Privacy-aware Federated Learning for Accelerating Biomedical and Clinical Time-to-Event Analysis

Kumulative Dissertation

An der Universität Hamburg eingereichte Dissertation zur Erlangung des akademischen Grades

Dr. rer. nat.

von Julian Alexander Späth
Geboren am 23.02.1993 in Tübingen

Fakultät für Mathematik, Informatik und Naturwissenschaften
Fachbereich Informatik

2024

Tag der Disputation: 27.09.2024
Erster Gutachter: Prof. Dr. Jan Baumbach
Zweiter Gutachter: Prof. Dr. Richard Röttger
Dritter Gutachter: Prof. Dr. Dominik Heider

Abstract

The digitalization of health care leads to the accumulation of huge amounts of biomedical data that is used in clinical research and studies to uncover therapies, treatments, or novel biomarkers. One important set of tools in clinical research is time-to-event analysis. These kinds of algorithms are used to analyze censored data. For such data, the exact time of an event is not known, since the event does not necessarily occur during observation time. These and other biomedical and clinical datasets are typically collected centrally at a single institution and then analyzed using statistical methods or machine learning. For gathering larger amounts of data, data sharing with a central institution is necessary. However, current privacy regulations make it difficult to share sensitive data with other institutions and gather them at a central instance. To address this issue, recently, a novel approach known as federated learning was introduced. Federated learning enables the application of machine learning on geographically distributed datasets. Therefore, the raw data of each institution stays locally and only model parameters or summary statistics are shared with a central aggregator. Despite recent advances in this field, there are still only a few accessible and privacy-preserving solutions for biomedical research, especially in time-to-event analysis.

The results of this cumulative dissertation are based on three main publications. The first publication introduces Partea, a platform for privacy-aware time-to-event analysis. Partea incorporates the most commonly employed time-to-event techniques and makes them accessible through a graphical user interface without requiring any programming expertise. The second publication describes FeatureCloud, a federated learning platform that goes beyond time-to-event analysis and enables both the use and development of federated learning algorithms by providing the necessary infrastructure. Finally, in the third publication, FeatureCloud was used to develop and evaluate a federated survival support vector machine for the analysis of distributed time-to-event data.

The developed methods and tools in this work extend existing approaches for analyzing time-to-event data on decentralized datasets and are directly accessible to researchers, statisticians, and clinicians. Furthermore, the dissertation demonstrates that federated learning algorithms possess the capability

to attain a comparable level of accuracy on distributed datasets as compared to the original algorithms that solely operate on centrally collected datasets. By providing a broader set of algorithms, implementing privacy-enhancing technologies and providing user-friendly interfaces, the results of this dissertation increase the accessibility of federated learning in biomedical and clinical research environments and reduce the hurdles of complex federated learning infrastructures.

Zusammenfassung

Die Digitalisierung des Gesundheitswesens führt zur Anhäufung riesiger Mengen biomedizinischer Daten. Diese Daten werden in der klinischen Forschung und in Studien verwendet, um Therapien, Behandlungen oder neue Biomarker zu erforschen. Ein wichtiges Instrument in der klinischen Forschung ist die Ereigniszeitanalyse. Diese Art von Algorithmen wird verwendet, um zensierte Daten zu analysieren. Bei solchen Daten ist der genaue Zeitpunkt eines Ereignisses nicht bekannt, da das Ereignis nicht unbedingt während der Beobachtungszeit eintritt. Diese und andere biomedizinische und klinische Datensätze werden in der Regel zentral in einer einzigen Einrichtung gesammelt und dann mit statistischen Methoden oder maschinellem Lernen analysiert. Für die Erfassung größerer Datenmengen ist das Teilen von Daten mit einer zentralen Einrichtung erforderlich. Die derzeitigen Datenschutzbestimmungen erschweren jedoch die Weitergabe sensibler Daten an andere Einrichtungen und deren Sammlung an einer zentralen Stelle. Um dieses Problem zu lösen, wurde kürzlich ein neuartiger Ansatz eingeführt, der als Federated Learning bekannt ist. Federated Learning ermöglicht die Anwendung von maschinellem Lernen auf verteilten Datensätzen. Dabei verbleiben die Daten jeder Einrichtung lokal und nur die Modellparameter oder zusammenfassende Statistiken werden mit einem zentralen Institut ausgetauscht. Trotz der jüngsten Fortschritte in diesem Bereich gibt es immer noch nur wenige zugängliche und privatsphäreschützende Lösungen für die biomedizinische Forschung, insbesondere für die Ereigniszeitanalyse.

Die Ergebnisse dieser kumulativen Dissertation stützen sich auf drei Hauptpublikationen. Die erste Veröffentlichung stellt Partea vor, eine Plattform für privatsphäreschützende Ereigniszeitanalysen. Partea unterstützt die am häufigsten verwendeten Ereigniszeitanalyse-Methoden und macht sie über eine grafische Benutzeroberfläche zugänglich, ohne dass Programmierkenntnisse erforderlich sind. Die zweite Veröffentlichung beschreibt FeatureCloud, eine Plattform für Federated Learning, die über die reine Ereigniszeitanalyse hinausgeht und sowohl die Nutzung als auch die Entwicklung von Algorithmen für Federated Learning ermöglicht, indem sie die notwendige Infrastruktur bereitstellt. In der dritten Publikation wurde schließlich FeatureCloud verwendet, um eine Survival Support Vektor Maschine für die verteilte Ereigniszeitanalyse zu entwickeln und zu evaluieren.

Die entwickelten Methoden und Werkzeuge dieser Arbeit erweitern bestehende Ansätze der Ereigniszeitanalyse für den Einsatz auf dezentral gespeicherten Datensätzen und sind für Forscher, Statistiker und Mediziner direkt zugänglich. Darüber hinaus zeigt die Dissertation, dass Federated Learning in der Lage ist, auf dezentral gespeicherten Datensätzen eine vergleichbare Genauigkeit erreichen zu können wie die ursprünglichen Algorithmen, die ausschließlich auf zentral gesammelten Datensätzen arbeiten. Die in dieser Arbeit bereitgestellte Palette an Algorithmen, die implementierten Methoden zum Schutz der Privatsphäre und die benutzerfreundlichen Schnittstellen erhöhen die Zugänglichkeit von Federated Learning in der biomedizinischen und klinischen Forschung und verringern die Hürden komplexer Infrastrukturen.

Acknowledgements

I would like to express my deepest appreciation to my supervisor, Prof. Dr. Jan Baumbach. He gave me the possibility to work in this interesting research field and provided an outstanding and supportive working environment throughout my whole time in the lab.

Further, I am extremely grateful to the committee and reviewers of my dissertation, who take their valuable time for reading and grading.

Particularly helpful to me during my entire time was the FeatureCloud team. We worked together as a real team, helped each other to solve problems, and had a great time together. Here, I want to express a special thanks to Julian Matschinske, who was an extraordinary helpful colleague, who always shared his profound knowledge and accelerated my skills enormously.

I further would like to thank the University of Hamburg and the Cosy.Bio lab for its valuable feedback, as well as the great research and social environment it provided for me.

Thanks should also go out to the proofreaders of my dissertation, Zakaria Louadi, Florence López, and Julian Matschinske.

Furthermore, special thanks to the Brigham and Women's Hospital and Harvard Medical School for offering me a doctoral research exchange and giving me invaluable insights into a clinical research environment. In particular, I want to thank Joseph Loscalzo, M.D, Ph.D., and my advisors Ruisheng Wang and Vladislav Elgart at the department of medicine. I want to further acknowledge the Technical University of Munich and the chair of experimental bioinformatics, where my Ph.D. journey began.

Finally, I thank my family, friends, and girlfriend, who supported me the whole time.

Publication record

Main publications of this dissertation

1. **Julian Späth***, Julian Matschinske, Frederick K. Kamanu, Sabina A. Murphy, Olga Zolotareva, Mohammad Bakhtiari, Elliott M. Antman, Joseph Loscalzo, Alissa Brauneck, Louisa Schmalhorst, Gabriele Buchholtz, and Jan Baumbach. **Privacy-aware multi-institutional time-to-event studies**. PLOS Digital Health, 1(9):1–16, 09 2022. doi: https://doi.org/10.1371/journal.pdig.0000101

2. Julian Matschinske*, **Julian Späth***, Mohammad Bakhtiari, Niklas Probul, Mohammad Mahdi Kazemi Majdabadi, Reza Nasirigerdeh, Reihaneh Torkzadehmahani, Anne Hartebrodt, Balazs-Attila Orban, Sándor-József Fejér, Olga Zolotareva, Supratim Das, Linda Baumbach, Josch K Pauling, Olivera Tomašević, Béla Bihari, Marcus Bloice, Nina C Donner, Walid Fdhila, Tobias Frisch, Anne-Christin Hauschild, Dominik Heider, Andreas Holzinger, Walter Hötzendorfer, Jan Hospes, Tim Kacprowski, Markus Kastelitz, Markus List, Rudolf Mayer, Mónika Moga, Heimo Müller, Anastasia Pustozerova, Richard Röttger, Christina C Saak, Anna Saranti, Harald H H W Schmidt, Christof Tschohl, Nina K Wenke, and Jan Baumbach. **The featurecloud platform for federated learning in biomedicine: Unified approach**. J Med Internet Res, 25:e42621, Jul 2023. doi: https://doi.org/10.2196/42621

3. **Julian Späth***, Zeno Sewald, Niklas Probul, Magali Berland, Mathieu Almeida, Nicolas Pons, Emmanuelle Le Chatelier, Pere Ginès, Cristina Solé, Adrià Juanola, Josch Pauling, and Jan Baumbach. **Privacy-preserving federated survival support vector machines for cross-institutional time-to-event analysis: Algorithm development and validation**. JMIR AI, 3:e47652, Mar 2024. doi: https://doi.org/10.2196/47652

*(shared) first author(s)

Supporting publications of this dissertation

4. Marisol Salgado-Albarrán*, **Julian Späth**, Rodrigo González-Barrios, Jan Baumbach, and Ernesto Soto-Reyes. **Ctcfl regulates the pi3k-akt pathway and it is a target for personalized ovarian cancer therapy**. npj Systems Biology and Applications, 8(1):5, Feb 2022. doi: https://doi.org/10.1038/s41540-022-00214-z

5. Olga Zolotareva*, Reza Nasirigerdeh, Julian Matschinske, Reihaneh Torkzadehmahani, Mohammad Bakhtiari, Tobias Frisch, Julian Späth, David B Blumenthal, Amir Abbasinejad, Paolo Tieri, Georgios Kaissis, Daniel Rückert, Nina K Wenke, Markus List, and Jan Baumbach. **Flimma: a federated and privacy-aware tool for differential gene expression analysis**. Genome biology, 22(1):1–26, 2021. doi: https://doi.org/10.1186/s13059-021-02553-2

6. Reza Nasirigerdeh*, Reihaneh Torkzadehmahani, Julian Matschinske, Tobias Frisch, Markus List, **Julian Späth**, Stefan Weiss, Uwe Völker, Esa Pitkänen, Dominik Heider, Nina Kerstin Wenke, Georgios Kaissis, Daniel Rueckert, Tim Kacprowski, and Jan Baumbach. **splink: a hybrid federated tool as a robust alternative to meta-analysis in genome-wide association studies**. Genome Biology, 23(1):1–24, 2022. doi: https://doi.org/10.1186/s13059-021-02562-1

7. Reihaneh Torkzadehmahani*, Reza Nasirigerdeh, David B Blumenthal, Tim Kacprowski, Markus List, Julian Matschinske, **Julian Spaeth**, Nina Kerstin Wenke, and Jan Baumbach. **Privacy-Preserving artificial intelligence techniques in biomedicine**. Methods Inf. Med., 61(S 01):e12–e27, June 2022. doi: https://doi.org/10.1055/s-0041-1740630

8. Han Cao*, Youcheng Zhang, Jan Baumbach, Paul R Burton, Dominic Dwyer, Nikolaos Koutsouleris, Julian Matschinske, Yannick Marcon, Sivanesan Rajan, Thilo Rieg, Patricia Ryser-Welch, **Julian Späth**, The COMMITMENT Consortium, Carl Herrmann, and Emanuel Schwarz. **dsMTL: a computational framework for privacy-preserving, distributed multi-task machine learning**. Bioinformatics, 38(21):4919–4926, 09 2022. doi: https://doi.org/10.1093/bioinformatics/btac616

Other publications

9. **Julian Späth***, Rui-Sheng Wang, Maeve Humphrey, Jan Baumbach, and Joseph Loscalzo. **Machine learning–based integration of network features and chemical structure of compounds for sars-cov-2 drug effect analysis**. CPT: Pharmacometrics & Systems Pharmacology, 13(2):257–269, 2024. doi: https://doi.org/10.1002/psp4.13076

10. Gihanna Galindez*, Julian Matschinske*, Tim Daniel Rose*, Sepideh Sadegh*, Marisol Salgado-Albarrán*, **Julian Späth***, Jan Baumbach, and Josch Konstantin Pauling. **Lessons from the covid-19 pandemic for advancing computational drug repurposing strategies**. Nature Computational Science, 1(1):33–41, Jan 2021. doi: https://doi.org/10.1038/s43588-020-00007-6

11. Sepideh Sadegh*, Julian Matschinske*, David B. Blumenthal, Gihanna Galindez, Tim Kacprowski, Markus List, Reza Nasirigerdeh, Mhaned Oubounyt, Andreas Pichlmair, Tim Daniel Rose, Marisol Salgado-Albarrán, **Julian Späth**, Alexey Stukalov, Nina K. Wenke, Kevin Yuan, Josch K. Pauling, and Jan Baumbach. **Exploring the sars-cov-2 virus-host-drug interactome for drug repurposing**. Nature Communications, 11(1):3518, Jul 2020. doi: https://doi.org/10.1038/s41467-020-17189-2

Contents

1. Introduction

1.1 Motivation

The digitalization of healthcare, accelerated through electronic health records (EHRs) and enhancements in sequencing technologies for measuring molecular (OMICS) data, drastically increased the amount of biomedical data that is available today [1]. Especially high-dimensional OMICS data, such as genomics (DNA sequence), transcriptomics (RNA transcripts), or proteomics (protein expression), have become a critical part of clinical research, diagnosis, and therapies. For example, in cancer research, OMICS technologies enabled the detection and research of various novel disease biomarkers [2].

The huge amount of data that is generated all over the world requires efficient analysis techniques to generate new knowledge and accelerate research [3]. Similar to other industries, artificial intelligence (AI) and statistics are essential in biomedical data analysis [4]. However, large sample sizes are needed to perform accurate analyses and apply machine learning (ML) that are often unavailable at a single institution. This is especially the case in time-to-event analysis, a common type of analysis in clinical trials. Here, individuals are observed over time until an event of interest, such as death or relapse, occurs [5]. As the event of interest does not necessarily occur during observation time for all participating individuals, large sample sizes are even more critical for obtaining accurate results [6].

Until now, the standard approach for collecting large amounts of data for analysis is central data collection. One example of these data collections are data repositories, such as the International Cancer Genome Consortium [7] or the Cancer Genome Atlas [8]. These repositories collect data from various institutions to increase the sample size and make it available for research. For this, institutions are encouraged to share their local data with a central entity. However, sharing sensitive medical data comes with the risk of privacy leakage and patients losing control over their data [9]. Threats of re-identification do still exist even if data is anonymized [10]. Therefore, sharing data often leads to conflicts with current privacy regulations, such as the Health Insurance Portability and Accountability Act (HIPAA) in the United States of America,

or the General Data Protection Regulation (GDPR) in the European Union (EU) [11, 12]. Being compliant with these regulations comes with a high level of bureaucratic and anonymization efforts. Another popular approach for increasing data size is meta-analysis. Meta-analysis collects published data from comparable studies and incorporates their statistics into a new analysis. Unfortunately, this approach leads to inaccuracies as only summaries and not the raw data itself can be used. Another problem is, that many studies, especially in time-to-event analysis, publish their results with deficiencies in method reporting, making it difficult to assess the quality of the resulting statistics [13].

Recent research has shown that Federated Learning (FL) has a high potential to overcome the lack of data accessibility without sharing sensitive raw data [14, 15]. In Federated Learning (FL), only model parameters are exchanged with a central entity and not the sensitive raw data itself. These locally trained model parameters are then aggregated to a global model at the central entity. It could be shown that this aggregation results in only little, or in some cases even without any accuracy loss, compared to centralized data analysis [16]. The invention of FL led to a lot of research and developments in biomedicine, such as the analysis of EHRs [17, 18, 19], Coronavirus disease [20, 21, 22], genome-wide association studies (GWASs) [23, 24, 25], differential gene expression analysis [26], and many more [27, 28]. Also in time-to-event analysis, decentralized and (hybrid) FL approaches were developed and published recently [29, 30, 31, 32]. While the current FL approaches work in principle, they are still impracticable or inaccessible for researchers and physicians without training in federated analyses. One reason for that is the need for specific infrastructure that handles the exchange of insensitive parameters between clients. This also relates to the second issue, the complicated deployment of FL algorithms. All clients need to run the same version of the algorithm on various operating systems and environments that might largely differ across clients. Additionally, the data should also be available in a compatible format across clients. Finally, there are often no user interfaces available to run FL pipelines, making it impractical for users without programming knowledge. As privacy gets more and more in the focus of patients and the public, FL is already mentioned as an important future direction for AI in healthcare [33, 27]. Therefore, there will be a massive need for practical and accessible solutions.

Considering the high potential of FL and its existing hurdles, making FL available and accessible to the research community, physicians, and bioinformaticians is the primary goal of this dissertation. As time-to-event analysis has a particular need for large sample sizes and is one of the most common types of clinical data analysis, it is the main application and focus of this dissertation. The work of this dissertation focuses on three publications: Publication 1 (section 3.1) introduces a platform for privacy-aware time-to-event analysis called Partea [34]. Partea allows the execution of federated time-to-event anal-

ysis in combination with privacy-enhancing technologies (PETs), without any programming knowledge and without additional infrastructure. Partea provides the most popular time-to-event analysis models and makes them applicable in a FL scenario using a graphical user interface (GUI). By doing so, users can perform collaborative time-to-event studies with other institutions without sharing their raw data and without notable quality loss.

To even go beyond time-to-event analysis, Publication 2 (section 3.2) demonstrates a unified platform for all kinds of FL algorithms in biomedicine, called FeatureCloud [35]. The platform addresses all stages of an FL lifecycle: algorithm development, deployment, and execution. On the one hand, FeatureCloud provides all necessary functions for developers to implement FL algorithms and publish them in the integrated AI store. On the other hand, it allows clinicians or researchers to execute federated workflows with other collaboration partners without any programming knowledge by choosing from various federated algorithms. These algorithms can be assembled into workflows and executed via a user-friendly interface. Thereby, the platform solves several hurdles of federated infrastructures, handling complex deployment, project coordination, and network communication. As proof-of-concepts, the algorithms developed in Publication 1 have also been implemented as apps in FeatureCloud. However, Publication 2 goes beyond the application of federated time-to-event analysis. It rather focuses on the implementation and execution of all kinds of machine learning algorithms, such as linear and logistic regression, random forest (RF), and deep learning (DL).

Finally, Publication 3 (section3.3 introduces a federated implementation of the survival support vector machine (SVM) and its application in the FeatureCloud platform [35]. Here, an app for the survival SVM was developed and published in the FeatureCloud AI store, and compared its results in various scenarios for different numbers of clients and datasets with the centralized approach.

1.2 Outline

This section briefly describes the outline of this dissertation to provide a broad overview of its content.

Chapter 2 contains all relevant background information for this dissertation. It describes the basic concepts of time-to-event analysis as an essential tool in clinical data analysis (section 2.1). Moreover, the chapter introduces the background about FL (section 2.2), PETs (section 2.3), and hybrid FL (section 2.4). Finally, the chapter ends with an overview of the state-of-the-art, a description of the open challenges (section 2.5), and a collection of the objectives of this dissertation (section 2.6).

The three main publications of this cumulative dissertation are listed in Chapter 3. Besides the full citations of the publications, the summary, author contributions are depicted, as well as further information regarding the publications are depicted in detail.

Thereafter, the results and importance of the publications are summarized and discussed in Chapter 4. Here, the collected objectives are compared with the actual results, the implications of the results are discussed, and an outlook into the future is given.

2. Background

This chapter introduces the basics of time-to-event analysis (section 2.1), FL (section 2.2), privacy-enhancing technologies (PETs) (section 2.3), and combinations of FL and PETs called hybrid FL (section 2.4). Finally, it provides an overview of the state of the art and open challenges in federated time-to-event analysis (section 2.5 and ends with a collection of the objectives and contributions of this dissertation (section 2.6).

2.1 Time-to-Event Analysis

As already mentioned in the introduction, time-to-event analysis, sometimes also called survival analysis, is an important tool in clinical research. It refers to the analysis of data that observes the time until a particular event occurs, such as death or relapse of a patient. Such data is frequently collected in clinical trials where participants are observed over a period of time. A unique feature of this data is that it suffers from censorship, distinguishing time-to-event analyses from other statistical methods [36, 37]. Censoring can be categorized into three main types [38]:

1. **Right censoring** occurs when the event of interest has not occurred up to the time point of censoring. This is the case, for example, for individuals who drop out early or if the study terminates before observing the event of interest.

2. **Left censoring** occurs for individuals whose events of interest happened before observation time. This happens, if the event of interest already occurred before the observation.

3. **Interval censoring** occurs if the exact time of the event of interest is unknown, but the time range in which it occurs is known.

This work focuses on right censoring, the most common type in clinical trials [39]. Here, the events of interest, such as death or relapse, do not necessarily occur during observation time, meaning that the observed survival time is less

than the actual survival time. Naively using the observed survival time will underestimate the actual underlying survival probability of the population, as the event of interest for censored individuals might happen at a later time point or, in extreme cases, never. Furthermore, ignoring censored individuals in the analysis is incorrect, as this usually causes an underestimation of the actual survival probability [40]. The concept of right censorship is explained in more detail in Figure 2.1. As depicted in the figure, not considering censorship in the analysis leads to underestimating the underlying survival distribution, as all information on the right side of the dashed line would not be considered.

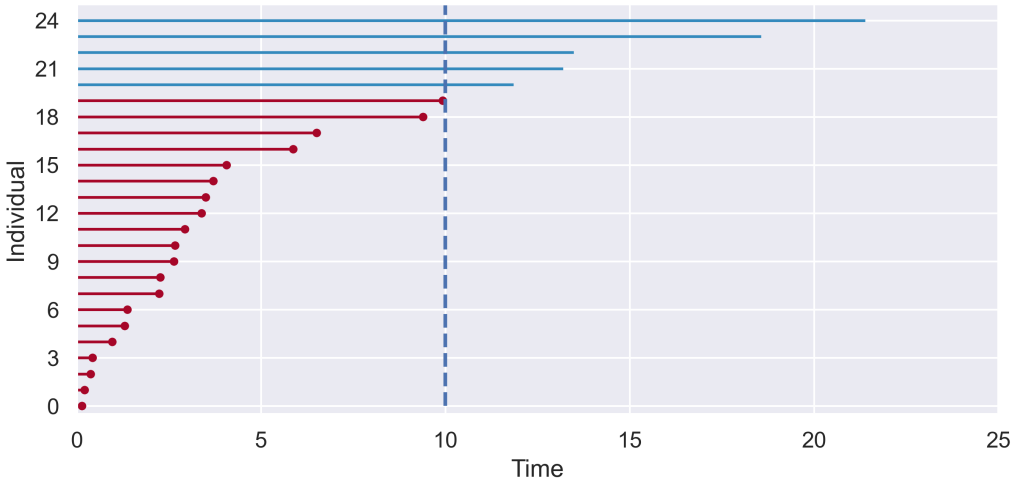


Figure 2.1: Example of right-censorship. Considering the end of observation at time ten, only for the red individuals did the event occur during the observation time. The blue lines represent right-censored individuals. Without considering the censorship, the information on the right side of the observation point (dashed line) would get lost, leading to an underestimation of the actual underlying survival distribution.

The following subsections will introduce the commonly used algorithms and state-of-the-art methods for time-to-event analysis.

2.1.1 Non-parametric Approaches

Non-parametric survival analysis approaches consider the population's time and event information and have no predictive features. Generally, three primary methods are applied in biomedical time-to-event analysis: the survival function $S(t)$ (A), the hazard function $h(t)$ (B), and the log-rank test (C), shown in Figure 2.2.

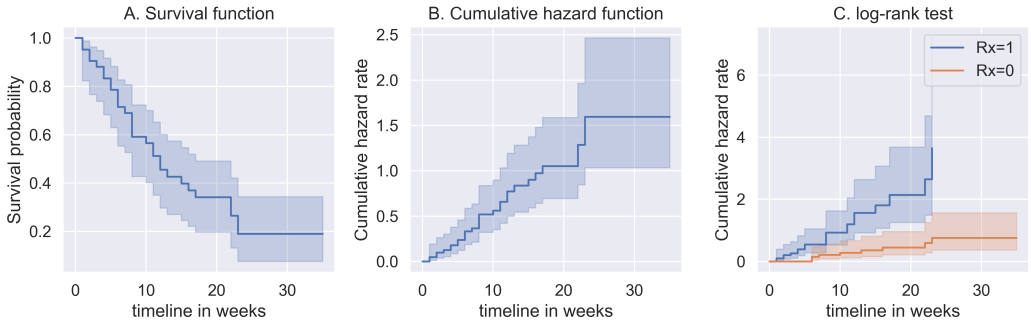


Figure 2.2: Non-parametric survival analyses. The survival function shows the survival probability of the population (y-axis) over time (x-axis). B. The CHF shows the hazard rate of the population (y-axis) over time (x-axis). C. The log-rank test is a statistical test for significance in survival analysis. It compares the CHF of two subpopulations and calculates a p-value to estimate the significance of the difference. In this example, the population that received treatment ($Rx = 0$) has a significantly lower hazard rate than the control group ($Rx = 1$).

Survival Function

The survival function $S(t)$ gives the probability for an individual that the event will happen later than a time point t. In practice, the survival function is a step function ranging from study start 0 until end $t - 1$ on the x-axis and from 0 to 1 on the y-axis (Figure 2.2 A). Each step indicates an event that occurred at this time point, causing a drop in survival probability [5]. A non-parametric survival function is estimated using the Kaplan-Meier estimation [41]. An equation of the Kaplan-Meier estimator is provided in Equation 2.1, with d_j representing the number of events that occurred at a time point j, and n_j representing the number of individuals at risk before time point j [42].

$$S(t_j) = S(t_j - 1)(1 - \frac{d_j}{n_j}) \tag{2.1}$$

Hazard Function

In contrast to the survival function, the hazard function $h(t)$ estimates failing instead of non-failing. Instead of a probability, it gives the instantaneous potential for an event to happen, given that the individual has survived up to time t [5]. It is generally estimated by the Nelson-Aalen algorithm for the cumulative hazard function (CHF), which is the integral of the hazard function between 0 and t [43]. As shown in Figure 2.3, the survival function, hazard function, and CHF are related [5]. Knowing one of the functions is enough to calculate the others.

Figure 2.3: Relation between hazard and survival. Taking the negative logarithm of the survival function results in the cumulative hazard function. The hazard function is simply the derivative of the cumulative hazard function.

Log-Rank Test

The log-rank test statistic is a very common statistical test to compare the survival and hazards functions of two populations [44]. It is a non-parametric test that does not make any assumptions about the underlying distribution and is based on the expected (E_i) and observed (O_i) numbers of events in both groups. The formula is stated in Equation 2.2. The statistic follows a chi-squared distribution, from which, ultimately, a p-value can be obtained to indicate its significance.

$$\chi^2 = \sum_{i=1}^{g} \frac{(O_i - E_i)^2}{E_i} \tag{2.2}$$

In the two-group case, rejecting the null hypothesis implies that the ratio of the hazard rates of the two groups is different from one [42]. As shown in Figure 2.2 C, the log-rank test statistic allows a quantitative comparison of two survival or hazard curves. In this example, the placebo group ($Rx = 1$) has a significantly higher hazard curve ($p < 0.005$) than the treatment arm ($Rx = 0$), indicating a positive effect for the evaluated treatment.

Even though the log-rank statistic is one of the most used algorithms in survival analysis, it has several limitations that should be considered before using it [45]. One such limitation is that the log-rank test statistic has an increased chance for type II errors if the survival curves of the two groups cross [46]. A type II error occurs if an actual, true effect is not detected.

2.1.2 Cox Proportional Hazard Model

Non-parametric approaches are very popular as they are easy to understand, allow the estimation of survival probabilities and hazard rates, and allow the comparison of different populations, such as a treatment and control arm in

a clinical study. However, non-parametric approaches do not provide any information about the underlying effects. These algorithms do not take any covariates into account. For this, the Cox proportional hazard model provides a way to regress one or more covariates against the outcome variable time, considering the censorship of the data [47, 48]. The formula of the Cox model is shown in Equation 2.3, with $h(t)$ being the hazard function, $h_0(t)$ the baseline hazard function, the covariates x, and the regression coefficients [49].

$$h(t) = h_0(t) * \exp\{\beta_1 x_1 + \beta_2 x_2 + ... + \beta_p x_p\} \tag{2.3}$$

As its name implies, the Cox proportional hazards model assumes that the hazard curves for each model are proportional. The equation is solved by an iterative approach using the Newton-Raphson algorithm by updating parameters to maximize the likelihood function until convergence [50].

The resulting hazard ratio of the Cox proportional hazards model are typically visualized in a plot of the coefficients and their 95% interval, as shown in Figure 2.4. In this example, the covariate sex has a slight positive effect (0.31); however, its confidence intervals (CIs) shows this effect is insignificant. For the covariate logWBC (logarithm of the count of white blood cells), and the treatment arm Rx, the effect is much higher than for sex (1.68 and 1.5). Moreover, the CI do not cross the zero line. Therefore, the two covariates, logWBC and Rx, can be considered to have a significant effect on the censored outcome time.

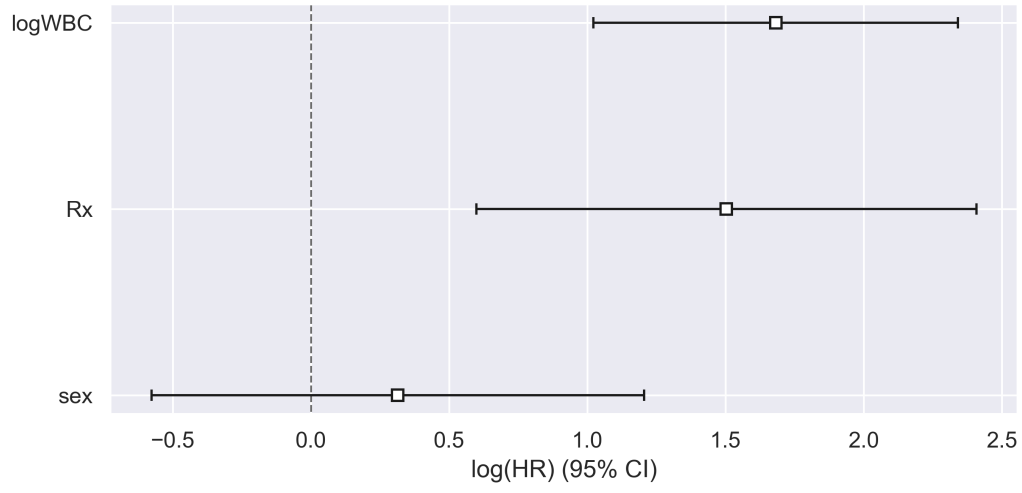


Figure 2.4: Logarithm of hazard ratios (log(HR)). The $log(HR)$ (white box) depicts the actual effect strength of the covariate. The effect is significant if the CI does not cross the dashed line at 0.

2.1.3 Machine Learning Approaches

While the previously explained methods are heavily used in practice and provide sufficient statistical results for most problems they have problems with high dimensional data, as it is often the case for OMICS data [51, 52]. Therefore, many of the classic ML methods have been extended to time-to-event analysis models. In the following subsections, two popular models will be briefly described: the random survival forest (RSF), and the survival SVM.

Random Survival Forest

The RSF is an extension of the popular random forest (RF) algorithm for the application to right-censored time-to-event data [53, 54]. It follows the original approach by Breiman et al. from 2001, but uses a splitting rule that involves the time-to-event and the censoring status to grow a tree [55]. An RF consists of multiple decision trees. These trees consist of a root node, daughter nodes, and leaf nodes. Each node splits the data into subgroups based on a splitting criterion. After the forest was created, a new sample can be put into each tree, predicting the label for this sample based on the majority vote of the tree's leafs.

The broad description of the algorithm, as implemented in the original R package by Ishwaran et al. in 2007 is depicted in Algorithm 1 [56].

Algorithm 1: Random Survival Forest [56]

1. Draw B bootstrap samples from the original data.
2. Grow a tree for each bootstrapped data set. At each node of the tree, randomly select p predictors (covariates) for splitting on. Split on a predictor using a survival splitting criterion. A node is split on that predictor which maximizes survival differences across daughter nodes.
3. Grow the tree to full size under the constraint that a terminal node should have no less than n unique deaths.
4. Calculate an ensemble cumulative hazard estimate by combining information from the B trees. One estimate for each individual in the data is calculated.
5. Compute an out-of-bag error rate for the ensemble derived using the first b trees, where $b = 1, ..., ntree$.

In the first step, B bootstrap samples are generated from the data. Bootstrapping is a method where random samples are drawn from the original data, usually with replacement. For each bootstrap sample, a tree is generated, leading to a forest of B trees is created on different subset of samples to increase diversity of the trees and reduce overfitting. Trees are created in the following way (step 2): At each node, a random set of p covariates are selected. For these

p covariates, a survival splitting criterion is applied to identify the covariate and threshold with the best split. A commonly used criterion is the log-rank test, which then splits the node on the covariate and threshold that maximizes the survival differences across the daughter nodes. This is repeated until the tree has reached full size or the terminal node constrained has been reached (step 3). Finally, the ensemble CHF is calculated for each tree leaf, averaging the CHF calculated by the Nelson-Aalen estimator of each tree (step 4). To validate the accuracy of the tree, the out-of-bag error rate can be calculated, or it can be validated on an external dataset (step 5).

