (Feb. 2022), p. 49. DOI: 10.1186/s12911-022-01771-3.

[95] Bridget G. Kelly and Maniza Habib. “Missed period? The significance of period-tracking applications in a post-Roe America”. In: *Sexual and Reproductive Health Matters* 31.4 (Dec. 2023), p. 2238940. DOI: 10.1080/26410397.2023.2238940.

[96] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. 2009.

[97] Stacy Kowalczyk and Kalpana Shankar. “Data sharing in the sciences”. In: *Annual review of information science and technology* 45.1 (2011), pp. 247–294.

[98] Ludmila I. Kuncheva and Christopher J. Whitaker. “Measures of Diversity in Classifier Ensembles and Their Relationship with the Ensemble Accuracy”. In: *Machine Learning* 51.2 (May 1, 2003), pp. 181–207. DOI: 10.1023/A:1022859003006.

[99] Isabel Kusche. “Possible harms of artificial intelligence and the EU AI act: fundamental rights and risk”. In: *Journal of Risk Research* 0.0 (May 2024), pp. 1–14. DOI: 10.1080/13669877.2024.2350720.

[100] S. L. Lauritzen and D. J. Spiegelhalter. “Local Computations with Probabilities on Graphical Structures and Their Application to Expert Systems”. In: *Journal of the Royal Statistical Society: Series B (Methodological)* 50.2 (1988), pp. 157–194. DOI: 10.1111/j.2517-6161.1988.tb01721.x.

[101] Steffen L. Lauritzen. “The EM algorithm for graphical association models with missing data”. en. In: *Computational Statistics & Data Analysis* 19.2 (Feb. 1995), pp. 191–201. DOI: 10.1016/0167-9473(93)E0056-A.

[102] Li Li et al. “A review of applications in federated learning”. en. In: *Computers & Industrial Engineering* 149 (Nov. 2020), p. 106854. DOI: 10.1016/j.cie.2020.106854.

[103] X. Li et al. “Multi-site fMRI analysis using privacy-preserving federated learning and domain adaptation: ABIDE results”. In: *Medical Image Analysis* 65 (Oct. 2020). DOI: 10.1016/j.media.2020.101765.

[104] Xiaoxiao Li et al. *FedBN: Federated Learning on Non-IID Features via Local Batch Normalization*. May 11, 2021. DOI: 10.48550/arXiv.2102.07623.

[105] Wei-Chao Lin, Yu-Hsin Lu, and Chih-Fong Tsai. “Feature selection in single and ensemble learning-based bankruptcy prediction models”. In: *Expert Systems* 36.1 (Aug. 2019), e12335. DOI: 10.1111/exsy.12335.

[106] Bo Liu et al. *When Machine Learning Meets Privacy: A Survey and Outlook*. Nov. 2020. DOI: 10.48550/arXiv.2011.11819.

[107] Y. Liu et al. “Privacy-Preserving Traffic Flow Prediction: A Federated Learning Approach”. In: *IEEE Internet of Things Journal* 7.8 (Apr. 2020), pp. 7751–7763. DOI: 10.1109/JIOT.2020.2991401.

[108] Yang Liu et al. “Federated Forest”. In: *IEEE Transactions on Big Data* (May 2020), pp. 1–1. DOI: 10.1109/TBDATA.2020.2992755.

[109] Yue Liu et al. “Data quantity governance for machine learning in materials science”. In: *National Science Review* 10.7 (2023), nwad125.

[110] Taylor Lorenz and Drew Harwell. “Facebook paid GOP firm to malign TikTok”. en-US. In: *Washington Post* (Mar. 2022).

[111] Mi Luo et al. “No Fear of Heterogeneity: Classifier Calibration for Federated Learning with Non-IID Data”. In: *Advances in Neural Information Processing Systems*. Vol. 34. Curran Associates, Inc., Dec. 2021, pp. 5972–5984.

[112] Christoph Lutz, Christian Pieter Hoffmann, and Giulia Ranzini. "Data capitalism and the user: An exploration of privacy cynicism in Germany". en. In: *New Media & Society* 22.7 (July 2020), pp. 1168–1187. DOI: 10.1177/1461444820912544.

[113] Lingjuan Lyu et al. "Towards Fair and Privacy-Preserving Federated Deep Models". In: *IEEE Transactions on Parallel and Distributed Systems* 31.11 (Nov. 2020), pp. 2524–2541. DOI: 10.1109/TPDS.2020.2996273.

[114] Sapna Maheshwari and Amanda Holpuch. *Why the U.S. Is Forcing TikTok to Be Sold or Banned*. Apr. 2024.

[115] E. J. Martin and X. W. Zhu. "Collaborative Profile-QSAR: A Natural Platform for Building Collaborative Models among Competing Companies". In: *Journal of Chemical Information and Modeling* 61.4 (Apr. 2021), pp. 1603–1616. DOI: 10.1021/acs.jcim.0c01342.

[116] Cecily Mauran. *Meta is using your posts to train AI. It's not easy to opt out.* en. May 2024.

[117] Scott McLachlan et al. "Bayesian networks in healthcare: Distribution by medical condition". In: *Artificial intelligence in medicine* 107 (2020), p. 101912.

[118] Brendan McMahan et al. "Communication-Efficient Learning of Deep Networks from Decentralized Data". en. In: *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*. ISSN: 2640-3498. PMLR, Apr. 2017, pp. 1273–1282.

[119] Maryam Mehrnezhad and Teresa Almeida. "My sex-related data is more sensitive than my financial data and I want the same level of security and privacy: User Risk Perceptions and Protective Actions in Female-oriented Technologies". In: *Proceedings of the 2023 European Symposium on Usable Security*. EuroUSEC '23. New York, NY, USA: Association for Computing Machinery, 2023, pp. 1–14. DOI: 10.1145/3617072.3617100.

[120] Maryam Mehrnezhad et al. "Vision: Too Little too Late? Do the Risks of FemTech already Outweigh the Benefits?" In: *Proceedings of the 2022 European Symposium on Usable Security*. EuroUSEC '22. New York, NY, USA: Association for Computing Machinery, Sept. 2022, pp. 145–150. DOI: 10.1145/3549015.3554204.

[121] Joseph Menn and Naomi Nix. "Big Tech funds the very people who are supposed to hold it accountable". en-US. In: *Washington Post* (Dec. 2023).

[122] Guanhong Miao et al. "Learning from vertically distributed data across multiple sites: An efficient privacy-preserving algorithm for Cox proportional hazards model with variable selection". eng. In: *Journal of Biomedical Informatics* 149 (Jan. 2024), p. 104581. DOI: 10.1016/j.jbi.2023.104581.

[123] Malcolm S. Mitchell, Mimi C. Yu, and Theresa L. Whiteside. "The tyranny of statistics in medicine: a critique of unthinking adherence to an arbitrary p value". en. In: *Cancer Immunology, Immunotherapy* 59.8 (Aug. 2010), pp. 1137–1140. DOI: 10.1007/s00262-010-0859-4.

[124] Arturo Moncada-Torres et al. "VANTAGE6: an open source priVAcy preserviNg federaTed leArninG infrastructurE for Secure Insight eXchange". In: *AMIA Annual Symposium Proceedings* 2020 (Jan. 2021), pp. 870–877.

[125] Arvind Narayanan and Vitaly Shmatikov. *How To Break Anonymity of the Netflix Prize Dataset*. arXiv:cs/0610105. Nov. 2007. DOI: 10.48550/arXiv.cs/0610105.

[126] Ignavier Ng and Kun Zhang. "Towards Federated Bayesian Network Structure Learning with Continuous Optimization". en. In: *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*. ISSN: 2640-3498. 2022, pp. 8095–8111.

[127] M. N. H. Nguyen et al. “Distributed and Democratized Learning: Philosophy and Research Challenges”. In: *IEEE Computational Intelligence Magazine* 16.1 (Feb. 2021), pp. 49–62. DOI: 10.1109/MCI.2020.3039068.

[128] Helen Nissenbaum. “Privacy as Contextual Integrity Symposium: Technology, Values, and the Justice System”. eng. In: *Washington Law Review* 79.1 (2004), pp. 119–158.

[129] Helen Nissenbaum. *Privacy in Context: Technology, Policy, and the Integrity of Social Life*. en. Stanford University Press, Nov. 2009. DOI: 10.1515/9780804772891.

[130] Rodrigo Ochigame. *How Big Tech Manipulates Academia to Avoid Regulation*. en-US. Dec. 2019.

[131] D. Opitz and R. Maclin. “Popular Ensemble Methods: An Empirical Study”. en. In: *Journal of Artificial Intelligence Research* 11 (Aug. 1999), pp. 169–198. DOI: 10.1613/jair.614.

[132] L. Ouyang, Y. Yuan, and F. Y. Wang. “Learning Markets: An AI Collaboration Framework Based on Blockchain and Smart Contracts”. In: *IEEE Internet of Things Journal* (Oct. 2020), pp. 1–1. DOI: 10.1109/JIOT.2020.3032706.

[133] Michael Palmer. *Data is the New Oil*. Nov. 2005.

[134] Charis Papaevangelou. “Funding Intermediaries: Google and Facebook’s Strategy to Capture Journalism”. In: *Digital Journalism* 12.2 (Feb. 2024), pp. 234–255. DOI: 10.1080/21670811.2022.2155206.

[135] Payal V. Parmar et al. “Survey of Various Homomorphic Encryption algorithms and Schemes”. en. In: *International Journal of Computer Applications* 91.8 (Apr. 2014), pp. 26–32. DOI: 10.5120/15902-5081.

[136] Katy Parsons. *Why taking Facebook quizzes is a really bad idea — CBC News*. en. Jan. 2020.