To use the RSF to predict the survival probability of a new, unseen sample, the sample is propagated down the tree and ends up in a unique leaf node. The ensemble CHF of this leaf node can finally be used to predict the survival probability of the new sample.

Survival Support Vector Machine

Another popular machine learning method are SVMs [57]. In contrast to RFs they are not based on bootstrapping and decision tress, but try to find the optimal hyperplane with the maximum margin to separate two classes in the binary classification problem. Similarly, for regression, it tries to fit the hyperplane to the data within a certain margin, to capture as many data points as possible. SVMs are very popular in biomedical research, especially due to its kernel trick for high dimensional data. By mapping the input features into high-dimensional spaces, they can efficiently model complex, non-linear patterns in the data.

A very efficient method for using SVMs for survival analysis was proposed by Pölster et al. in 2015 [58]. The authors suggested three efficient training algorithms for linear survival SVMs: a ranking-based, a regression-based, and a combined approach. Equations 2.4 and 2.5 show the objective to minimize the formula for the regression-based survival SVM:

$$\arg\min_{\omega,b} f_{Regr}(\omega, b) = \frac{1}{2}\omega^T\omega + \frac{\gamma}{2}\sum_{i=0}^{n}(\mathbf{x}i_{\omega,b}(\gamma_i, x_i, \delta_i))^2 \tag{2.4}$$

$$\delta_{\omega,b}(y_i, \mathbf{x}_i, \delta_i) = \begin{cases} \max(0, y_i - \omega^T\mathbf{x}_i - b) & \text{if } \delta_i = 0 \\ y_i - \omega^T\mathbf{x}_i - b & \text{if } \delta_i = 1 \end{cases} \tag{2.5}$$

The formula uses the training data consisting of a feature vector $\mathbf{x}_i$, survival time $y_i \geq 0$, and the event indicator $\delta_i \in 0, 1$. The parameter γ determines the amount of regularization applied inversely. Minimizing the objective is finally performed via a truncated Newton optimization [59]. The pseudocode for the training of a survival SVM is depicted in Algorithm 2.

Algorithm 2: Survival SVM [60]

Data: Training data $D = (\mathbf{x}_i, y_i, \delta_i)_{i-1}^{n},$ hyper-parameter $\gamma > 0$.
Result: coefficients ω.
Randomly resolve ties in survival times $y_i \forall i \in 1, ..., n$;
$\omega_0 \leftarrow 0$;
$t \leftarrow 0$;
while *not converged* **do**
  Use conjugate gradient to determine search direction
  $\mathbf{u} = (\frac{\partial^2 f}{\partial \omega \partial \omega^T})1{-}1\frac{\partial}{\partial \omega}$ with $\omega = \omega^T$;
  Choose step size μ by backtracking line search;
  Update $\omega^{t+1} \leftarrow \omega^t + \mu\mathbf{u}$;
  $t \leftarrow t + 1$;
end
$\omega \leftarrow \omega^t$;

2.1.4 Validation of Time-to-Event Models

For evaluating time-to-event models, such as the survival SVM, RSF, or Cox proportional hazard model, common metrics for regression are not very practical, as they were not developed for censored samples. To address this issue, the concordance index (c-index) was developed as a generalization of the area under the receiver operating characteristic curve by Harrell et al. [61]. A c-index of 0.5 indicates a randomly predicting model, and 1.0 is a perfect model. As shown in the simplified Equation 2.6, the c-index is the probability of concordance between observed and predicted survival based on each pair of individuals [62].

$$C = \frac{\text{number of concordant pairs} + 0.5 * \text{number of tied pairs}}{\text{total number of pairs}} \tag{2.6}$$

To obtain the concordant pairs, first, only comparable pairs are considered. A pair is comparable if the shorter survival time is not censored. If both are censored, the survival time is equal. A concordant pair is then a pair where the subject with the earlier event is having a greater risk and discordant otherwise [63]. If both subjects have a tied risk, it is defined as a tied pair.

2.2 Federated Learning

The introduced time-to-event analysis algorithms in section 2.1 were designed for the application on single, centralized datasets. FL has the goal to enable the application and training of algorithms on data that is geographically distributed. It was first introduced in 2017 [14, 15]. As shown in Figure 2.5, a basic FL

workflow consists of a central aggregation server (orange box) and two or more participating clients (blue boxes) [27]. (1) The central aggregation server initially shares an initialized global model with all clients. This global model can be any ML model containing model parameters or weights. (2) After receiving the global model, each client uses its data to update it to a new, local model by using the same optimizer as in a centralized analysis. (3) The updated models are sent to the global aggregation server, which (4) aggregates the models (typically their parameters) with an aggregation approach, such as a weighted average. If the algorithm has multiple iterations, the whole process starts again from the beginning. Once all iterations are completed, the final aggregated model will be shared with all clients.

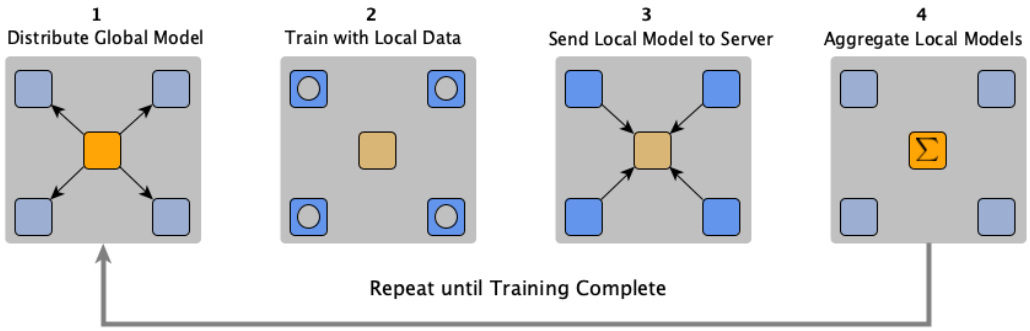


Figure 2.5: Basic FL workflow with central aggregation server. (1) Central aggregation server (orange box) shares the initialized global model with all participating clients (blue boxes). (2) Each client trains its local model (gray circle) using its local data. (3) The clients send their local models to the aggregation server. (4) The aggregation server aggregates the local models into a common global model. If multiple iterations are needed, the process starts again from (1). (Figure inspired by [27]).

FL can be broadly categorized into two main approaches, mostly dependent on the number of participating clients:

1. Cross-device FL: This type of FL aims to train ML models across many participating (even millions) devices (clients), such as mobile phones.

2. Cross-silo FL: Focusing on only a few clients, such as hospitals or companies. These institutions can use FL to train ML models while the data stays at each site, and only model parameters are exchanged with a central aggregator.

Both approaches share similar properties but have different challenges to face, especially in the context of privacy and scalability [64]. In addition to

the differentiation between cross-device and cross-silo, FL can be performed on either horizontally partitioned or vertically partitioned data. Horizontal FL trains a model on samples distributed across the clients with the same feature sets. Vertical FL learns a model on distributed feature sets for the same set of samples. In this dissertation, FL refers to on horizontal cross-silo FL if not stated otherwise.

Algorithm 3: FederatedAveraging

Data: The K clients are indexed by k; B is the local minibatch size, E is the number of local epochs, C is a random fraction of clients, and n is the learning rate. [14]
Result: Global weights ω
Server Executes:
Initialize ω
ClientUpdate(k, ω): // Run on client k
for *each round* $t = 1, 2, ..$ **do**
  $m \leftarrow \max(C_K, 1)$
  $S_t \leftarrow$ (random set of m clients)
  for *each client* $k \in S_t$ ***in parallel*** **do**
    $\omega_{t+1}^k \leftarrow$ **ClientUpdate**(k, ω_t)
  end
  $\omega_{t+1}^k \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} \omega_{t+1}^k$
end
$B \leftarrow$ (split P_k into batches of size B)
for *each local epoch* i *from* 1 *to* E **do**
  for *batch* $b \in B$ **do**
    $\omega \leftarrow \omega - \eta \Delta l(\omega, b)$
  end
end
return ω to server

Depending on these categories, an aggregation approach needs to be chosen. When FL was introduced first by McMahan et al. in 2017, the authors proposed an aggregation approach called FederatedAveraging (FedAvg), shown as pseudocode in Algorithm 3 [15]. FedAvg was originally implemented for communication-efficient DL in cross-device scenarios, and therefore supports some desirable properties for this kind of FL, such as being communication efficient and applicable to various types of optimization algorithms. First, a server initializes the weights globally. Then, in each iteration, a random subset of clients is used to perform the learning to reduce communication overhead. Each client receives the global weights from the server, locally updates them based on their local data, and sends them to the server for averaging. The local update is performed in batches and for a specific number of epochs, such that not necessarily each epoch needs to be averaged.

As FL has the property of only exchanging parameters rather than the raw data (in FedAvg, only local weights are exchanged), it is often advertised as a method for privacy-preserving ML. However, its basic properties do not guarantee security and privacy [65]. The model parameters of the local models could still reveal insights into the original data on which they were trained. Further, it is open to specific attacks on distributed architectures, such as malicious servers or inference attacks. The central aggregator is a potential target, as it knows about all local models in each iteration of every client. This problem can partly be solved by using hybrid approaches that combine FL with PETs [66], which will be explained in detail in the next chapters 2.3 and 2.4.

2.3 Privacy-Enhancing Technologies

PETs are defined as additional measures to reduce the risk of privacy leakage in a system without losing its functionality [67]. These measures do not necessarily need to be technical. Examples of non-technical PETs are compliance, or consent. In the context of ML, the most prominent PETs are purely technical: differential privacy (DP), homomorphic encryption (HE), secure multi-party computation (SMPC), and confidential computing [68]. The PETs mentioned can also be applied in combination with FL (hybrid FL).

2.3.1 Differential Privacy

DP, first introduced by Dwork et al. in 2006, is a concept that adds a controlled amount of random noise to the data to ensure and quantify the privacy of individuals in a database[69, 70]. The most common DP approaches, (ϵ)-DP and (ϵ, δ)-DP, draw random noise from a Laplacian or Gaussian distribution. The amount of the controlled noise is determined by the sensitivity of the function and the parameters ϵ and δ.

Sensitivity is a measure of how much a function's output changes when one entry of the database is removed. Formally, the L1 sensitivity of a function $f : D^n \rightarrow R^d$ is defined as the smallest number $S(f)$ such that the following holds for all $x, x' \in D^n$ that differ in a single entry:

$$f(x) - f(x') \leq S(f) \tag{2.7}$$

For example, when considering a function that counts the number of death events in a study cohort, the sensitivity of this function would be one since removing one study participant could, at most, change the function output by one.

The parameter ϵ quantifies the privacy loss of an individual, considering the dataset would be released. It handles the level of privacy needed for the DP

mechanism. Together with the sensitivity, it defines the scale of the Laplacian or Gaussian distribution. For a reasonable trade-off between privacy and accuracy, the scale is proposed to be sensitivity divided by ϵ. As the strong requirements of (ϵ)-DP are often inapplicable in practice due to the high amount of noise added to the calculation, (ϵ, δ)-DP introduced an additional privacy parameter δ. The parameter allows softening the ϵ-privacy by representing a probability that the data of an individual will still be revealed after DP. Consequently, it should be set to a very low value to maintain a high level of privacy while still improving the accuracy of the analysis.

2.3.2 Homomorphic Encryption

In contrast to the noise-adding mechanism in DP, HE enables calculations on encrypted data directly without decrypting it. The first fully homomorphic encryption (FHE) scheme was introduced by Gentry in 2009 [71]. Fully means that the HE scheme allows any type of computation on encrypted data. However, it comes with high computational complexity. Only recently, practical tools and frameworks for FHE emerged and are used in practice, such as Microsoft's SEAL library [72] or Google's FHE compiler [73]. Still, FHE requires high computational resources and special hardware to be practically usable. A more practical solution is partial homomorphic encryption (PHE), with the downside of only allowing for restricted calculations, such as addition or multiplication. One example of PHE is the popular RSE algorithm from 1978 [74].

With HE, it is now possible to send a client's encrypted data to a server. The server performs the calculation directly on the encrypted data and broadcasts the result to the clients. After decrypting the result, the clients obtain the global result in clear text. As only encrypted data leaves the client's site, there is no risk of leaking any local privacy here.

2.3.3 Secure Multi-Party Computation

SMPC aims to perform multi-party computations securely without revealing the local data of the participants, based on an underlying secret sharing protocol [75]. Shamir introduced the first efficient secret-sharing scheme in 1979 [76]. Shamir's secret sharing is a (k, n)-threshold scheme, where only k out of n shares are needed to recover the secret. This threshold also makes it practical for numerous participants, as the number of shares to be created does not increase exponentially anymore.

A still widespread approach for a low number of participants is linear secret-sharing schemes, such as additive secret sharing, based on an (n, n)-threshold [77]. Here, all n shares are required to reconstruct the secret. The secret of each participant is split into n shares, so the sum of all n shares equals the

secret. The sum of the shares of all participants reveals the actual sum of all clients. The same is also possible with multiplication [78].

2.4 Hybrid Federated Learning

As FL alone is not necessarily privacy-preserving, it must be combined with one or more PETs to increase privacy and security. This combination of FL and PETs is also known as hybrid FL. The properties of the different hybrid approaches are shown and compared in Figure 2.6, as proposed in earlier work already [66].

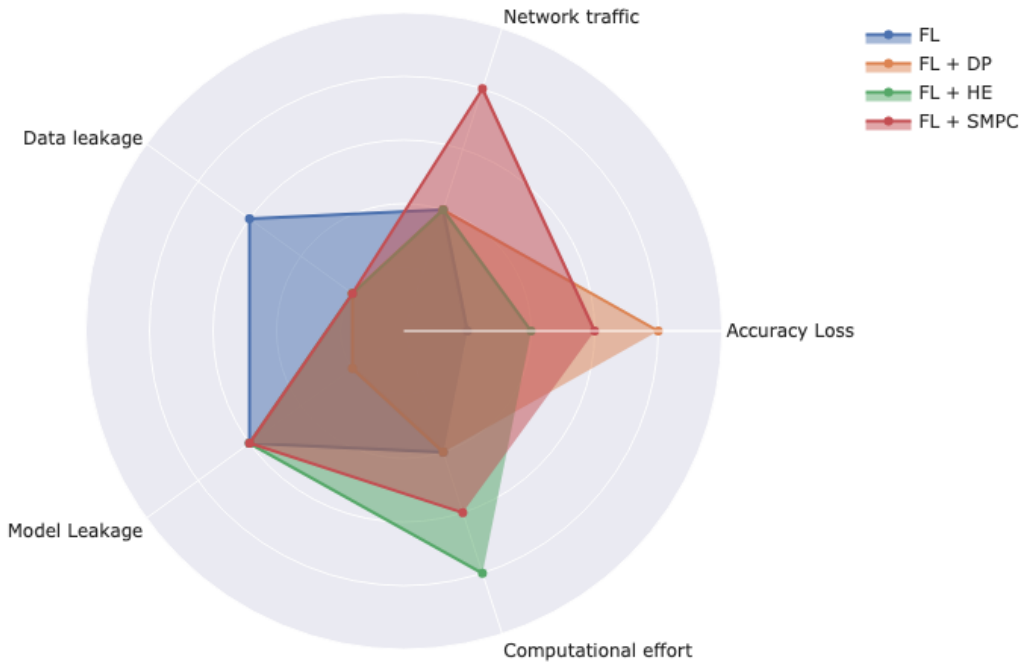


Figure 2.6: Hybrid approaches in comparison. To reduce data leakage, FL (blue) can be combined with various PETs, such as DP (yellow), HE (green), and SMPC (red). The different combinations differ in network traffic, accuracy loss, computational effort and model leakage.

FL without any additional PET (blue) has a higher data leakage than hybrid approaches. Even though only insensitive model parameters are exchanged, these might potentially leak sensitive information about the local model and local data. A combination of FL with DP, HE, or SMPC can reduce the leakage of local data. The potential data leakage by the final model is decreased by DP, as this is the only approach where noise is used to disturb the actual model

parameters. However, this also leads to the highest accuracy loss for FL + DP. FL + HE needs the highest computational effort, while FL + SMPC has the highest network traffic due to the exchange of shards between the clients. Both come with slight accuracy loss, in contrast to only FL due to floating-point precision.

It can be seen, that FL should be combined with PETs to avoid data leakage and keep the local data private. Which PET to choose depends mainly on the application case, as all hybrid approaches have some advantages and disadvantages against others. In a recent publication by Soykan et al., the authors conducted a survey on PETs for collaborative ML and concluded with a guideline to ease the choice [68]. The guideline considers two main parts: requirements and constraints. The requirements of the collaborative ML setup encourage thinking about the collaboration model, ML algorithm, data distribution, threat models, and domain needs. The constraints should be evaluated concerning communication rounds, transmitted data size, computation capabilities, accuracy, computation environment, and the number of device dropouts. For example, if only a few hospitals with low sample sizes participate in a collaboration study, DP might not be the PET of choice. For medical studies, accuracy is crucial, and for low sample sizes and few participants, the amount of noise to keep the data private might result in an impracticable model. Instead, an SMPC approach might be preferred, as with only a few participants, the communication overhead is still acceptable with only a few participants.

2.5 State-of-the-Art and Open Challenges

While the list of generic FL frameworks is quite long, the number of tools for federated time-to-event analysis is rather low. The special need to consider censored data samples in the analysis makes general FL algorithms not simply applicable to this kind of data.

2.5.1 Web Service for Distributed Cox Model Learning

One of the first algorithms for a distributed Cox model was proposed with WebDISCO in 2015 [30]. Its pseudocode is shown in Algorithm 4. A major advantage of WebDISCO is that it produces nearly identical solutions compared to centralized analysis due to its iterative update of the weights. To achieve that, each client calculates its local statistics using the current global $\boldsymbol{\beta}$ and sends them to the global server. The global server aggregates these statistics to update $\boldsymbol{\beta}$ by calculating the first and second derivatives of the likelihood function and broadcasting them to the clients. This is repeated until convergence occurs.

WebDISCO was implemented as a web service, and therefore the authors took care of deployment. However, the web service is not available anymore, and also the code of the application can no longer be found. Therefore, the algorithm is only of theoretical value but cannot be easily used in biomedical environments. Another issue is that they do not use any additional PETs to secure the exchanged local statistics and avoid any leakage of private information.

Algorithm 4: WebDISCO [30]

Local initialization for all sites: Each site initializes index subsets R_i^k and D_i^k based on their local data. Each site sends the aggregated statistics $\sum_{i=1}^{D} \sum_{l \in D_i^k} z_r^l$ to the global server to avoid additional communication overhead, as this value is unchanged during the whole learning process.

Global initialization: The global server requests distinct event times from each site to initialize the parameters D and $|D_i^k|$. Additionally, the global server aggregates the incoming statistics from all sites as $\hat{z}_r = \sum_{k=1}^{M} \sum_{i=1}^{D} \sum_{l \in D_i^K} z_r^l$. The server initializes $\boldsymbol{\beta}^0$ and broadcasts it to each size

while *parameters not converged* **do**

  for *all sites (parallel update)* **do**

    Receive an updated $\boldsymbol{\beta}^\tau$ from the global server

    Calculate the following aggregated statistics: $\sum_{l \in R_i^k} \exp(\boldsymbol{\beta}^T \mathbf{z}^l)$, $\sum_{l \in R_i^k} z_q^l \exp(\boldsymbol{\beta}^T \mathbf{z}^l)$ and $\sum_{l \in R_i^k} z_r^l z_q^l \exp(\boldsymbol{\beta}^T \mathbf{z}^l)$

    Send these statistics back to the global server

  end

  Calculate the first and second derivatives of the likelihood function using the statistics received from each site

  Update $\boldsymbol{\beta}^{\tau+1}$ using the Newton-Raphson algorithm and send it back to each site

end

Send converged model parameters to each site.

2.5.2 One-Shot Distributed Algorithm for Cox Model

Another recent approach for a distributed Cox model was proposed recently, which is called "One-shot distributed algorithm for Cox model (ODAC)" [79]. Compared to an iterative calculation, its one-shot approach improves runtime and communication (Algorithm 5) overhead drastically while still outperforming meta-analysis approaches.

Even though most of the examples in the publication achieved almost identical results compared to centralized analysis, some of them did not, which might lead to problems in some heterogeneous datasets. Moreover, ODAC does

not implement any PETs to reduce privacy leakage of the exchanged models. Therefore, exchanged model parameters might leak sensitive information.

Algorithm 5: ODAC [79]

(1) Initialization:
for *all sites (parallel update)* **do**
  Fit a Cox model and obtain the local estimate $\hat{\beta}_k$ and the variance estimate $\hat{V}_k$; **broadcast** $\hat{\beta}_k, \hat{V}_k$, and the set of unique event time points in site k.
end
(2) Local surrogate estimator:
for *all sites (parallel update)* **do**
  obtain $\tilde{\beta}$ and all the unique event time points across all sites $t_1...t_d$; calculate and broadcast the intermediate summary-level statistics $U_j(T)$, $W_j(T)$ and $Z_j(T)$; construct the surrogate likelihood $\tilde{L}_k(\beta)$ by treating the k-th site as the local site; obtain and broadcast $\tilde{\beta}_k$ and the variance $\tilde{V}_k$
end
(3) Evidence synthesis:
Obtain $\tilde{\beta}$ by plugging in $\tilde{\beta}$ and $\tilde{V}_k$
return $\tilde{\beta}$

2.5.3 Federated Analytics Multiparty Homomorphic Encryption

This federated approach is based on multi-party homomorphic encryption that enables the computation and encryption of local patient-level statistics and the homomorphic aggregation of these results. The FAMHE version of the Kaplan-Meier estimator achieved identical survival curves compared to centralized computation, with still good runtime and scalability. Remembering the formula of the Kaplan-Meier estimator in Equation 2.1, each client creates a vector with the number of censored event times c_j, the number of events d_j and the numbers of individuals at risk n_j at the time t_j for $t_j = 0, ..., T$. These vectors are encrypted and collectively aggregated, obtaining the exact survival curve.

This sophisticated approach introduces PETs to the federated computation and therefore ensures the privacy of the exchanged data. However, it does not consider that the resulting survival curve might still reveal patient-level data [80, 81]. The code is not publicly available as its license does not allow for open-source distribution, making it unusable and inaccessible for general FL in biomedical environments.

2.5.4 Survival Models for Federated Individual Patient Meta-Analysis in DataSHIELD

DataSHIELD is an early approach and framework from 2014 that aims to facilitate distributed research settings [82]. Recently, an approach for Cox models has also been proposed, called dsSurvival. The dsSurvival framework is more of a meta-analysis approach than an actual FL. A complete Cox model is built at each site, and the model statistics are sent to a global server. An interesting approach of dsSurvival is that these statistics are stored on the global server and can be used again in future meta-analyses. Another advantage of the approach is that they provide an R package and open-source code. However, the approach does not lead to results comparable to centralized analysis, does not implement univariate approaches, and has no UI.

2.5.5 Open Challenges

A closer look at the state-of-the-art approaches in federated time-to-event analysis, listed in Table 2.1, shows that there is already some research in the field. There are various approaches with different techniques, from federated meta-analysis to federated homomorphic encryption or distributed Cox regression as a web server. However, none of the tools actually provides a complete solution.

Table 2.1: Existing solutions for privacy-preserving time-to-event analysis. ACC=accuracy, FL=Federated Learning, PET=privacy-enhancing technology, KM=Kaplan-Meier estimator, NA=Nelson-Aalen estimator, LRT=log-rank test, COX=Cox model, OS=open source, GUI=graphical user interface

Method	ACC	FL	PET	KM	NA	LRT	COX	OS	GUI
WebDISCO [30]	★★★	✓	✗	✗	✗	✗	✓	✗	✗
ODAC [79]	★★☆	✓	✗	✗	✗	✗	✓	✓	✗
FAMHE [31]	★★★	✓	✓	✓	✗	✗	✗	✗	✗
AusCAT [32]	★★★	✓	✗	✓	✗	✗	✓	✓	✓
dsSurvival [83]	★☆☆	✓	✗	✗	✗	✗	✓	✓	✗

The combination of FL and PETs is highly important, as even local models can be considered as personal data in the GDPR. Unfortunately, FAMHE, the only solution that uses PETs, does not provide any code or deployed version and is therefore not directly usable by researchers and clinicians. None of the tools covers all the most widely used time-to-event algorithms (Kaplan-Meier estimator, Nelson-Aalen estimator, log-rank test, and Cox proportional hazard model) but rather concentrates on one specific algorithm. This is impractical, as in clinical time-to-event studies, there is often a need to apply multiple

approaches to enhance the interpretation of the results. In the current state-of-the-art, researchers would need to switch between different tools. Moreover, except for WebDISCO and AuSCAT, no UIs are provided for better usability and accessibility of the algorithms. Most of the tools leave the responsibility of deployment to the researchers or their IT departments. This increases the hurdles to using this technique, potentially leading researchers to just stick to the centralized methods they are used to already.

2.6 Objectives and Contributions

As shown in this section, there is an actual lack of usable, accessible, and privacy-preserving FL solutions for time-to-event analysis. This dissertation aims to fill this gap.

The main objective of this dissertation is to increase the applicability and accessibility of FL in biomedical research and bring privacy-aware algorithms to researchers in biomedicine. Due to its high importance and occurrence in clinical studies, and the lack of accessible and privacy-aware methods, time-to-event analysis was chosen as the main algorithmic field for this dissertation. In the following chapter 3, three publications are presented that aim to fulfill critical parts of the objectives.

In the first publication, a tool for multi-institutional time-to-event analysis is proposed, called Partea, providing a complete, accessible, and privacy-aware all-in-one tool for clinical studies and other areas. Here, the author of this dissertation was the first author and responsible for the development, analysis, evaluation, and writing.

The second publication is based on the promising results of Partea and other tools that were developed recently, such as sPLINK [25] or Flimma [26]. FeatureCloud, a unified platform for FL in biomedicine, is presented, which eases both the development and deployment of all kinds of federated algorithms as well as their execution in clinical infrastructures. In this publication, first authorship was shared between the author of this dissertation and Julian Matschinske, as the development and evaluation of this platform were a gigantic effort. The leading contribution of the author of this dissertation was the development of the interactive web interface, the AI store for FL applications and implementation of several apps, and the accuracy evaluation of the algorithms compared to their central machine learning algorithms.

The third publication presents the first available federated survival SVM, which was entirely developed using FeatureCloud and is accessible through the platform. In this publication, the author of this dissertation is the first author and was responsible for the algorithm elaboration and evaluation, app development, as well as the evaluation and writing.

3. Publications

3.1 Publication 1: Partea

The article **Privacy-aware multi-institutional time-to-event analysis** was published online at **PLOS Digital Health** on September 6, 2022. The full publication is available in Appendix A.1

Citation

Julian Späth, Julian Matschinske, Frederick K. Kamanu, Sabina A. Murphy, Olga Zolotareva, Mohammad Bakhtiari, Elliott M. Antman, Joseph Loscalzo, Alissa Brauneck, Louisa Schmalhorst, Gabriele Buchholtz, and Jan Baumbach. Privacy-aware multi-institutional time-to-event studies. PLOS Digital Health, 1(9):1–16, 09 2022. doi: `https://doi.org/10.1371/journal.pdig.0000101`

Summary

Time-to-event analysis is an essential tool in clinical studies for analyzing censored data, in which an event of interest is not always observed during observation time. Collecting large amounts of data for these clinical studies is necessary to improve the quality of the study, and often requires the collaboration of multiple institutions. As strict data regulation rules complicate study collaborations, new machine learning technologies, such as federated learning have shown high potential for enabling privacy-aware and accurate decentralized analyses across institutions. However, especially for time-to-event analysis, federated implementations do not exist for common algorithms or are not available for the research community.

To address these issues, the authors developed the intuitive web app Partea that offers an intuitive environment for the federated execution of the most popular time-to-event methods: survival function, hazard function, log-rank test, and Cox model. Our evaluations on several benchmark datasets and a real-world dataset from a previous clinical study show that all federated implementations achieve highly similar, or even identical results as the centralized methods.

With the support of privacy-enhancing technologies, the exchanged local model parameters are encrypted and not even visible to the global aggregator. Data leakage from published survival curves is hindered by adding random noise to the calculation, using differential privacy. All algorithms can be accessed using graphical user interfaces, removing expertise in federated infrastructure and programming knowledge. Therefore, as the first tool of its kind, Partea accelerates the accessibility of privacy-aware time-to-event analysis.

Availability

The entire source code of the Partea platform is available on GitHub (`https://github.com/federated-partea`). The lung [84], veteran [85], and colon [86] benchmark datasets are available through the R survival package. The rossi [87] benchmark dataset is available via the *lifelines* Python package. Data from the ENGAGE-TIMI 48 Trial is not publicly available but is discussed in detail in the original publication of Giugliano et al. [88].

Author contribution

The author of this thesis conceptualized this work and was mainly responsible for the content. He developed and implemented the federated algorithms and evaluated them on the benchmark datasets. With the help of Julian Matschinske, the author implemented the web platform and deployed the tools. Olga Zolotareva and Mohammad Bakhtiari provided constructive feedback in the context of federated learning and supported writing the manuscript. Frederick K. Kamanu and Sabina A. Murphy performed the federated and centralized analysis on the ENGAGE-TIMI 48 trial data. Elliott M. Antman and Joseph Loscalzo provided medical expertise to make the platform usable in clinical environments and supported writing the manuscript. Alissa Brauneck, Louisa Schmalhorst, and Gabriele Buchholtz supported the work from a legal perspective and performed in the GDPR evaluation. Prof. Dr. Jan Baumbach supervised the project, supported the writing of the manuscript, and provided valuable feedback. The author wrote the main parts of the manuscript, finalized it, and produced the figures.

Rights and permissions

Supplementary material

Supplementary data are available online at PLOS Digital Health https://journals.plos.org/digitalhealth/article?id=10.1371/journal.pdig.0000101#sec014

3.2 Publication 2: FeatureCloud

The article **The FeatureCloud Platform for Federated Learning in Biomedicine: A Unified Approach** was published online at the **Journal of Medical Internet Research (JMIR)** on July 12th, 2023. The full publication is available in Appendix A.2

Citation

Julian Matschinske, Julian Späth, Mohammad Bakhtiari, Niklas Probul, Mohammad Mahdi Kazemi Majdabadi, Reza Nasirigerdeh, Reihaneh Torkzadehmahani, Anne Hartebrodt, Balazs-Attila Orban, Sándor-József Fejér, Olga Zolotareva, Supratim Das, Linda Baumbach, Josch K Pauling, Olivera Tomašević, Béla Bihari, Marcus Bloice, Nina C Donner, Walid Fdhila, Tobias Frisch, Anne-Christin Hauschild, Dominik Heider, Andreas Holzinger, Walter Hötzendorfer, Jan Hospes, Tim Kacprowski, Markus Kastelitz, Markus List, Rudolf Mayer, Mónika Moga, Heimo Müller, Anastasia Pustozerova, Richard Röttger, Christina C Saak, Anna Saranti, Harald H H W Schmidt, Christof Tschohl, Nina K Wenke, and Jan Baumbach. The featurecloud platform for federated learning in biomedicine: Unified approach. J Med Internet Res, 25:e42621, Jul 2023. doi: `https://doi.org/10.2196/42621`

Summary

While vast amounts of data are gathered in healthcare nowadays, this data is usually distributed among different institutions and thus not usable for machine learning (ML) algorithms, requiring centralized datasets. Sharing highly sensitive patient data is still difficult due to privacy regulations, such as the GDPR in the EU or HIPAA in the US, new technologies were developed recently to address this problem. Federated Learning (FL) is a solution that enables the training of distributed ML models without sharing confidential data itself, but rather exchanging locally trained model parameters with a global aggregator. Despite advancements, FL implementation remains complex and time-consuming, requiring advanced technical skills and complex infrastructures. While there are many frameworks for centralized ML, there is a gap in practical solutions for FL that focus on both developing and executing federated algorithms.

To address this issue, the authors developed FeatureCloud, an all-in-one FL platform for biomedicine and beyond. The platform addresses two audiences: developers and researchers. Our platform enables a straightforward way for developers to implement FL algorithms. Using the platform's application programming interface, developers do not need to worry about infrastructure,

execution, and deployment of the algorithms but rather about the federated method itself. After implementation, developers can publish their method as an app in the AI Store, making it available for researchers to use. Researchers can then compose workflows out of various apps and run their ML pipeline on data distributed among different institutions that they can invite to the computation.

To increase the privacy of the exchanged data, FeatureCloud incorporates privacy-enhancing technology (PET), such as secure aggregation, to secure the locally exchanged model parameters. Further, the implementation and infrastructure of the platform are explained in detail. The authors show, that for several apps, highly similar results to centralized models are achieved. Further, the results are more generalizable and accurate compared to institutions that train models solely on their data.

Availability

The Survey of Health, Aging and Retirement in Europe (SHARE) data are distributed by SHARE-European Research Infrastructure Consortium (ERIC) to registered users through the SHARE Research Data Center. We used only data from the 8 waves [89]. Except for the SHARE data, all our data sets, including the Indian Liver Patient Dataset [90], Breast Invasive Carcinoma data set [91], Boston data set [92], and Diabetes data set [93], and scripts used for our evaluation results are available in our GitHub repository (`https://github.com/FeatureCloud/evaluation`). To increase interpretation and reproducibility, we followed the minimum information about clinical artificial intelligence modeling (ML-CLAIM) reporting standard (Norgeot et al. [94]). The filled-out ML-CLAIM clinical checklist is also available in our GitHub repository.

Author contribution

As this publication was the major publication of the FeatureCloud consortium, many authors contributed to this work. The main work in developing the FeatureCloud platform, the AI store, and the manuscript was done by Julian Matschinske and the author of this thesis. In detail, the author of this thesis was mainly responsible for the frontend and backend development of the AI store and many federated learning apps such as linear and logistic regression, various time-to-event analysis apps (Kaplan-Meier Estimator, Nelson-Aalen Estimator, Random Survival Forest, and Survival SVM), and useful apps for federated machine learning pipelines (Cross-validation, Evaluation). Further, the author of this thesis equally contributed as a first author to the publication, being primarily involved in the writing, the comparison to existing work, and the evaluation of the accuracy for different datasets and algorithms. The other

first author, Julian Matschinske, mainly contributed to writing the publication, explaining the methodology, infrastructure, and functions of the platform, and evaluating the network traffic and runtime.

Rights and permissions

Additional supplementary material

Supplementary data are available online at JMIR `https://www.jmir.org/2023/1/e42621#app1`

3.3 Publication 3: Federated Survival SVM

The article **Privacy-Preserving Federated Survival Support Vector Machines for Cross-Institutional Time-To-Event Analysis: Algorithm Development and Validation** was published online at **JMIR AI (JAI)** on March 29th, 2024. The full publication is available in Appendix A.3

Citation

Julian Späth, Zeno Sewald, Niklas Probul, Magali Berland, Mathieu Almeida, Nicolas Pons, Emmanuelle Le Chatelier, Pere Ginès, Cristina Solé, Adrià Juanola, Josch Pauling, and Jan Baumbach. Privacy-preserving federated survival support vector machines for cross-institutional time-to-event analysis: Algorithm development and validation. JMIR AI, 3:e47652, Mar 2024. doi: https://doi.org/10.2196/47652

Summary

Federated Learning has become an efficient tool to enable multi-institutional data analysis without sharing the actual raw data. However, many algorithms are still not available and accessible for federated learning in clinical analysis. In this publication, the authors aimed to develop and validate a privacy-preserving and federated survival SVM algorithm that is easily applicable and accessible. The authors extended the centralized survival SVM algorithm to work in a federated environment and implemented it as an app in the FeatureCloud platform, which is freely and openly available in the FeatureCloud AI store. Finally, the authors evaluated the algorithm on three benchmark datasets, one real-world microbiome dataset, and a synthetic large sample size dataset for various number of clients and compared it to the centralized method. The authors could show that the federated version is highly similar to the centralized version for all datasets and scenarios, with only minimal differences in c-index and model weights of the final model. Furthermore, they showed the high importance of including more sites in the analysis to increase sample size.

Availability

The data sets generated and analyzed during this study are available in the GitHub repository (`https://github.com/julianspaeth/federated-survival-svm`). The code for the implementation of the federated survival SVM is available in the GitHub repository (`https://github.com/FeatureCloud/fc-survival-svm`). The microbiome data set is not publicly available due to privacy regulations but is available from the corresponding author on reasonable request.

Author contribution

The author of this thesis conceptualized this work and was mainly responsible for the content. He developed and implemented the federated survival SVM algorithm and FeatureCloud app together with Zeno Sewald. The writing was mainly performed by the author of this thesis, with the help of Niklas Probul. Preprocessing of the microbiome dataset, as well as explaining the methodology of the dataset, was performed by Magali Berland, Mathieu Almeida, Nicolas Pons, and Emmanuelle Chatelier. Pere Ginès, Cristina Solé, and Adrià Juanola were responsible for the generation and provision of the microbiome dataset. Josch Pauling and Jan Baumbach supervised the work and provided valuable feedback on the conception and development of the work, as well as support in writing the publication.

Rights and permissions

Additional supplementary material

Supplementary data are available online at JMIR AI `https://ai.jmir.org/2024/1/e47652`

4. Discussion and Outlook

The large amount of biomedical data generated by the digitalization of healthcare, as well as cheaper and faster sequencing technologies, have drastically accelerated biomedical research in the last decade. Especially, the application of ML algorithms enabled the extraction of knowledge from large amounts of biomedical data and the identification of novel biomarkers or therapies. This valuable biomedical data is now available and used for data analysis and research. Although this biomedical is now being collected systematically, it often highly sensitive regarding privacy and scattered across different institutions. This makes the data difficult to access, as it cannot be easily shared with other institutions, nor easily collected centrally due to privacy regulations worldwide. In time-to-event analysis, which is a common evaluation tool in clinical studies, data is frequently collected from different institutions to increase sample size (multicenter studies). Collecting this data at a central institution to enable ML and other data analyses comes with high bureaucratic efforts. For a long time, ML research solely focused on analyzing centrally collected data. This focus shifted in 2017 with the introduction of FL as a novel concept for applying ML on geographically distributed datasets [14]. Since then, lots of research has been performed to address the open challenges of FL, such as communication efficiency or accuracy [64, 95].

Unfortunately, most of the existing FL solutions are still not accessible to researchers without profound knowledge regarding federated infrastructure or deployment. This is a massive hurdle, especially for clinicians and biomedical researchers without proper training in informatics and programming. Another issue is that most of the existing FL tools solely focused on DL algorithms. While DL is certainly one of the most popular ML approaches today, it is not the best choice for each problem. For example, it is not appropriate for datasets with low sample sizes, due to factors like overfitting or sampling bias [96]. However, a small sample size is often the standard in biomedical research or clinical studies. Datasets with hundreds of thousands of samples are the exception, even when applying FL to make use of the data from multiple data centers. Another issue is the fact that DL requires state-of-the-art GPU hardware, which is usually not available in hospitals or small research labs. This is one reason why in biomedical or clinical research, other ML methods

are sometimes preferred over DL. Methods such as RF, SVM, and differential expression analysis, or GWAS for the analysis of OMICS data, as well as time-to-event analysis algorithms for clinical studies, are broadly used and work efficiently even without GPU hardware. A considerable gap in FL research is therefore the focus on various biomedical and clinical algorithms such as time-to-event analysis. In the past, only a few theoretical FL approaches were developed, and practical, usable solutions do still not exist.

The purpose of this dissertation is to fill this gap and make FL available for clinicians and researchers working in the biomedical field. To achieve this, several commonly used ML and time-to-event analysis algorithms have been extended to be applicable in a federated environment. Additionally, an easy-to-use infrastructure has been provided to make them available and accessible for the research community. A high focus in the implementation of the algorithms was on protecting the privacy of the potentially sensitive data. Therefore, the results of this dissertation support additional PETs to securely exchange model parameters using additive secret sharing or hiding the sensitive results of survival curves using DP.

The performed research was published in multiple publications, of which three are part of this dissertation. In publication 1 (section 3.1), a platform for privacy-aware time-to-event analysis (Partea) has been implemented. The platform incorporates various federated implementations of the most commonly used time-to-event algorithms (survival curves, hazard rate, log-rank test, and Cox proportional hazard model), and makes them available in an intuitive user interface. In publication 2 (section 3.2), the FeatureCloud platform for FL in biomedicine has been introduced, which enables the development of FL methods without taking care of the necessary infrastructure. The integrated AI store takes care of the deployment of the developed methods, making them accessible for the actual users for running FL workflows. In publication 3 (section 3.3), the FeatureCloud platform has been used to implement and evaluate a federated survival SVM. All federated algorithms implemented in the three publications were evaluated in depth on various datasets and different client scenarios. They achieved highly similar or in some cases even identical results to their corresponding centralized approach.

The results of this dissertation extend the current research field of FL, dominated by DL and theoretical approaches, by introducing new platforms, tools, and algorithms for federated clinical time-to-event analysis and biomedicine. Further, the results of this dissertation show that many algorithms that were not yet considered in FL research produce accurate results while maintaining a high level of privacy. The developed algorithms are not just introduced on a theoretical and mathematical level, but are accessible through intuitive web apps without programming knowledge by using GUIs. For this reason, they are available for clinicians and researchers without profound knowledge about

FL infrastructure and programming. As shown in Table 4.1, the developed methods in this dissertation extend the current state-of-the-art (as previously described in section 2.5) by combining various algorithms in one tool, providing open-source code for maintainability and expendability, as well as improving accessibility through GUIs.

Table 4.1: Developed solutions of this dissertation compared to the existing solutions in federated time-to-event analysis. ACC=accuracy, FL=Federated Learning, PET=privacy-enhancing technology, KM=Kaplan-Meier estimator, NA=Nelson-Aalen estimator, LRT=log-rank test, COX=Cox model, OS=open source, GUI=graphical user interface, *Accuracy depends on the implementation

Method	ACC	FL	PET	KM	NA	LRT	COX	OS	GUI
WebDISCO [30]	★★★	✓	✗	✗	✗	✗	✓	✗	✗
ODAC [79]	★★☆	✓	✗	✗	✗	✗	✓	✓	✗
FAMHE [31]	★★★	✓	✓	✓	✗	✗	✗	✗	✗
AusCAT [32]	★★★	✓	✗	✓	✗	✗	✓	✓	✓
dsSurvival [83]	★☆☆	✓	✗	✗	✗	✗	✓	✓	✗
Partea [34]	★★★	✓	✓	✓	✓	✓	✓	✓	✓
FeatureCloud [35]	★★★*	✓	✓	✓	✓	✓	✓	✓	✓

Moreover, the results of the dissertation, indicate that centralized data collection in time-to-event analysis or biomedical data analysis is not necessarily needed to obtain accurate results. FL can be a serious alternative to keeping control over sensitive institutional data, such as patient data, but still collaborating with other institutions or providing this data for analyses and studies.

Besides the achievements, there are still limitations that could not be addressed in this dissertation. One limitation that was not specifically addressed is communication efficiency. The hybrid approaches used in this dissertation are based on consists of a combination of FL and SMPC. As currently an additive secret sharing scheme is used, communication efficiency could be enhanced by replacing it with a more efficient scheme, such as Shamir's secret sharing.

Another problem that still exists is data harmonization between different institutions. FL in general expects datasets in a common structure across different participants. This might affect the names of the features, the order of the features, or the metric, in which a certain feature is documented (e.g., height of a patient in meters or centimeters). For analyses based on raw data, such as sequencing data, preprocessing could be part of the FL pipeline already. As an example, a common preprocessing could be implemented as an app in FeatureCloud to harmonize and standardize the input data for the FL algorithm. However, in most cases the participating institutions need to make sure that

their datasets are harmonized. A possibility to achieve this is to define data standards before the analysis starts and communicate them to all participants. For studies, this is often declared in a data collection strategy. Clearly, future research should focus on simplifying this problem to ensure compatible data across the participants. A related issue is fairness in FL, where a trained FL might be biased towards the dataset of a specific participating institution. This was neither a focus of this dissertation, but is an important topic that is currently investigated in the research community and should be further in the future [97, 98, 99].

Finally, a critical focus in the future should be the privacy of the patient data and the accordance of the methods to privacy regulations. Currently, there are still no clear regulatory requirements for FL. As the exchanged, local models trained on personal data are themselves personal data according to GDPR, the combination with PETs is required [100]. Although the results of this dissertation use PETs already, further research should focus more on the concordance with regulatory requirements and to create a better picture of which measures are appropriate to satisfy the requirements.

As shown in this dissertation, privacy-aware FL has the potential to change the way data analysis and machine learning is performed in clinical (time-to-event) studies and research in the future. Each institution can keep control over their sensitive data, and studies and analyses can be performed collaboratively with other institutions without raw data sharing. This can possibly accelerate the access to larger sample sizes and more diverse datasets, especially when considering privacy regulations. One application, where this could play a major role in the future, is the research of rare diseases. Patients might be scattered across the whole world, and obtaining a large enough sample size is challenging even when considering the data of a whole country.

Bibliography

[1] Kris A Wetterstrand. DNA sequencing costs: Data from the NHGRI genome sequencing program (GSP). `www.genome.gov/sequencingcostsdata`, March 2019. Accessed: 2023-1-24.

[2] Nishu Dalal, Rekha Jalandra, Minakshi Sharma, Hridayesh Prakash, Govind K Makharia, Pratima R Solanki, Rajeev Singh, and Anil Kumar. Omics technologies for improved diagnosis and treatment of colorectal cancer: Technical advancement and major perspectives. *Biomedicine & Pharmacotherapy*, 131:110648, 2020.

[3] Christine M Micheel, Sharly J Nass, Gilbert S Omenn, Board on Health Care Services, Board on Health Sciences Policy, and Institute of Medicine. *Omics-Based Clinical Discovery: Science, Technology, and Applications*. National Academies Press (US), March 2012.

[4] Adam Bohr and Kaveh Memarzadeh. The rise of artificial intelligence in healthcare applications. *Artificial Intelligence in Healthcare*, page 25, 2020.

[5] David G Kleinbaum and Mitchel Klein. *Survival Analysis: A Self-Learning Text, Third Edition*. Springer New York, August 2011.

[6] Jaclyn M Beca, Kelvin KW Chan, David MJ Naimark, and Petros Pechlivanoglou. Impact of limited sample size and follow-up on single event survival extrapolation for health technology assessment: a simulation study. *BMC Medical Research Methodology*, 21(1):1–12, 2021.

[7] Junjun Zhang, Rosita Bajari, Dusan Andric, Francois Gerthoffert, Alexandru Lepsa, Hardeep Nahal-Bose, Lincoln D Stein, and Vincent Ferretti. The international cancer genome consortium data portal. *Nature biotechnology*, 37(4):367–369, 2019.

[8] Robert L Grossman, Allison P Heath, Vincent Ferretti, Harold E Varmus, Douglas R Lowy, Warren A Kibbe, and Louis M Staudt. Toward a shared vision for cancer genomic data. *New England Journal of Medicine*, 375(12):1109–1112, 2016.

[9] Elliott Antman. Data sharing in research: benefits and risks for clinicians. *BMJ*, 348, 2014.

[10] Luc Rocher, Julien M Hendrickx, and Yves-Alexandre De Montjoye. Estimating the success of re-identifications in incomplete datasets using generative models. *Nature communications*, 10(1):1–9, 2019.

[11] W Nicholson Price and I Glenn Cohen. Privacy in the age of medical big data. *Nature medicine*, 25(1):37–43, 2019.

[12] Becky McCall. What does the gdpr mean for the medical community? *The Lancet*, 391(10127):1249–1250, 2018.

[13] Sarah Batson, Gemma Greenall, and Pollyanna Hudson. Review of the reporting of survival analyses within randomised controlled trials and the implications for meta-analysis. *PLoS One*, 11(5):e0154870, 2016.

[14] Brendan McMahan and Daniel Ramage. Federated learning: Collaborative machine learning without centralized training data. *Google Research Blog*, 3, 2017.

[15] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data. In Aarti Singh and Jerry Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 1273–1282. PMLR, 2017.

[16] Micah J Sheller, Brandon Edwards, G Anthony Reina, Jason Martin, Sarthak Pati, Aikaterini Kotrotsou, Mikhail Milchenko, Weilin Xu, Daniel Marcus, Rivka R Colen, et al. Federated learning in medicine: facilitating multi-institutional collaborations without sharing patient data. *Scientific reports*, 10(1):1–12, 2020.

[17] Theodora S Brisimi, Ruidi Chen, Theofanie Mela, Alex Olshevsky, Ioannis Ch Paschalidis, and Wei Shi. Federated learning of predictive models from federated electronic health records. *International journal of medical informatics*, 112:59–67, 2018.

[18] Songtao Lu, Yawen Zhang, and Yunlong Wang. Decentralized federated learning for electronic health records. In *2020 54th Annual Conference on Information Sciences and Systems (CISS)*, pages 1–5. IEEE, 2020.

[19] Trung Kien Dang, Xiang Lan, Jianshu Weng, and Mengling Feng. Federated learning for electronic health records. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 13(5):1–17, 2022.

[20] Ittai Dayan, Holger R Roth, Aoxiao Zhong, Ahmed Harouni, Amilcare Gentili, Anas Z Abidin, Andrew Liu, Anthony Beardsworth Costa, Bradford J Wood, Chien-Sung Tsai, et al. Federated learning for predicting clinical outcomes in patients with covid-19. *Nature medicine*, 27(10):1735–1743, 2021.

[21] Adnan Qayyum, Kashif Ahmad, Muhammad Ahtazaz Ahsan, Ala Al-Fuqaha, and Junaid Qadir. Collaborative federated learning for healthcare: Multi-modal covid-19 diagnosis at the edge. *IEEE Open Journal of the Computer Society*, 3:172–184, 2022. doi:10.1109/OJCS.2022.3206407.

[22] Sadaf Naz, Khoa T Phan, and Yi-Ping Phoebe Chen. A comprehensive review of federated learning for covid-19 detection. *International Journal of Intelligent Systems*, 37(3):2371–2392, 2022.

[23] Scott D Constable, Yuzhe Tang, Shuang Wang, Xiaoqian Jiang, and Steve Chapin. Privacy-preserving gwas analysis on federated genomic datasets. In *BMC medical informatics and decision making*, volume 15 (Suppl 5), pages 1–9. BioMed Central, 2015.

[24] Md Nazmus Sadat, Md Momin Al Aziz, Noman Mohammed, Feng Chen, Xiaoqian Jiang, and Shuang Wang. Safety: secure gwas in federated environment through a hybrid solution. *IEEE/ACM transactions on computational biology and bioinformatics*, 16(1):93–102, 2018.

[25] Reza Nasirigerdeh, Reihaneh Torkzadehmahani, Julian Matschinske, Tobias Frisch, Markus List, Julian Späth, Stefan Weiss, Uwe Völker, Esa Pitkänen, Dominik Heider, et al. splink: a hybrid federated tool as a robust alternative to meta-analysis in genome-wide association studies. *Genome Biology*, 23(1):1–24, 2022.

[26] Olga Zolotareva, Reza Nasirigerdeh, Julian Matschinske, Reihaneh Torkzadehmahani, Mohammad Bakhtiari, Tobias Frisch, Julian Späth, David B Blumenthal, Amir Abbasinejad, Paolo Tieri, et al. Flimma: a federated and privacy-aware tool for differential gene expression analysis. *Genome biology*, 22(1):1–26, 2021.

[27] Nicola Rieke, Jonny Hancox, Wenqi Li, Fausto Milletarì, Holger R Roth, Shadi Albarqouni, Spyridon Bakas, Mathieu N Galtier, Bennett A Landman, Klaus Maier-Hein, Sébastien Ourselin, Micah Sheller, Ronald M Summers, Andrew Trask, Daguang Xu, Maximilian Baust, and M Jorge Cardoso. The future of digital health with federated learning. *NPJ Digit Med*, 3:119, September 2020.

[28] Jie Xu, Benjamin S Glicksberg, Chang Su, Peter Walker, Jiang Bian, and Fei Wang. Federated learning for healthcare informatics. *Journal of Healthcare Informatics Research*, 5:1–19, 2021.

[29] Shipeng Yu, Glenn Fung, Romer Rosales, Sriram Krishnan, R Bharat Rao, Cary Dehing-Oberije, and Philippe Lambin. Privacy-preserving cox regression for survival analysis. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1034–1042, 2008.

[30] Chia-Lun Lu, Shuang Wang, Zhanglong Ji, Yuan Wu, Li Xiong, Xiaoqian Jiang, and Lucila Ohno-Machado. Webdisco: a web service for distributed cox model learning without patient-level data sharing. *Journal of the American Medical Informatics Association*, 22(6):1212–1219, 2015.

[31] David Froelicher, Juan R Troncoso-Pastoriza, Jean Louis Raisaro, Michel A Cuendet, Joao Sa Sousa, Hyunghoon Cho, Bonnie Berger, Jacques Fellay, and Jean-Pierre Hubaux. Truly privacy-preserving federated analytics for precision medicine with multiparty homomorphic encryption. *Nature communications*, 12(1):5910, 2021.

[32] Christian Rønn Hansen, Gareth Price, Matthew Field, Nis Sarup, Ruta Zukauskaite, Jørgen Johansen, Jesper Grau Eriksen, Farhannah Aly, Andrew McPartlin, Lois Holloway, et al. Larynx cancer survival model developed through open-source federated learning. *Radiotherapy and Oncology*, 176:179–186, 2022.

[33] Fei Wang and Anita Preininger. Ai in health: state of the art, challenges, and future directions. *Yearbook of medical informatics*, 28(01):016–026, 2019.

[34] Julian Späth, Julian Matschinske, Frederick K. Kamanu, Sabina A. Murphy, Olga Zolotareva, Mohammad Bakhtiari, Elliott M. Antman, Joseph Loscalzo, Alissa Brauneck, Louisa Schmalhorst, Gabriele Buchholtz, and Jan Baumbach. Privacy-aware multi-institutional time-to-event studies. *PLOS Digital Health*, 1(9):1–16, 09 2022. doi:10.1371/journal.pdig.0000101.

[35] Julian Matschinske, Julian Späth, Mohammad Bakhtiari, Niklas Probul, Mohammad Mahdi Kazemi Majdabadi, Reza Nasirigerdeh, Reihaneh Torkzadehmahani, Anne Hartebrodt, Balazs-Attila Orban, Sándor-József Fejér, Olga Zolotareva, Supratim Das, Linda Baumbach, Josch K Pauling, Olivera Tomašević, Béla Bihari, Marcus Bloice, Nina C Donner, Walid Fdhila, Tobias Frisch, Anne-Christin Hauschild, Dominik Heider, Andreas Holzinger, Walter Hötzendorfer, Jan Hospes, Tim Kacprowski,

Markus Kastelitz, Markus List, Rudolf Mayer, Mónika Moga, Heimo Müller, Anastasia Pustozerova, Richard Röttger, Christina C Saak, Anna Saranti, Harald H H W Schmidt, Christof Tschohl, Nina K Wenke, and Jan Baumbach. The featurecloud platform for federated learning in biomedicine: Unified approach. *J Med Internet Res*, 25:e42621, Jul 2023. doi:10.2196/42621.

[36] Kwan-Moon Leung, Robert M Elashoff, and Abdelmonem A Afifi. Censoring issues in survival analysis. *Annual review of public health*, 18(1):83–104, 1997.

[37] Ritesh Singh and Keshab Mukhopadhyay. Survival analysis in clinical trials: Basics and must know areas. *Perspect. Clin. Res.*, 2(4):145–148, October 2011.

[38] Christiana Kartsonaki. Survival analysis. *Diagn. Histopathol.*, 22(7):263–270, July 2016.

[39] Stephen W Lagakos. General right censoring and its impact on the analysis of survival data. *Biometrics*, pages 139–156, 1979.

[40] DC Watt, TC Aitchison, RM MacKie, and JM Sirel. Survival analysis: the importance of censored observations. *Melanoma research*, 6(5):379–385, 1996.

[41] Edward L Kaplan and Paul Meier. Nonparametric estimation from incomplete observations. *Journal of the American statistical association*, 53(282):457–481, 1958.

[42] Taane G Clark, Michael J Bradburn, Sharon B Love, and Douglas G Altman. Survival analysis part i: basic concepts and first analyses. *British journal of cancer*, 89(2):232–238, 2003.

[43] Odd Aalen. Nonparametric inference for a family of counting processes. *Ann. Stat.*, 6(4):701–726, 1978.

[44] Nathan Mantel et al. Evaluation of survival data and two new rank order statistics arising in its consideration. *Cancer Chemother Rep*, 50(3):163–170, 1966.

[45] Eric Le Bourg. Limitations of log-rank tests for analysing longevity data in biogerontology. *Biogerontology*, 15(4):401–405, August 2014.

[46] George Bouliotis and Lucinda Billingham. Crossing survival curves: alternatives to the log-rank test, 2011.

[47] David R Cox. Regression models and life-tables. *Journal of the Royal Statistical Society: Series B (Methodological)*, 34(2):187–202, 1972.

[48] Robert Tibshirani. Regression shrinkage and selection via the lasso. *J. R. Stat. Soc.*, 58(1):267–288, January 1996.

[49] Mike J Bradburn, Taane G Clark, Sharon B Love, and Douglas Graham Altman. Survival analysis part ii: multivariate data analysis–an introduction to concepts and methods. *British journal of cancer*, 89(3):431–436, 2003.

[50] A Clifford Cohen, John D Kalbfleisch, and Ross L Prentice. The statistical analysis of failure time data. *J. Am. Stat. Assoc.*, 77(378):497, June 1982.

[51] Stephen Salerno and Yi Li. High-dimensional survival analysis: Methods and applications. *Annual review of statistics and its application*, 10:25–49, 2023.

[52] Annette Spooner, Emily Chen, Arcot Sowmya, Perminder Sachdev, Nicole A Kochan, Julian Trollor, and Henry Brodaty. A comparison of machine learning methods for survival analysis of high-dimensional clinical data for dementia prediction. *Scientific reports*, 10(1):20410, 2020.

[53] Hemant Ishwaran, Udaya B Kogalur, Eugene H Blackstone, and Michael S Lauer. Random survival forests. *aoas*, 2(3):841–860, September 2008.

[54] Hemant Ishwaran, Udaya B Kogalur, Xi Chen, and Andy J Minn. Random survival forests for high-dimensional data. *Stat. Anal. Data Min.*, 4(1):115–132, February 2011.

[55] Leo Breiman. Random forests. *Mach. Learn.*, 45(1):5–32, October 2001.

[56] Hemant Ishwaran and Udaya B Kogalur. Random survival forests for r. *R news*, 7(2):25–31, 2007.

[57] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20:273–297, 1995.

[58] Sebastian Pölsterl, Nassir Navab, and Amin Katouzian. Fast training of support vector machines for survival analysis. In *Machine Learning and Knowledge Discovery in Databases*, pages 243–259. Springer International Publishing, 2015.

[59] Ron S Dembo and Trond Steihaug. Truncated-newtono algorithms for large-scale unconstrained optimization. *Mathematical Programming*, 26(2):190–212, 1983.

[60] Sebastian Pölsterl. scikit-survival: A library for Time-to-Event analysis built on top of scikit-learn. *J. Mach. Learn. Res.*, 21(212):1–6, 2020.

[61] Frank E Harrell, Robert M Califf, David B Pryor, Kerry L Lee, and Robert A Rosati. Evaluating the yield of medical tests. *Jama*, 247(18):2543–2546, 1982.

[62] Taane G Clark, Michael J Bradburn, Sharon B Love, and DG2394469 Altman. Survival analysis part iv: further concepts and methods in survival analysis. *British journal of cancer*, 89(5):781–786, 2003.

[63] Enrico Longato, Martina Vettoretti, and Barbara Di Camillo. A practical perspective on the concordance index for the evaluation and selection of prognostic time-to-event models. *Journal of Biomedical Informatics*, 108:103496, 2020.

[64] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, Rafael G L D'Oliveira, Hubert Eichner, Salim El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adrià Gascón, Badih Ghazi, Phillip B Gibbons, Marco Gruteser, Zaid Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konecný, Aleksandra Korolova, Farinaz Koushanfar, Sanmi Koyejo, Tancrède Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer Özgür, Rasmus Pagh, Hang Qi, Daniel Ramage, Ramesh Raskar, Mariana Raykova, Dawn Song, Weikang Song, Sebastian U Stich, Ziteng Sun, Ananda Theertha Suresh, Florian Tramèr, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X Yu, Han Yu, and Sen Zhao. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021.

[65] Georgios A Kaissis, Marcus R Makowski, Daniel Rückert, and Rickmer F Braren. Secure, privacy-preserving and federated machine learning in medical imaging. *Nature Machine Intelligence*, 2(6):305–311, June 2020.

[66] Reihaneh Torkzadehmahani, Reza Nasirigerdeh, David B Blumenthal, Tim Kacprowski, Markus List, Julian Matschinske, Julian Spaeth, Nina Kerstin Wenke, and Jan Baumbach. Privacy-Preserving artificial intelligence techniques in biomedicine. *Methods Inf. Med.*, 61(S 01):e12–e27, June 2022.

[67] G W Van Blarkom, John J Borking, and J G Eddy Olk. Handbook of privacy and privacy-enhancing technologies. *Privacy Incorporated Software Agent (PISA) Consortium, The Hague*, 198:14, 2003.

[68] Elif Ustundag Soykan, Leyli Karaçay, Ferhat Karakoç, and Emrah Tomur. A survey and guideline on privacy enhancing technologies for collaborative machine learning. *IEEE Access*, 10:97495–97519, 2022.

[69] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis, 2006.

[70] Cynthia Dwork. Differential privacy, 2006.

[71] Craig Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, STOC '09, pages 169–178, New York, NY, USA, May 2009. Association for Computing Machinery.

[72] Microsoft SEAL (release 4.1). `https://github.com/Microsoft/SEAL`, January 2023.

[73] Shruthi Gorantala, Rob Springer, Sean Purser-Haskell, William Lam, Royce Wilson, Asra Ali, Eric P Astor, Itai Zukerman, Sam Ruth, Christoph Dibak, Phillipp Schoppmann, Sasha Kulankhina, Alain Forget, David Marn, Cameron Tew, Rafael Misoczki, Bernat Guillen, Xinyu Ye, Dennis Kraft, Damien Desfontaines, Aishe Krishnamurthy, Miguel Guevara, Irippuge Milinda Perera, Yurii Sushko, and Bryant Gipson. A general purpose transpiler for fully homomorphic encryption. Cryptology ePrint Archive, Paper 2021/811, 2021.

[74] Ronald L Rivest, Adi Shamir, and Leonard Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[75] Ronald Cramer, Ivan Bjerre Damgard, and Jesper Buus Nielsen. Secure multiparty computation and secret sharing, 2015.

[76] Adi Shamir. How to share a secret, 1979.

[77] Daniel Escudero. An introduction to Secret-Sharing-Based secure multiparty computation. *Cryptology ePrint Archive*, 2022.

[78] Daniel Kales. Secret sharing. `https://www.iaik.tugraz.at/wp-content/uploads/teaching/mfc/secret_sharing.pdf`. Accessed: 2023-1-20.

[79] Rui Duan, Chongliang Luo, Martijn J. Schuemie, Jiayi Tong, C. Jason Liang, Howard H. Chang, Mary Regina Boland, Jiang Bian, Hua Xu, John H. Holmes, Christopher B. Forrest, Sally C. Morton, Jesse A. Berlin, Jason H. Moore, Kevin B. Mahoney, and Yong Chen. Learning from

local to global: An efficient distributed algorithm for modeling time-to-event data. *Journal of the American Medical Informatics Association*, 27:1028–1036, 7 2020. doi:10.1093/jamia/ocaa044.

[80] Patricia Guyot, AE Ades, Mario JNM Ouwens, and Nicky J Welton. Enhanced secondary analysis of survival data: reconstructing the data from published kaplan-meier survival curves. *BMC medical research methodology*, 12:1–13, 2012.

[81] Zhihui Liu, Benjamin Rich, and James A Hanley. Recovering the raw data behind a non-parametric survival curve. *Systematic reviews*, 3:1–10, 2014.

[82] Amadou Gaye, Yannick Marcon, Julia Isaeva, Philippe LaFlamme, Andrew Turner, Elinor M Jones, Joel Minion, Andrew W Boyd, Christopher J Newby, Marja-Liisa Nuotio, et al. Datashield: taking the analysis to the data, not the data to the analysis. *International journal of epidemiology*, 43(6):1929–1944, 2014.

[83] Soumya Banerjee, Ghislain N Sofack, Thodoris Papakonstantinou, Demetris Avraam, Paul Burton, Daniela Zöller, and Tom RP Bishop. dssurvival: Privacy preserving survival models for federated individual patient meta-analysis in datashield. *BMC Research Notes*, 15(1):197, 2022.

[84] Charles Lawrence Loprinzi, John A Laurie, H Sam Wieand, James E Krook, Paul J Novotny, John W Kugler, Joan Bartel, Marlys Law, Marilyn Bateman, and Nancy E Klatt. Prospective evaluation of prognostic variables from patient-completed questionnaires. north central cancer treatment group. *Journal of Clinical Oncology*, 12(3):601–607, 1994.

[85] John D Kalbfleisch and Ross L Prentice. The statistical analysis of failure time data. *Wiley, New York*, 1980.

[86] John A Laurie, Charles G Moertel, Thomas R Fleming, Harry S Wieand, John E Leigh, Jebal Rubin, Greg W McCormack, James B Gerstner, James E Krook, and James Malliard. Surgical adjuvant therapy of large-bowel carcinoma: an evaluation of levamisole and the combination of levamisole and fluorouracil. the north central cancer treatment group and the mayo clinic. *Journal of Clinical Oncology*, 7(10):1447–1456, 1989.

[87] Peter H Rossi, Richard A Berk, and Kenneth J Lenihan. Money, work and crime: some experimental results, 1980.

[88] Robert P Giugliano, Christian T Ruff, Eugene Braunwald, Sabina A Murphy, Stephen D Wiviott, Jonathan L Halperin, Albert L Waldo,

Michael D Ezekowitz, Jeffrey I Weitz, Jindřich Špinar, et al. Edoxaban versus warfarin in patients with atrial fibrillation. *New England Journal of Medicine*, 369(22):2093–2104, 2013.

[89] Axel Börsch-Supan. Survey of health, ageing and retirement in europe (SHARE) wave 8, February 2022. COVID-19 Survey 1. Release version: 8.0.0. SHARE-ERIC. Data set.

[90] Bendi Ramana and N. Venkateswarlu. ILPD (Indian Liver Patient Dataset). UCI Machine Learning Repository, 2012. DOI: https://doi.org/10.24432/C5D02C.

[91] W. Nick Street, W. H. Wolberg, and O. L. Mangasarian. Nuclear feature extraction for breast tumor diagnosis. In Raj S. Acharya and Dmitry B. Goldgof, editors, *Biomedical Image Processing and Biomedical Visualization*, volume 1905, pages 861 – 870. International Society for Optics and Photonics, SPIE, 1993. doi:10.1117/12.148698.

[92] David Harrison and Daniel L Rubinfeld. Hedonic housing prices and the demand for clean air. *Journal of Environmental Economics and Management*, 5(1):81–102, 1978. URL: https://www.sciencedirect.com/science/article/pii/0095069678900062, doi:10.1016/0095-0696(78)90006-2.

[93] Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani. Least angle regression. *The Annals of Statistics*, 32(2):407 – 499, 2004. doi:10.1214/009053604000000067.