[137] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. en. Elsevier, June 2014.

[138] David Peloquin et al. "Disruptive and avoidable: GDPR challenges to secondary research uses of data". en. In: *European Journal of Human Genetics* 28.6 (June 2020), pp. 697–705. DOI: 10.1038/s41431-020-0596-x.

[139] Viraj Prabhu et al. "Open Set Medical Diagnosis". In: *arXiv preprint arXiv:1910.02830* (Oct. 2019).

[140] Tech Transparency Project. *Zuckerberg and Meta Reach Deep into Academia*. en-us. Dec. 2023.

[141] *Proposal for a Regulation of the European Parliament and of the Council laying down harmonised rules on Artificial Intelligence ( Artificial Intelligence Act) and amending certain Union legislative Acts. COM/2021/206 final*. en. Apr. 2021.

[142] Jenny Radesky et al. "Digital Advertising to Children". In: *Pediatrics* 146.1 (July 2020), e20201681. DOI: 10.1542/peds.2020-1681.

[143] Pablo Ramirez-Hereza et al. "Score-based Bayesian network structure learning algorithms for modeling radioisotope levels in nuclear power plant reactors". In: *Chemometrics and Intelligent Laboratory Systems* 237 (2023), p. 104811.

[144] *Register berispingen — Autoriteit Persoonsgegevens*. nl.

[145] Ikhlaq ur Rehman. "Facebook-Cambridge Analytica data harvesting: What you need to know". In: *Library Philosophy and Practice* (2019), pp. 1–11.

[146] Matthias Reisser et al. "Federated Mixture of Experts". In: (July 2021), arXiv:2107.06724.

[147] Alexey Reznichenko and Paul Francis. "Private-by-Design Advertising Meets the Real World". In: *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. CCS '14. New York, NY, USA: Association for Computing Machinery, Nov. 2014, pp. 116–128. DOI: 10.1145/2660267.2660305.

[148] Beate Roessler and Judith DeCew. “Privacy”. In: *The Stanford Encyclopedia of Philosophy*. Ed. by Edward N. Zalta and Uri Nodelman. Winter 2023. Metaphysics Research Lab, Stanford University, 2023.

[149] Lior Rokach. “Taxonomy for characterizing ensemble methods in classification tasks: A review and annotated bibliography”. In: *Computational Statistics & Data Analysis* 53.12 (Oct. 2009), pp. 4046–4072. DOI: 10.1016/j.csda.2009.07.017.

[150] Ty Roush. *Meta Launching Paid Subscriptions To Use Facebook And Instagram Ad-Free In Europe*. en. Oct. 2023.

[151] Omer Sagi and Lior Rokach. “Ensemble learning: A survey”. In: *WIREs Data Mining and Knowledge Discovery* 8.4 (Feb. 2018), e1249. DOI: 10.1002/widm.1249.

[152] B. Scheenstra et al. “A big data-driven eHealth approach to prevent, detect, and reduce atherosclerotic cardiovascular disease burden”. In: *European Journal of Preventive Cardiology* 29.Supplement_1 (2022), zwac056–305.

[153] Bart Scheenstra et al. “Digital Health Solutions to Reduce the Burden of Atherosclerotic Cardiovascular Disease Proposed by the CARRIER Consortium”. en. In: *JMIR Cardio* 6.2 (Oct. 2022), e37437. DOI: 10.2196/37437.

[154] Jeffrey C. Schlimmer. “Concept acquisition through representational adjustment”. en. In: (July 1987).

[155] Thomas Schneider and Amos Treiber. “A Comment on Privacy-Preserving Scalar Product Protocols as Proposed in “SPOC””. In: *IEEE Transactions on Parallel and Distributed Systems* 31.3 (Mar. 2020), pp. 543–546. DOI: 10.1109/TPDS.2019.2939313.

[156] Zhandos Sembay. *Seer Breast Cancer Data*. July 2021. DOI: 10.5281/zenodo.5120960.

[157] Giovanni Seni and John F. Elder. "Ensemble Complexity". In: *Ensemble Methods in Data Mining: Improving Accuracy Through Combining Predictions*. Ed. by Giovanni Seni and John F. Elder. Synthesis Lectures on Data Mining and Knowledge Discovery. Cham: Springer International Publishing, 2010, pp. 108–123. DOI: 10.1007/978-3-031-01899-2_6.

[158] Saqib Shah. *How to opt out of Instagram and Facebook training AI on your photos*. en. June 2024.

[159] Donald Shaw. *Meta Shatters Lobbying Record as House Passes TikTok Ban*. en. Apr. 2024.

[160] J. Shi et al. "MODES: model-based optimization on distributed embedded systems". In: *Machine Learning* 110.6 (June 2021), pp. 1527–1547. DOI: 10.1007/s10994-021-06014-6.

[161] Laura Shipp and Jorge Blasco. "How private is your period?: A systematic analysis of menstrual app privacy policies". en. In: *Proceedings on Privacy Enhancing Technologies* 2020.4 (Oct. 2020), pp. 491–510. DOI: 10.2478/popets-2020-0083.

[162] Nir Shlezinger et al. "Collaborative Inference via Ensembles on the Edge". In: *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. ISSN: 2379-190X. June 2021, pp. 8478–8482. DOI: 10.1109/ICASSP39728.2021.9414740.

[163] Erez Shmueli and Tamir Tassa. "Mediated Secure Multi-Party Protocols for Collaborative Filtering". In: *ACM Transactions on Intelligent Systems and Technology* 11.2 (Apr. 30, 2020), pp. 1–25. DOI: 10.1145/3375402.

[164] Reza Shokri et al. "Membership Inference Attacks Against Machine Learning Models". In: *2017 IEEE Symposium on Security and Privacy (SP)*. ISSN: 2375-1207. May 2017, pp. 3–18. DOI: 10.1109/SP.2017.41.

[165] Lok Siu and Claire Chun. “Yellow Peril and Techno-orientalism in the Time of Covid-19: Racialized Contagion, Scientific Espionage, and Techno-Economic Warfare”. In: *Journal of Asian American Studies* 23.3 (2020), pp. 421–440.

[166] Manel Slokom, Peter-Paul de Wolf, and Martha Larson. “When Machine Learning Models Leak: An Exploration of Synthetic Training Data”. en. In: *Privacy in Statistical Databases*. Ed. by Josep Domingo-Ferrer and Maryline Laurent. Cham: Springer International Publishing, 2022, pp. 283–296. DOI: 10.1007/978-3-031-13945-1_20.

[167] Jack W. Smith et al. “Using the ADAP Learning Algorithm to Forecast the Onset of Diabetes Mellitus”. In: *Proceedings of the Annual Symposium on Computer Application in Medical Care* (Nov. 1988), pp. 261–265.

[168] Daniel J. Solove. “Conceptualizing Privacy”. In: *California Law Review* 90.4 (2002), pp. 1087–1155. DOI: 10.2307/3481326.

[169] Peter Spirtes et al. “An algorithm for fast recovery of sparse causal graphs”. In: *Social Science Computer Review* (1991), pp. 62–72.

[170] Sonali Srivastava, Terhi-Anna Wilska, and Jussi Nyrhinen. “Awareness of digital commercial profiling among adolescents in Finland and their perspectives on online targeted advertisements”. In: *Journal of Children and Media* 17.4 (Oct. 2023), pp. 559–578. DOI: 10.1080/17482798.2023.2257813.

[171] Latanya Sweeney. “k-ANONYMITY: A MODEL FOR PROTECTING PRIVACY”. In: *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 10.05 (Oct. 2002), pp. 557–570. DOI: 10.1142/S0218488502001648.

[172] Xiaoqing Tan, Chung-Chou H. Chang, and Lu Tang. “A Tree-based Federated Learning Approach for Personalized Treatment Effect Estimation from Heterogeneous Data Sources”. In: *arXiv* (Mar. 1, 2021), arXiv:2103.06261.

[173] Julia Tar. *EU data protection body says Meta's 'pay or OK' model is not OK*. en-GB. Apr. 2024.

[174] Fadi Thabtah. "Autism Spectrum Disorder Screening: Machine Learning Adaptation and DSM-5 Fulfillment". In: *Proceedings of the 1st International Conference on Medical and Health Informatics 2017*. ICMHI '17. New York, NY, USA: Association for Computing Machinery, 2017, pp. 1–6. DOI: 10.1145/3107514.3107515.

[175] Zhiyi Tian et al. "A Comprehensive Survey on Poisoning Attacks and Countermeasures in Machine Learning". In: *ACM Computing Surveys* 55.8 (Dec. 2022), 166:1–166:35. DOI: 10.1145/3551636.

[176] Nguyen Truong et al. "Privacy preservation in federated learning: An insightful survey from the GDPR perspective". en. In: *Computers & Security* 110 (Nov. 2021), p. 102402. DOI: 10.1016/j.cose.2021.102402.

[177] Jaideep Vaidya and Chris Clifton. "Privacy preserving association rule mining in vertically partitioned data". In: *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. KDD '02. New York, NY, USA: Association for Computing Machinery, July 2002, pp. 639–644. DOI: 10.1145/775047.775142.

[178] Florian Van Daalen et al. "Federated Bayesian Network Ensembles". In: *2023 Eighth International Conference on Fog and Mobile Edge Computing (FMEC)*. IEEE. 2023, pp. 22–33.

[179] Florian Van Daalen et al. "Federated Bayesian Network Ensembles". In: *2023 Eighth International Conference on Fog and Mobile Edge Computing (FMEC)*. Tartu, Estonia: IEEE, Sept. 2023, pp. 22–33. DOI: 10.1109/FMEC59375.2023.10306230.

[180] Dinusha Vatsalan et al. "Privacy-Preserving Record Linkage for Big Data: Current Approaches and Research Challenges". en. In: *Handbook of Big Data Technologies*. Ed. by Albert Y. Zomaya

and Sherif Sakr. Cham: Springer International Publishing, 2017, pp. 851–895. DOI: 10.1007/978-3-319-49340-4_25.

[181] Pieter Verdegem. "Dismantling AI capitalism: the commons as an alternative to the power concentration of Big Tech". In: *AI & society* 39.2 (2024), pp. 727–737.

[182] D. Verma et al. "Policy based Ensembles for applying ML on Big Data". In: *2019 IEEE International Conference on Big Data (Big Data)*. Dec. 2019, pp. 4038–4044. DOI: 10.1109/BigData47090.2019.9006193.

[183] Dinesh C. Verma et al. "Policy-based ensembles for multi domain operations". In: *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications II*. Vol. 11413. SPIE, Apr. 2020, pp. 48–56. DOI: 10.1117/12.2558727.

[184] Shikha Verma. "Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy". en. In: *Vikalpa: The Journal for Decision Makers* 44.2 (June 2019), pp. 97–98. DOI: 10.1177/0256090919853933.

[185] Jayakorn Vongkulbhisal, Phongtharin Vinayavekhin, and Marco Visentini-Scarzanella. "Unifying Heterogeneous Classifiers With Distillation". English. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, June 2019, pp. 3170–3179. DOI: 10.1109/CVPR.2019.00329.

[186] Hongrui Wang et al. "A Bayesian network approach for condition monitoring of high-speed railway catenaries". In: *IEEE Transactions on Intelligent Transportation Systems* 21.10 (2019), pp. 4037–4051.

[187] Huanyu Wang and Elena Dubrova. "Federated Learning in Side-Channel Analysis". en. In: *Information Security and Cryptology – ICISC 2020*. Ed. by Deukjo Hong. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2021, pp. 257–272. DOI: 10.1007/978-3-030-68890-5_14.

[188] Junxiao Wang et al. “Protect Privacy from Gradient Leakage Attack in Federated Learning”. In: *IEEE INFOCOM 2022 - IEEE Conference on Computer Communications*. London, United Kingdom: IEEE, May 2022, pp. 580–589. DOI: 10.1109/INFOCOM48880.2022.9796841.

[189] Robert Wargaski. “Privacy Paradox or Privacy Apathy? Exploring the Relationship between Social Media Usage and Public Opinion on Government Usage of Data Collection Programs”. en. In: *Aresty Rutgers Undergraduate Research Journal* 1.4 (Dec. 2022). DOI: 10.14713/arestyrurj.v1i4.213.

[190] Samuel Warren and Louis Brandeis. “The right to privacy”. In: *Harvard Law Review* 4.5 (1890), pp. 193–220.

[191] Wenqi Wei et al. *A Framework for Evaluating Gradient Leakage Attacks in Federated Learning*. Apr. 23, 2020. DOI: 10.48550/arXiv.2004.10397.

[192] Sarah Myers West. “Data Capitalism: Redefining the Logics of Surveillance and Privacy”. en. In: *Business & Society* 58.1 (Jan. 2019), pp. 20–41. DOI: 10.1177/0007650317718185.

[193] Alan F. Westin. “Privacy and freedom”. In: *Washington and Lee Law Review* 25.1 (1968), p. 166.

[194] Simon Wiedemann, Klaus-Robert Müller, and Wojciech Samek. “Compact and Computationally Efficient Representation of Deep Neural Networks”. In: *IEEE Transactions on Neural Networks and Learning Systems* 31.3 (Mar. 2020), pp. 772–785. DOI: 10.1109/TNNLS.2019.2910073.

[195] Rebecca Wright and Zhiqiang Yang. “Privacy-preserving Bayesian network structure computation on distributed heterogeneous data”. In: *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. KDD ’04. New York, NY, USA: Association for Computing Machinery, Aug. 2004, pp. 713–718. DOI: 10.1145/1014052.1014145.

[196] Xi-Zhu Wu, Song Liu, and Zhi-Hua Zhou. "Heterogeneous Model Reuse via Optimizing Multiparty Multiclass Margin". en. In: *Proceedings of the 36th International Conference on Machine Learning*. ISSN: 2640-3498. PMLR, May 2019, pp. 6840–6849.

[197] *Xenophobic machines: Discrimination through unregulated use of algorithms in the Dutch childcare benefits scandal*. en. Oct. 2021.

[198] Runhua Xu, Nathalie Baracaldo, and James Joshi. *Privacy-Preserving Machine Learning: Methods, Challenges and Directions*. arXiv:2108.04417 [cs]. Sept. 2021. DOI: 10.48550/arXiv.2108.04417.

[199] Liu Yang et al. "Federated Recommendation Systems". en. In: *Federated Learning: Privacy and Incentive*. Ed. by Qiang Yang, Lixin Fan, and Han Yu. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2020, pp. 225–239. DOI: 10.1007/978-3-030-63076-8_16.

[200] Qiang Yang et al. "Federated Machine Learning: Concept and Applications". en. In: *arXiv:1902.04885 [cs]* (Feb. 2019). arXiv: 1902.04885.

[201] Zhiqiang Yang and R.N. Wright. "Privacy-Preserving Computation of Bayesian Networks on Vertically Partitioned Data". en. In: *IEEE Transactions on Knowledge and Data Engineering* 18.9 (Sept. 2006), pp. 1253–1264. DOI: 10.1109/TKDE.2006.147.

[202] Andrew C. Yao. "Protocols for secure computations". In: *23rd Annual Symposium on Foundations of Computer Science (sfcs 1982)*. ISSN: 0272-5428. Nov. 1982, pp. 160–164. DOI: 10.1109/SFCS.1982.38.

[203] Yufeng Zhan et al. "A Survey of Incentive Mechanism Design for Federated Learning". In: *IEEE Transactions on Emerging Topics in Computing* 10.2 (Apr. 2022), pp. 1035–1044. DOI: 10.1109/TETC.2021.3063517.

[204] Yufeng Zhan et al. “Incentive Mechanism Design for Federated Learning: Challenges and Opportunities”. In: *IEEE Network* 35.4 (July 2021), pp. 310–317. DOI: 10.1109/MNET.011.2000627.

[205] Rongying Zhao et al. “Development strategy and collaboration preference in S&T of enterprises based on funded papers: a case study of Google”. en. In: *Scientometrics* 121.1 (Oct. 2019), pp. 323–347. DOI: 10.1007/s11192-019-03182-0.

[206] Zhiqiang Yang and R.N. Wright. “Improved Privacy-Preserving Bayesian Network Parameter Learning on Vertically Partitioned Data”. In: *21st International Conference on Data Engineering Workshops (ICDEW'05)*. Tokyo, Japan, 2005, pp. 1196–1196. DOI: 10.1109/ICDE.2005.230.

# Scientific and Societal Impact

First and foremost the results of this thesis have been several scientific articles. In addition to these articles the results of this thesis have been shared in the following ways. Each technical solution that was developed has been published in an open source repository. This has resulted in the following output:

- Privacy preserving $n$-party protocol library, written in java. This library can be used as a building block in the development of any future algorithms.
  `https://github.com/MaastrichtU-CDS/n-scalar-product-protocol`
- VertiBayes implementation in java as well as a python wrapper for deployment using the Vantage6 framework. The Java implementation can be run independently, or used as a library for further development. The Vantage6 wrapper allows it to be used within Vantage6, this allows a user to take advantage of the framework with respect to aspects such as user management removing the need to implement this directly into VertiBayes.
  - Java implementation: `https://github.com/MaastrichtU-CDS/vertibayes`
  - Vantage6 wrapper: `https://github.com/MaastrichtU-CDS/vertibayes_vantage6`
- Federated Bayesian Network Ensemble implementation in java as well as a python wrapper for deployment using the Vantage6 framework. Again the java implementation can be run independently or used as a library for further development.
  `https://github.com/MaastrichtU-CDS/bayesianEnsemble`

- Verticox+ implemented in a mix of python and java, as well as a python wrapper for deployment using the Vantage6 framework: `https://github.com/CARRIER-project/verticox`

Additionally, the work done for this thesis has resulted in numerous improvements to the Vantage6 framework. These improvements include both new functionalities as well as general improvements to security and performance.

In addition to distributing the software several workshops and guest lectures have been held on the topics of Vantage6 and federated learning. These had the twin-purpose of educating people on the topics, as well as to evangelize the Vantage6 framework.

We developed a demo application based on the $n$-party protocol for HealthRI 2022 in Utrecht, the Netherlands. This demo application was designed to give individuals without a technical background, such as policy makers, a basic understanding of federated learning in a vertical setting. The focus of the demonstration was on letting participants play themselves with a simple example to naturally help develop understanding of the topic.

At MIE 2023 in Gothenburg, Sweden, a workshop was given on the topic of Federated learning. For this workshop we presented the challenges we encountered within the CARRIER project up to this point, as well as our technical solutions. This was followed by a panel discussion on federated learning with a mix of experts where we provided the technical expertise. In addition to this workshop the aforementioned HealthRI demo was available for the entire conference for attendants.

A guest lecture was given as part of the MegaData: Federated Machine Learning summer school 2023 for Tartu University, Estonia. Subsequent editions of the summer school have started to include Vantage6 more broadly into their curriculum, this highlights the appeal of Vantage6.

Another guest lecture was given for the Privacy Engineering Track for the IT Law Master program at Istanbul Bilgi University, Turkey. This lecture aimed to familiarize lawyers with the basic concepts of federated learning.