[94] Beau Norgeot, Giorgio Quer, Brett K Beaulieu-Jones, Ali Torkamani, Raquel Dias, Milena Gianfrancesco, Rima Arnaout, Isaac S Kohane, Suchi Saria, Eric Topol, Ziad Obermeyer, Bin Yu, and Atul J Butte. Minimum information about clinical artificial intelligence modeling: the mi-claim checklist. *Nature medicine*, 26(9):1320—1324, September 2020. URL: https://europepmc.org/articles/PMC7538196, doi:10.1038/s41591-020-1041-y.

[95] Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

[96] Hui Wang, Ivo Duentsch, Gongde Guo, and Sadiq Ali Khan. Special issue on small data analytics. *International Journal of Machine Learning and Cybernetics*, 14(1):1–2, 2023.

[97] Wei Du, Depeng Xu, Xintao Wu, and Hanghang Tong. Fairness-aware agnostic federated learning. In *Proceedings of the 2021 SIAM International Conference on Data Mining (SDM)*, pages 181–189. SIAM, 2021.

[98] Yahya H Ezzeldin, Shen Yan, Chaoyang He, Emilio Ferrara, and A Salman Avestimehr. Fairfed: Enabling group fairness in federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 7494–7502, 2023.

[99] Lingjuan Lyu, Xinyi Xu, Qian Wang, and Han Yu. Collaborative fairness in federated learning. *Federated Learning: Privacy and Incentive*, pages 189–204, 2020.

[100] Alissa Brauneck, Louisa Schmalhorst, Mohammad Mahdi Kazemi Majdabadi, Mohammad Bakhtiari, Uwe Völker, Christina Caroline Saak, Jan Baumbach, Linda Baumbach, and Gabriele Buchholtz. Federated machine learning in data-protection-compliant research. *Nature Machine Intelligence*, 5(1):2–4, 2023.

Acronyms

AI artificial intelligence. 1, 2, 3, 22, 28, 31, 34

c-index concordance index. 12

CHF cumulative hazard function. 7, 11

CI confidence interval. 9

DL deep learning. 3, 14, 33, 34

DP differential privacy. 15, 16, 17, 18, 34

EHR electronic health record. 1, 2

EU European Union. 2, 27

FedAvg FederatedAveraging. 14, 15

FHE fully homomorphic encryption. 16

FL Federated Learning. 2, 3, 5, 12, 13, 14, 15, 17, 18, 20, 21, 22, 27, 33, 34, 35, 36, 51

GDPR General Data Protection Regulation. 2, 21, 27, 36

GUI graphical user interface. 3, 21, 34, 35

GWAS genome-wide association study. 2, 34

HE homomorphic encryption. 15, 16, 17, 18

HIPAA Health Insurance Portability and Accountability Act. 1, 27

ML machine learning. 1, 10, 13, 15, 18, 27, 28, 33, 34

PET privacy-enhancing technology. 3, 5, 15, 17, 18, 19, 20, 21, 28, 34, 35, 36

PHE partial homomorphic encryption. 16

RF random forest. 3, 10, 11, 34

RSF random survival forest. 10, 11

SMPC secure multi-party computation. 15, 16, 17, 18, 35

SVM support vector machine. 3, 10, 11, 12, 22, 31, 34, 55

UI user interface. 21, 22

List of Figures

List of Tables

List of Algorithms

Appendices

A Publications of this dissertation

A.1 Publication 1

PLOS DIGITAL HEALTH

RESEARCH ARTICLE

Privacy-aware multi-institutional time-to-event studies

Julian Späth[1]*, Julian Matschinske[1], Frederick K. Kamanu[2], Sabina A. Murphy[2], Olga Zolotareva[1,3], Mohammad Bakhtiari[1], Elliott M. Antman[4], Joseph Loscalzo[4], Alissa Brauneck[5], Louisa Schmalhorst[5], Gabriele Buchholtz[5], Jan Baumbach[1]

1 Institute for Computational Systems Biology, University of Hamburg, Hamburg, Germany, **2** TIMI Study Group, Division of Cardiovascular Medicine, Department of Medicine, Brigham and Women's Hospital, Harvard Medical School, Boston, Massachusetts, United States of America, **3** Chair of Proteomics and Bioanalytics, TUM School of Life Sciences, Technical University of Munich, Munich, Germany, **4** Department of Medicine, Brigham and Women's Hospital, Harvard Medical School, Boston, Massachusetts, United States of America, **5** Faculty of Legal Sciences, University of Hamburg, Hamburg, Germany

* julian.alexander.spaeth@uni-hamburg.de

Citation: Späth J, Matschinske J, Kamanu FK, Murphy SA, Zolotareva O, Bakhtiari M, et al. (2022) Privacy-aware multi-institutional time-to-event studies. PLOS Digit Health 1(9): e0000101. https://doi.org/10.1371/journal.pdig.0000101

Editor: Thomas Schmidt, University of Southern Denmark, DENMARK

Received: May 31, 2022

Accepted: August 6, 2022

Published: September 6, 2022

Data Availability Statement: The entire source code of the Partea platform is available on GitHub (https://github.com/federated-partea). The lung, veteran, and colon benchmark datasets are available through the R survival package. The rossi benchmark dataset is available via the lifelines Python package. Data from the ENGAGE-TIMI 48 Trial is not publicly available but is discussed in detail in the original publication of Giugliano et al.

Funding: This project has received funding from the European Union's Horizon 2020 research and innovation program (https://ec.europa.eu/info/

Abstract

Clinical time-to-event studies are dependent on large sample sizes, often not available at a single institution. However, this is countered by the fact that, particularly in the medical field, individual institutions are often legally unable to share their data, as medical data is subject to strong privacy protection due to its particular sensitivity. But the collection, and especially aggregation into centralized datasets, is also fraught with substantial legal risks and often outright unlawful. Existing solutions using federated learning have already demonstrated considerable potential as an alternative for central data collection. Unfortunately, current approaches are incomplete or not easily applicable in clinical studies owing to the complexity of federated infrastructures. This work presents privacy-aware and federated implementations of the most used time-to-event algorithms (survival curve, cumulative hazard rate, log-rank test, and Cox proportional hazards model) in clinical trials, based on a hybrid approach of federated learning, additive secret sharing, and differential privacy. On several benchmark datasets, we show that all algorithms produce highly similar, or in some cases, even identical results compared to traditional centralized time-to-event algorithms. Furthermore, we were able to reproduce the results of a previous clinical time-to-event study in various federated scenarios. All algorithms are accessible through the intuitive web-app *Partea* (https://partea.zbh.uni-hamburg.de), offering a graphical user interface for clinicians and non-computational researchers without programming knowledge. *Partea* removes the high infrastructural hurdles derived from existing federated learning approaches and removes the complexity of execution. Therefore, it is an easy-to-use alternative to central data collection, reducing bureaucratic efforts but also the legal risks associated with the processing of personal data to a minimum.

research-and-innovation/funding/funding-opportunities/funding-programmes-and-open-calls/horizon-2020_en) under grant agreements No. 826078 (JS, JM, MB, JB) and No. 777111 (JS, JB). This publication reflects only the authors' view and the European Commission is not responsible for any use that may be made of the information it contains. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: The authors have declared that no competing interests exist.

Author summary

Collecting data centrally from different sites in the clinical time-to-event analysis is still challenging due to the high bureaucratic effort and strict data protection laws such as the GDPR. However, huge datasets are needed to extract valuable insights from the data by applying statistical and machine learning approaches. Current approaches are still incomplete: they often do not address privacy issues in any depth, have inaccessible user interfaces, do not cover multiple algorithms, or are not open-source. In contrast, the approach we present in this work is an open-source tool for privacy-aware time-to-event analysis (*Partea*) that can be intuitively used by clinicians, statisticians, and other researchers. It allows the users to run state-of-the-art privacy-aware time-to-event analysis on data distributed between multiple sites through an easy-to-use interface and solves technical and legal issues for the underlying technologies.

1. Introduction

Time-to-event analysis is a standard tool in clinical trials to model censored data [1]. In these data, the event of interest (e.g. death or relapse) is not necessarily observed until the end of the study, making usual statistical methods unemployable [2]. Time-to-event analysis is often applied in clinical trials that are designed to identify significant survival-related biomarkers or compare the efficacy of drugs [3–5]. As with many statistical analyses, large sample sizes are needed to produce reliable results and reduce bias. These large sample sizes are usually not available at a single institution. Therefore, different research institutions frequently participate in joint studies using a central data collection strategy. Owing to strict privacy regulations, such as the European General Data Protection Regulation (GDPR), collecting data centrally from different institutions is challenging, imposes substantial bureaucratic burdens, and might even be illegal in some cases [6,7]. Common approaches in clinical data sharing, such as de-identification or anonymization, come with a trade-off between data privacy and data quality [8]. If de-identification is not sufficiently strong, re-identification attacks can still reveal sensitive patient information [9,10]. Successful re-identification of shared anonymized data would harm data subjects in their fundamental right to privacy, thereby exposing the associated researchers to severe legal penalties. This is but one example of how crucial privacy-aware analysis of sensitive biomedical data is for the analysis of clinical studies.

Federated learning (FL) was developed to overcome these obstacles by enabling data analysis on geographically distributed data and keeping the sensitive data private [11,12]. FL allows the training of statistical models without sharing the raw data that contains private information about patients. Only summary statistics or model parameters, so-called local models, are shared with a trusted central aggregator [13]. These local models also fall under GDPR rules if they are generated from personal data. Still, FL systems can add technical security measures to make aggregation possible in a way that would not be the case with the data itself. One fundamental measure is encryption, preventing the aggregation server from being able to mount reconstruction attacks. Moreover, a combination of FL and privacy-enhancing technologies (PETs), such as additive secret sharing or differential privacy (DP), is needed to increase the privacy and security of the whole analysis, reduce the need for trust in the aggregation server, and ensure compliance with data protection laws [14–16]. Such a combination of FL and PETs is often called a hybrid approach. FL or hybrid implementations of various algorithms have already been shown to deliver accurate results in different biomedical applications, such as

genome-wide association studies [17,18], differential gene expression analysis [19], the analysis of electronic health records [20], or the prediction of patient outcomes with COVID-19 [21].

For time-time-to-event analysis, the first privacy-preserving and federated approaches were already developed in recent years. A concept of a distributed time-to-event regression was published by Lu et al. in 2015 [22]. *WebDISCO* was a web platform for distributed Cox proportional hazards models without patient-level data sharing. Another approach for calculating federated survival functions using multi-party homomorphic encryption was published by Froelicher et al. in 2021, being the first hybrid approach with an enhanced focus on privacy [23]. The current approaches already show the high potential of FL for time-to-event analysis, however, they do not offer fully extensive solutions. *WebDISCO* is not maintained any longer and does not consider PETs, requiring a high trust in the aggregating server and making it a potential point of cyber-attack [24]. Also, it only supports the Cox proportional hazards model and no other time-to-event algorithms. Froelicher et al. strongly focused on the privacy of the raw data and the exchanged model parameters. However, while their approach offers a strong level of privacy, the resulting survival curves can still leak information about the included patients without much effort [25]. Also, they solely focused on one type of algorithm, the Kaplan-Meier estimator. Another disadvantage is that their tool is unavailable to the general public, and their implementation is not open-source. A comprehensive toolset of widely used time-to-event algorithms is needed that is straightforward to understand and intuitive to set up and use. Ideally, it should reduce technical hurdles to a minimum, achieving similar results to the centralized approaches while preserving the patients' privacy and being GDPR compliant. Furthermore, when it comes to privacy-aware methods, open-source solutions have tremendous advantages by revealing the source code and therefore increasing the trust in the software. Also, open-source software enables future maintenance, security updates, community-driven development, and code usage in other projects. From a data privacy perspective, the open-source approach has the potential to maintain privacy through faster discovery and remedy of vulnerabilities. At the same time, it poses the risk of hackers exploiting their access to the code. However, from a technical point of view, this risk is not necessarily higher than in closed-source software [26].

To address the existing problems, we propose easily applicable, privacy-aware, federated implementations of the most widely used algorithms in clinical time-to-event studies: survival function, cumulative hazard function, log-rank test, and Cox proportional hazards model. Our implementations are based on a hybrid approach of FL and additive secret sharing to increase the privacy of FL by hiding the shared local statistics and model parameters from the global aggregator [27]. We extended the federated survival function, cumulative hazard function, and log-rank test by a previously published approach to render the resulting outputs differentially private and reduce the privacy leakage of published data [28]. Moreover, we extended the federated Cox proportional hazards model to support L1- and L2 penalization, which was not supported before in *WebDISCO*. We demonstrate that our approach performs as well as centralized approaches. Additionally, we reproduced a multi-institutional clinical study with centralized data collection with very high similarity. All methods are accessible through the open-source platform *Partea* (https://partea.zbh.uni-hamburg.de), enabling complete transparency about the implementations and allowing for further maintenance and extendibility by the community. The platform provides an entire federated infrastructure and makes privacy-aware multi-institutional time-to-event analysis accessible and ready for clinicians, statisticians, and bioinformaticians without deeper technical knowledge. It also incorporates PETs that represent the state-of-the-art in data privacy and ensure sufficient data protection to enable GDPR compliance even in large, complex collaborations. The entire source code is available on GitHub (https://github.com/federated-partea).

2. Materials and methods

2.1 Implementation

In this work, we implemented a hybrid approach combining FL and additive secret sharing to enable privacy-aware multi-institutional time-to-event analysis without central data collection. The FL architecture consists of local clients handling sensitive data analysis at each participating site and a global aggregation server that receives the local parameters from each site to incorporate them into a common, global model. At the beginning of each project, the public keys of each site are exchanged with all other sites. After that, our workflow consists of five major steps, as illustrated in Fig 1. (1). Each site creates a secret of its exchanging parameters for each other site, which, summed up together, will reveal the actual parameters again. Each secret is encrypted by the public key of a certain site, and can therefore only be decrypted by this site. (2) The server collects all secrets and distributes them to the corresponding sites. (3) Each site decrypts the received secrets using its private key and sums them up. (4) The summed-up parameters still do not reveal any information and are sent to the aggregation server. (5) Finally, the server sums up the received sums of each site to obtain the actual global aggregate and broadcast it to the local sites. For algorithms with an iterative approach, such as the Cox proportional hazards model, the whole process from step (1) to (5) is repeated until convergence or a stopping criterion has been reached.

The main advantage of this hybrid combination of FL and additive secret sharing is that participants and the aggregating server can only see the global aggregate of the calculation. They are not able to identify or reconstruct any of the exchanged parameters by still maintaining almost identical results. With this approach, we implemented privacy-aware and federated methods for the most commonly used time-to-event algorithms in clinical trials: the Kaplan-

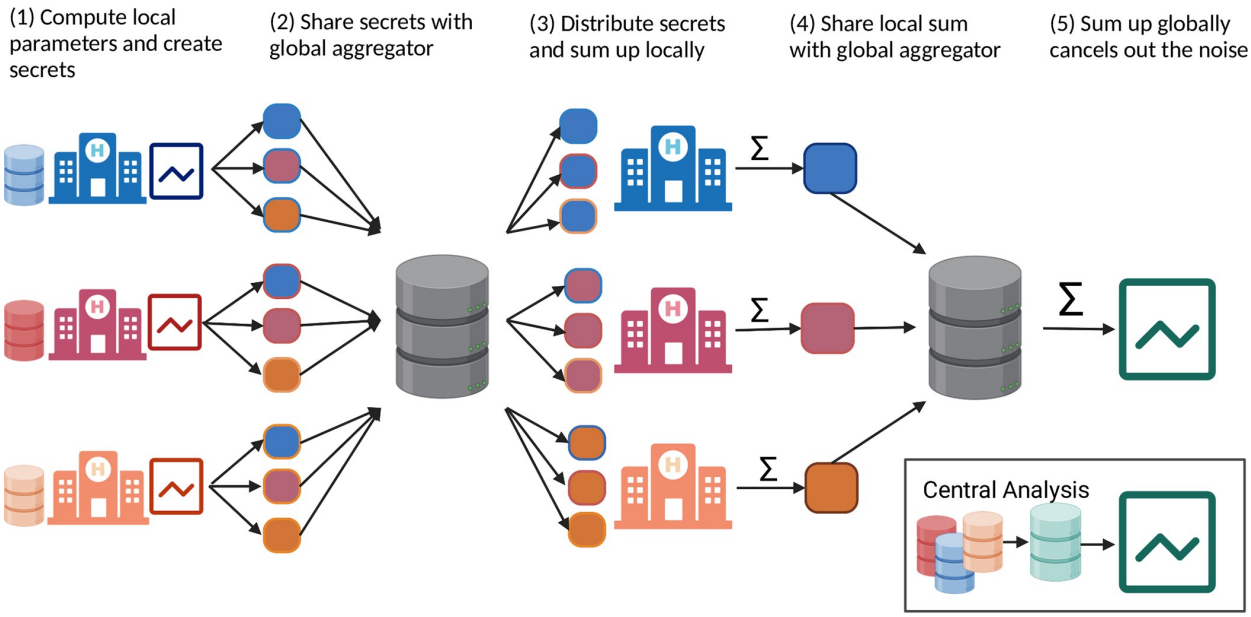


Fig 1. Hybrid federated learning workflow using additive secret sharing. Each institution calculates its local statistics and creates a secret for each participant (1). The global aggregation server receives the secrets and distributes them to the corresponding participants (2). Each local client decrypts the secrets and sums them up (3). The sum is shared with the global aggregation server (4), which sums them up again, revealing the final global aggregation (5). Created with Biorender.com.

https://doi.org/10.1371/journal.pdig.0000101.g001

Meier estimator for estimating the survival function [29], the Nelson-Aalen estimator for estimating the cumulative hazard function [30], the log-rank test for the comparison of two individual cohorts [31], and the Cox proportional hazards model for time-to-event regression [32,33].

Previous work has shown that it is possible to reconstruct the time of the event and event status directly from the survival function [34,35]. This potential leak in privacy also occurs in survival functions computed on centrally collected datasets. DP can be used to address this potential limitation. In DP, random noise is added to a model to hide the characteristics of individual data points. The noise level is chosen to prevent re-identification but not change the global properties of the dataset [36,37]. Therefore, we integrated the functionality of differentially private survival functions, cumulative hazard function, and log-rank test as proposed by Gondara *et al*. in 2020 into our approach. The authors added random Laplacian noise to the number of events, subjects at risk, and censored individuals for each time point [28].

All algorithms are accessible through the "***Partea***—**P**rivacy-**A**wa**R**e **T**ime-to-**E**vent **A**nalysis" platform, making them easily applicable in clinical trials. *Partea* consists of three main parts: (1) a global web frontend (Angular) to create federated projects, invite participants and visualize the results; (2) a local client application running on all major operating systems (Ubuntu, macOS, Windows) for local computations on sensitive data; (3) and a server for handling the data communication (Django). Through its intuitive user interface, *Partea* is not only applicable to statisticians or (bio)informaticians but can also be used by clinicians or biologists without programming knowledge. After creating a new study and adjusting several initial settings, the study coordinator can invite other participants by sharing unique invitation tokens. With these tokens, an invited participant can join the project through the local *Partea* client, choose its local dataset, and follow the progress of the federated study through the web app. After every participant has joined and the clients are running, the study coordinator can start the federated analysis through the web app. After the run, all results are available through the web app and can be explored interactively or downloaded.

2.2 Federated time-to-event analysis algorithms

2.2.1 Survival function, cumulative hazard function, and log-rank test. The survival function S(t) and cumulative hazard function H(t) are defined as:

$$S(t) = \prod_{t_i \leq t} \left[1 - \frac{d_i}{n_i}\right], H(t) = \sum_{t_i \leq t} \frac{d_i}{n_i}$$

In our federated approach, each participating site k calculates the number of events d_{ik} and the number of individuals at risk n_{ik} locally for each time point t_i and shares the resulting matrix m_k with the global aggregator. The aggregator sums up d_i and n_i of all K sites, leading to the formula for the federated survival function, and federated cumulative hazard function $H_{fed}(t)$:

$$S_{fed}(t) = \prod_{t_i \leq t} \left[1 - \left(\frac{\sum_{k \leq K} d_{ik}}{\sum_{k \leq K} n_{ik}}\right)\right], H_{fed}(t) = \sum_{t_i \leq t} \left(\frac{\sum_{k \leq K} d_{ik}}{\sum_{k \leq K} n_{ik}}\right)$$

Only counts of the observed events and of the individuals at risk are exchanged with the server to calculate the global survival function $S_{fed}(t)$ and cumulative hazard function $H_{fed}(t)$ by the global aggregator. Using the additive secret sharing scheme for data exchange, the aggregating server can only see the aggregated, global matrix m instead of all local matrices m_k that are being received, leading to a similar level of privacy as the centralized approach.

We further extended the approach to allow for the comparison of different cohorts or study groups using the log-rank test. For this comparison, each site needs an additional column in their input data indicating the corresponding group or cohort. For each group c, a separate matrix m_{ck} is calculated locally and aggregated to a global matrix m by the global aggregator. Based on this strategy, a pairwise federated log-rank test statistic X^2_{fed} can be calculated centrally at the aggregator using the expected (E) and observed (O) values of each group pair A and B:

$$X^2_{fed} = \frac{(O_A - E_A)^2}{E_A} + \frac{(O_B - E_B)^2}{E_B} \; with$$

$$O_A = \sum_i \left(\sum_{k \leq K} d^A_{ik} \right), O_B = \sum_i \left(\sum_{k \leq K} d^B_{ik} \right) \; and$$

$$E_A = \sum_i \frac{\sum_{k \leq K} n^A_{ik} * d_{ik}}{\sum_{k \leq K} n_{ik}}, E_B = \sum_i \frac{\sum_{k \leq K} n^B_{ik} * d_{ik}}{\sum_{k \leq K} n_{ik}},$$

2.2.2 Federated Cox proportional hazards model. Further, we reimplemented the WebDISCO [22] approach for the Cox proportional hazards model to enable federated time-to-event regression and extended it by the additive secret sharing scheme. Our implementation is based on lifelines, an open-source, state-of-the-art Python package for time-to-event analysis [38].

As *WebDISCO* did not address any normalization, we extended the approach with a federated z-score normalization. For this purpose, two exchanges with the server are needed. The local mean m_k and the local number of samples n_k for each site k and covariate are calculated and shared with the global aggregator to calculate the global mean m and share it with the local sites. Thereafter, each site uses the global mean m to compute their local $\Sigma_i \|X_i - m\|^2$ and shares it with the global aggregator to calculate the global standard deviation. This result is broadcast to the local sites again and used to normalize their local data, resulting in the formula for federated z-score normalization:

$$X_{fed,norm_k} = \frac{X_k - m_{fed}}{\sigma_{fed}}, m_{fed} = \frac{\sum_{k \leq K} m_k * n_k}{\sum_{k \leq K} n_k}, \sigma_{fed} = \sqrt{\left(\frac{\sum_{k \leq K} \sum_i \|X_{ik} - m_{fed}\|^2}{(\sum_{k \leq K} n_k) - 1} \right)}$$

We perform the initialization similar to as was done in the *WebDISCO* approach. After normalization, each site initializes the model statistics based on its local data. These statistics remain the same for the entire training process and are aggregated to the initialized global statistics on the aggregation server.

- D_k: distinct event times of site k
- d_k: number of events at each time point i of site k
- z_k: sum of the covariates over all individuals with an event that occurred at site k

resulting in the global aggregates:

$$D = U_{k \leq K} D_k, d = \sum_{k \leq K} d_k, z = \sum_{k \leq K} z_k$$

Furthermore, the beta vector containing the coefficient values is initialized with zeros. In our hybrid approach, instead of sharing the distinct event times of each site k we share a

common, predefined timeline. This hides the actual distinct event times of each site from the global aggregator and assures higher privacy.

Iteratively, until convergence, the global beta vector is broadcasted to the clients and the local statistics are calculated and shared again with the global aggregator:

$$\sum_{l \in R_i} \exp(\beta^T z^l), \sum_{l \in R_i} z_q^l * \exp(\beta^T z^l), \sum_{l \in R_i} z_r^l * z_q^l * \exp(\beta^T z^l)$$

with R_i being the index set of individuals who are at risk for failure at the time i. The global aggregator then calculates the first and second-order derivates of the likelihood function, updates the beta vector according to the Newton-Raphson method, and if convergence is not achieved, a new iteration starts.

We further extended the *WebDISCO* approach, to make use of *lifelines* penalized regression functionality and allow the use of both L1 and L2 penalties by specifying the l1-ratio (α) and penalty (p):

$$\frac{1}{2} p((1-\alpha) * \|\beta\|_2^2 + \alpha * \|\beta\|_1)$$

After convergence, the final coefficients of the model are known and can be used to prepare the final plots and statistics, such as p-values and hazard ratios for each covariate.

3. Results

3.1 Benchmark evaluation

To evaluate the performance of our approach, we ran analyses on four benchmark datasets that are commonly used in time-to-event analysis: US Veterans' Administration lung cancer study data [39] (Veteran, 137 samples), NCCTG lung cancer data [40] (Lung, 168 samples), criminal recidivism data [41] (Rossi, 432 samples), and chemotherapy for Stage B/C colon cancer trial data [42] (Colon, 888 samples). More details about the datasets can be found in the supporting information, **S1 Text.** Each dataset was split randomly and equally into 3, 5, and 10 parts to simulate various federated scenarios with different numbers of sites and sample sizes. For this, we simulated a federated environment using docker. Each site's local client was executed in a separate docker container to simulate network communication and different environments of the local datasets realistically.

3.1.1 Survival function. We calculated the survival function for each federated scenario using the federated approach (FL) and the hybrid approach of FL and additive secret sharing (sFL). We compared it to the central survival function estimated using *lifelines*, a state-of-the-art Python package for time-to-event analysis [38]. Both the federated and hybrid approaches resulted in identical survival functions compared to the central analysis (*lifelines*) for all evaluated datasets and scenarios of varying numbers of participants. The resulting survival curves are shown in **Fig 2**. Owing to the same underlying statistics, this also proves that our FL and sFL approach of the Nelson-Aalen estimator and the log-rank test provide identical results compared to the central analysis.

3.1.2 Differentially private survival functions. We also included the functionality for differentially private survival functions and evaluated the approach by comparing DP survival functions to the actual non-DP survival functions. The main goal of this evaluation was to suggest the privacy loss metric epsilon of the DP computation for future time-to-event analyses. Note that this evaluation is independent of the federated computation, as both provide identical results. In the method for differentially private survival function estimation by Gondara et al. in 2020, they show that the sensitivity of the survival function estimation is 1. With this

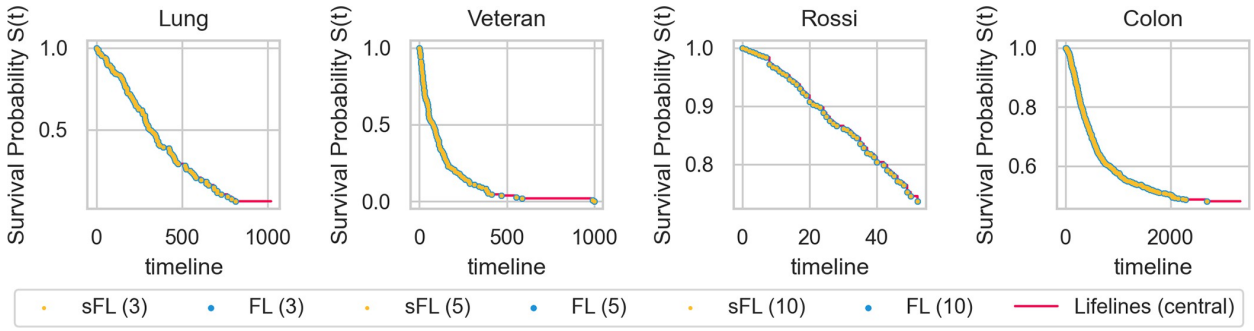


Fig 2. Evaluation of the survival function on benchmark datasets. For both the hybrid approach of FL and additive secret sharing (sFL, yellow) and the federated-only approach (FL, blue), identical survival functions are achieved compared to *lifelines*' Kaplan-Meier estimator (lifelines, red) for all four datasets and the various number of participants.

https://doi.org/10.1371/journal.pdig.0000101.g002

sensitivity and a variable privacy loss metric epsilon, the amount of noise is calculated, which is added to the survival function to guarantee a certain amount of privacy.

The smaller the privacy loss metric epsilon is, the more privacy is assured by the algorithm. For each dataset, we ran 1000 simulations with different epsilons (3, 2, 1, and 0.75). We next compared each differentially private function to the original non-DP function by applying a log-rank test to test whether two functions significantly differ from each other. A significant log-rank test means that the DP function is not similar to the actual function, making it inaccurate in clinical studies. The results of the log-rank tests are depicted in the supporting information, S1 Table.

As shown in Fig 3, the smaller epsilon is, the greater the resulting survival function differs from the original non-DP survival function. This finding, and the fact that smaller sample sizes are more affected by noise, match the observations in the original publication by Gondara et al. and is a common observation when applying DP.

Using epsilons of 3 and 2 resulted in 100% non-significant differences between the differentially private and non-differentially private survival function, using the log-rank test for all datasets. Except for the two datasets with smaller sample sizes (Lung and Veteran), epsilon equal to 1 and 0.75 led to significantly different survival functions in very few cases (worst being Veteran with an epsilon of 0.75, resulting in 2.4% significantly different functions). This observation indicates that, only in some rare cases, an epsilon of 1 and smaller can lead to too much noise if the sample sizes are small. Again, our results of the DP survival function evaluation are transferable to the Nelson-Aalen estimator and log-rank test as they are all calculated using the same underlying statistics.

Based on this analysis, we suggest three predefined epsilons to reduce complexity and understandability for users: "high DP" with an epsilon of 0.75, which can be applied if more than 400 samples are available; "medium DP" with an epsilon of 1, which can be applied with more than 200 samples; for smaller sample sizes, "low DP" for which an epsilon of 3 should be used.

3.1.3 Cox proportional hazards model. Similarly to the evaluation of the survival function, we simulated a federated scenario using the Cox proportional hazards model to compare the resulting logarithmized hazard ratio (HR) and its 95% confidence interval (CI). Fall datasets and the various number of participants, our federated-only approach, and the hybrid approach resulted in almost identical hazard ratios and corresponding CI for all covariates. A detailed overview of the comparison for each covariate and dataset is shown in Fig 4.