During the MAASTRO science day 2024, in Maastricht, the Netherlands, a presentation on the CARRIER project was given to an audience of medical researchers and policy makers. This presentation focused on illustrating the practical problems a federated learning project encounters. The take away message of this presentation is that the biggest obstacles are not technical problems, but are bureaucratic and political in nature.

Additionally, during the werkorientatiedag 2024 for the Bachelor course Gezondheidswetenschappen at Maastricht University a presentation and workshop on federated learning and datasharing was given to students. The goal of the werkorientatiedag is to help students decide on their future specialization; our presentation represented one possible direction they can take their education & future career in.

The ultimate goal of the CARRIER project is to provide early detection, followed by personalized intervention, of patient with cardiovascular problems. Within this project it was our responsibility to make it technically feasible to train a model in a privacy preserving manner on vertically partitioned data. While the necessary tools have been developed, no model has been trained yet. This has been due to various delays which are bureaucratic, legal, and organizational in nature.

Based on these experiences several lessons have been learned for future projects, which will be formalized in a project report that can be disseminated within Statistics Netherlands, as well as more broadly with other governmental agencies. This report, exploring the potential of federated learning for data sharing within Statistics Netherlands, was one of the goals of the CARRIER project. As data sharing is one of the core tasks of Statistics Netherlands, the lessons learned will be

very valuable and will contribute towards unlocking the information stored within their datasets for a broader public.

In addition to this report we wished to train a model in a federated way. This model was to be used for the early detection of cardiovascular disease. While this will no longer be part of this thesis we still intend to execute the federated analysis. The results of this analysis will lead to further publications. In addition this will result in a model which will be implemented within the broader CARRIER project. A successful CARRIER project will ultimately contribute to a healthier population, as cardiovascular disease will be detected and treated at an earlier stage.

# Summary/Samenvatting

## 1 English

Federated learning is a field of machine learning in which models are built in a decentralized manner without the need to directly share data. This approach allows researchers to work with data that would normally be difficult to access due to legal, ethical, and practical concerns.

Historically federated learning has largely been focused on horizontal scenarios; scenarios where the different parties collect the same data about different individuals. Vertical scenarios; scenarios where different parties collect different types of data belonging to the same individuals, has not received as much attention. In this thesis we presented several algorithms that can be applied in vertical scenarios. It should be noted that each of these algorithms also generalizes to the horizontal or a hybrid setting.

In chapter 2 we introduce the privacy preserving scalar product protocol. This protocol can be used to answer basic queries about the dataset, such as how many individuals fulfil a specific requirement, in a privacy preserving manner even when data is split over multiple parties. We then show how this protocol can be generalized to scenarios with more than two parties. This protocol serves as a secure building block for the more complex analysis we introduce in later chapters.

In chapter 3 we use the protocol introduced in chapter 2, alongside synthetic data, to train a Bayesian network in a vertically partitioned scenario. This gives us access to a popular and powerful model in a vertically split federated setting with an arbitrary number of parties. The model's popularity stems from its high level of explainability, and is easy to understand without needing a technical background. As

such it is of great value in a medical setting, where explainability, and ease of understanding, are very important.

In chapter 4 we explore the topic of ensemble learning within the context of federated learning using a literature review. Ensembles are a natural fit for the split environment federated scenarios present. They naturally provide a level of privacy preservation, avoid the complex technical solutions required by alternative federated learning techniques, and can potentially make use of the natural differences between the datasets of different parties. During our review we discover that ensemble learning is currently underutilized within the context of federated learning.

In chapter 5 we devise our own ensemble learning algorithm, incorporating the VertiBayes algorithm introduced in chapter 3. Our experiments show that this is indeed a viable alternative with significant advantages in the right circumstances. The reduction in computational complexity and communication overhead results in a considerably shorter training time when compared to VertiBayes. It retains a similar performance, and may even outperform VertiBayes. Additionally, it is easier to use when classifying new individuals in a federated setting. Lastly, it provides some additional privacy guarantees. However, the ensemble model is more difficult to interpret.

In chapter 6 we move on to the Verticox+ algorithm, an extension of the original Verticox algorithm which can train a Cox proportional hazard model in a vertically partitioned setting. Our extension improves the privacy guarantees of the original algorithm by removing the original's assumption that the event-attribute is be known at every party. This assumption is unrealistic to hold in a realistic scenario, removing this assumption removes a very large limitation. Our improved version retains the same performance, and while it theoretically introduces an increased time complexity, the practical consequences are limited due to the bottleneck being elsewhere.

Finally in chapter 7 we discuss how privacy is currently viewed within the scientific community and how these views are shaped by various

institutes. We note how big organizations, both private for profit organizations, as well as public nonprofits, heavily influence how privacy is viewed. By funding the research in the field, as well as via lobbying efforts, they shape the discussion on privacy. While this does not have to be malicious in nature, it brings with it several issues. For example, these institutes often prefer to focus on researching new privacy preserving methods to enable their projects, but do not wish to discuss the ethical and philosophical implications of the data sharing projects themselves. This contrasts with the wishes of the individual data-subjects who often have very different priorities. Consequently much of the current research is arguably focused on the wrong things. We conclude that a greater focus on the needs, and wishes of the data subjects is needed. This requires a cultural shift among researchers, for example to move the focus away from technical solutions. Additional it may also require changes in the legal frameworks.

## 2 Nederlands

Federated learning is een veld binnen kunstmatige intelligentie waarin modellen op een gedecentraliseerde manier worden gebouwd zonder data direct hoeft te worden gedeeld. Deze aanpak maakt het mogelijk voor onderzoekers om te werken met data die normaal gesproken moeilijk te gebruiken is vanwege legale, ethische, en praktische beperkingen.

Historisch gezien is federated learning gefocust op horizontale scenario's; scenario's waarin verschillende partijen dezelfde data verzamelen over verschillende individuen. Verticale scenario's; scenario's waarin verschillende partijen verschillende soorten data verzamelen met betrekking tot dezelfde individuen, hebben tot nu toe minder aandacht gekregen. In deze thesis presenteren we verscheidene algoritmes die kunnen worden toegepast in verticale scenario's. Deze algoritmes zijn dusdanig flexibel dat zij ook kunnen worden gebruikt in een horizontale of hybride setting.

In hoofdstuk 2 introduceren we het privacy beschermende inwendig product protocol (privacy preserving scalar product protocol). Dit protocol kan worden gebruikt om simpele vragen, zoals hoeveel individuen in de dataset vervullen bepaalde criteria, over een dataset te beantwoorden op een manier die de privacy waarborgt, ook wanneer de data is gesplitst over meerdere partijen. We laten dan zien hoe dit protocol kan worden gegeneraliseerd naar een scenario met een willekeurig aantal partijen. Dit protocol dient als als basis voor de meer complexe analyses die wij later introduceren.

In hoofdstuk 3 gebruiken we het protocol uit hoofdstuk 2, samen met synthetische data, om een Bayesian netwerk te trainen in een verticaal gepartioneerd scenario met een willekeurig aantal partijen. Dit geeft ons toegang tot dit populaire en krachtige model type in een verticaale gepartitioneerde setting. Dit model is populair dankzij de hoge mate van uitlegbaarheid omdat het gemakkelijk het te begrijpen is zonder een technische achtergrond nodig te hebben. Als zodanig is het een zeer nuttig model in een medische setting, waar uitlegbaarheid, en begrijpbaarheid, van groot belang zijn.

In hoofdstuk 4 duiken we in het gebruik van ensemble learning binnen de context van federated learning aan de hand van een literatuur review. Ensembles zijn van nature zeer geschikt om met de gesplitste data in gefedereerde scenario's te werken. Zij beschermen inherent de privacy tot op zekere hoogte, ze vermijden de technische en ingewikkelde oplossingen die andere federated technieken vereisen, en ze kunnen mogelijk gebruik maken van de verschillen in de datasets van de verschillende partijen. Uit onze review blijkt dat de sterke punten van ensemble learning momenteel niet volledig worden benut in de context van federated learning.

In hoofdstuk 5 ontwikkelen we ons eigen ensemble learning algoritme. We gebruiken hiervoor het VertiBayes algoritme uit hoofdstuk 3. Onze experimenten laten zien dat dit inderdaad een geschikt alternatief is met significante voordelen in de juiste omstandigheden. Dankzij de lagere complexiteit en communicatie

overhead is het ensemble model significant sneller te trainen. Het behoudt een vergelijkbare nauwkeurigheid, en kan zelfs betere resultaten geven dan VertiBayes. Daarnaast is het makkelijker te gebruiken wanneer men ook de classificatie van nieuwe individuen op een federatieve manier wil uitvoeren. Daarbovenop geeft het ook licht betere privacy garanties. In vergelijking tot VertiBayes resulteert het wel in een minder interpreteerbaar model.

In hoofdstuk 6 introduceren we het Verticox+ algoritme, een uitbreiding van het originele Verticox algoritme waarmee een Cox proportional hazard model in een verticale setting kan worden getraind. Onze uitbreiding verbetert de privacy garanties van het originele algoritme omdat onze implementatie de aanname van het origineel - dat het event-attribuut bij elke partij bekent is - niet meer nodig heeft. Deze aanname is onrealistisch in de praktijk. Het verwijderen van deze aanname maakt Verticox+ veel breder toepasbaar dan zijn voorganger. Onze verbeterde versie behoudt dezelfde nauwkeurigheid, en hoewel in theorie de tijdscomplexiteit van het algoritme slechter is, zijn de consequenties in de praktijk beperkt omdat het knelpunt elders zit.

Uiteindelijk bespreken we in hoofdstuk 7 hoe privacy momenteel wordt gezien in de wetenschappelijke gemeenschap, en hoe verschillende instituties de discussie over privacy beïnvloeden. Grote organisaties, zowel privéprivate, op winst gerichte organisaties, als publieke non-profits, beïnvloeden sterk hoe privacy wordt gezien. Door onderzoek over het onderwerp te financieren, en daarnaast mogelijk door het uitvoeren van lobbyactiviteiten, vormen zij hoe privacy wordt besproken en onderzocht. Hoewel dit geen kwaadwillende insteek hoeft te hebben, brengt dit toch een aantal problemen met zich mee. Het is bijvoorbeeld zo dat deze instituten zich vaak focussen op het ontwikkelen van nieuwe technieken om privacy te beschermen zodat zij hun projecten daadwerkelijk kunnen uitvoeren. Maar zij zullen minder geneigd zijn om de ethische en filosofische implicaties van deze projecten te bespreken. De individuen wiens data daadwerkelijk wordt gebruikt hebben

daarentegen vaak erg andere prioriteiten. Als gevolg hiervan is veel van het huidige onderzoek gefocust op de verkeerde dingen. We concluderen dat een grotere focus op de wensen en prioriteiten van individuen nodig is. Dit vereist een verandering binnen de onderzoekscultuur, bijvoorbeeld om de focus te verleggen van puur technische oplossingen naar de meer ethische en filosofische vraagstukken. Daarnaast zal het mogelijk veranderingen in de legale raamwerken vereisen.

# Acknowledgments

This thesis was made possible thanks to the feedback, support, and advice of many people, not least the members of the assessment committee who devoted considerable time to read and review my work.

I would like to thank my promoter, Andre Dekker, for his invaluable support and guidance over the course of my PhD. He encouraged me to pursue my research interests, both within the context of the CARRIER project and on unrelated topics. In addition, he encouraged me to develop outside of science and develop myself as a teacher, which I have always been interested in. I feel very lucky to have been part of his research group.

My copromoter Inigo Bermejo was a constant source of encouragement and has helped me tremendously with writing my research. I am very grateful for all the help I have received with turning my incoherent thoughts into clear and concise papers.

Many thanks also to Lianne Ippel who supported me on behalf of Statistics Netherlands. Your feedback on my papers was invaluable and you have been key in moving things forward within the project.

A big thank you to all the amazing people I met in the Clinical Data Science group at MAASTRO. It has been a wonderful group to work in. Especially Leonard, Ralph, & Johan. You encouraged me to start various side projects and develop connections that I would not have managed on my own. I would also like to thank Marine for her help with cleaning up my messiest paper. Additionally, I would like to thank Djura for her help with the various technical challenges we faced. Anniek, Simone, and Hannah deserve special mention for their show of outstanding bravery by volunteering to be taught by me, knowing fully what they were signing up for. Hopefully, my lessons have been

useful. Lastly, I would like to thank everyone else at CDS who helped make this such a positive experience.

My work would not have been possible without the support of my friends & family. Listening to me complain about the various ups and downs of PhD life has been vital. I would especially like to thank my wife, without her I would not have had the courage to return to academia. Thank you for always pushing me forward.

Florian van Daalen
Maastricht
2025-03-28

# Published work

## 1 Published original research


1. Florian van Daalen et al. *Federated Ensembles: a literature review*. en. Dec. 2022. DOI: 10.21203/rs.3.rs-2350540/v1

2. Florian van Daalen et al. "Privacy Preserving n-Party Scalar Product Protocol". In: *IEEE Transactions on Parallel and Distributed Systems* 34.4 (Apr. 2023), pp. 1060–1066. DOI: 10.1109/TPDS.2023.3238768

3. Florian van Daalen et al. "VertiBayes: learning Bayesian network parameters from vertically partitioned data with missing values". en. In: *Complex & Intelligent Systems* (Apr. 2024). DOI: 10.1007/s40747-024-01424-0

4. Florian Van Daalen et al. "Federated Bayesian Network Ensembles". In: *2023 Eighth International Conference on Fog and Mobile Edge Computing (FMEC)*. IEEE. 2023, pp. 22–33

5. Bart Scheenstra et al. "Digital Health Solutions to Reduce the Burden of Atherosclerotic Cardiovascular Disease Proposed by the CARRIER Consortium". en. In: *JMIR Cardio* 6.2 (Oct. 2022), e37437. DOI: 10.2196/37437

6. B. Scheenstra et al. "A big data-driven eHealth approach to prevent, detect, and reduce atherosclerotic cardiovascular disease burden". In: *European Journal of Preventive Cardiology* 29.Supplement_1 (2022), zwac056–305

7. Florian van Daalen et al. *A Response to: A Note on "Privacy Preserving n-Party Scalar Product Protocol"*. 2024

8. Florian van Daalen et al. "A Bayesian Network Approach to Lung Cancer Screening: Assessing the Impact of Data Quantity, Quality, and the Combination of Data from Danish Electronic Health Records". en. In: *Cancers* 16.23 (Jan. 2024), p. 3989. DOI: 10.3390/cancers16233989

9. Margrethe Bang Henriksen et al. "Lung Cancer Detection Using Bayesian Networks: A Retrospective Development and Validation Study on a Danish Population of High-Risk Individuals". In: *Cancer Medicine* 14.3 (Jan. 2025), e70458. DOI: 10.1002/cam4.70458

## 2 Submitted and currently under review

1. Verticox+: Improving Privacy Guarantees, Florian van Daalen, Graduate Student Member IEEE, Djura Smits, Lianne Ippel, Andre Dekker, and Inigo Bermejo

2. A critique of current approaches to privacy in machine learning, Florian van Daalen, Marine Jacquemin, Johan van Soest, Nina Stahl, David Townend, Andre Dekker, Inigo Bermejo

3. Multinomial Classification Certainty: a new uncertainty metric for multinomial outcome prediction, Florian van Daalen, Ralph Brecheisen, Leonard Wee, Andre Dekker, Inigo Bermejo

4. Uncertainty Quantification in radiomics to classify patients based on WHO pathological risk grades, Hannah Mary Thomas, Florian van Daalen, Leonard Wee

## 3 Poster sessions and presentations

1. Vantage6 poster demonstration utilizing the $n$-party scalar product protocol, 7th Health-RI conference, Utrecht, The Netherlands, October 2022, ($Poster$)

2. Privacy Enhancing Technologies (PETs), IT Law Master program Istanbul Bilgi, University, Istanbul, Turkey, April 2023, ($Oral\ presentation$)

3. Privacy Preserving Analysis Using Vantage6, MegaData: Federated Machine learning Summerschool, Tartu, Estonia, August 2023, ($Oral\ presentation$)

4. Vantage6 poster demonstration utilizing the $n$-party scalar product protocol, 33th conference Medical Informatics Europe Conference (MIE), Gothenburg, Sweden, May 2023, ($Poster$)

5. Caring is Sharing – Exploiting Value in Data for Health and Innovation, 33th conference Medical Informatics Europe Conference (MIE), Gothenburg, Sweden, May 2023, ($Oral\ presentation$, $panel\ discussion$, $workshop$)

6. Privacy Preserving Analysis Using Vantage6, MegaData: Federated Machine learning Summerschool, Tartu, Estonia, August 2023, ($Oral\ presentation$)

7. Federated Bayesian Network Ensembles, The 1st International Symposium on Federated Learning Technologies and Applications (FLTA), Tartu, Estonia, September 2023, ($Oral\ presentation$)

8. federated learning on vertically distributed data in practice (CARRIER-project), MAASTRO science day, Maastricht, The Netherlands, May 2024, ($Oral\ presentation$)

## 4 Prizes

1. ”Best Paper award”, The 1st International Symposium on Federated Learning Technologies and Applications (FLTA), Tartu, Estonia, September 18-20, 2023. Awarded for the paper entitled ”Federated Bayesian Network Ensembles”.

# About the author

Florian van Daalen was born on 9 Juli in Waalwijk, The Netherlands. Upon finishing his secondary education at Willem van Oranje College in Waalwijk, he first studied knowledge engineering, followed by Artificial Intelligence at Maastricht university.

He finished his Master's degree in 2014. His thesis explored the application of Ensemble Learning to deal with time dependencies in a dataset. Following his graduation, he turned his thesis into a full-fledged publication while working as a business engineer and software engineer at Blueriq. In 2021 he returned to his alma mater as a PhD candidate to work on privacy preserving Federated Learning for the CARRIER project within the GROW school for Oncology and Developmental Biologoy at the Faculty of Health, Medicine and Life Sciences. As part of his research, he developed various algorithms to train machine learning models in a privacy preserving manner when data is split over various parties who are unwilling, or unable, to directly share their data.

His interest remains in privacy preserving Federated Learning, with a focus on the real world applications and consequences of data sharing.