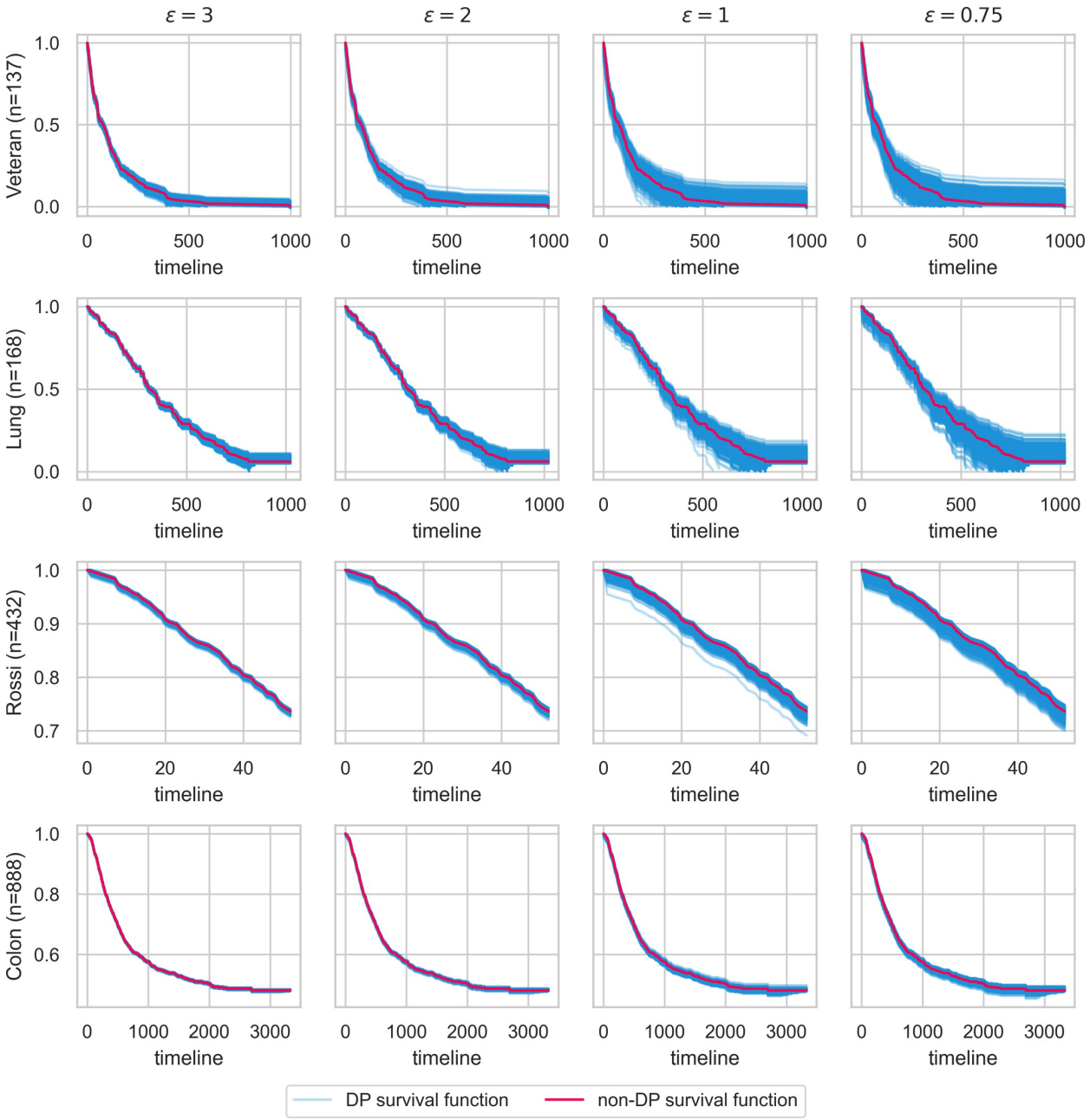


Fig 3. Comparison of DP survival functions against the non-DP baseline. The non-DP survival function (red) is used as a baseline against 1000 runs of DP survival functions for different epsilons and datasets. The resulting DP survival functions (blue) become noisier with decreasing epsilon. Note that the influence of the noise increases with decreasing sample size.

https://doi.org/10.1371/journal.pdig.0000101.g003

The evaluation shows that the federated-only approach is identical to the centralized Cox proportional hazards model. The hybrid approach with additive secret sharing is slightly more inaccurate because we not only transmit the timeline of the actual samples. Instead, a time range is used, including intermediate time points not existing in the local datasets. This assures

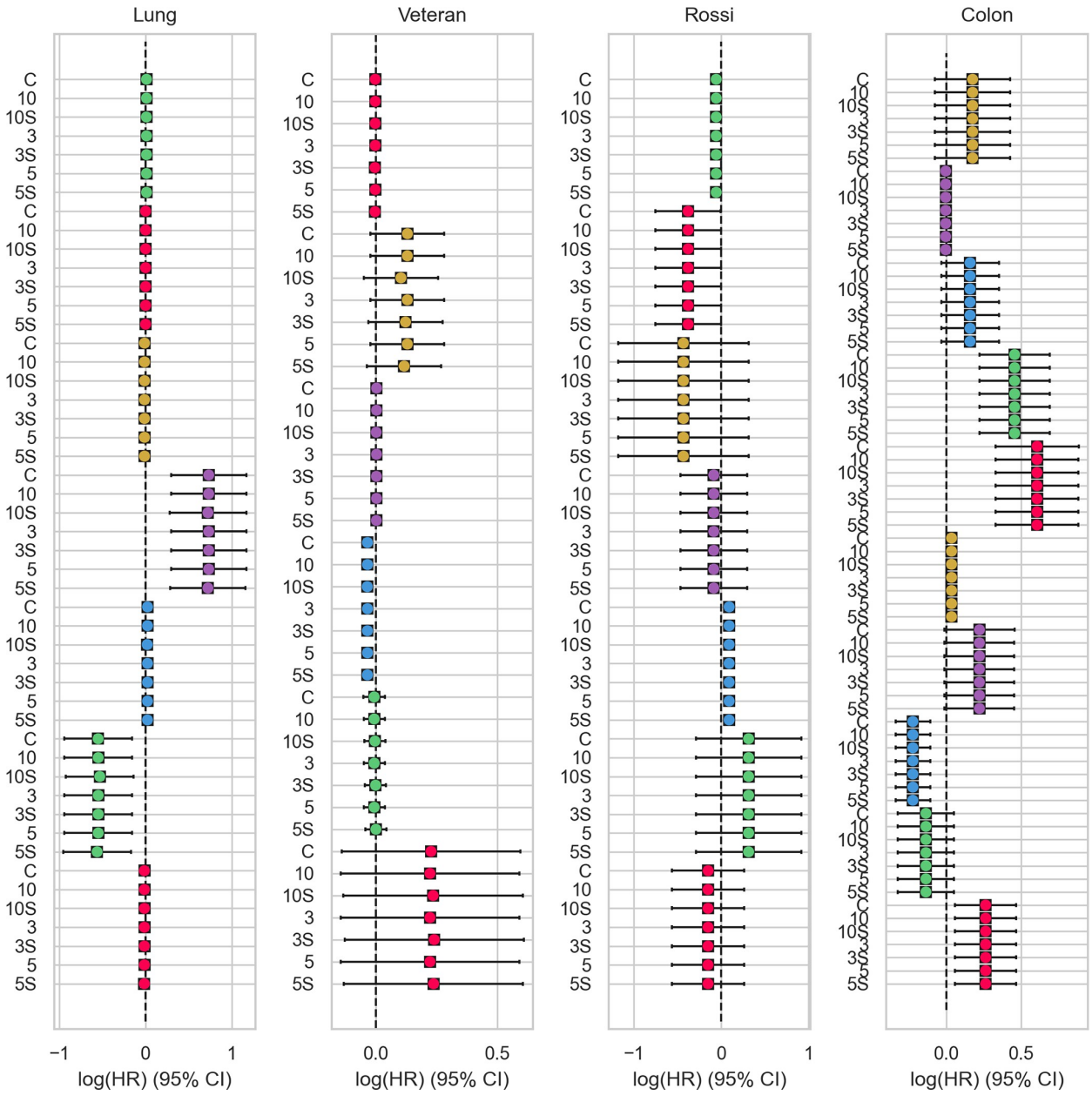


Fig 4. Evaluation of the Cox proportional hazards model on benchmark datasets. For each dataset, we compared the logarithmized hazard ratio and corresponding 95% CI of our algorithms for 3, 5, and 10 clients with the results of the centralized *lifelines* model. For all covariates (distinguished by colors), the federated-only (3, 5, 10) and hybrid approach (S3, S5, S10) resulted in almost identical results compared to the centralized calculation using *lifelines* (C).

https://doi.org/10.1371/journal.pdig.0000101.g004

more privacy, as what timepoints are derived from which site is inapparent in the data exchange. As we show on the four benchmark datasets, this slight inaccurateness does not influence the overall interpretation of the results, which remain close to the centralized or federated-only results.

3.2 Reproduction of a clinical study

To show the practical benefit of our framework for actual clinical time-to-event studies, we attempted to reproduce the results of the ENGAGE-TIMI 48 study conducted by the TIMI study group [43]. The study data were collected by the TIMI study group as part of a phase three, randomized, double-blind, double-dummy, parallel-group, multi-center, multi-national study, ENGAGE-TIMI 48, and contains more than 21,000 participants [43] from initially more than 1,300 sites. ENGAGE-TIMI 48 compared two different doses of edoxaban, a direct oral factor Xa inhibitor, with warfarin to evaluate the long-term efficacy and safety in patients with atrial fibrillation. Analyses were performed using a Cox proportional hazards model comparing each edoxaban dose group to warfarin and included the two randomization stratification factors. For our analysis, we split the centralized dataset equally into 3, 5, and 10 sites. We used the federated-only and hybrid Cox proportional hazards model to reproduce the results for the five outcome variables: stroke or systemic embolic event (Stroke/see), stroke, see or death from cardiovascular causes (Cv death/stroke/see), major adverse cardiac event (Mace), Stroke, and All-cause death.

Fig 5 depicts the logarithmized hazard ratios for each of the five outcome variables (columns) and covariates (indicated by different colors), calculated with our federated only (3, 5,

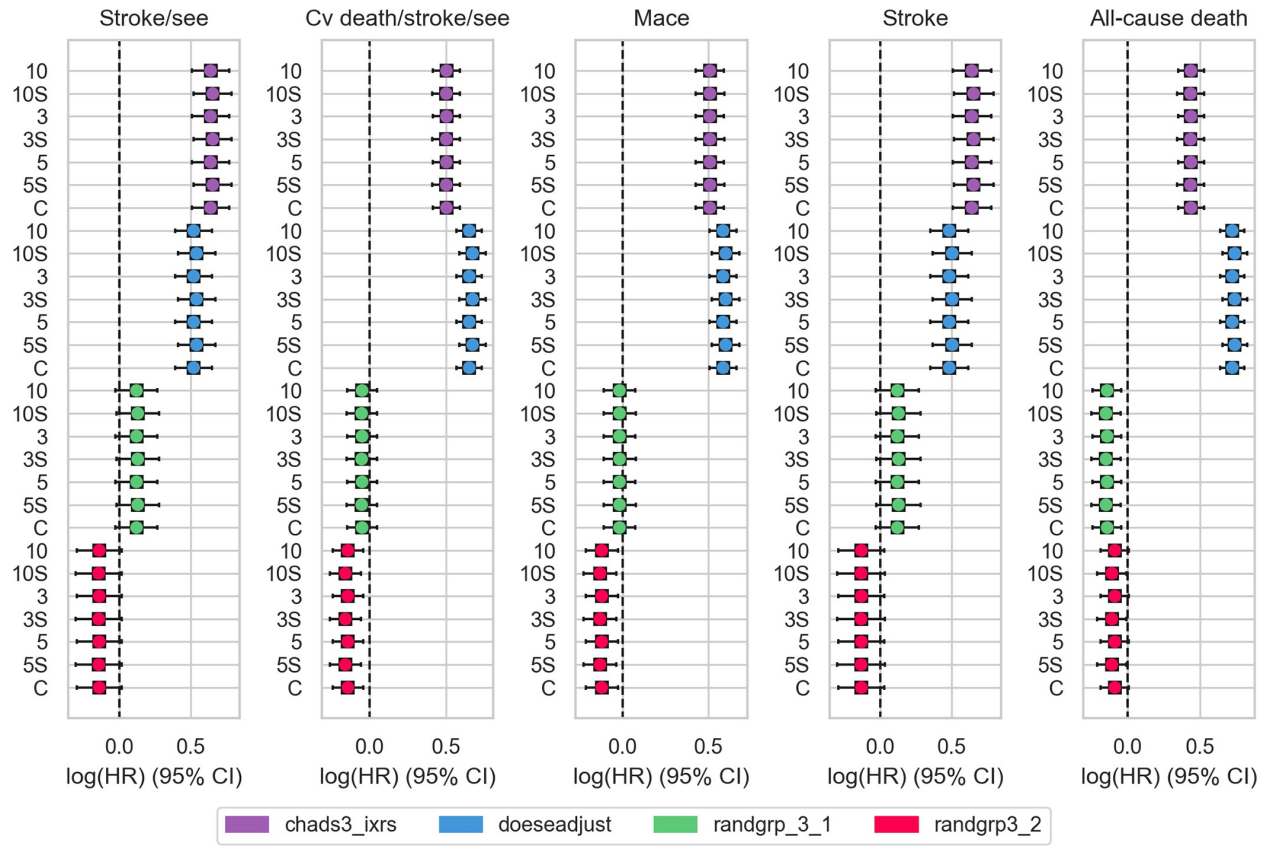


Fig 5. Federated time-to-event analysis of the ENGAGE-TIMI 48 study. Reproducing the ENGAGE-TIMI 48 study using 3, 5, and 10 clients (number) of both federated-only and hybrid approaches (number + S) compared with the results of the central analysis (C). The different covariates (colors) were regressed against five outcome variables (each subplot). The results of all five outcome variables and covariates are highly similar to the central analysis performed using the *lifelines* package.

https://doi.org/10.1371/journal.pdig.0000101.g005

10) and hybrid Cox proportional hazards model (3S, 5S, 10S). As apparent in the plots, our results are highly similar to the centralized calculation (C). The number of sites over which the data was distributed does not play any role. This shows that the federated, as well as the hybrid Cox proportional hazard model could accurately reproduce the analysis of the ENGAGE-TIMI 48 study, indicating the high potential of our approaches for future multi-institutional time-to-event studies.

4. Discussion

Clinical time-to-event studies are mainly performed on centralized data from one or more institutions. If multiple institutions participate in one study, a complicated data collection strategy is needed with high bureaucratic hurdles and legal pitfalls regarding the privacy of the utilized data. Prior work has already shown the potential of privacy-aware distributed analysis techniques in time-to-event analysis. However, current approaches are not complete. They either are not accessible, only support one kind of algorithm, do not integrate PETs, or are not open-source. In this work, we introduced a hybrid approach of FL and additive secret sharing for the most widely used time-to-event analysis algorithms: the Kaplan-Meier estimator for survival functions, the Nelson-Aalen estimator for cumulative hazard functions, the log-rank test, and the Cox proportional hazards model. All algorithms are bundled in our open-source platform *Partea*, making them easily accessible for usage in clinical trials and increasing trust and maintenance by having a published code-base. Our analyses on several benchmark datasets and the reproduction of a previous clinical study show highly similar results compared to central time-to-event studies. Our platform *Partea* has the possibility of being an intuitive and privacy-aware alternative to central data collection for future multi-institutional time-to-event studies with geographically distributed datasets.

The hybrid approach of FL and additive secret sharing can currently be considered state-of-the-art, privacy-aware, and potentially GDPR compliant. However, evaluating the GDPR compliance of machine learning systems is not trivial, owing to unclear criteria and definitions and the lack of jurisprudence. Even though the status of local and global models as personal data is still uncertain, it is very likely that the GDPR at least remains applicable to local models trained on personal data [15,44]. Likewise, the extent to which the addition of PETs such as additive secret sharing and DP is sufficient to result in GDPR compliance is not conclusively resolved. These questions will need answers from the courts or legislators in the future. The flexibility of *Partea's* open-source architecture allows it to rapidly be adapted and extended with community input in response to regulatory changes.

Open-source systems like *Partea* have further benefits compared to closed-source systems regarding maintainability and transparency. The source code is openly visible, so everyone can see how personal data is processed, increasing users' trust. This is also relevant for controllers of personal data, who are legally obligated under Article 32 of the GDPR to protect this data according to the state of art in technical and organizational measures. In the case of open source, the controller can show which PETs are used, how the program treats the data, what is sent around, and whether it holds its promises. In case of any security or privacy issues, users have a much higher chance of discovering this (legally relevant) breach and of holding the data controller accountable. Another advantage is that security gaps can be identified more quickly by the community. One downside of open-source is that it may also facilitate the identification of the existing security breaches for the attackers. However, there are indications that this does not lead to more attacks. In fact, experience has also shown that the security through obscurity of closed-source systems is very brittle [26].

In addition to these promising results, a few problems need to be considered, both technical and legal. Our hybrid scheme is only applicable if at least three sites participate in the analysis. In addition, it is apparent that the results of the hybrid approach differ slightly more from the centralized and federated-only analysis, mainly owing to the stronger privacy mechanism we implemented in this approach. Instead of sharing the exact distinct event times occurring at each site, we use a predefined timeline, including times that do not appear in the local dataset. This approach prevents the global aggregator from reading the event times of a single site. However, users can still use the federated-only approach if they trust the global aggregation server or work with non-sensitive data.

Another problem appearing in most of the federated learning tools is data harmonization. Our algorithms include several preprocessing steps for the computation (e.g. standardization of the data in the Cox proportional hazards model). Also, we allow for a detailed description of the data, such as the used time format (days, weeks, months, years) or the naming of the time and event columns. However, especially in the case of the Cox proportional hazards model, *Partea* expects similar datasets, meaning that besides the event and time columns, all other columns should be harmonized between different sites. The automatic harmonization of data is not trivial and out of the scope of *Partea*. However, this also encourages the participating sites to communicate and discuss a thorough study design upfront which might be even advantageous over an automatic data harmonization.

Our approaches can be easily extended in future work. As already mentioned, by offering an open-source platform and algorithms, Partea can be quickly adapted to potential changes in privacy regulations. Also, subsequent analyses such as checking the proportional hazard assumption based on scaled Schoenfeld residuals could be integrated in the future [45,46]. Furthermore, our platform could be easily extended by further privacy-aware time-to-event analysis implementations, such as Random Survival Forests [47] or Survival Support Vector Machines [48,49]. We believe that through the combination of extendibility through its open-source code, the strong focus on privacy, its accessibility, and its support of the most used time-to-event analysis algorithms, *Partea* has the potential to become the new gold standard in multi-institutional time-to-event analyses and provides various advantages to current solutions.

Supporting information

S1 Text. Dataset.
(DOCX)

S1 Table. Log-rank test comparison of DP survival functions and non-DP survival functions. For all datasets, using an epsilon of 3 and 2 resulted in non-significant p-values of the log rank-test, indicating that the DP survival functions are still relatable to the original one. Only for the small sample size datasets Veteran and Lung, small epsilons of 1 and 0.75 resulted in significant differences between in less than 2.5% of the curves.
(DOCX)

Author Contributions

Conceptualization: Julian Späth, Sabina A. Murphy, Elliott M. Antman, Joseph Loscalzo, Jan Baumbach.

Data curation: Frederick K. Kamanu, Sabina A. Murphy.

Formal analysis: Julian Späth, Frederick K. Kamanu.

Funding acquisition: Jan Baumbach.

Investigation: Alissa Brauneck, Louisa Schmalhorst, Gabriele Buchholtz.

Methodology: Julian Späth, Julian Matschinske, Sabina A. Murphy, Olga Zolotareva, Mohammad Bakhtiari.

Project administration: Jan Baumbach.

Resources: Sabina A. Murphy, Elliott M. Antman, Joseph Loscalzo.

Software: Julian Späth, Julian Matschinske, Olga Zolotareva, Mohammad Bakhtiari.

Visualization: Julian Späth.

Writing – original draft: Julian Späth.

Writing – review & editing: Julian Späth, Julian Matschinske, Frederick K. Kamanu, Sabina A. Murphy, Olga Zolotareva, Mohammad Bakhtiari, Elliott M. Antman, Joseph Loscalzo, Alissa Brauneck, Louisa Schmalhorst, Gabriele Buchholtz, Jan Baumbach.

References

1. Singh R, Mukhopadhyay K. Survival analysis in clinical trials: Basics and must know areas. Perspect Clin Res. 2011; 2: 145–148. https://doi.org/10.4103/2229-3485.86872 PMID: 22145125
2. Prinja S, Gupta N, Verma R. Censoring in clinical trials: review of survival analysis techniques. Indian J Community Med. 2010; 35: 217–221. https://doi.org/10.4103/0970-0218.66859 PMID: 20922095
3. Grein J, Ohmagari N, Shin D, Diaz G, Asperges E, Castagna A, et al. Compassionate use of remdesivir for patients with severe Covid-19. N Engl J Med. 2020; 382: 2327–2336. https://doi.org/10.1056/NEJMoa2007016 PMID: 32275812
4. Levy D, Kenchaiah S, Larson MG, Benjamin EJ, Kupka MJ, Ho KKL, et al. Long-term trends in the incidence of and survival with heart failure. N Engl J Med. 2002; 347: 1397–1402. https://doi.org/10.1056/NEJMoa020265 PMID: 12409541
5. Liu L, Chen Z, Shi W, Liu H, Pang W. Breast cancer survival prediction using seven prognostic biomarker genes. Oncol Lett. 2019; 18: 2907–2916. https://doi.org/10.3892/ol.2019.10635 PMID: 31452771
6. Antman E. Data sharing in research: benefits and risks for clinicians. BMJ. 2014; 348: g237. https://doi.org/10.1136/bmj.g237 PMID: 24458978
7. Aichroth P, Battis V, Dewes A, Dibak C, Doroshenko V, Geiger B, et al. Anonymisierung und Pseudonymisierung von Daten für Projekte des maschinellen Lernens- Eine Handreichung für Unternehmen. In: Bitkom [Internet]. 2020 [cited 11 Aug 2022]. Available: https://www.bitkom.org/sites/default/files/2020-10/201002_lf_anonymisierung-und-pseudonymisierung-von-daten.pdf. German.
8. Lo B. Sharing clinical trial data: maximizing benefits, minimizing risk. JAMA. 2015; 313: 793–794. https://doi.org/10.1001/jama.2015.292 PMID: 25594500
9. Hansson MG, Lochmüller H, Riess O, Schaefer F, Orth M, Rubinstein Y, et al. The risk of re-identification versus the need to identify individuals in rare disease research. Eur J Hum Genet. 2016; 24: 1553–1558. https://doi.org/10.1038/ejhg.2016.52 PMID: 27222291
10. McGuire AL, Gibbs RA. No longer de-identified. SCIENCE-NEW YORK THEN WASHINGTON-. 2006; 312: 370.
11. Kairouz P, Brendan McMahan H, Avent B, Bellet A, Bennis M, Bhagoji AN, et al. Advances and Open Problems in Federated Learning. arXiv [cs.LG]. 2019. Available: http://arxiv.org/abs/1912.04977
12. Yang Q, Liu Y, Chen T, Tong Y. Federated machine learning. ACM Trans Intell Syst Technol. 2019; 10: 1–19.
13. Sheller MJ, Edwards B, Reina GA, Martin J, Pati S, Kotrotsou A, et al. Federated learning in medicine: facilitating multi-institutional collaborations without sharing patient data. Sci Rep. 2020; 10: 12598. https://doi.org/10.1038/s41598-020-69250-1 PMID: 32724046
14. Kaissis GA, Makowski MR, Rückert D, Braren RF. Secure, privacy-preserving and federated machine learning in medical imaging. Nature Machine Intelligence. 2020; 2: 305–311.

15. Rossello S, Díaz Morales R, Muñoz-González L. Data protection by design in AI? The case of federated learning. 30 May 2021 [cited 11 Aug 2022]. Available: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3879613

16. Bonawitz K, Kairouz P, McMahan B, Ramage D. Federated Learning and Privacy: Building privacy-preserving systems for machine learning and data science on decentralized data. ACM QUEUE. 2021; 19: 87–114.

17. Constable SD, Tang Y, Wang S, Jiang X, Chapin S. Privacy-preserving GWAS analysis on federated genomic datasets. BMC Med Inform Decis Mak. 2015; 15 Suppl 5: S2. https://doi.org/10.1186/1472-6947-15-S5-S2 PMID: 26733045

18. Nasirigerdeh R, Torkzadehmahani R, Matschinske J, Frisch T, List M, Späth J, et al. sPLINK: a hybrid federated tool as a robust alternative to meta-analysis in genome-wide association studies. Genome Biol. 2022; 23: 32. https://doi.org/10.1186/s13059-021-02562-1 PMID: 35073941

19. Zolotareva O, Nasirigerdeh R, Matschinske J, Torkzadehmahani R, Bakhtiari M, Frisch T, et al. Flimma: a federated and privacy-aware tool for differential gene expression analysis. Genome Biol. 2021; 22: 338. https://doi.org/10.1186/s13059-021-02553-2 PMID: 34906207

20. Brisimi TS, Chen R, Mela T, Olshevsky A, Paschalidis IC, Shi W. Federated learning of predictive models from federated Electronic Health Records. Int J Med Inform. 2018; 112: 59–67. https://doi.org/10.1016/j.ijmedinf.2018.01.007 PMID: 29500022

21. Dayan I, Roth HR, Zhong A, Harouni A, Gentili A, Abidin AZ, et al. Federated learning for predicting clinical outcomes in patients with COVID-19. Nat Med. 2021; 27: 1735–1743. https://doi.org/10.1038/s41591-021-01506-3 PMID: 34526699

22. Lu C-L, Wang S, Ji Z, Wu Y, Xiong L, Jiang X, et al. WebDISCO: a web service for distributed cox model learning without patient-level data sharing. J Am Med Inform Assoc. 2015; 22: 1212–1219. https://doi.org/10.1093/jamia/ocv083 PMID: 26159465

23. Froelicher D, Troncoso-Pastoriza JR, Raisaro JL, Cuendet MA, Sousa JS, Cho H, et al. Truly privacy-preserving federated analytics for precision medicine with multiparty homomorphic encryption. Nat Commun. 2021; 12: 5910. https://doi.org/10.1038/s41467-021-25972-y PMID: 34635645

24. Torkzadehmahani R, Nasirigerdeh R, Blumenthal DB, Kacprowski T, List M, Matschinske J, et al. Privacy-Preserving Artificial Intelligence Techniques in Biomedicine. Methods Inf Med. 2022. https://doi.org/10.1055/s-0041-1740630 PMID: 35062032

25. Liu Z, Rich B, Hanley JA. Recovering the raw data behind a non-parametric survival curve. Syst Rev. 2014; 3: 151. https://doi.org/10.1186/2046-4053-3-151 PMID: 25551437

26. Clarke Dorwin, Nash. Is open source software more secure? Homeland Security/Cyber Security.

27. Cramer R, Damgard IB, Nielsen JB. Secure Multiparty Computation and Secret Sharing. 2015. https://doi.org/10.1017/cbo9781107337756

28. Gondara L, Wang K. Differentially Private Survival Function Estimation. In: Doshi-Velez F, Fackler J, Jung K, Kale D, Ranganath R, Wallace B, et al., editors. Proceedings of the 5th Machine Learning for Healthcare Conference. PMLR; 07–08 Aug 2020. pp. 271–291.

29. Kaplan EL, Meier P. Nonparametric Estimation from Incomplete Observations. J Am Stat Assoc. 1958; 53: 457–481.

30. Aalen O. Nonparametric Inference for a Family of Counting Processes. Ann Stat. 1978; 6: 701–726.

31. Mantel N. Evaluation of survival data and two new rank order statistics arising in its consideration. Cancer Chemother Rep. 1966; 50: 163–170. PMID: 5910392

32. Cox DR. Regression models and life-tables. J R Stat Soc. 1972; 34: 187–202.

33. Tibshirani R. Regression shrinkage and selection via the lasso. J R Stat Soc. 1996; 58: 267–288.

34. Guyot P, Ades AE, Ouwens MJNM, Welton NJ. Enhanced secondary analysis of survival data: reconstructing the data from published Kaplan-Meier survival curves. BMC Med Res Methodol. 2012; 12: 9. https://doi.org/10.1186/1471-2288-12-9 PMID: 22297116

35. Wei Y, Royston P. Reconstructing Time-to-event Data from Published Kaplan–Meier Curves. Stata J. 2017; 17: 786–802. PMID: 29398980

36. Dwork C. Differential Privacy. Automata, Languages and Programming. 2006. pp. 1–12. https://doi.org/10.1007/11787006_1

37. Dwork C, McSherry F, Nissim K, Smith A. Calibrating noise to sensitivity in private data analysis. Theory of cryptography conference. Springer; 2006. pp. 265–284.

38. Davidson-Pilon C. lifelines: survival analysis in Python. J Open Source Softw. 2019; 4: 1317.

39. Cohen AC, Kalbfleisch JD, Prentice RL. The statistical analysis of failure time data. J Am Stat Assoc. 1982; 77: 497.

40. Loprinzi CL, Laurie JA, Wieand HS, Krook JE, Novotny PJ, Kugler JW, et al. Prospective evaluation of prognostic variables from patient-completed questionnaires. North Central Cancer Treatment Group. J Clin Oncol. 1994; 12: 601–607.

41. Rossi PH, Berk RA, Lenihan KJ. Money, work and crime: some experimental results. New York: Academic Press; 1980.

42. Laurie JA, Moertel CG, Fleming TR, Wieand HS, Leigh JE, Rubin J, et al. Surgical adjuvant therapy of large-bowel carcinoma: an evaluation of levamisole and the combination of levamisole and fluorouracil. The North Central Cancer Treatment Group and the Mayo Clinic. J Clin Oncol. 1989; 7: 1447–1456.

43. Giugliano RP, Ruff CT, Braunwald E, Murphy SA, Wiviott SD, Halperin JL, et al. Edoxaban versus warfarin in patients with atrial fibrillation. N Engl J Med. 2013; 369: 2093–2104. https://doi.org/10.1056/NEJMoa1310907 PMID: 24251359

44. Truong N, Sun K, Wang S, Guitton F, Guo Y. Privacy preservation in federated learning: An insightful survey from the GDPR perspective. Computer Security. 2021;110. https://doi.org/10.1016/j.cose.2021.102402

45. Schoenfeld D. Chi-squared goodness-of-fit tests for the proportional hazards regression model. Biometrika. 1980; 67: 145–153.

46. Schoenfeld D. Partial residuals for the proportional hazards regression model. Biometrika. 1982; 69: 239–241.

47. Ishwaran H, Kogalur UB, Blackstone EH, Lauer MS. Random survival forests. aoas. 2008; 2: 841–860.

48. Van Belle V, Pelckmans K, Suykens JAK, Van Huffel S. Support vector machines for survival analysis. Proceedings of the Third International Conference on Computational Intelligence in Medicine and Healthcare (CIMED2007). 2007. pp. 1–8.

49. Pölsterl S, Navab N, Katouzian A. Fast Training of Support Vector Machines for Survival Analysis. Machine Learning and Knowledge Discovery in Databases. Springer International Publishing; 2015. pp. 243–259.

A.2 Publication 2

Original Paper

The FeatureCloud Platform for Federated Learning in Biomedicine: Unified Approach

Julian Matschinske[1*], MSc; Julian Späth[1*], MSc; Mohammad Bakhtiari[1], MSc; Niklas Probul[1], MSc; Mohammad Mahdi Kazemi Majdabadi[1], MSc; Reza Nasirigerdeh[2], MSc; Reihaneh Torkzadehmahani[2], MSc; Anne Hartebrodt[3], MSc, PhD; Balazs-Attila Orban[4], MSc; Sándor-József Fejér[4], MSc; Olga Zolotareva[2], PhD; Supratim Das[1], MSc; Linda Baumbach[5], PhD; Josch K Pauling[2], PhD; Olivera Tomašević[6], MSc; Béla Bihari[4], MSc; Marcus Bloice[7], MSc; Nina C Donner[8], PhD; Walid Fdhila[9], PhD; Tobias Frisch[3], PhD; Anne-Christin Hauschild[10], Prof Dr; Dominik Heider[11], Prof Dr; Andreas Holzinger[7], Prof Dr; Walter Hötzendorfer[12], Dr; Jan Hospes[12], Mag iur; Tim Kacprowski[13], Prof Dr; Markus Kastelitz[12], PhD; Markus List[2], PhD; Rudolf Mayer[9], MSc; Mónika Moga[4], PhD; Heimo Müller[7], PhD; Anastasia Pustozerova[9], MSc; Richard Röttger[3], Prof Dr; Christina C Saak[1], PhD; Anna Saranti[7], PhD; Harald H H W Schmidt[14], Prof Dr; Christof Tschohl[12], Dr; Nina K Wenke[1], PhD; Jan Baumbach[1], Prof Dr

[1]University of Hamburg, Hamburg, Germany
[2]Technical University Munich, Munich, Germany
[3]University of Southern Denmark, Odense, Denmark
[4]Gnome Design SRL, Sfântu Gheorghe, Romania
[5]University Medical Center Hamburg-Eppendorf, Hamburg, Germany
[6]University of Novi Sad, Novi Sad
[7]Medical University of Graz, Graz, Austria
[8]Concentris Research Management gGmbH, Fürstenfeldbruck, Germany
[9]SBA Research gGmbH, Vienna, Austria
[10]University Medical Center Göttingen, Göttingen, Germany
[11]Philipps-University of Marburg, Marburg, Germany
[12]Research Institute AG & Co KG, Vienna, Austria
[13]Technical University Braunschweig and Hannover Medical School, Brunswick, Germany
[14]Maastricht University, Maastricht, Netherlands
[*]these authors contributed equally

Corresponding Author:
Julian Matschinske, MSc
University of Hamburg
Notkestrasse 9
Hamburg, 22607
Germany
Phone: 49 40 42838 ext 7640
Email: julian.matschinske@uni-hamburg.de

Abstract

Background: Machine learning and artificial intelligence have shown promising results in many areas and are driven by the increasing amount of available data. However, these data are often distributed across different institutions and cannot be easily shared owing to strict privacy regulations. Federated learning (FL) allows the training of distributed machine learning models without sharing sensitive data. In addition, the implementation is time-consuming and requires advanced programming skills and complex technical infrastructures.

Objective: Various tools and frameworks have been developed to simplify the development of FL algorithms and provide the necessary technical infrastructure. Although there are many high-quality frameworks, most focus only on a single application case or method. To our knowledge, there are no generic frameworks, meaning that the existing solutions are restricted to a particular type of algorithm or application field. Furthermore, most of these frameworks provide an application programming

interface that needs programming knowledge. There is no collection of ready-to-use FL algorithms that are extendable and allow users (eg, researchers) without programming knowledge to apply FL. A central FL platform for both FL algorithm developers and users does not exist. This study aimed to address this gap and make FL available to everyone by developing FeatureCloud, an all-in-one platform for FL in biomedicine and beyond.

Methods: The FeatureCloud platform consists of 3 main components: a global frontend, a global backend, and a local controller. Our platform uses a Docker to separate the local acting components of the platform from the sensitive data systems. We evaluated our platform using 4 different algorithms on 5 data sets for both accuracy and runtime.

Results: FeatureCloud removes the complexity of distributed systems for developers and end users by providing a comprehensive platform for executing multi-institutional FL analyses and implementing FL algorithms. Through its integrated artificial intelligence store, federated algorithms can easily be published and reused by the community. To secure sensitive raw data, FeatureCloud supports privacy-enhancing technologies to secure the shared local models and assures high standards in data privacy to comply with the strict General Data Protection Regulation. Our evaluation shows that applications developed in FeatureCloud can produce highly similar results compared with centralized approaches and scale well for an increasing number of participating sites.

Conclusions: FeatureCloud provides a ready-to-use platform that integrates the development and execution of FL algorithms while reducing the complexity to a minimum and removing the hurdles of federated infrastructure. Thus, we believe that it has the potential to greatly increase the accessibility of privacy-preserving and distributed data analyses in biomedicine and beyond.

(J Med Internet Res 2023;25:e42621) doi: 10.2196/42621

KEYWORDS

privacy-preserving machine learning; federated learning; interactive platform; artificial intelligence; AI store; privacy-enhancing technologies; additive secret sharing

Introduction

The Problem of Scattered Data

Machine learning (ML) and artificial intelligence (AI) have increased in popularity over the last decade, leading to discoveries in various fields, including biomedicine [1-3]. The utility of ML and AI models depends on the size and quality of the available training data. However, data sources are often scattered across multiple facilities, and privacy regulations restrict data sharing, rendering large-scale, centralized ML infeasible. Particularly in biomedicine, the collection of molecular and clinical data is becoming ubiquitous with the successful applications of ML in diagnostics [4] or drug discovery [5]. Privacy concerns hinder even faster advances because of the small sample size of the individual data sets available, such as in the case of rare diseases.