# Appendices

## 1 Full 3-party naive calculation Privacy Preserving N-party scalar product protocol

The full calculation can be expanded as follows:

$$
\begin{gathered}
\varphi(\hat{\mathbf{A}} \cdot \hat{\mathbf{C}} \cdot \mathbf{B}) + r_b - v_2 - \varphi(\mathbf{R_a} \cdot \hat{\mathbf{B}} \cdot \hat{\mathbf{C}}) + r_a \\
- \varphi(\mathbf{R_c} \cdot \hat{\mathbf{A}} \cdot \hat{\mathbf{B}}) + r_c + v_2 \\
= \\
\varphi((\mathbf{A} + \mathbf{R_a}) \cdot (\mathbf{C} + \mathbf{R_c}) \cdot \mathbf{B}) - \varphi((\mathbf{B} + \mathbf{R_b}) \cdot (\mathbf{C} + \mathbf{R_c}) \cdot \mathbf{R_a}) \\
- \varphi((\mathbf{A} + \mathbf{R_a}) \cdot (\mathbf{B} + \mathbf{R_b}) \cdot \mathbf{R_c}) + r_a + r_b + r_c \\
= \\
\varphi(\mathbf{A} \cdot \mathbf{B} \cdot \mathbf{C} + \mathbf{A} \cdot \mathbf{B} \cdot \mathbf{R_c} + \mathbf{B} \cdot \mathbf{C} \cdot \mathbf{R_a} + \mathbf{B} \cdot \mathbf{R_a} \cdot \mathbf{R_c}) \\
- \varphi(\mathbf{B} \cdot \mathbf{C} \cdot \mathbf{R_a} + \mathbf{B} \cdot \mathbf{R_a} \cdot \mathbf{R_c} + \mathbf{C} \cdot \mathbf{R_a} \cdot \mathbf{R_b} + \mathbf{R_a} \cdot \mathbf{R_b} \cdot \mathbf{R_c}) \\
- \varphi(\mathbf{A} \cdot \mathbf{B} \cdot \mathbf{R_c} + \mathbf{A} \cdot \mathbf{R_b} \cdot \mathbf{R_c} + \mathbf{B} \cdot \mathbf{R_a} \cdot \mathbf{R_c} + \mathbf{R_a} \cdot \mathbf{R_b} \cdot \mathbf{R_c}) \\
+ r_a + r_b + r_c \\
= \\
\varphi(\mathbf{A} \cdot \mathbf{B} \cdot \mathbf{C}) + \varphi(\mathbf{A} \cdot \mathbf{B} \cdot \mathbf{R_c}) + \varphi(\mathbf{B} \cdot \mathbf{C} \cdot \mathbf{R_a}) + \varphi(\mathbf{B} \cdot \mathbf{R_a} \cdot \mathbf{R_c}) \\
- \varphi(\mathbf{B} \cdot \mathbf{C} \cdot \mathbf{R_a}) - \varphi(\mathbf{B} \cdot \mathbf{R_a} \cdot \mathbf{R_c}) - \varphi(\mathbf{C} \cdot \mathbf{R_a} \cdot \mathbf{R_b}) \\
- \varphi(\mathbf{R_a} \cdot \mathbf{R_b} \cdot \mathbf{R_c}) - \varphi(\mathbf{A} \cdot \mathbf{B} \cdot \mathbf{R_c}) - \varphi(\mathbf{A} \cdot \mathbf{R_b} \cdot \mathbf{R_c}) \\
- \varphi(\mathbf{B} \cdot \mathbf{R_a} \cdot \mathbf{R_c}) - \varphi(\mathbf{R_a} \cdot \mathbf{R_b} \cdot \mathbf{R_c}) + r_a + r_b + r_c \\
= \\
\varphi(\mathbf{A} \cdot \mathbf{B} \cdot \mathbf{C}) - \varphi(\mathbf{C} \cdot \mathbf{R_a} \cdot \mathbf{R_b}) - \varphi(\mathbf{A} \cdot \mathbf{R_b} \cdot \mathbf{R_c}) \\
- \varphi(\mathbf{B} \cdot \mathbf{R_a} \cdot \mathbf{R_c}) - \varphi(\mathbf{R_a} \cdot \mathbf{R_b} \cdot \mathbf{R_c}) + r_a + r_b + r_c
\end{gathered}
$$

## 2 Full 3-party example Privacy Preserving N-party scalar product protocol

Practical example of the $n$-party scalar protocol: 3 parties Alice, Bob, & Claire with the following data. Data $\mathbf{A}$: $\begin{bmatrix}100\\010\\001\end{bmatrix}$ Data $\mathbf{B}$: $\begin{bmatrix}000\\010\\001\end{bmatrix}$ Data $\mathbf{C}$: $\begin{bmatrix}100\\000\\001\end{bmatrix}$

This means we are dealing with an $n$-party protocol where $n = 3$. The target value would be: $\varphi(\mathbf{A} \cdot \mathbf{B} \cdot \mathbf{C}) = 1$

Using the $n$-scalar protocol the calculation will look as follows: First the trusted third party Merlin generates the following three random matrices:

$$\mathbf{R_a}: \begin{bmatrix}172 & 0 & 0\\ 0 & 243 & 0\\ 0 & 0 & 136\end{bmatrix} \quad \mathbf{R_b}: \begin{bmatrix}274 & 0 & 0\\ 0 & 356 & 0\\ 0 & 0 & 180\end{bmatrix} \quad \mathbf{R_c}: \begin{bmatrix}341 & 0 & 0\\ 0 & 357 & 0\\ 0 & 0 & 69\end{bmatrix}$$

Merlin then calculates: $\varphi(\mathbf{R_a} \cdot \mathbf{R_b} \cdot \mathbf{R_c}) = 48643124$ Merlin then splits $\varphi(\mathbf{R_a} \cdot \mathbf{R_b} \cdot \mathbf{R_c})$ into three secret shares: $r_a = 8015322, r_b = 10543269$, & $r_c = 30084533$.

Alice then calculates $\hat{\mathbf{A}} = \mathbf{A}+\mathbf{R_a} = \begin{bmatrix}173 & 0 & 0\\ 0 & 244 & 0\\ 0 & 0 & 137\end{bmatrix}$ and shares the result with the others. Bob then calculates $\hat{\mathbf{B}} = \mathbf{B}+\mathbf{R_b} = \begin{bmatrix}274 & 0 & 0\\ 0 & 357 & 0\\ 0 & 0 & 181\end{bmatrix}$ and shares the result with the others. Claire then calculates $\hat{\mathbf{C}} = \mathbf{C}+\mathbf{R_c} = \begin{bmatrix}342 & 0 & 0\\ 0 & 357 & 0\\ 0 & 0 & 70\end{bmatrix}$ and shares the result with the others. Alice generates a

random value $v_2 = 3$, after which Alice calculates:

$$
\begin{gathered}
u_1 \\
= \\
\hat{\mathbf{B}} \cdot \hat{\mathbf{C}} \cdot \mathbf{A} + (n-1) \cdot r_a - v_2 \\
= \\
\varphi(\begin{bmatrix} 274 & 0 & 0 \\ 0 & 357 & 0 \\ 0 & 0 & 181 \end{bmatrix} \cdot \begin{bmatrix} 342 & 0 & 0 \\ 0 & 357 & 0 \\ 0 & 0 & 70 \end{bmatrix} \cdot \begin{bmatrix} 100 \\ 010 \\ 001 \end{bmatrix}) + (3-1) \cdot 8015322 - 3 \\
= \\
233827 + 16030644 - 3 \\
= \\
16264468
\end{gathered}
$$

Bob then calculates

$$
\begin{gathered}
u_2 \\
= \\
u_1 - \varphi(\hat{\mathbf{A}} \cdot \hat{\mathbf{C}} \cdot \mathbf{R_b}) + (n-1) r_b \\
= \\
u_1 - \varphi(\begin{bmatrix} 173 & 0 & 0 \\ 0 & 244 & 0 \\ 0 & 0 & 137 \end{bmatrix} \cdot \begin{bmatrix} 342 & 0 & 0 \\ 0 & 357 & 0 \\ 0 & 0 & 70 \end{bmatrix} \cdot \begin{bmatrix} 274 & 0 & 0 \\ 0 & 356 & 0 \\ 0 & 0 & 180 \end{bmatrix} \cdot) \\
+ (3-1) \cdot 10543269 \\
= \\
16264468 - 48948132 + 21086538 \\
= \\
-11597126
\end{gathered}
$$

Claire then calculates

$$
\begin{gathered}
u_3 \\
= \\
u_2 - \varphi(\hat{\mathbf{A}} \cdot \hat{\mathbf{B}} \cdot \mathbf{R_c}) + (n-1) r_c \\
= \\
u_2 - \varphi(\begin{bmatrix} 173 & 0 & 0 \\ 0 & 244 & 0 \\ 0 & 0 & 137 \end{bmatrix} \cdot \begin{bmatrix} 274 & 0 & 0 \\ 0 & 357 & 0 \\ 0 & 0 & 181 \end{bmatrix} \cdot \begin{bmatrix} 341 & 0 & 0 \\ 0 & 357 & 0 \\ 0 & 0 & 69 \end{bmatrix}) \\
+ (3-1) \cdot 30084533 \\
= \\
-11597126 - 48972631 + 60169066 \\
= \\
-400691
\end{gathered}
$$

At this point $u_3$ is equal to the following:

$$
\begin{gathered}
u_3 = \varphi(\mathbf{A} \cdot \mathbf{B} \cdot \mathbf{C}) - \varphi(\mathbf{A} \cdot \mathbf{R_b} \cdot \mathbf{R_c}) - \varphi(\mathbf{B} \cdot \mathbf{R_a} \cdot \mathbf{R_c}) \\
- \varphi(\mathbf{C} \cdot \mathbf{R_a} \cdot \mathbf{R_b}) - v_2
\end{gathered}
$$

The leftover terms in $\varphi(\mathbf{A} \cdot \mathbf{R_b} \cdot \mathbf{R_c}) - \varphi(\mathbf{B} \cdot \mathbf{R_a} \cdot \mathbf{R_c})$
$- \varphi(\mathbf{C} \cdot \mathbf{R_a} \cdot \mathbf{R_b})$ need to be solved separately using their own 2-party scalar product protocol. Once these have been solved separately Claire calculates the following. For the sake of readability we introduce a helper variable $h$ here.

$$
\begin{gathered}
h \\
= \\
u_3 + \varphi(\mathbf{A} \cdot \mathbf{R_b} \cdot \mathbf{R_c}) + \varphi(\mathbf{B} \cdot \mathbf{R_a} \cdot \mathbf{R_c}) + \varphi(\mathbf{C} \cdot \mathbf{R_a} \cdot \mathbf{R_b}) \\
= \\
u_3 + \varphi(\begin{bmatrix}100\\010\\001\end{bmatrix} \cdot \begin{bmatrix}274 & 0 & 0\\ 0 & 356 & 0\\ 0 & 0 & 180\end{bmatrix} \cdot \begin{bmatrix}341 & 0 & 0\\ 0 & 357 & 0\\ 0 & 0 & 69\end{bmatrix}) + \varphi(\begin{bmatrix}000\\010\\001\end{bmatrix} \\
\cdot \begin{bmatrix}172 & 0 & 0\\ 0 & 243 & 0\\ 0 & 0 & 136\end{bmatrix} \cdot \begin{bmatrix}341 & 0 & 0\\ 0 & 357 & 0\\ 0 & 0 & 69\end{bmatrix}) \\
+ \varphi(\begin{bmatrix}100\\000\\001\end{bmatrix} \cdot \begin{bmatrix}172 & 0 & 0\\ 0 & 243 & 0\\ 0 & 0 & 136\end{bmatrix} \cdot \begin{bmatrix}274 & 0 & 0\\ 0 & 356 & 0\\ 0 & 0 & 180\end{bmatrix}) \\
= \\
-400691 + 232946 + 96135 + 71608 \\
= \\
-2
\end{gathered}
$$

Alice then calculates $h + v_2 = -2 + 3 = 1$ which is our final result and corresponds to our expected result.

# 3 GIT repository Privacy Preserving N-party scalar product protocol

An implementation of the $n$-party protocol in both java and in python can be found in the following git repo: `https://github.com/MaastrichtU-CDS/n-scalar-product-protocol`

Table 1: Experimental results vertically split 3-party scenarios where attributes were randomly split across parties. '*' Indicates the best performing model, '†' indicates the second best performing model.

| | | AUC | | | | | |
|---|---|---|---|---|---|---|---|
| Name | Missing Data Level | FBNE | Party 1 | Party 2 | Party 3 | Central | VertiBayes |
| Alarm population size: 10000 | 0 | 0,884* | 0,790 | 0,669 | 0,561 | 0,790† | 0,790† |
| Asia population size: 10000 | 0 | 0,997* | 0,824 | 0,873 | 0,902 | 0,996† | 0,986 |
| | 0.05 | 0,739† | 0,657 | 0,651 | 0,664 | 0,736 | 0,750* |
| | 0.1 | 0,622† | 0,607 | 0,519 | 0,583 | 0,621 | 0,704* |
| | 0.3 | 0,418† | 0,380 | 0,394 | 0,407 | 0,417 | 0,569* |
| Autism population size: 704 | 0 | 0,934† | 0,784 | 0,787 | 0,740 | 0,832 | 0,977* |
| | 0.05 | 0,797* | 0,742 | 0,664 | 0,434 | 0,732 | 0,780† |
| | 0.1 | 0,734* | 0,686 | 0,605 | 0,447 | 0,694 | 0,730† |
| | 0.3 | 0,497† | 0,492 | 0,388 | 0,369 | 0,489 | 0,627* |
| Diabetes population size: 768 | 0 | 0,801* | 0,659 | 0,647 | 0,675 | 0,789† | 0,783 |
| | 0.05 | 0,753* | 0,654 | 0,630 | 0,598 | 0,728 | 0,740† |
| | 0.1 | 0,695† | 0,613 | 0,614 | 0,560 | 0,678 | 0,804* |
| | 0.3 | 0,445† | 0,407 | 0,383 | 0,382 | 0,438 | 0,552* |
| Mushroom population size: 8124 | 0 | 0,989* | 0,818 | 0,987 | 0,589 | 0,988† | 0,987 |

# 4 Experimental results FBNE

The remaining results of the experiments ran for FBNE are contained in the following pages.

Table 2: Experimental results vertically split 3-party scenarios where attributes were manually split across parties. '*' Indicates the best performing model, '†' indicates the second best performing model.

| | | AUC | | | | | |
|---|---|---|---|---|---|---|---|
| Name | Missing Data Level | FBNE | Party 1 | Party 2 | Party 3 | Central | VertiBayes |
| Autism population size: 704 | 0 | 0,920* | 0,843 | 0,730 | 0,811 | 0,830† | 0,829 |
| | 0.05 | 0,797* | 0,742 | 0,664 | 0,434 | 0,732 | 0,780† |
| | 0.1 | 0,734* | 0,686 | 0,605 | 0,447 | 0,694 | 0,730† |
| | 0.3 | 0,497† | 0,492 | 0,388 | 0,369 | 0,489 | 0,627* |
| Mushroom population size: 8124 | 0 | 0,991* | 0,881 | 0,986 | 0,680 | 0,988† | 0,986 |

Table 3: Experimental results hybrid split 3-party scenarios where hybrid split attributes can fully incorporated into the local models. '*' Indicates the best performing model, '†' indicates the second best performing model.

| | | AUC | | | | | |
|---|---|---|---|---|---|---|---|
| Name | Missing Data Level | FBNE | Party 1 | Party 2 | Party 3 | Central | VertiBayes |
| Asia population size: 10000 | 0 | 0,996† | 0,885 | 0,929 | 0,929 | 0,995† | 0,999* |
| | 0.05 | 0,743 | 0,709 | 0,717 | 0,722 | 0,745† | 0,766* |
| | 0.1 | 0,623† | 0,612 | 0,554 | 0,555 | 0,619 | 0,669* |
| | 0.3 | 0,419† | 0,401 | 0,410 | 0,412 | 0,419† | 0,567* |
| Autism population size: 704 | 0 | 0,903* | 0,807 | 0,817 | 0,822 | 0,849† | 0,847 |
| | 0.05 | 0,793* | 0,724 | 0,710 | 0,715 | 0,753 | 0,777† |
| | 0.1 | 0,740† | 0,671 | 0,667 | 0,671 | 0,682 | 0,749* |
| | 0.3 | 0,527† | 0,481 | 0,493 | 0,493 | 0,503 | 0,747* |
| Diabetes population size: 768 | 0 | 0,811* | 0,731 | 0,695 | 0,700 | 0,779† | 0,776 |
| | 0.05 | 0,692† | 0,604 | 0,608 | 0,607 | 0,673 | 0,734* |
| | 0.1 | 0,755* | 0,670 | 0,670 | 0,673 | 0,726 | 0,752† |
| | 0.3 | 0,456† | 0,416 | 0,403 | 0,404 | 0,439 | 0,697* |
| Iris population size: 150 | 0 | 0,939* | 0,889† | 0,879 | 0,886 | 0,883 | 0,771 |
| | 0.05 | 0,892* | 0,783 | 0,835 | 0,830 | 0,876† | 0,736 |
| | 0.1 | 0,788* | 0,702 | 0,719 | 0,724† | 0,713 | 0,704 |
| | 0.3 | 0,653† | 0,588 | 0,595 | 0,599 | 0,662* | 0,611 |

Table 4: Experimental results horizontally split 3-party scenarios where records are randomly split across parties. '*' Indicates the best performing model, '†' indicates the second best performing model.

| | | AUC | | | | | |
|---|---|---|---|---|---|---|---|
| Name | Missing Data Level | FBNE | Party 1 | Party 2 | Party 3 | Central | VertiBayes |
| Asia population size: 10000 | 0 | 0,995* | 0,995* | 0,995* | 0,995* | 0,995* | 0,987 |
| | 0.05 | 0,741† | 0,741† | 0,741† | 0,741† | 0,730 | 0,763* |
| | 0.1 | 0,623† | 0,623† | 0,623† | 0,623† | 0,618 | 0,674* |
| | 0.3 | 0,418† | 0,418† | 0,418† | 0,418† | 0,417 | 0,568* |
| Autism population size: 704 | 0 | 0,889* | 0,836 | 0,836 | 0,836 | 0,838† | 0,829 |
| | 0.05 | 0,794* | 0,736 | 0,736 | 0,736 | 0,754 | 0,780† |
| | 0.1 | 0,724† | 0,687 | 0,687 | 0,687 | 0,688 | 0,749* |
| | 0.3 | 0,544† | 0,493 | 0,494 | 0,493 | 0,494 | 0,625* |
| Diabetes population size: 768 | 0 | 0,775 | 0,780† | 0,780† | 0,780† | 0,786* | 0,778 |
| | 0.05 | 0,730† | 0,727 | 0,727 | 0,727 | 0,720 | 0,753* |
| | 0.1 | 0,674† | 0,673 | 0,673 | 0,673 | 0,667 | 0,733* |
| | 0.3 | 0,448† | 0,439 | 0,437 | 0,438 | 0,439 | 0,648* |
| Iris population size: 150 | 0 | 0,960* | 0,896 | 0,897† | 0,890 | 0,890 | 0,761 |
| | 0.05 | 0,875 | 0,876 | 0,877† | 0,875 | 0,879* | 0,768 |
| | 0.1 | 0,826* | 0,703 | 0,703 | 0,703 | 0,709 | 0,676 |
| | 0.3 | 0,632 | 0,666* | 0,666* | 0,666* | 0,664† | 0,626 |

Table 5: Experimental results horizontally split 2-party scenarios where records are randomly split across parties. Varies levels of bias were introduced in this experiment where the level of bias corresponds to the probability of an individual with first class label to be assigned to party 1. '*' Indicates the best performing model, '†' indicates the second best performing model.