Federated Learning and Privacy-Enhancing Technologies

One way to overcome these challenges is federated learning (FL). FL allows distributed data analysis by only exchanging model parameters and local models instead of sensitive raw data [6]. Hence, analyses can benefit from considerably larger data sets and be exploited with a lower risk of revealing primary data. FL can be divided into several subcategories that address different problems in decentralized computation and differ in their requirements [7]. First, FL can be categorized according to how the data are distributed among the clients. Horizontal FL addresses the training of a model on distributed data that has the same features but different samples. Vertical FL, in contrast, trains a model for the same samples but distributed features. Second, FL is distinguished by the number of clients that participate. Training a model on decentralized data from several organizations or data silos, such as hospitals or companies, is called cross-silo FL. If model training involves thousands or millions of clients, such as mobile phones or internet of things devices, we speak of cross-device FL. A typical FL setup consists of several clients and a central aggregator. Each client updates a local model based on its local data and sends it to a central aggregator. Here, the local models are aggregated into a common global model by an aggregation function, such as federated average [6]. This global model is then broadcasted to each client again. The entire process is repeated for the iterative algorithms.

Although other techniques, such as homomorphic encryption (HE), also allow for the analysis of distributed data by enabling calculations on encrypted data directly, they are computationally expensive compared with FL. In addition, they often require drastic changes to their original ML algorithm. In contrast, FL alone cannot always fulfill strict privacy requirements [8,9]. Therefore, to improve data privacy, FL can be combined with privacy-enhancing technologies (PETs) [10], such as secure aggregation [11] or differential privacy (DP) [12,13]. A recent study demonstrated that federated algorithms could achieve comparable or identical results compared with centralized ML [14-18].

Prior Work

Several frameworks have recently been developed to make FL available for a broader user group. Backend frameworks provide developers with methods to simplify the implementation of federated and privacy-aware algorithms [19-22]. They are limited to users with a strong background in software development or programming experience. Such skills are usually not expected from clinical experts and researchers, which considerably restricts their usability. All-in-one frameworks bring privacy-aware analyses to users without in-depth programming skills by providing a graphical user interface (GUI) [23-26]. However, most existing all-in-one frameworks are either not extendible or highly specific, focusing on a certain

type of algorithm (eg, deep learning [DL] only) or application (eg, neuroimaging and genomics).

Existing Shortcomings

Although the available frameworks demonstrate that FL is applicable and accelerates research in health care or biomedicine, the focus on 1 specific application or algorithm is also a huge restriction, especially in the collaboration of different fields. To the best of our knowledge, a generic, low-code, and open-source platform that can be driven and extended openly by the community to cover different algorithms and fields has been unavailable. However, such a platform is needed to enable FL across different applications and to make it applicable for users without technical knowledge of FL infrastructure or coding skills.

Goal of This Work

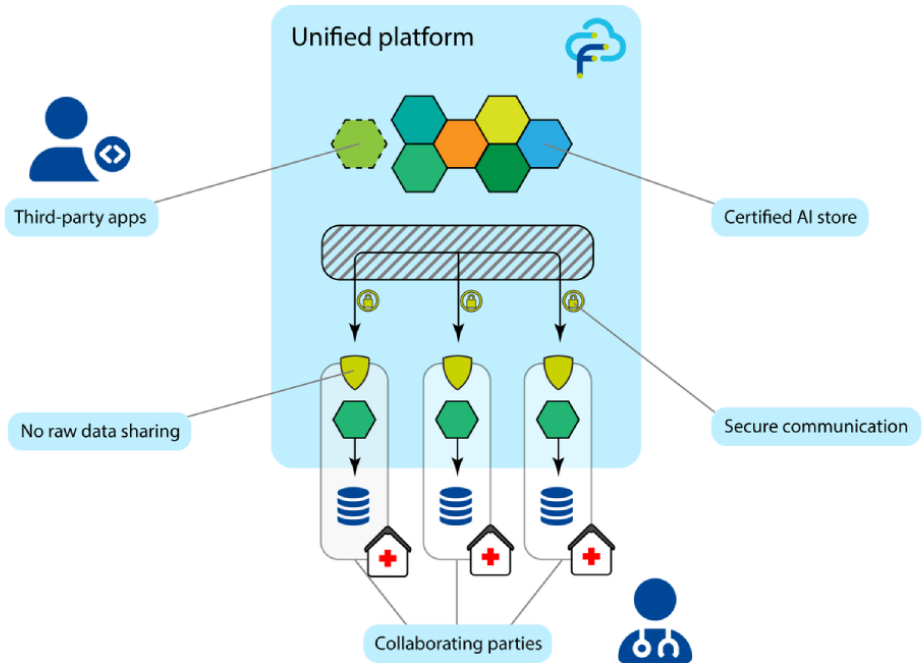
To close this gap, we present FeatureCloud, a comprehensive platform covering all the required steps from project coordination and workflow execution for the development of algorithms for cross-silo FL [27]. It incorporates and facilitates the development and deployment of federated algorithms and alleviates the technical difficulties of end users by providing a complete and ready-to-use infrastructure. Contrary to existing programming frameworks, FeatureCloud provides a running all-in-one platform that eliminates the need for developers and users to arrange a server deployment to conduct a federated study.

Methods

Overview

FeatureCloud was developed as a unified platform to increase the accessibility of FL for two large user groups as follows: (1) end users running FL algorithms to train ML models on distributed data sets and (2) developers implementing federated algorithms for statistics or ML that are not easily accessible in federated environments yet. As illustrated in Figure 1, the interface between developers and end users is our integrated AI store. Application developers can easily implement their own applications and publish them in the AI store, making them easily accessible to end users. Out of a broad collection of applications in the AI store, end users can assemble tailored workflows, invite collaborators, and perform FL on geographically distributed data. Therefore, FeatureCloud provides a complete infrastructure, including secure state-of-the-art communication, no raw data sharing, and several mechanisms to keep the actual data private.

Figure 1. Outline of the FeatureCloud system. Medical institutions collaborate in a federated study with all primary or raw data remaining at their original location. FeatureCloud handles the distribution, execution, and communication of certified artificial intelligence (AI) applications from the FeatureCloud AI store and addresses developers and end users.



Implementation

In this section, we present our implementation of the FeatureCloud platform: its system architecture, the FeatureCloud application programming interface (API) for developers, and the FL scheme and PETs used. Furthermore, we present the FL algorithms used for the evaluation of our platform.

System Architecture

FeatureCloud was developed as a system consisting of several interacting parts distributed between the participants and a central server. The central components include the backend (Python and Django), frontend (Angular), and Docker registry. The local components include the controller (Golang), the Docker engine, and the application instances (Docker images). Figure 2 shows the system components and the communication channels between them. Further details regarding their

implementation and technology used can be found in Multimedia Appendix 1.

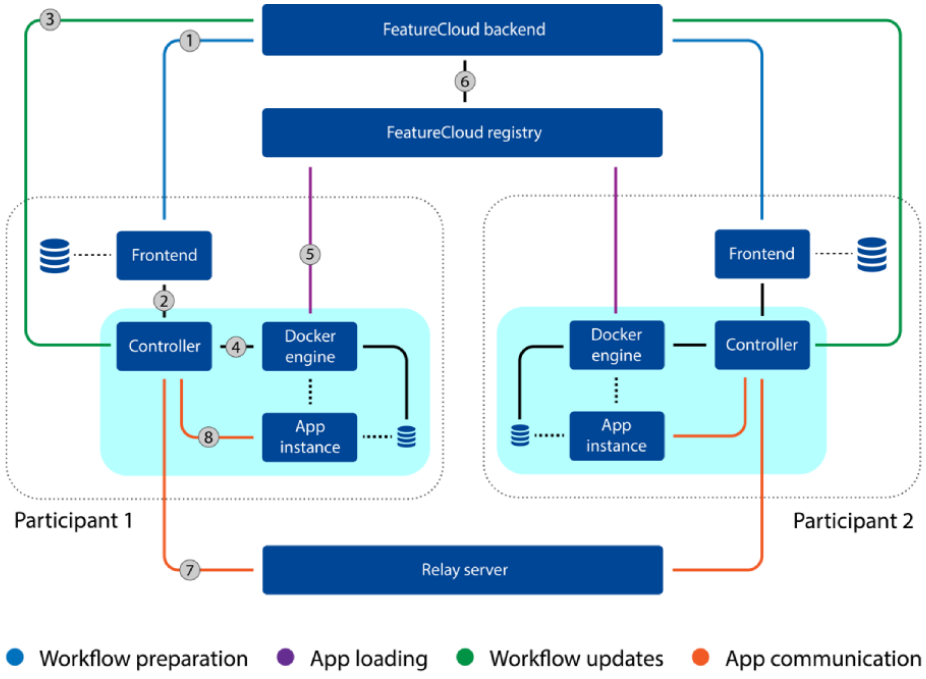
The frontend is a web application running on a web browser. It uses the FeatureCloud backend API (link 1 in Figure 2) to offer all the features of the AI store and for collaborative project management. It is also connected to the controller to allow for monitoring and handing over data for workflow runs (link 2 in Figure 2).

The controller is responsible for orchestrating the local part of the workflow execution. It receives information via the FeatureCloud backend API (link 3 in Figure 2), indicating which applications to execute next, and reports about the progress. Contrary to the relay server traffic, this traffic only contains metainformation about the execution and no data used in the algorithms themselves. It uses the Docker API (link 4 in Figure 2) to instruct the Docker engine to manage containers that serve as isolated application instances and pulls the images of the required applications for a workflow from the Docker registry (link 5 in Figure 2). When pushing new application versions, the Docker registry ensures that the user is entitled to do so by verifying their credentials through the backend (link 6 in Figure 2). In addition, the controller is an integral part of the security and privacy system of FeatureCloud. It handles local data processing and is the only part of FeatureCloud that has access to the local computer system. The controller runs in a Docker container to prevent random access to data on the system. Therefore, it only has access to selected data sets that were actively chosen by a system administrator or a user through a FeatureCloud application.

The participants of a federated workflow must also agree on a common relay server. The relay server, implemented in Go, is responsible for transmitting all traffic of the federated algorithms via a secure socket connection (link 7 in Figure 2). This central communication hub is aware of all the participants and their roles in the federated execution. It follows the required communication pattern, sending aggregated models to all the participants and local model parameters to the coordinating party only. Although FeatureCloud provides a relay server instance used by default, it is possible to use a private instance to completely shield the traffic from anyone outside the collaboration by adjusting the configuration file for the controller.

As FeatureCloud applications are a dynamic system component, partly contributed by external developers, it is necessary to isolate their implementation. This is achieved by using Docker, which ensures that they cannot access system resources other than required, especially the filesystem and network, and allows for limiting resource use, such as central processing unit or memory. They receive their input data inside a Docker volume and communicate with the *controller* through a defined API (link 8 in Figure 2). This API is the main interface between externally developed applications and the FeatureCloud system. It is http based and requires the application to act as a web server, which means that it needs to wait for the controller to query for data and cannot actively send data by itself; thus, active network access can be forbidden.

Figure 2. System architecture of FeatureCloud with 2 participants. The controller, frontend, Docker engine, and application instances run locally at each participant's site. The FeatureCloud backend and Docker registry are running on FeatureCloud servers. The relay server can be run on a separate server, or participants can use a provided instance from FeatureCloud. The components are connected via transmission control protocol/IP connections (straight lines). All links are http based, except for link 7, which uses a raw socket connection. Links 1 to 3 use JSON for serialization, and links 4 to 6 use the Docker application programming interface.



The FeatureCloud API for Developers

To avoid restricting end users to the current selection of applications, FeatureCloud invites external developers to implement their own federated applications and publish them in our AI store. A FeatureCloud application is a program isolated inside a Docker container that communicates with other instances using the FeatureCloud API [28]. Several templates and example applications are provided to further facilitate the implementation by directly explaining the API with code.

In addition to the AI store and the API, FeatureCloud provides tools to accelerate the development of federated applications. When developing a new federated method, application developers can directly start with the federation of the AI logic by using an existing template. To verify that the API has been implemented correctly, a simulation tool aids the developer in testing their application before publishing. Each test run specifies the number of participants, test data, and communication channels and subsequently starts the corresponding instances, simulating a real-world execution on multiple machines. During the test run, it shows logs and results for each participant and the network traffic to monitor the execution and identify bugs and potential communication bottlenecks.

After the development phase, applications can be published in the FeatureCloud AI store. Developers need to fill out a form prompting all relevant information about the application, which is displayed to the end users and used for the search and filter functions. Subsequently, they can push their Docker image into the Docker registry of the FeatureCloud platform. For end users collaborating with the developer, who explicitly enables uncertified applications, it is already usable and can be tested in a real-world scenario. For other end users, we enforce a certification process to increase the hurdle for malicious applications and maintain high privacy standards in the AI store. To this end, the developer must provide the necessary documentation and details regarding the implemented privacy mechanism. Furthermore, the application's source code must be accessible so that the application can be exhaustively tested and vetted by the FeatureCloud team and community for possible privacy leaks. When the certification process has been successfully completed by a member of the FeatureCloud consortium according to a defined checklist (Multimedia Appendix 1), the application will be displayed in the AI store and can be used by all end users. If the certification process is unsuccessful, the developer is notified and requested to address the issues raised. Upon each update of an application, a new certification procedure is triggered.

As FeatureCloud does not impose restrictions on the types of algorithms it supports, the running environment of the federated applications is kept very general. It allows the implementation of any type of ML algorithm and an optional custom GUI for user interaction in the form of a web-based frontend. This GUI can be used to receive input parameters, indicate the current progress, or display the results. No direct internet access is granted to the applications to avoid security risks.

FL Scheme and PETs

FL generally involves two possibly alternating operations as follows: (1) local optimization and (2) global aggregation. In FeatureCloud, all running instances of a federated application have 1 of 2 roles (participant and coordinator) performing the respective federated operation. FeatureCloud expects precisely 1 coordinator and an arbitrary number of participants, leading to a star-based architecture. We chose this architecture over others because it mirrors the general design of a FL scheme with a central aggregator and clients with local data sets.

After the local learning operation has been completed by a participant, it sends the local parameters to the coordinator. The coordinator collects these parameters and aggregates them into a collective (global) model, which is shared with the participants again. Depending on the type of ML algorithm, these 2 operations can alternate multiple times, for example, until convergence or a predefined number of iterations has been reached (Figure S1 in Multimedia Appendix 1). For some algorithms (eg, random forest [RF] and linear regression), only 1 iteration is necessary. However, this strict separation between optimization and aggregation is not actively enforced by FeatureCloud. In many cases, aggregation can start after the first parameters have been received, thereby increasing efficiency through parallelization of the computation. During the implementation of a federated application, the distinction between the coordinator and the participant is of conceptual relevance. However, in practice, the coordinator can also obtain local data that can be used for training. Therefore, FeatureCloud allows the coordinator to simultaneously adopt the role of a participant.

Although FL improves privacy, it can still leak information to the coordinator, who can see all individual models before aggregating them. Local updates of the model based on a previously distributed global model may reveal information regarding the primary data [29]. Secure aggregation techniques can address this problem. In FeatureCloud, we integrated additive secret sharing as a mitigation method to obtain the global sum without revealing the local submodels. Application developers can use this method with minimal or no added complexity to their algorithms. More details can be found in Multimedia Appendix 1.

Federated Algorithms

Comparing Federated Algorithms

As there are unique challenges for federating individual algorithms, each ML model needs to be developed independently and, therefore, needs to be based on a different underlying federation mechanism. This means that each algorithm has challenges regarding effectiveness, privacy, or scalability that need to be solved by the application developers. For the evaluation of our platform in this work, we used 4 FeatureCloud FL applications: the linear and logistic regression applications, a RF, and a DL application.

Federated Linear and Logistic Regression

For the implementation of the linear and logistic regression applications, the methods introduced by Nasirigerdeh et al [17]

have been adapted from genome-wide association studies (GWAS) to a general ML use case. For linear regression, the local X^TX and X^TY matrices are computed by each participant individually, where X is the feature matrix and Y is the label vector. Then, they are sent to the coordinator, aggregating the local matrices to the global matrices by adding them. Using these global matrices, the coordinator can calculate the beta vector through the federated method in such a way that it is identical to the beta vector calculated through the nonfederated method.

Logistic regression was implemented as an iterative approach. On the basis of the current beta vector, the local gradient and Hessian matrices of each participant are calculated and shared with the coordinator in each iteration. The coordinator aggregates the matrices again by adding them, updates the beta vector, and broadcasts it back to the participants. This process is repeated until convergence or the maximum number of iterations (prespecified for each execution) is achieved.

Internally, the scikit-learn model API has been used to implement the applications [30,31]. In the performance evaluation, we used the default scikit-learn hyperparameters for the linear regression models. For logistic regression, the penalty was set to none; the maximum number of iterations was set to 10,000; and the "lbgs" solver was used to fit the models.

Federated RF

We used the popular RF classifier and RF regressor as the second algorithm for our evaluation. As an ensemble algorithm, RF can be easily federated in a naive manner [32]. Our implementation trains multiple classification or regression decision trees on the local primary data of each participant. The fitted trees are then transmitted to the coordinator and merged into a global RF. To account for the different number of samples for each participant, each of them contributes a portion of the merged RF proportional to the number of samples. To achieve a similar behavior as the centralized implementation, the size of the merged RF is kept constant, meaning that an increasing number of participants decreases the number of required trees per participant. The federated computation occurs in three steps, each involving data exchange as follows: (1) participants indicate the number of samples and receive the total number of samples; (2) participants train the required number of trees, and the aggregator merges them into a global RF; and (3) participants receive the aggregated model to evaluate its performance on their data and share the results to obtain a global summary.

As the aim is not to achieve the highest possible accuracy but to compare the federated version with the nonfederated version, the hyperparameters were set to the default values of sklearn, namely, 100 decision trees, Gini impurity minimization as the splitting rule, and feature sampling equal to the square root of the features. Prepruning parameters such as maximum depth, minimum samples per node, and other constraints were not applied.

Federated DL

Our federated DL application is based on the federated average algorithm [6]. In the training phase, the weights and biases update is performed iteratively, where each iteration implies the parameter aggregation performed in three steps as follows: (1) the local weights and biases are computed by every participant individually and shared with the coordinator, (2) the coordinator averages the parameters and broadcasts them back to participants, and (3) the participants receive the new values of weights and biases and update the weights and biases of their model accordingly. After the final number of iterations is reached, the model performance of each participant is independently assessed using their data. The local weights and biases update is performed with the back-propagation algorithm, applied to data batches of a specified size. The neural network model architecture and training were implemented using the PyTorch library [33]. The application enables the implementation of any architecture and provides a centralized version of a PyTorch code. The application also enables federated transfer learning to be applied to a pretrained model, whose specified layers are trained in the same federated fashion.

Results

The results comprise the unified platform and an evaluation demonstrating the technical capabilities of FeatureCloud to run different workflows. The platform consists of the open AI store, development and debugging tools, and an execution environment for federated workflows.

Unified Platform

The unified platform (Figure 1) provides developers with an API to quickly develop privacy-enhancing FL applications. This supports a hybrid communication scheme for FL and secure aggregation (additive secret sharing). The integrated AI store is the interface between developers and end users, displaying and describing all available applications. Developers can publish (deploy) their applications in the AI store that are then available for use in federated workflows for the end users, for example, biomedical researchers. They can quickly create projects, assemble federated workflows with the applications from the AI store, invite other sites to the study, and view and download the results of each run. The interface of end users with the complicated federated architecture is reduced to only a web frontend and the FeatureCloud controller, running in the background and responsible for the local processing of sensitive data. Moreover, all applications and the entire architecture of FeatureCloud are open source, making it the first unified and open-source FL platform that considers all steps including development, deployment, and execution.

AI Store

The integrated AI store provides an intuitive and user-friendly interface for biomedical researchers and developers. It offers a variety of applications and displays basic information about them, including short descriptions, keywords, end-user ratings, and certification status. Users can easily find applications of interest via a textual search and filter them by type (preprocessing, analysis, and evaluation) and their privacy-enhancing techniques (FL, DP, and HE). End users can review the applications and provide feedback. The application pages display a method summary, description, user reviews, developer name, and contact details to report bugs. Each

application provides either a GUI or a configuration file to set the application parameters and adapt them to different contexts. This reduces technical details and makes applications user friendly for end users, independent of their background. When users add applications to their library, they can assemble them into a workflow and manage the execution with other collaborators on the FeatureCloud website without having to download any additional software.

The AI store has a broad selection of popular ML models, as listed in Table 1. The applications are categorized into preprocessing, analysis, and evaluation. Some analysis applications, such as linear regression and RF, are generic and suitable for different data types and application scenarios. These applications can be easily integrated into a federated workflow with preprocessing and evaluation applications, such as a federated standardization of the input data and a final evaluation of the trained classifier with several performance metrics. Other applications, such as the sPLINK [17] application for federated GWAS, integrate all the necessary steps of an application-specific workflow and do not require combination with other applications.

Table 1. Applications in the FeatureCloud artificial intelligence (AI) store[a].

Application	Type	Description
Ada boost	Machine learning	Classification model based on boosting trees
CACS[b] forest	Machine learning	Random forest classifying patients into their CACS
Cox PH[c] model	Survival analysis	Survival regression based on the lifelines library
Cross-validation	Preprocessing	Local splits for a k-fold cross-validation
Deep learning	Machine learning	Deep neural networks implemented in PyTorch
Evaluation (Classification)	Evaluation	Evaluation with various classification metrics (eg, accuracy)
Evaluation (Regression)	Evaluation	Evaluation with various regression metrics (eg, mean squared error)
Evaluation (survival)	Evaluation	Evaluation of survival or time-to-event predictions
Flimma	Differential expression	Differential expression analysis based on limma-voom
Graph-guided random forest	Machine learning	Random forest classification, regression, and survival based on graphs
Kaplan-Meier estimator	Survival analysis	Survival function estimation and log-rank test
Linear regression	Machine learning	Regression model
Logistic regression	Machine learning	Classification model
Nelson-Aalen estimator	Survival analysis	Hazard function estimation and log-rank test
Normalization	Preprocessing	Standardizing input data
One-hot encoder	Preprocessing	One-hot encoding for categorical variables
Random forest	Machine learning	Classification and regression model based on decision trees
Random survival forest	Survival analysis	Survival prediction based on scikit-survival
SVD[d]	Machine learning	SVD for dimensionality reduction
sPLINK[e]	GWAS[f]	GWAS based on PLINK
Survival SVM[g]	Survival analysis	Survival prediction based on scikit-survival

[a]The growing list of applications available in the AI store covers preprocessing, analysis, and evaluation. All-in-one applications cover the entire workflow for a more specific domain and can be executed without other applications.

[b]CACS: coronary artery calcification score.

[c]PH: proportional hazard.

[d]SVD: singular value decomposition.

[e]sPLINK: secure PLINK.

[f]GWAS: genome-wide association studies.

[g]SVM: support vector machine.

Multi-institutional Federated Workflows

FeatureCloud offers easy project management for the execution of FL workflows. In these workflows, users can select from a large variety of applications in the AI store and connect them to the entire workflow. Before collectively running a federated workflow, all collaborating sites (participants) must download and start the client-side FeatureCloud controller on their machines. It only requires Docker, which is freely available for all the major operating systems. Users also need to create an

account on the FeatureCloud website, which serves as a web frontend and is used to coordinate the FeatureCloud system (refer to the *Methods* section and Multimedia Appendix 1 for details on the architecture). Each collaborative execution of applications is organized into so-called projects on the web frontend. They contain a description of the planned analysis, connect the collaborating partners by allowing invited participants to join, and show the current status of the workflow (Figure S2 in Multimedia Appendix 1).

Workflows are composed of 1 or multiple applications from the AI store that are to be executed consecutively. Each application produces intermediate results that serve as input for the consecutive application. Intermediate results are maintained on the respective machines and are not shared with other participants. The last application produces the final results, which are then shared with all the project participants. During the execution of a workflow, its progress can be monitored on the FeatureCloud website, showing the current stage, computational progress, and intermediate results from each application. Applications can provide their own user interface, allowing for user interaction if necessary and for showing specific reports. Users can monitor application logs and react in case something unexpected occurs (eg, stop and rerun the workflow with other data or a different configuration). When the last application in the workflow successfully completes its computation, the final results are automatically shared with all project participants. Intermediate results and application logs remain available on the local machines to allow for later verification. For example, the results may include a report showing the effectiveness of the trained model and the model itself. The latter can also be used outside of FeatureCloud. For example, if a project fails because a participant drops out, it can be restarted quickly after the problem has been solved. During the entire process, no programming knowledge or command-line interaction is required, making the system especially suited for medical personnel without technical education.

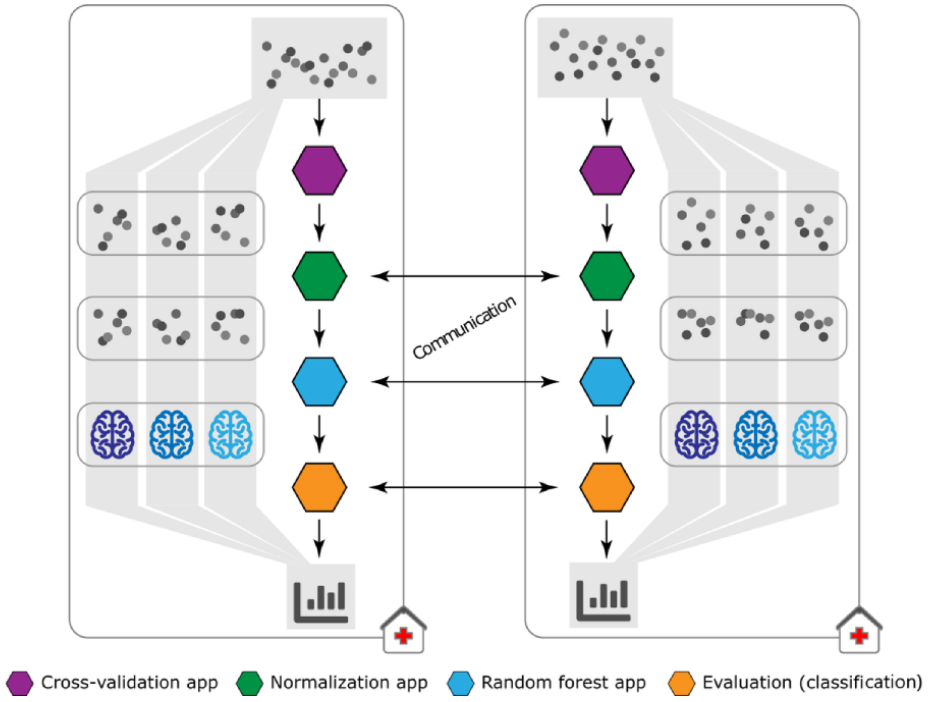
Evaluation

Methods and Data Sets

To evaluate the practical applicability of FeatureCloud, multiple workflows operating on different data sets were created. Except for DL, each workflow consists of a cross-validation (CV) application (10-fold CV), a standardization application, a model training application, and a final evaluation application (Figure 3). For DL, we evaluated a 20% test set, as this is more common for big data to reduce the training time. Individual applications are data-type agnostic and are suitable for various applications. Classification analyses were performed on the Indian Liver Patient Dataset [34] with 579 samples and 10 features and the Cancer Genome Atlas Breast Invasive Carcinoma [35] data set with 569 samples and 20 features. For regression analyses, they were evaluated on the Diabetes [36] data set with 442 samples and 10 features and the Boston [37] house prices data set with 506 samples and 13 features, both provided by scikit-learn [30]. Finally, for DL regression, we used a large data set from the Survey of Health, Aging, and Retirement in Europe [38], with 12 questionnaire variables and the target 12-item critical assessment of protein structure prediction quality of life score. After dropping samples with “Refusal” and “Don’t know” type values in those 12 variables and nonavailable 12-item critical assessment of protein structure prediction quality of life score, we were left with 42,894 (91.79%) out of 46,733 samples. Further details regarding the network architecture are provided in Multimedia Appendix 1.

For each workflow, we split the central data set into 5 participants with uneven data distribution. Participants 1, 2 and 3, and 4 and 5 each had 10% (4289/42,894), 15% (6434/42,894), and 30% (12,868/42,894) of the samples, respectively. We used the F_1-score to evaluate the classification models and the root mean squared error for the regression models, as both are common metrics used to evaluate ML models. Furthermore, we also investigated the scalability concerning runtime and network traffic for 2 to 8 participants as well as a larger number of participants and iterations.

Figure 3. Workflow structure used for evaluation. The first application (purple—Cross-Validation) creates splits for cross-validation (CV). All following applications perform their tasks on each split individually, in a federated fashion, only transmitting model parameters. The gray dots represent intermediate training and test data. The second application (green—"Normalization") performs normalization, and the third application (blue—"Random Forest") trains the models, generating a global model based on the output of the normalization application. The resulting global model is evaluated in the evaluation application (orange—"Evaluation [Classification]"). The evaluation results are finally aggregated to obtain an evaluation report based on the initial CV splits.



Performance

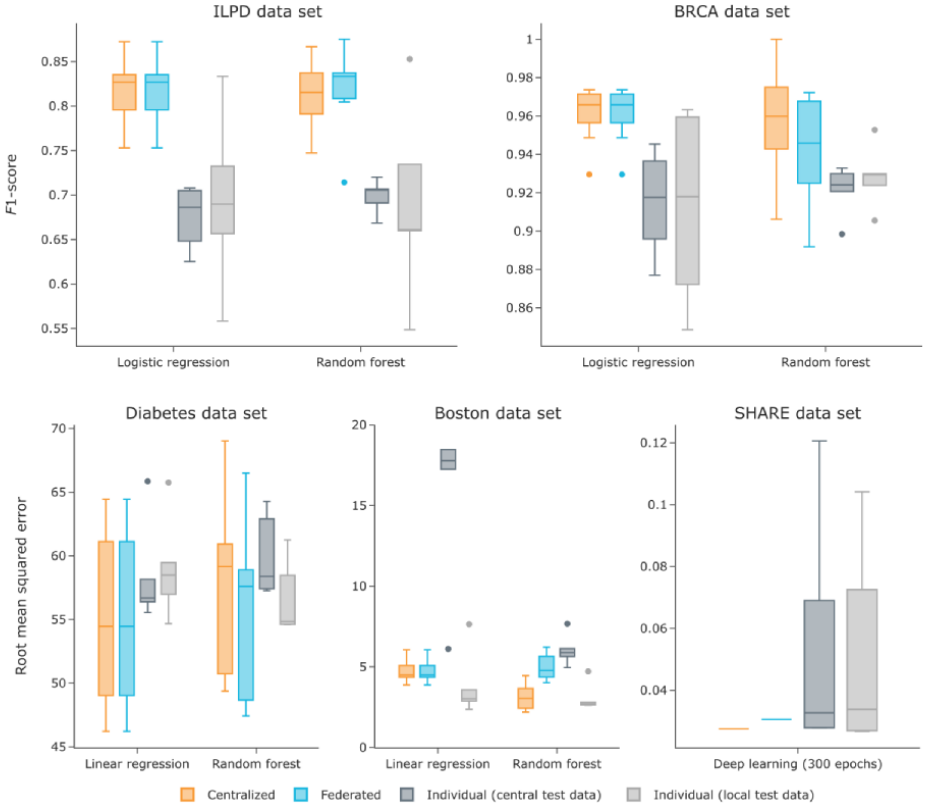
Previous studies have shown that FL can achieve similar performance to centralized learning in many scenarios [14,15,39]. To verify the approach used in FeatureCloud, we compared the performance of 4 federated FeatureCloud applications integrated into an ML workflow with their corresponding centralized scikit-learn [30] models. The results are shown in Figure 4. For logistic regression and linear regression, the FeatureCloud workflow achieved a performance identical to that of scikit-learn, which is consistent with the previous results of federated linear and logistic regression applications [17,40]. A similar performance was achieved for the RF regression and classification models. Owing to the simple aggregation method that combines the local trees into 1 global tree, identical results were not obtained or expected. Owing to the bootstrapping mechanism and its attached randomness, the federated RF sometimes performs slightly better than the centralized approach. As a final example, our federated DL model trained in 300 epochs produced a very close root mean squared error compared with the centralized model.

Furthermore, we compared the federated models with the individual models trained and evaluated by each participant (10-fold CV, except DL). Here, we distinguish between the central evaluation of the models on the overall test splits (central test data), identical to the test splits for the centralized and federated models, and the local evaluation of the models on the local test splits only (local test data). As shown in Figure 4, the local evaluation performance varies widely but is worse on average than the federated models. For classification, the local evaluation performed worse than the federated models. However, for the regression models, the locally evaluated models of the individual participants sometimes outperformed the centralized model. Nevertheless, compared with the central test data, it is obvious that these models did not generalize well and only performed well for the individual participants with a very small test set. This can be deceptive, as in this case, even the 10-fold CV cannot be trusted. Furthermore, our DL model evaluated on a 20% test set performs much more reliably than individual client models, which can have drastically worse results than the federated or centralized models. This highlights the effectiveness of FL, as these models use more training and test data, resulting in more generalized models. Our RF application is based on a previously published implementation [32] and confirms that our platform yields comparable results, including scenarios in which the data are neither independent nor identically distributed (nonindependent and identically distributed). It performed much more reliably than only using individual client data.

As an additional example of clinical data analysis, we evaluated the Kaplan-Meier estimator application that implements an already published approach for federated survival curves and a log-rank test for multi-institutional time-to-event analyses [18]. The application, implemented and run in FeatureCloud, produced identical results to the centralized analysis (Table S1 in Multimedia Appendix 1) on the lung cancer data set of the North Central Cancer Treatment Group [41]. Similarly, we evaluated the Flimma application for differential gene expression analysis [16] as an example of biomedical data on a subset of 152 breast cancer expressions from the Cancer Genome Atlas

repository [42] with 20,536 features. Our Flimma application produced highly similar results to those of the centralized analysis (Figure S3 in Multimedia Appendix 1). These 2 examples further show that FeatureCloud has the capability of implementing and running different approaches and bringing them into a production system.

Figure 4. Performance evaluation of federated artificial intelligence methods. The box plots show the results of a 10-fold cross-valuation for the different classification and regression models and data sets in multiple settings. Only the deep learning model was evaluated on a test set. The centralized results are shown in orange, the corresponding federated results in blue, and the individual results obtained locally at each participant in gray. Each model was evaluated on the entire test set (dark gray) such as the centralized and federated models and on the individual (local) parts of the test set (light gray). The federated logistic and linear regressions perform in identical fashion to their centralized versions, and the federated random forest and deep learning models perform in similar fashion to their centralized versions. BRCA: Breast Invasive Carcinoma; ILDP: Indian Liver Patient Dataset; SHARE: Survey of Health, Aging and Retirement in Europe.



Runtime and Network Traffic

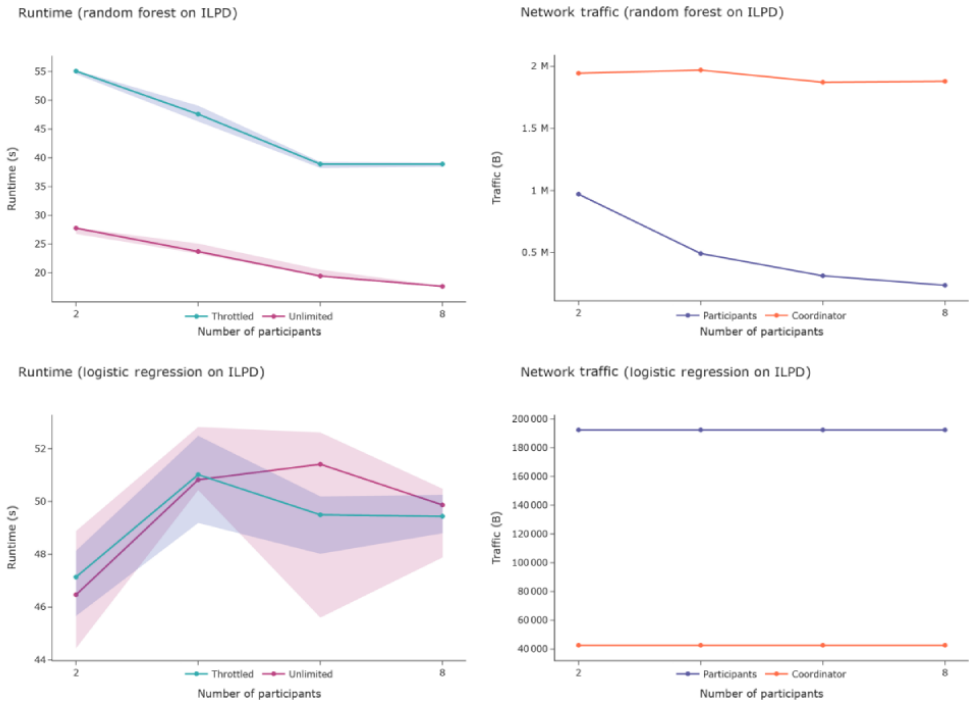
Multiple executions with varying numbers of clients were performed to assess the scalability of the FeatureCloud platform and the federated methods. RF and linear regression classifiers were chosen as the iterative and noniterative methods, respectively, and both were applied to the Indian Liver Patient Dataset. Both were tested with 2, 4, 6, and 8 clients and the same number of samples to ensure comparability across the executions. To investigate the impact of network bandwidth on runtime, all executions were performed on a normal and throttled internet connection with a maximum transmission of 100 kB per second.

Figure 5 shows that runtime mildly increases for logistic regression but decreases for RF. This is because the logistic regression models are of equal size for all clients, whereas the size of the RF models depends on the number of trees. In our implementation of federated RF, the global model is of a fixed size (100 trees), which means that each client contributes a portion that decreases with a higher number of participants. The throttling bandwidth significantly increases the runtime for RF but leaves the runtime for logistic regression almost unaffected. This is because the transmitted data for RF are more extensive and come in 1 chunk, whereas logistic regression requires approximately 10 iterations, each exchanging a few parameters. The centralized versions take 2 to 3 seconds to complete for both logistic regression and RF, implying that their federated versions take 10 to 20 times longer to complete.

In this setting, an increasing number of participating parties has a weak impact on the duration of the aggregation part for these methods, compared with the total runtime. The local computations occur in parallel such that an increasing number of participants does not have a huge impact. However, because the aggregation step cannot be completed before all participants send their models, the runtime of each aggregation step depends on the slowest participant, which poses a potential problem for large federations. FeatureCloud primarily focuses on being used in a tightly regulated medical research environment. Therefore, there is currently no automatic “matchmaking” in place, but all participants must join each project actively. In this context, running an analysis with data sets of >8 participants is still an uncommon scenario. To demonstrate its scalability and robustness for more sophisticated scenarios, we evaluated the

FeatureCloud platform using the logistic regression application for 1, 5, 10, 15, 20, 25, and 30 clients on simulated data, with each client containing 1000 samples and 1, 5, and 10 iterations. Our analysis shows that the FeatureCloud platform is also computationally suitable for larger numbers of clients and higher numbers of iterations, confirming the results of our runtime analysis for a small number of clients (Figure S4 in Multimedia Appendix 1).

Figure 5. Runtime and network traffic. The left plots show runtime for unlimited and throttled connections, the right plots show network traffic for the coordinator and participants evaluated on the ILPD. The lines represent the median values measured over 10 executions. The areas show the 25% and 75% quartiles to illustrate variance across the executions. ILPD: Indian Liver Patient Dataset; s:second; B: byte; M: million.



Discussion

In this section, we summarize our main findings and provide a discussion about its comparison with prior work, its limitations, the potential for future work, and conclusions of our work.

Summary of Results

In this study, we presented the FeatureCloud platform, a comprehensive platform for the application and development of privacy-preserving FL workflows in biomedicine and beyond. Through its high generalization, it allows the application of various ML workflows to a variety of data types. In addition, it offers prebuilt solutions for common-use cases in the form of applications in the AI store or application templates for developers. The concept of freely composing applications in a workflow is challenging because of the need for a standard data format, which is not always available and can reduce flexibility. The same applies to the initial data, which need to be provided in a form that is processable and understandable by the desired application. As FL adaptation is still in its early stages, it is necessary to understand which functionality and types of data will be used, which ML techniques prove to be most prevalent in federated settings, and which challenges arise when using the platform. Therefore, several assumptions can be made in advance.

Comparison With Prior Work

One main goal of FeatureCloud was to keep the platform as flexible and extensible as possible, to align new functionality closely to the demand of its users. The possibility of integrating additional PETs, such as DP or additive secret sharing, on the application layer of the API demonstrates the versatility of this approach. Although the current implementation of additive secret sharing has a quadratic increase in network traffic, it shows that flexible communication can be achieved through asymmetrical encryption and can serve as a blueprint for similar scenarios and future developments.

The prediction performance of our FL workflows is consistent with the current research, with some performing equally well compared with the central implementations (linear and logistic regression and normalization) or highly similar (RF). Computational and communication overheads are acceptable for an ordinary FL. In our opinion, it plays a smaller role than the additional overhead related to human-to-human coordination of federated projects. We demonstrated that the currently available applications and the platform scale well for up to 8 participants.

The main novelty, in contrast to prior work, is the high flexibility of the AI store, ranging from prebuilt task-centered applications, such as GWAS, to generic method-centered applications, such as RF. Therefore, we address a broad spectrum of end users and developers. Less experienced users without deeper

methodological or statistical knowledge benefit from the ease of use of a task-centered application. Advanced users can tailor the workflow to their needs. In contrast, application developers can use our API to develop FL applications that can be easily deployed into the AI store and reach a broad user base. They are incentivized to build their applications to be compatible with existing ones (eg, a new AI method that processes data preprocessed by an existing normalization application) to maximize their utility. Thus, the FeatureCloud AI store aims to become an ecosystem for FL, driving collaborative research.

Limitations

In addition to the huge potential of FeatureCloud, some issues still need to be addressed. Our secure aggregation approach, directly implemented into the developer API, only applies to ≥3 participants. Its application on workflows with only 2 participants would allow the coordinator to reveal the local parameters of the other participant and therefore has no benefit. In addition, as it is currently implemented, our additive secret-sharing approach only supports addition and multiplication and is, therefore, not applicable to more complex types of calculations. Although the open AI store accelerated the development and deployment of FL applications and workflows, it is the responsibility of the application developers to provide proof that their implementations provide accurate results. FeatureCloud certifies applications that provide a reasonable amount of privacy and security measures but cannot check the prediction quality of every application. However, through its open-source design, the community can exchange experiences, provide feedback, and enhance applications and algorithms to keep them up to date with the current state of the art.

Future Work

The generic and extendable design of FeatureCloud makes it highly interesting for future studies. FeatureCloud envisions being driven by an emerging community whose features are closely aligned to their needs. As FeatureCloud is entirely open source, it can be quickly maintained and extended and it can accelerate the development, deployment, and execution of privacy-preserving FL workflows in biomedicine and other areas. FeatureCloud applications can be developed by anyone using the developer API and easy-to-start templates. One part could focus on integrating more PETs into the API for the application developers to ease their use and increase adoption in federated algorithms. Although FeatureCloud already integrates an additive secret-sharing scheme, there are many more PETs, such as DP or HE schemes, that can be implemented. Other potential enhancements could focus on nonlinear workflows, the integration of the AIMe registry [43] into the certification process of FeatureCloud applications, and reducing Docker dependency by also supporting other secure containerization systems such as Singularity [44]. To address the problem of data harmonization and preprocessing of different formats at different sites, it may be useful to add a federated database with a common ontology to the FeatureCloud controller [45]. Through this, the problem of different data formats between sites is solved, as the input data for workflows can be directly created from the database. Integrating local data into this database can be performed using predefined Extract-Transform-Load scripts for the most common data formats and standards.

Conclusions

In conclusion, FeatureCloud provides an all-in-one platform for privacy-preserving FL. In contrast to other FL frameworks, FeatureCloud considers every aspect of FL from development and deployment to the execution and project planning of federated analyses. Furthermore, it is highly generic to support all types of algorithms and is not restricted to only DL or a certain application. Thus, we believe that it has a huge potential to accelerate the development of FL workflows and the application of federated analyses in biomedicine.

Acknowledgments

The FeatureCloud project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement 826078. This publication reflects only the authors' view, and the European Commission is not responsible for any use that may be made of the information it contains. This work was further supported by the German Federal Ministry of Education and Research (BMBF)–funded German Network for Bioinformatics Infrastructure (de.NBI) Cloud within the de.NBI (031A532B, 031A533A, 031A533B, 031A534A, 031A535A, 031A537A, 031A537B, 031A537C, 031A537D, and 031A538A). This work was further supported by the German BMBF within the framework of "CLINSPECT-M" (grant FKZ161L0214A). This work was partly developed as part of the Center for Female Artificial Intelligence project funded by the German BMBF under grant 01IS21079. This work was further funded by the German BMBF under grant 16DTM100A. JKP was funded by the Bavarian State Ministry of Education and the Arts in the framework of the Bavarian Research Institute for Digital Transformation (bidt, grant LipiTUM). This study uses data from the Survey of Health, Aging and Retirement in Europe (SHARE) Wave 8 (refer to the study by Börsch-Supan et al [38] for methodological details). The SHARE data collection has been funded by the European Commission, Directorate-General for Research and Innovation (DG RTD) through FP5 (QLK6-CT-2001-00360), FP6 (SHARE-I3: RII-CT-2006-062193, COMPARE: CIT5-CT-2005-028857, and SHARELIFE: CIT4-CT-2006-028812), FP7 (SHARE-PREP: GA N°211909, SHARE-LEAP: GA N°227822, SHARE M4: GA N°261982, and Data Service Infrastructure for the Social Sciences and Humanities (DASISH): GA N°283646), and Horizon 2020 (SHARE-DEV3: GA N°676536, SHARE-COHESION: GA N°870628, SERISS: GA N°654221, SSHOC: GA N°823782, and SHARE-COVID19: GA N°101015924); and by DG Employment, Social Affairs & Inclusion through VS 2015/0195, VS 2016/0135, VS 2018/0285, VS 2019/0332, and VS 2020/0313. Additional funding from the German Ministry of Education and Research, the Max Planck Society for the Advancement of Science, the US National Institute on Aging (U01_AG09740-13S2, P01_AG005842, P01_AG08291, P30_AG12815,

R21_AG025169, Y1-AG-4553-01, IAG_BSR06-11, OGHA_04-064, HHSN271201300071C, and RAG052527A), and various national funding sources is gratefully acknowledged [46].

Data Availability

The Survey of Health, Aging and Retirement in Europe (SHARE) data are distributed by SHARE-European Research Infrastructure Consortium (ERIC) to registered users through the SHARE Research Data Center. We used only data from the 8 waves [38]. Except for the SHARE data, all our data sets, including the Indian Liver Patient Dataset [34], Breast Invasive Carcinoma data set [35], Boston data set [37], and Diabetes data set [36], and scripts used for our evaluation results are available in our GitHub repository [47]. To increase interpretation and reproducibility, we followed the minimum information about clinical artificial intelligence modeling (ML-CLAIM) reporting standard (Norgeot et al [48]). The filled-out ML-CLAIM clinical checklist is also available in our GitHub repository.

Conflicts of Interest

None declared.

Multimedia Appendix 1

Additional information containing figures and descriptions related to software architecture and implementation.
[DOCX File , 1262 KB-Multimedia Appendix 1]

References

1. Mamoshina P, Vieira A, Putin E, Zhavoronkov A. Applications of deep learning in biomedicine. Mol Pharm 2016 May 02;13(5):1445-1454 [doi: 10.1021/acs.molpharmaceut.5b00982] [Medline: 27007977]
2. Yu KH, Beam AL, Kohane IS. Artificial intelligence in healthcare. Nat Biomed Eng 2018 Oct;2(10):719-731 [doi: 10.1038/s41551-018-0305-z] [Medline: 31015651]
3. Malle B, Giuliani N, Kieseberg P, Holzinger A. The more the merrier - federated learning from local sphere recommendations. In: Proceedings of the 1st International Cross-Domain Conference on Machine Learning and Knowledge Extraction. 2017 Presented at: CD-MAKE '17; August 29-September 1, 2017; Reggio, Italy p. 367-373 URL: https://link.springer.com/chapter/10.1007/978-3-319-66808-6_24 [doi: 10.1007/978-3-319-66808-6_24]
4. Esteva A, Kuprel B, Novoa RA, Ko J, Swetter SM, Blau HM, et al. Corrigendum: dermatologist-level classification of skin cancer with deep neural networks. Nature 2017 Jun 28;546(7660):686 [doi: 10.1038/nature22985] [Medline: 28658222]
5. Chan HC, Shan H, Dahoun T, Vogel H, Yuan S. Advancing drug discovery via artificial intelligence. Trends Pharmacol Sci 2019 Oct;40(10):801 [doi: 10.1016/j.tips.2019.07.013] [Medline: 31451243]
6. McMahan HB, Moore E, Ramage D, Hampson S, Arcas BA. Communication-efficient learning of deep networks from decentralized data. In: Proceedings of the 20th International Conference on Artificial Intelligence and Statistics. 2017 Presented at: AISTATS '17; April 20-22, 2017; Ft. Lauderdale, FL, USA p. 1273-1282 URL: http://proceedings.mlr.press/v54/mcmahan17a?ref=https://githubhelp.com
7. Kairouz P, McMahan HB, Avent B, Bellet A, Bennis M, Nitin Bhagoji A, et al. Advances and open problems in federated learning. Found Trends Mach Learn 2021 Jun 23;14(1–2):1-210 [FREE Full text] [doi: 10.1561/2200000083]
8. Tomsett R, Chan KS, Chakraborty S. Model poisoning attacks against distributed machine learning systems. In: Proceedings of the 2019 SPIE Defense and Commercial Sensing. 2019 Presented at: SDCS '19; April 14-18, 2019; Baltimore, MD, USA URL: https://www.spiedigitallibrary.org/conference-proceedings-of-spie/11006.toc [doi: 10.1117/12.2520275]
9. Usynin D, Ziller A, Makowski M, Braren R, Rueckert D, Glocker B, et al. Adversarial interference and its mitigations in privacy-preserving collaborative machine learning. Nat Mach Intell 2021 Sep 17;3(9):749-758 [FREE Full text] [doi: 10.1038/s42256-021-00390-3]
10. Acar A, Aksu H, Uluagac AS, Conti M. A survey on homomorphic encryption schemes: theory and implementation. ACM Comput Surv 2019 Jul 31;51(4):79 [FREE Full text] [doi: 10.1145/3214303]
11. Bonawitz KA, Ivanov V, Kreuter B, Marcedone A, McMahan HB, Patel S, et al. Practical secure aggregation for privacy-preserving machine learning. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. 2017 Presented at: CCS '17; October 30-November 3, 2017; Dallas, TA, USA p. 1175-1191 URL: https://dl.acm.org/doi/10.1145/3133956.3133982 [doi: 10.1145/3133956.3133982]
12. Dwork C, McSherry F, Nissim K, Smith A. Calibrating noise to sensitivity in private data analysis. In: Proceedings of the 3rd Theory of Cryptography Conference. 2006 Presented at: TCC '06; March 4-7, 2006; New York, NY, USA p. 265-284 URL: https://link.springer.com/chapter/10.1007/11681878_14 [doi: 10.1007/11681878_14]
13. Dwork C, Roth A. The algorithmic foundations of differential privacy. Found Trends Theor Comput Sci 2014 Aug 11;9(3-4):211-407 [FREE Full text] [doi: 10.1561/0400000042]

14. Nilsson A, Smith S, Ulm G, Gustavsson E, Jirstrand M. A performance evaluation of federated learning algorithms. In: Proceedings of the 2nd Workshop on Distributed Infrastructures for Deep Learning. 2018 Presented at: DIDL '18; December 10-11, 2018; Rennes, France p. 1-8 URL: https://dl.acm.org/doi/10.1145/3286490.3286559 [doi: 10.1145/3286490.3286559]
15. Lee GH, Shin SY. Federated learning on clinical benchmark data: performance assessment. J Med Internet Res 2020 Oct 26;22(10):e20891 [FREE Full text] [doi: 10.2196/20891] [Medline: 33104011]
16. Zolotareva O, Nasirigerdeh R, Matschinske J, Torkzadehmahani R, Bakhtiari M, Frisch T, et al. Flimma: a federated and privacy-aware tool for differential gene expression analysis. Genome Biol 2021 Dec 14;22(1):338 [FREE Full text] [doi: 10.1186/s13059-021-02553-2] [Medline: 34906207]
17. Nasirigerdeh R, Torkzadehmahani R, Matschinske J, Frisch T, List M, Späth J, et al. sPLINK: a hybrid federated tool as a robust alternative to meta-analysis in genome-wide association studies. Genome Biol 2022 Jan 24;23(1):32 [FREE Full text] [doi: 10.1186/s13059-021-02562-1] [Medline: 35073941]
18. Späth J, Matschinske J, Kamanu FK, Murphy SA, Zolotareva O, Bakhtiari M, et al. Privacy-aware multi-institutional time-to-event studies. PLOS Digit Health 2022 Sep;1(9):e0000101 [FREE Full text] [doi: 10.1371/journal.pdig.0000101] [Medline: 36812603]
19. Ryffel T, Trask A, Dahl M, Wagner B, Mancuso J, Rueckert D, et al. A generic framework for privacy preserving deep learning. arXiv Preprint posted online on November 9, 2018 [FREE Full text] [doi: 10.48550/arXiv.1811.04017]
20. Konczyk J. Federated Learning with TensorFlow. Birmingham, UK: Packt Publishing; 2019.
21. Train on the edge with federated learning. XayNet. URL: https://www.xaynet.dev/ [accessed 2023-05-12]
22. Yang L, Tao F, Tianjian C, Qian X, Qiang Y. An industrial grade federated learning framework. The Journal of Machine Learning Research 2021 Aug:10320-10325 [FREE Full text]
23. Gazula H, Kelly R, Romero J, Verner E, Baker BT, Silva RF, et al. COINSTAC: collaborative informatics and neuroimaging suite toolkit for anonymous computation. J Open Source Softw 2020 Oct 25;5(54):2166-2169 [FREE Full text] [doi: 10.21105/joss.02166]
24. Silva S, Altmann A, Gutman B, Lorenzi M. Fed-BioMed: a general open-source frontend framework for federated learning in healthcare. In: Proceedings of the 2nd MICCAI Workshop on Domain Adaptation and Representation Transfer, and Distributed and Collaborative Learning. 2020 Presented at: DART '20 and DCL '20; October 4–8, 2020; Lima, Peru p. 201-210 URL: https://link.springer.com/chapter/10.1007/978-3-030-60548-3_20 [doi: 10.1007/978-3-030-60548-3_20]
25. Owkin. URL: https://owkin.com/ [accessed 2023-05-12]
26. Melloddy. URL: https://www.melloddy.eu/ [accessed 2023-05-12]
27. FeatureCloud - Privacy-Preserving AI. URL: https://featurecloud.ai [accessed 2023-06-02]
28. FeatureCloud AI developer API (1.1.0). FeatureCloud AI. URL: https://featurecloud.ai/assets/api/redoc-static.html [accessed 2023-05-12]
29. Lyu L, Yu H, Yang Q. Threats to federated learning: a survey. arXiv Preprint posted online on March 4, 2020 [FREE Full text] [doi: 10.48550/arXiv.2003.02133]
30. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: machine learning in Python. J Mach Learn Res 2011;12(85):2825-2830 [FREE Full text]
31. Buitinck L, Louppe G, Blondel M, Pedregosa F, Mueller A, Grisel O, et al. API design for machine learning software: experiences from the scikit-learn project. In: Proceedings of the 2013 European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases. 2013 Presented at: ECML PKDD '13; September 23-27, 2013; Prague, Czech Republic p. 108-122
32. Hauschild AC, Lemanczyk M, Matschinske J, Frisch T, Zolotareva O, Holzinger A, et al. Federated random forests can improve local performance of predictive models for various healthcare applications. Bioinformatics 2022 Apr 12;38(8):2278-2286 [doi: 10.1093/bioinformatics/btac065] [Medline: 35139148]
33. Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, et al. PyTorch: an imperative style, high-performance deep learning library. In: Proceedings of the 32nd Conference on Neural Information Processing Systems. 2019 Presented at: NeurIPS '19; December 8-14, 2019; Vancouver, Canada URL: https://proceedings.neurips.cc/paper/2019/hash/bdbca288fee7f92f2bfa9f7012727740-Abstract.html
34. UC Irvine Machine Learning Repository. URL: https://doi.org/10.24432/C5D02C [accessed 2023-05-12]
35. Street WN, Wolberg WH, Mangasarian OL. Nuclear feature extraction for breast tumor diagnosis. In: Proceedings of the 1993 IS&T/SPIE'S Symposium on Symposium on Electronic Imaging: Science and Technology. 1993 Presented at: IS&T '93; January 31-February 5, 1993; San Jose, CA, USA URL: https://www.spiedigitallibrary.org/conference-proceedings-of-spie/1905/1/Nuclear-feature-extraction-for-breast-tumor-diagnosis/10.1117/12.148698.short [doi: 10.1117/12.148698]
36. Efron B, Hastie T, Johnstone I, Tibshirani R. Least angle regression. Ann Stat 2004 Apr;32(2):407-451 [FREE Full text] [doi: 10.1214/009053604000000067]
37. Harrison Jr D, Rubinfeld DL. Hedonic housing prices and the demand for clean air. J Environ Econ Manage 1978 Mar;5(1):81-102 [FREE Full text] [doi: 10.1016/0095-0696(78)90006-2]
38. Börsch-Supan A. Survey of Health, Ageing and Retirement in Europe (SHARE) Wave 8. COVID-19 Survey 1. Release version: 8.0.0. Survey of Health, Ageing and Retirement in Europe (SHARE). 2022 Feb 10. URL: https://share-eric.eu/data/data-set-details/share-corona-survey-1 [accessed 2023-05-12]

39. Sheller MJ, Edwards B, Reina GA, Martin J, Pati S, Kotrotsou A, et al. Federated learning in medicine: facilitating multi-institutional collaborations without sharing patient data. Sci Rep 2020 Jul 28;10(1):12598 [FREE Full text] [doi: 10.1038/s41598-020-69250-1] [Medline: 32724046]
40. McMahan B, Ramage D. Federated learning: collaborative machine learning without centralized training data. Google Research. 2017 Apr 06. URL: https://ai.googleblog.com/2017/04/federated-learning-collaborative.html [accessed 2023-05-12]
41. Loprinzi CL, Laurie JA, Wieand HS, Krook JE, Novotny PJ, Kugler JW, et al. Prospective evaluation of prognostic variables from patient-completed questionnaires. north central cancer treatment group. J Clin Oncol 1994 Mar;12(3):601-607 [doi: 10.1200/JCO.1994.12.3.601] [Medline: 8120560]
42. Liu J, Lichtenberg T, Hoadley KA, Poisson LM, Lazar AJ, Cherniack AD, Cancer Genome Atlas Research Network, et al. An integrated TCGA pan-cancer clinical data resource to drive high-quality survival outcome analytics. Cell 2018 Apr 05;173(2):400-16.e11 [FREE Full text] [doi: 10.1016/j.cell.2018.02.052] [Medline: 29625055]
43. Matschinske J, Alcaraz N, Benis A, Golebiewski M, Grimm DG, Heumos L, et al. The AIMe registry for artificial intelligence in biomedical research. Nat Methods 2021 Oct;18(10):1128-1131 [FREE Full text] [doi: 10.1038/s41592-021-01241-0] [Medline: 34433960]
44. Kurtzer GM, Sochat V, Bauer MW. Singularity: scientific containers for mobility of compute. PLoS One 2017 May 11;12(5):e0177459 [FREE Full text] [doi: 10.1371/journal.pone.0177459] [Medline: 28494014]
45. Sheth AP, Larson JA. Federated database systems for managing distributed, heterogeneous, and autonomous databases. ACM Comput Surv 1990 Sep 01;22(3):183-236 [FREE Full text] [doi: 10.1145/96602.96604]
46. Survey of Health, Aging and Retirement in Europe. URL: https://share-eric.eu/ [accessed 2023-05-12]
47. Matschinske J, Späth J. Evaluation - FeatureCloud. GitHub. URL: https://github.com/FeatureCloud/evaluation [accessed 2023-05-12]
48. Norgeot B, Quer G, Beaulieu-Jones BK, Torkamani A, Dias R, Gianfrancesco M, et al. Minimum information about clinical artificial intelligence modeling: the MI-CLAIM checklist. Nat Med 2020 Sep;26(9):1320-1324 [FREE Full text] [doi: 10.1038/s41591-020-1041-y] [Medline: 32908275]

Abbreviations

AI: artificial intelligence
API: application programming interface
CV: cross-validation
DL: deep learning
DP: differential privacy
FL: federated learning
GUI: graphical user interface
GWAS: genome-wide association studies
HE: homomorphic encryption
ML: machine learning
PET: privacy-enhancing technology
RF: random forest

Edited by T Leung; submitted 12.09.22; peer-reviewed by A Brauneck, P Mitrovic, A Wang, Z Zrubka, J Seth; comments to author 20.12.22; revised version received 13.01.23; accepted 26.02.23; published 12.07.23

Please cite as:
Matschinske J, Späth J, Bakhtiari M, Probul N, Kazemi Majdabadi MM, Nasirigerdeh R, Torkzadehmahani R, Hartebrodt A, Orban BA, Fejér SJ, Zolotareva O, Das S, Baumbach L, Pauling JK, Tomašević O, Bihari B, Bloice M, Donner NC, Fdhila W, Frisch T, Hauschild AC, Heider D, Holzinger A, Hötzendorfer W, Hospes J, Kacprowski T, Kastelitz M, List M, Mayer R, Moga M, Müller H, Pustozerova A, Röttger R, Saak CC, Saranti A, Schmidt HHHW, Tschohl C, Wenke NK, Baumbach J
The FeatureCloud Platform for Federated Learning in Biomedicine: Unified Approach
J Med Internet Res 2023;25:e42621
URL: https://www.jmir.org/2023/1/e42621
doi: 10.2196/42621
PMID:

A.3 Publication 3

Original Paper

Privacy-Preserving Federated Survival Support Vector Machines for Cross-Institutional Time-To-Event Analysis: Algorithm Development and Validation

Julian Späth[1], MSc; Zeno Sewald[2], BSc; Niklas Probul[1], MSc; Magali Berland[3], PhD; Mathieu Almeida[3], PhD; Nicolas Pons[3], PhD; Emmanuelle Le Chatelier[3], PhD; Pere Ginès[4,5,6,7], MD, PhD; Cristina Solé[4,5,6], MD; Adrià Juanola[4,5,6], MD, PhD; Josch Pauling[2], PhD; Jan Baumbach[1], Prof Dr

[1]Institute for Computational Systems Biology, University of Hamburg, Hamburg, Germany

[2]LipiTUM, Chair of Experimental Bioinformatics, TUM School of Life Sciences, Technical University of Munich, Freising, Germany

[3]MetaGenoPolis, INRAE, Université Paris-Saclay, Jouy-en-Josas, France

[4]Liver Unit, Hospital Clínic de Barcelona, Barcelona, Spain

[5]Institut d'Investigacions Biomèdiques August Pi i Sunyer (IDIBAPS), Barcelona, Spain

[6]Centro de Investigacion en Red de Enfermedades hepaticas y Digestivas (CIBEReHD), Madrid, Spain

[7]Faculty of Medicine and Health Sciences, University of Barcelona, Barcelona, Spain

Corresponding Author:
Julian Späth, MSc
Institute for Computational Systems Biology
University of Hamburg
Notkestrasse 9
Hamburg, 22607
Germany
Phone: 49 15750665331
Email: julian.alexander.spaeth@uni-hamburg.de

Abstract

Background: Central collection of distributed medical patient data is problematic due to strict privacy regulations. Especially in clinical environments, such as clinical time-to-event studies, large sample sizes are critical but usually not available at a single institution. It has been shown recently that federated learning, combined with privacy-enhancing technologies, is an excellent and privacy-preserving alternative to data sharing.

Objective: This study aims to develop and validate a privacy-preserving, federated survival support vector machine (SVM) and make it accessible for researchers to perform cross-institutional time-to-event analyses.

Methods: We extended the survival SVM algorithm to be applicable in federated environments. We further implemented it as a FeatureCloud app, enabling it to run in the federated infrastructure provided by the FeatureCloud platform. Finally, we evaluated our algorithm on 3 benchmark data sets, a large sample size synthetic data set, and a real-world microbiome data set and compared the results to the corresponding central method.

Results: Our federated survival SVM produces highly similar results to the centralized model on all data sets. The maximal difference between the model weights of the central model and the federated model was only 0.001, and the mean difference over all data sets was 0.0002. We further show that by including more data in the analysis through federated learning, predictions are more accurate even in the presence of site-dependent batch effects.

Conclusions: The federated survival SVM extends the palette of federated time-to-event analysis methods by a robust machine learning approach. To our knowledge, the implemented FeatureCloud app is the first publicly available implementation of a federated survival SVM, is freely accessible for all kinds of researchers, and can be directly used within the FeatureCloud platform.

(JMIR AI 2024;3:e47652) doi: 10.2196/47652

KEYWORDS

federated learning; survival analysis; support vector machine; machine learning; federated; algorithm; survival; FeatureCloud; predict; predictive; prediction; predictions; Implementation science; Implementation; centralized model; privacy regulation

Introduction

Accessing data to apply machine learning (ML) in biomedical settings is still challenging [1]. Large amounts of data exist in clinical settings but are scattered across numerous institutions. Due to strict privacy regulations, such as the General Data Protection Regulation (GDPR), this data cannot be easily shared or collected at a central institution [2]. This causes hurdles for cross-institutional biomedical analyses that depend on highly sensitive patient data. One example is time-to-event analysis, aiming to find parameters that prolong or shorten the time until a particular event, such as death, occurs [3]. In these studies, the event of interest does not necessarily occur for all samples, increasing the need for large sample sizes [4]. Until today, the need for large sample sizes and heterogeneous data for time-to-event studies is still mainly solved through traditional data sharing, leading to the central collection of various deidentified and anonymized data sets from different centers. Since using anonymized data in the training of ML models tends to weaken model performance [5], this comes with a tradeoff of data privacy and data quality, accelerating the need for alternative methods that keep data private and ensure the quality of the data [6].

In recent years, federated learning (FL) has become a feasible alternative to central data collection by enabling the training of models on distributed data sets. Instead of sharing sensitive data with a central institution, in FL, only insensitive model parameters are shared with a central aggregation server [7,8]. Therefore, each participating party calculates its own model with local model parameters on their local data. These local model parameters are then shared with the aggregator and aggregated into a global model. Afterward, the global model is shared again with each participant and can be updated in another iteration. The first and probably most widely used aggregation approach is the federated average [9], calculating the weighted mean of the exchanged model parameters. Besides using different aggregation approaches, FL can also be distinguished between horizontal and vertical learning, as well as cross-device and cross-silo learning. Horizontal learning describes FL on data with the same features but different samples, while vertical learning performs on the same samples but with different features between the participating parties. Cross-device FL trains models across millions of participants (such as mobile phones), cross-silo FL, on the other hand, focuses on a few clients only, such as hospitals or research institutes [10].

Especially in combination with privacy-enhancing techniques (PETs), model parameters can be exchanged securely, such that a global aggregator or potential attacker cannot even see the local parameters of each participant [11]. This secure exchange of model parameters is necessary to comply with the GDPR, as even local models can be considered personal data [12]. Therefore, FL enables the training on a significantly larger data set compared with single-institution scenarios. While federated algorithms still often struggle with communication efficiency, the significantly increased amount of data can offset this performance issue, making FL a serious competitor to classical ML. Additionally, since FL models are trained on a larger variety of data, they typically generalize better than traditional ML models and even generalize faster in some cases [13,14]. Many FL approaches are already published for biomedical applications, such as medical imaging analysis, genome-wide association studies, or gene expression analysis [15-17].