| | | | AUC | | | | |
|---|---|---|---|---|---|---|---|
| Name | Bias Level | Missing Data Level | Ensemble | Party 1 | Part 2 | Central | VertiBayes |
| Asia population size: 10000 | 0.75 | 0 | 0,995† | 0,995† | 0,995† | 0,996* | 0,987 |
| | | 0.05 | 0,743† | 0,741 | 0,741 | 0,742 | 0,765* |
| | | 0.1 | 0,624† | 0,623 | 0,623 | 0,624† | 0,670* |
| | | 0.3 | 0,419† | 0,419† | 0,418 | 0,417 | 0,568* |
| | 0.85 | 0 | 0,996* | 0,995† | 0,995† | 0,996* | 0,986 |
| | | 0.05 | 0,741† | 0,741† | 0,741† | 0,737 | 0,763* |
| | | 0.1 | 0,622 | 0,623 | 0,623 | 0,628† | 0,671* |
| | | 0.3 | 0,419 | 0,419 | 0,419 | 0,420† | 0,568* |
| | 0.95 | 0 | 0,995† | 0,995† | 0,995† | 0,996* | 0,986 |
| | | 0.05 | 0,741 | 0,741 | 0,741 | 0,742† | 0,764* |
| | | 0.1 | 0,624 | 0,623 | 0,623 | 0,627† | 0,669* |
| | | 0.3 | 0,420† | 0,419 | 0,419 | 0,418 | 0,567* |
| Autism population size: 704 | 0.75 | 0 | 0,876* | 0,500 | 0,780 | 0,845† | 0,835 |
| | | 0.05 | 0,786* | 0,488 | 0,487 | 0,748 | 0,780† |
| | | 0.1 | 0,708† | 0,501 | 0,487 | 0,698 | 0,749* |
| | | 0.3 | 0,535† | 0,380 | 0,456 | 0,500 | 0,627* |
| | 0.85 | 0 | 0,880* | 0,500 | 0,770 | 0,848† | 0,835 |
| | | 0.05 | 0,779* | 0,464 | 0,464 | 0,731† | 0,779* |
| | | 0.1 | 0,722† | 0,445 | 0,450 | 0,688 | 0,746* |
| | | 0.3 | 0,527† | 0,427 | 0,454 | 0,493 | 0,630* |
| | 0.95 | 0 | 0,898* | 0,534 | 0,500 | 0,830 | 0,834† |
| | | 0.05 | 0,779* | 0,464 | 0,512 | 0,736† | 0,779* |
| | | 0.1 | 0,724† | 0,443 | 0,528 | 0,675 | 0,747* |
| | | 0.3 | 0,512† | 0,430 | 0,450 | 0,499 | 0,629* |
| Diabetes population size: 768 | 0.75 | 0 | 0,778 | 0,500 | 0,500 | 0,787* | 0,780† |
| | | 0.05 | 0,737† | 0,480 | 0,480 | 0,731 | 0,753* |
| | | 0.1 | 0,676† | 0,447 | 0,447 | 0,674 | 0,736* |
| | | 0.3 | 0,430 | 0,405 | 0,359 | 0,445† | 0,645* |
| | 0.85 | 0 | 0,781* | 0,500 | 0,500 | 0,776† | 0,781* |
| | | 0.05 | 0,730† | 0,480 | 0,480 | 0,730† | 0,754* |
| | | 0.1 | 0,675† | 0,447 | 0,447 | 0,678 | 0,736* |
| | | 0.3 | 0,412 | 0,349 | 0,406 | 0,444† | 0,645* |
| | 0.95 | 0 | 0,772 | 0,500 | 0,500 | 0,786* | 0,780† |
| | | 0.05 | 0,581 | 0,480 | 0,499 | 0,732† | 0,754* |
| | | 0.1 | 0,530 | 0,447 | 0,459 | 0,672† | 0,737* |
| | | 0.3 | 0,354 | 0,370 | 0,346 | 0,442† | 0,646* |
| Iris population size: 150 | 0.75 | 0 | 0,942* | 0,876 | 0,735 | 0,890† | 0,767 |
| | | 0.05 | 0,870† | 0,727 | 0,782 | 0,875* | 0,760 |
| | | 0.1 | 0,802* | 0,588 | 0,703 | 0,718† | 0,669 |
| | | 0.3 | 0,608 | 0,544 | 0,611 | 0,676* | 0,607 |
| | 0.85 | 0 | 0,950* | 0,782 | 0,692 | 0,896† | 0,779 |
| | | 0.05 | 0,870† | 0,625 | 0,793 | 0,879* | 0,766 |
| | | 0.1 | 0,746* | 0,604 | 0,712 | 0,710 | 0,678 |
| | | 0.3 | 0,636† | 0,478 | 0,571 | 0,669* | 0,600 |
| | 0.95 | 0 | 0,915* | 0,560 | 0,653 | 0,895† | 0,780 |
| | | 0.05 | 0,800† | 0,497 | 0,688 | 0,873* | 0,771 |
| | | 0.1 | 0,752* | 0,504 | 0,641 | 0,710† | 0,667 |
| | | 0.3 | 0,633† | 0,469 | 0,480 | 0,673* | 0,608 |

Table 6: Experimental results horizontally split 3-party scenarios where records are randomly split across parties. Varies levels of bias were introduced in this experiment where the level of bias corresponds to the probability of an individual with first class label to be assigned to party 1. '*' Indicates the best performing model, '†' indicates the second best performing model.

| | | | AUC | | | | | |
|---|---|---|---|---|---|---|---|---|
| Name | Bias Level | Missing Data Level | Ensemble | Party 1 | Party 2 | Party 3 | Central | VertiBayes |
| Asia population size: 10000 | 0.75 | 0 | 0,996* | 0,995† | 0,995† | 0,995† | 0,995† | 0,988 |
| | | 0.05 | 0,740 | 0,741 | 0,741 | 0,741 | 0,745† | 0,766* |
| | | 0.1 | 0,622 | 0,623 | 0,623 | 0,623 | 0,624† | 0,668* |
| | | 0.3 | 0,419† | 0,418 | 0,418 | 0,418 | 0,418 | 0,569* |
| | 0.85 | 0 | 0,996* | 0,995† | 0,995† | 0,995† | 0,996* | 0,986 |
| | | 0.05 | 0,741† | 0,741† | 0,741† | 0,741† | 0,740 | 0,769* |
| | | 0.1 | 0,625 | 0,623 | 0,623 | 0,623 | 0,628† | 0,671* |
| | | 0.3 | 0,419† | 0,418 | 0,418 | 0,418 | 0,417 | 0,568* |
| | 0.95 | 0 | 0,995† | 0,995† | 0,995† | 0,995† | 0,996* | 0,987 |
| | | 0.05 | 0,743 | 0,741 | 0,741 | 0,741 | 0,745† | 0,762* |
| | | 0.1 | 0,622 | 0,623 | 0,623 | 0,623 | 0,623† | 0,671* |
| | | 0.3 | 0,419† | 0,418 | 0,418 | 0,418 | 0,416 | 0,569* |
| Autism population size: 704 | 0.75 | 0 | 0,911* | 0,836 | 0,836 | 0,836 | 0,843† | 0,833 |
| | | 0.05 | 0,797* | 0,736 | 0,736 | 0,736 | 0,732 | 0,776† |
| | | 0.1 | 0,728† | 0,687 | 0,687 | 0,687 | 0,700 | 0,746* |
| | | 0.3 | 0,541† | 0,493 | 0,494 | 0,494 | 0,491 | 0,627* |
| | 0.85 | 0 | 0,905* | 0,836 | 0,836 | 0,836 | 0,832 | 0,842† |
| | | 0.05 | 0,810* | 0,736 | 0,736 | 0,736 | 0,732 | 0,785† |
| | | 0.1 | 0,739† | 0,687 | 0,688 | 0,687 | 0,687 | 0,744* |
| | | 0.3 | 0,541† | 0,493 | 0,494 | 0,494 | 0,492 | 0,623* |
| | 0.95 | 0 | 0,899* | 0,836 | 0,836 | 0,836 | 0,844† | 0,835 |
| | | 0.05 | 0,785* | 0,736 | 0,736 | 0,736 | 0,745 | 0,780† |
| | | 0.1 | 0,720† | 0,687 | 0,687 | 0,687 | 0,677 | 0,745* |
| | | 0.3 | 0,490 | 0,494 | 0,494 | 0,494 | 0,495† | 0,630* |
| Diabetes population size: 768 | 0.75 | 0 | 0,788* | 0,780 | 0,780 | 0,780 | 0,779 | 0,786† |
| | | 0.05 | 0,760† | 0,727 | 0,727 | 0,727 | 0,727 | 0,761* |
| | | 0.1 | 0,690† | 0,673 | 0,672 | 0,672 | 0,680 | 0,743* |
| | | 0.3 | 0,430 | 0,437 | 0,438† | 0,438† | 0,434 | 0,653* |
| | 0.85 | 0 | 0,785* | 0,780† | 0,780† | 0,780† | 0,776 | 0,777 |
| | | 0.05 | 0,745† | 0,727 | 0,727 | 0,727 | 0,726 | 0,747* |
| | | 0.1 | 0,674 | 0,673 | 0,673 | 0,673 | 0,676† | 0,737* |
| | | 0.3 | 0,394 | 0,437 | 0,438 | 0,439† | 0,438 | 0,648* |
| | 0.95 | 0 | 0,752 | 0,780† | 0,780† | 0,780† | 0,782* | 0,780† |
| | | 0.05 | 0,631 | 0,727 | 0,727 | 0,727 | 0,723† | 0,755* |
| | | 0.1 | 0,564 | 0,672 | 0,673 | 0,673† | 0,668 | 0,740* |
| | | 0.3 | 0,353 | 0,438 | 0,438 | 0,438 | 0,443† | 0,649* |