In addition to federated ML approaches, several federated time-to-event analysis algorithms have been introduced recently and confirmed their high potential for privacy-preserving analyses [18-21]. However, existing approaches solely cover traditional statistical methods such as the estimation of survival functions and the Cox proportional hazards model. Modern ML algorithms for survival analysis, such as survival Support Vector Machines (SVMs), are not yet available in a federated fashion, even though SVMs belong to one of the most popular ML methods. If algorithms are not available in federated scenarios, this might be a reason why researchers chose not to perform FL, if their favorite algorithms are not available. Many well-performing centralized algorithms are challenging to translate to a federated scenario while keeping sensitive data private. Another limitation of FL is communication efficiency. FL algorithms need to exchange the intermediate statistics with a central aggregator, which is especially inefficient for algorithms with many iterations. This inefficiency even increases when adding secure aggregation schemes, such as additive secret sharing. This PET ensures that only masked and encrypted model parameters are shared with the aggregating party, securing the local models from data leakage [18].

To address the lack of availability of federated time-to-event methods, we propose a privacy-preserving, horizontally federated, cross-silo survival SVM based on the survival analysis package *scikit-survival* [22]. Compared with other existing time-to-event methods, such as the Cox proportional hazard model, the survival SVM allows an actual prediction of the time until an event happens. It can be used to predict the risk of individual samples, which is not possible in univariate time-to-event algorithms and is not the aim of the Cox proportional hazards model. Therefore, to the best of our knowledge, it is the first freely available federated survival prediction method. We implemented the algorithm as an app in the FeatureCloud platform to make it publicly accessible and to minimize the hurdles of FL infrastructure [23]. Based on a combination of FL and additive secret sharing, we show on 3 benchmark data sets, that our approach achieves highly similar results compared with central data analysis. Additionally, we apply it to a set of real-world microbiome data sets to demonstrate its applicability to original clinical data.

Methods

Here, we propose the adapted algorithm for the federated survival SVM, describe its implementation as a FeatureCloud app, and explain how we evaluated its performance.

Federated Survival SVM

We extended the regression objective of *scikit-survival*'s FastSurvivalSVM without ranking to be applicable in federated environments [24]. As shown in Figure 1, instead of calculating the sum of the squared ζ-function centrally, it is calculated at each site, with the feature vector x_i, the survival time $y_i>0$, and the binary event indicator δ_i. Each site's local sum of squared ζ-function is then sent to a global aggregator and summed up to the global sum of squared ζ-function. The below equations show the central objective function and our corresponding federated objective function, with C being the set of all participating clients.

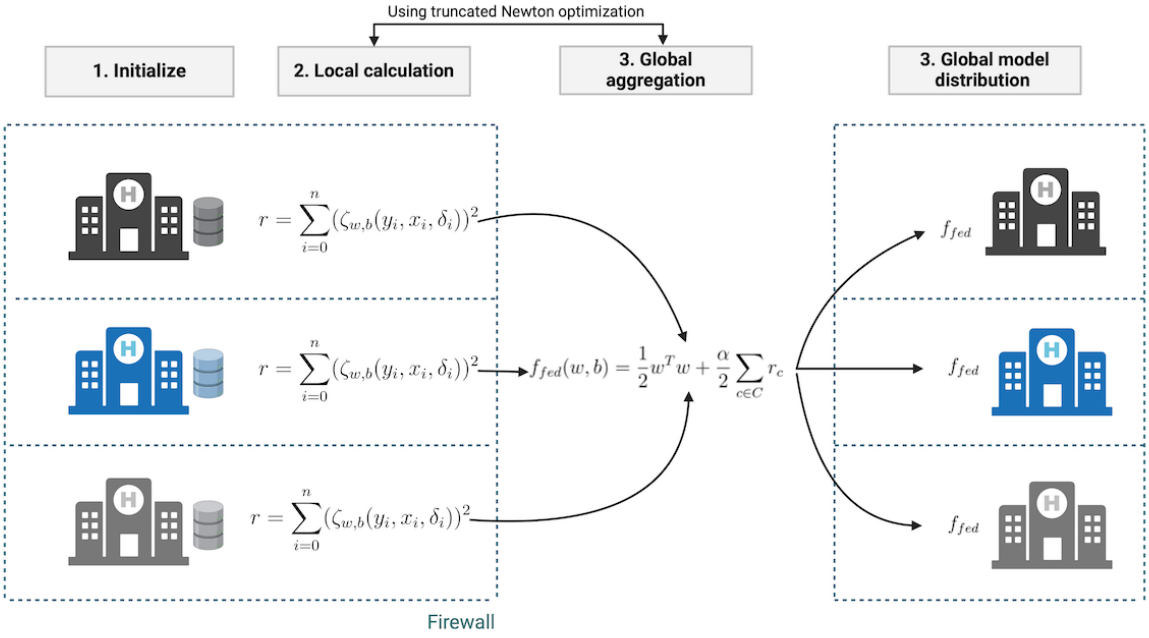
$$argmin_{w,b}\ f_{central}(\boldsymbol{\omega}, b) = \frac{1}{2}\boldsymbol{\omega}^T\boldsymbol{\omega} + \frac{\alpha}{2}\sum_{i=0}^{n}\left(\zeta_{\boldsymbol{\omega},b}(\boldsymbol{y_i}, \boldsymbol{x_i}, \boldsymbol{\delta_i})\right)^2$$

$$argmin_{w,b}\ f_{fed}(\omega, b) = \frac{1}{2}\boldsymbol{\omega}^T\boldsymbol{\omega} + \frac{\alpha}{2}\sum_{c\in C}\sum_{i=0}^{n_c}\left(\zeta_{\boldsymbol{\omega},b}(\boldsymbol{y_{c,i}}, \boldsymbol{x_{c,i}}, \boldsymbol{\delta_{c,i}})\right)^2$$

$$\zeta_{\boldsymbol{\omega},b}(\boldsymbol{y_{c,i}}, \boldsymbol{x_{c,i}}, \boldsymbol{\delta_{c,i}}) = \begin{cases} \max(0, \boldsymbol{y_i} - \boldsymbol{w}^T\boldsymbol{x_i} - b) & if\ \boldsymbol{\delta_i} = 0, \\ \boldsymbol{y_i} - \boldsymbol{w}^T\boldsymbol{x_i} - b & if\ \boldsymbol{\delta_i} = 1 \end{cases}$$

Mathematically, our federated formula leads to the same solution as the centralized calculation of the objective function. Similar to the centralized analysis, a truncated Newton method (such as Newton-CG) can be used to optimize the objective function. For this, in each iteration, the gradient and Hessian matrix of each client are also sent to the global aggregator to sum them up to the global gradient and Hessian matrix. To reduce potential privacy leakage from the exchanged data, the implementation of the federated algorithm should support a secure aggregation scheme that hides the locally exchanged data from attackers or the global aggregation server.

Figure 1. Federated calculation of a survival support vector machine (SVM). Each site calculates the sum of squares locally and sends it to the global aggregation server. The aggregation server aggregates the local sum of squares by summing them up to the global sum of squares. The objective function is minimized in a federated fashion by a truncated Newton approach. After convergence, the global model is distributed to all participating clients.



FeatureCloud

We developed an FL app on the FeatureCloud platform to make our approach publicly available. To develop this app, we used the app template and application programming interface provided by FeatureCloud [25]. Using the *scikit-survival* package and Python, we implemented our algorithm, put it into the FeatureCloud app template, and published it in the FeatureCloud artificial intelligence store. It can be used with other apps in a workflow or standalone using the platform. Our code is entirely open source.

In FeatureCloud, 1 participating client also takes the aggregating role and is called the coordinator. The app is implemented as a state machine, meaning that the app switches between states to perform different tasks. All states and their transitions are shown in Multimedia Appendix 1. After reading the local data and config files, minimizing the objective function using a federated Newton conjugate gradient is performed iteratively. Therefore, the local gradient and Hessian matrices are calculated and sent to the coordinator. The coordinator aggregates these data to obtain the global matrices, updates the weight vector ω, and broadcasts it to all clients. This is repeated until convergence.

A considerable advantage of the FeatureCloud platform is its native support of 2 very popular PETs, such as secure multiparty computation (SMPC). For applying SMPC, FeatureCloud supports a secure aggregation scheme for hiding locally exchanged parameters using additive secret sharing [26]. Through this, the exchanged local models are protected, and only the global aggregations are visible to attackers, clients, and the global aggregator. This is achieved by splitting the value that needs to be exchanged with the global aggregator into *n* shards, where *n* is the number of participating clients, and the sum of these *n* shards would result in the actual value [23]. Each shard is encrypted using a public key of each participant. These encrypted shards are shared with the global aggregator, sending them to the corresponding client holding the private key. The clients decrypt the received shards, sum them up, and send them back to the global aggregator, which sums up all received sums. This final sum results in the actual, nonhidden, global aggregate.

Ethical Considerations

According to German regulations, for our retrospective study performed on publicly available data or data with explicit consent, approval from an ethical committee was not required.

Evaluation

We evaluated our approach using the developed FeatureCloud app on 3 benchmark data sets, all available via the *scikit-survival* package. The breast cancer data set (BRCA) [27] contains the gene expression profiling of microarray experiments from 198 primary breast tumors, originally used to validate a 76-gene prognostic signature able to predict distant metastases in lymph node–negative patients with breast cancer. The German Breast Cancer Study Group 2 data set (GBSG2) [28] contains data from a multicenter randomized clinical trial to compare the effectiveness of 3 versus 6 cycles of cyclophosphamide, methotrexate, and fluorouracil on recurrence-free and overall survival of 686 women. The observed parameters were hormonal therapy (yes or no), age of the patients, menopausal status (pre vs post), tumor size (in mm), tumor grade, number of positive tumor nodes, progesterone receptor (in fmol), and estrogen, as well as the censoring indicator and recurrence-free survival time (in days). The Worcester Heart Attack Study data set (WHAS500) [29] contains data from 500 patients with acute myocardial infarction, collected during thirteen 1-year periods. Parameters were age, gender, initial heart rate, initial systolic and diastolic blood pressure, body mass index, history of cardiovascular disease, atrial fibrillation, cardiogenic shock, congestive heart complications, complete heart block, myocardial infarction order and type, vital status, and total length of follow-up.

Additionally, we evaluated our algorithm on a recent, high-dimensional gut microbiome data set from the Hospital Clinic of Barcelona, containing data from 150 patients with liver cirrhosis [30]. The data set was aimed at assessing the predicting role of the gut microbiome for the survival of the patients in the context of liver cirrhosis, using shotgun metagenomic sequencing performed on fecal DNA isolated from stool samples. A former version of the data has been previously analyzed with a different methodology [30]. For this study, the Metagenomic Species Pangenome (MSP) was used to identify and quantify microbial species associated with the IGC2 reference catalog [31]. MSPs are clusters of coabundant genes (minimum size >100 genes) used as a proxy for microbial species, reconstructed from 1601 metagenomes to 1990 MSP species [32]. MSP abundances were estimated as the mean abundance of their 100 marker genes, as far as at least 20% of these genes are detected. The MSP abundance table was then normalized in each sample by dividing its abundance by the sum of MSP abundances detected in the sample. Further details regarding the data sets are shown in Table 1.

Table 1. Overview of all data sets. Our 4 evaluation data sets differ greatly in the number of samples, features, events, and censored individuals. Features indicate the number of clinical variables or microbial species abundance in the data set; median follow-up indicates the median follow-up time of the patients in days; events indicate the number of patients for whom the event of interest was observed during observation time; and censored indicates the number of patients for whom the event of interest was not observed during observation time.

Data set	Samples, n	Features, n	Median follow-up (days)	Events, n	Censored, n	End point
BRCA	198	84	4384.0	51	147	Presence of metastases
GBSG2	686	11	1084.0	299	387	Recurrence-free survival
WHAS500	500	16	631.5	215	285	Death
Microbiome	150	1995	416.0	51	99	Death

[a]BRCA: breast cancer data set.

[b]GBSG2: German Breast Cancer Study Group 2 data set.

[c]WHAS500: Worcester Heart Attack Study data set.

We one-hot encoded nonbinary categorical features. For each data set, we created either 1 client (100%) as the centralized scenario, 3 clients (20%, 50%, and 30%) as the multicentric imbalanced scenario, and 5 clients (20% each) as the multicentric balanced scenario, and we split the data accordingly.

To evaluate the accuracy of our model, we used the Harrell concordance index, which was developed as a generalization of the area under the receiver operating characteristic curve for time-to-event models [33]. It corresponds to the probability of concordance between observed and predicted survival based on each pair of individuals. A c-index of 0.5 means that the model

performs as well as a random guess, and a c-index of 1.0 means that the model predicts perfectly well.

After preprocessing, we performed a 3 × 3-fold cross-validation (CV) for a FeatureCloud workflow consisting of a federated normalization, the federated survival SVM, and a federated survival evaluation (c-index). We then compared our results with the centralized analysis of every client and the merged data set (simulating a central data collection). Centralized analysis was performed using *scikit-survival*'s FastSurvivalSVM with a rank ratio of 0, α of 0.0001, true fit intercept, and a maximum of 50 iterations. The same hyperparameters were used for the federated analysis, respectively.

Privacy

FeatureCloud supports several properties to increase the privacy and security of the computations. One important step is that FL projects can be only executed with invited participants. For this, a unique and secret code is needed to join the project. Every participant can see the workflow and each individually executed FeatureCloud app that will run in the workflow. As FeatureCloud apps are open source, even the executed code of the apps can be examined.

The execution of apps and workflows in FeatureCloud is containerized and strictly monitored. Due to the containerization, individual apps are not allowed to establish a connection to the internet, which prevents the extraction of data from malicious code. Even though the communication between clients does not contain sensitive patient information, it is RSA (Rivest–Shamir–Adleman) encrypted through the standard HTTPS protocol. This prevents unauthorized third parties from gaining insights into parameters exchanged during training.

Exchanged parameters from each individual site are masked through the secure aggregation scheme, hiding the intermediate statistics from other participating clients and the global aggregator. This efficiently addresses the problem of local models considered as personal data in GDPR [18].

Our federated survival SVM app currently uses a hybrid approach of SMPC and FL. This hybrid approach increases the privacy of the exchanged local parameters from both participants and potential attackers, as explained in the methods section.

Differential privacy (DP) [34] is not yet supported by FeatureCloud but is currently in development and could be added to the algorithm as an additional layer to improve privacy. However, as the app trains a linear model, it is less prone to overfit, reducing the surface for potential membership and attribute inference attacks [35]. In DP, noise is added to the model parameters during the training process to guarantee a mathematically quantifiable amount of privacy for each sample. While this comes with large advantages regarding privacy, the application of DP has also various weaknesses. The addition of noise lowers the performance of the model significantly, especially when applying the amount of noise necessary for a meaningful level of privacy [36]. Further, this guarantee only is applicable for a limited number of interactions with the resulting model. As the final model is distributed to all participants, they can interact with the model arbitrarily, making the privacy guarantee void, thus not warranting an inclusion in this analysis.

A PET not supported by FeatureCloud currently is homomorphic encryption (HE), which allows the computation of the model on encrypted values, making sharing of data even more secure. While this is great in theory, it actually gains very little benefit in this analysis scenario. The data we share is already nonsensitive and through the use of SMPC, we can hide not only the data but the data's origin. This is why FeatureCloud currently supports SMPC instead of HE.

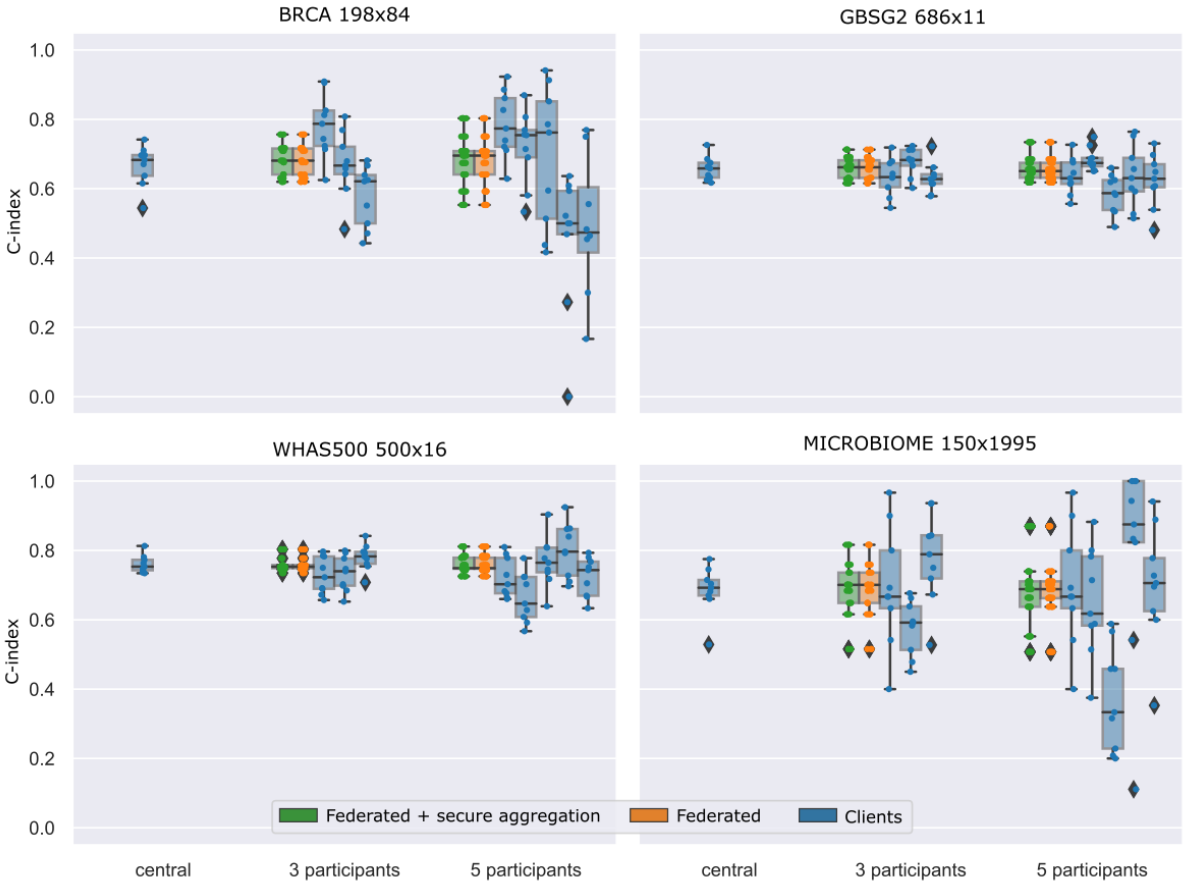
Our implementation of the federated survival SVM app uses all the functionalities offered by FeatureCloud and does not deviate from these best practices.

Results

Performance

Our workflow delivered a highly similar model performance and model parameters for all federated analyses compared with the ones performed on the corresponding centralized data sets. The resulting c-indices to estimate the performance of our time-to-event models are depicted in Figure 2 [33]. For each data set (subplot), we show a boxplot consisting of the evaluated c-index for each CV split of our federated workflow with secure aggregation (green), federated workflow without secure aggregation (orange), and centralized calculation for each individual client (blue). The CV results show that our federated as well as the federated and secure aggregation approach perform highly similar to the centralized estimates. The calculation of the federated c-index in FeatureCloud causes small deviances in the c-index between centralized and federated. This is because FeatureCloud calculates a local c-index and aggregates to the mean c-indices of all sites. Therefore, it does not lead to the same c-index as a central computation would. The mean c-indices for the 4 data sets are in the range between 0.658 (GSBG2) and 0.76 (WHAS500). In contrast to the accuracy, achieving very high c-indices is rather difficult and depends very much on the problem. In a bioinformatics context, the lowest c-index of 0.658 (GBSG2) can be considered as moderate. The model achieves discrimination between individuals with different survival outcomes. However, it might not be of clinical utility and needs further refinement. The c-index of 0.76 (WHAS500) on the other hand, can be considered as good and has predictive value. Improving the predictive value of the models and increasing c-index was out of the scope of this work. A complete table of the results is available in Multimedia Appendix 2.

Figure 2. Comparison of federated and centralized analysis. The boxplots show the evaluated c-indices (3 × 3-fold cross validation) of the central, 3 participants, and 5 participants analysis (rows). For each scenario, we compared the federated and secure aggregation approach (green), the federated-only approach (orange), and the performance of every single site (blue). BRCA: breast cancer data set; GBSG2: German Breast Cancer Study Group 2 data set; WHAS500: Worcester Heart Attack Study data set.



The model weights are nearly identical, with a maximum difference of only 0.001 and a mean difference of 0.0002 (Multimedia Appendices 1 and 3). These tiny differences between the weights of the central model and our model are negligible, as they do not change the overall prediction results and still lead to equal c-indices. The resulting model is therefore almost identical to the one that was trained on central data. A useful property of the linear survival SVM is, that the model weights can be used as a feature importance measure, which is also supported in our approach.

Besides calculating the feature importance from model weights directly, our federated survival SVM app uses Shapley additive explanations (SHAP), an explainable artificial intelligence framework for the interpretation of ML models [37]. Using SHAP, we compared the final models of the central, federated without secure aggregation, and federated with secure aggregation runs. For each data set, the SHAP shows highly similar model interpretations with a mean Pearson correlation of 0.991 between the central and the federated model without secure aggregation, and a mean Pearson correlation of 0.985 between the central model and the federated model with secure aggregation. A slightly worse correlation in the secure aggregation model is expected, as the masking of local parameters leads to floating-point issues. The worst correlation is shown in the microbiome data set (0.964), which can be explained by the high correlation between features in this data set. The results of the SHAP correlation analysis are listed in Multimedia Appendix 4 and the corresponding SHAP beeswarm plots are available in Multimedia Appendix 5.

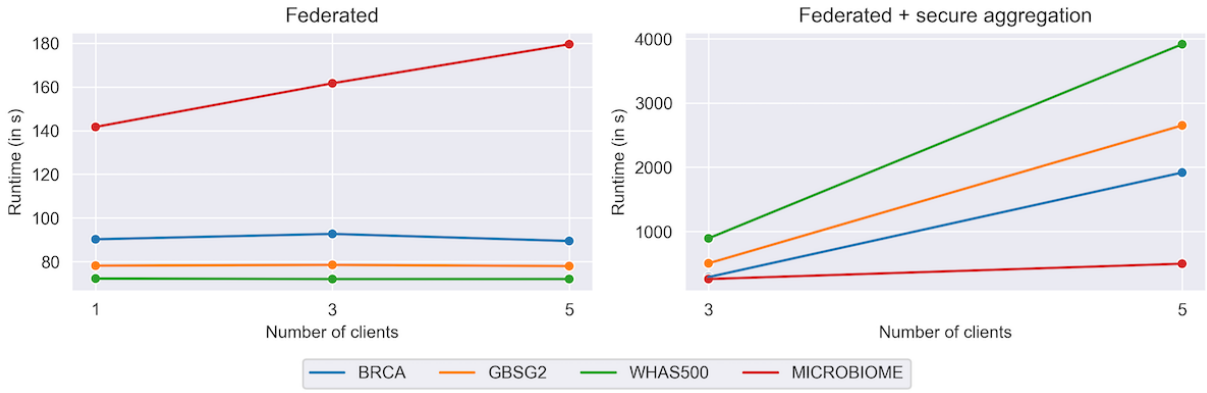
Our results further demonstrate the importance of large data sets, as the performance of the locally trained models on single clients (smaller sample size) shows a much higher variance than our federated models. If 5 institutes combine their small data sets, they can perform a much more reliable time-to-event analysis compared with isolated institutions. This further supports the high practical value of FL in real-world clinical time-to-event analysis, especially for institutions with small sample sizes, homogenous cohorts, or only a few patients with rare diseases.

Runtime

As shown in Figure 3, the runtime largely depends on the data set. In the case of FL, the number of iterations and, therefore, the number of data exchanges are the bottleneck. While the federated-only approach has linear runtime, the runtime of federated and secure aggregation is much worse and increases with an increasing number of clients. As described in the FeatureCloud publication, providing better privacy by hiding

the exchanged parameters from the global aggregator, the simple additive secret sharing grows quadratic with the number of participants. Especially when many iterations and data exchanges are needed, this has a bad influence on the runtime of the FL implementation.

Figure 3. Runtime analysis. The lines represent the runtime for each data set and the number of participating clients. The federated-only approach is depicted on the left, and the federated and secure aggregation approach is depicted on the right.



All results of the runtime analysis are shown in Multimedia Appendix 6. Additionally, we performed the runtime analysis on a data set with a large sample size. As real-world time-to-event data sets are difficult to find, we used a synthetically generated, published data set from an example colon data set with 15,564 samples [38]. Our results show that our method scales well for large sample sizes, as the number of iterations is the bottleneck in FL (Multimedia Appendix 7).

FeatureCloud App

The app we developed can easily be used within the FeatureCloud platform. For this, a project coordinator creates a project, selects the app, and invites collaborators. Each participant installs FeatureCloud and joins the project. The app expects 2 CSV files as input, one for the training data and another for the test data. A config file can be used to define hyperparameters and other descriptors, such as the time and event label columns. After the federated computation has finished, each client receives the globally trained model as a pickle file, as well as a prediction file containing all predictions on the local test data set. The app can also be used in a FeatureCloud workflow, supporting various preprocessing methods, such as CV, normalization, feature selection, one-hot encoding, and subsequent evaluation of survival models using the c-index.

The requirements for running the survival SVM app are the same as for executing the FeatureCloud platform. It requires a stable internet connection to exchange the incentive model parameters with the central aggregator and to run the app on the website. Docker needs to be installed on a Mac, Linux, or Windows computer with the corresponding requirements for running Docker [39]. Moreover, enough memory should be available to process the data set. This depends mainly on the data set size, and not on the algorithm itself.

Discussion

Principal Findings

Our federated survival SVM has been demonstrated to offer a highly viable alternative to centralized data collection in a time-to-event analysis. It achieves comparable levels of accuracy without compromising the privacy of highly sensitive patient data. This makes it a compelling solution for organizations seeking to safeguard sensitive data while still gaining the benefits of advanced analysis and the application of ML. Through its availability as a FeatureCloud app, the platform takes care of deployment and federated infrastructures, making it directly usable with little programming knowledge. The results of the real-world microbiome data set are promising and show that FL might be an accelerator in microbiome research and the analysis of time-to-event microbiome data sets. Using FL combined with additive secret sharing, our approach can be currently considered GDPR compliant and, therefore, practically usable in real clinical time-to-event studies [12].

Comparison to Existing Work

Only a few federated survival analysis approaches were developed in recent years, such as the distributed Cox proportional hazards model WebDISCO or an approach for federated survival curves using multiparty HE [18,20]. In a recent study about privacy-aware multi-institutional time-to-event analysis, it was criticized that the existing work was mainly focusing on theoretical solutions, rather than practical [21]. Therefore, lack of usability was a huge issue that was addressed by the authors, who developed the platform “Partea” [21]. The platform supports the Kaplan-Meier estimator for survival curve estimation [40], Nelson-Aalen estimator for cumulative hazard ratios [41], and Cox proportional hazards model for survival regression [42]. Compared with “Partea,” FeatureCloud does not only address the execution of FL algorithms, but also development. The FeatureCloud developer application programming interface for implementing FL algorithms that can be executed through FeatureCloud and published in the App Store is a huge advantage in terms of

development speed and also accessibility for the potential user group.

To our knowledge, the survival SVM FeatureCloud app is one of the first time-to-event analysis ML models implemented as a FL algorithm. This makes the accuracy (or c-index in our case) between the algorithms not directly comparable. However, similar to the existing solutions [20,21], our approach achieves almost identical results compared with the central algorithms.

Regarding runtime, univariate methods without iterations, such as Kaplan-Meier estimator, Nelson-Aalen estimator, or log-rank test are much more efficient in FL settings. However, these approaches cannot be used to analyze high dimensional data and multivariate settings. The efficiency of our approach is comparable to the iteratively trained Cox proportional hazard model, which is trained iteratively and requires communication and aggregation for every parameter update step.

Limitations

Our current approach does not support the more efficient ranking objective, as federated ranking is not trivial to implement. Instead, it is based on *scikit-survival*'s regression objective. Moreover, it solely supports the linear SVM and does not support the kernel SVM yet. Calculating a kernel matrix in a federated setting is not trivial, as it represents pairwise similarities (or distances) between the training data points. For supporting more complex, nonlinear relationships, this should be further investigated in the future. We still decided to implement and use a survival SVM in this work, as SVMs are very popular in health care and the only available time-to-event analysis ML model in *scikit-survival* that is not based on an ensemble approach. Ensemble models, such as random survival forests [43] or survival gradient boost, are both based on a set of survival trees. While ensemble models are also popular in time-to-event analysis, the federated aggregation of the local forests produces slightly worse results than centrally trained models in imbalanced scenarios [44]. A federated aggregation of each local tree, on the other hand, is computationally costly. The SVM in our implementation produces highly accurate results compared with central learning for model weights, c-index, and feature importance and can therefore lower the burden of applying FL in health care (eg, microbiome analysis), as the participants can be sure that the results are equal to the ones they would obtain in a central setting.

FeatureCloud currently only supports a simple additive secret-sharing scheme, increasing runtime for calculations with many clients and iterations. This could be solved in the future by using a more efficient secret-sharing scheme, such as Shamir secret sharing, that is currently not supported by FeatureCloud [45]. By using FeatureCloud as the execution platform, our approach does not solve the still existing open problems of FL, such as fairness, debugging, and communication efficiency (especially when using secret sharing) [46]. Furthermore, there are attacks on FL architectures that cannot be prevented through the existing methods, such as privacy inference from the global model, and model or data poisoning [47]. It is therefore recommended to use the algorithms and FeatureCloud platform only with trusted parties.

Another limitation that comes from the FeatureCloud platform is data standardization. Data formatting and standards need to be discussed and determined in advance by the participants of the federated analysis. However, FeatureCloud provides the possibility to include federated data preprocessing applications in the workflow. While this does not remove the need for external communication of data standards, such as included features and naming conventions, it makes it straightforward to guarantee the same format and preprocessing for the used data before the actual model training process. Possible applications include imputation, normalization, train or test splitting, and CV [48,49].

Conclusions

In conclusion, we developed an open-source federated survival SVM that performs time-to-event analysis on geographically distributed data sets without sharing sensitive raw data. It is freely available in the FeatureCloud App Store. The trained models are almost identical compared with centrally trained survival SVMs. This extends the palette of existing federated time-to-event analysis approaches by another algorithm that can be applied to various problems.

Acknowledgments

This project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement 826078. This publication reflects only the author's view and the European Commission is not responsible for any use that may be made of the information it contains (JB). This work was developed as part of the FeMAI project and is funded by the German Federal Ministry of Education and Research (BMBF) under grant 01IS21079 (NP) and by the Agence Nationale de la Recherche (ANR) under grant ANR-21-FAI1-0010. MB and MA were also supported by the grant ANR-11-DPBS-0001. JB was partially funded by his VILLUM Young Investigator Grant (13154). PG has received funds from the Instituto de Salud Carlos III through the Plan Estatal de Investigación Científica y Técnica y de Innovación, project references PI 16/00043 and PI 20/00579. These grants were cofunded by the European Regional Development Fund (FEDER) and also funded in part by an EU Horizon 20/20 Programme (H2020-SC1-2016-RTD), LIVERHOPE (731875). JKP is funded by the Bavarian State Ministry of Education and the Arts in the framework of the Bavarian Research Institute for Digital Transformation (bidt, grant LipiTUM)

Data Availability

The data sets generated and analyzed during this study are available in the GitHub repository [50]. The code for the implementation of the federated survival SVM is available in the GitHub repository [51]. The microbiome data set is not publicly available due to privacy regulations but is available from the corresponding author on reasonable request.

Conflicts of Interest

CS has recieved speakig fees from Abbive and Grifols. PG has received research funding from Gilead & Grifols. PG has consulted or attended advisory boards for Gilead, RallyBio, SeaBeLife, Merck, Sharp and Dohme (MSD), Ocelot Bio, Behring, Roche Diagnostics International and Boehringer Ingelheim, and received speaking fees from Pfizer.

Multimedia Appendix 1

State workflow of the survival support vector machine (SVM) FeatureCloud app and difference between coefficients.
[DOCX File , 244 KB-Multimedia Appendix 1]

Multimedia Appendix 2

C-indices of central, federated, and federated + secure aggregation analyses.
[XLSX File (Microsoft Excel File), 32 KB-Multimedia Appendix 2]

Multimedia Appendix 3

Coefficients of the trained survival support vector machines (SVMs).
[XLSX File (Microsoft Excel File), 243 KB-Multimedia Appendix 3]

Multimedia Appendix 4

Correlation of Shapley additive explanations (SHAP) values between central, federated, and federated + secure aggregation model.
[XLSX File (Microsoft Excel File), 10 KB-Multimedia Appendix 4]

Multimedia Appendix 5

Shapley additive explanations (SHAP) beeswarm plots for the different models.
[ZIP File (Zip Archive), 25020 KB-Multimedia Appendix 5]

Multimedia Appendix 6

Runtime of the federated survival support vector machine (SVM) training with 1, 3, and 5 clients.
[XLSX File (Microsoft Excel File), 11 KB-Multimedia Appendix 6]

Multimedia Appendix 7

Runtime of the federated survival support vector machine (SVM) with 1, 3, and 5 clients of a large sample size synthetic data set.
[XLSX File (Microsoft Excel File), 10 KB-Multimedia Appendix 7]

References

1. Adibuzzaman M, DeLaurentis P, Hill J, Benneyworth BD. Big data in healthcare—the promises, challenges and opportunities from a research perspective: a case study with a model database. AMIA Annu Symp Proc. 2017;2017:384-392. [Medline: 29854102]
2. Vlahou A, Hallinan D, Apweiler R, Argiles A, Beige J, Benigni A, et al. Data sharing under the general data protection regulation: time to harmonize law and research ethics? Hypertension. 2021;77(4):1029-1035. [FREE Full text] [doi: 10.1161/HYPERTENSIONAHA.120.16340] [Medline: 33583200]
3. Greenhouse JB, Stangl D, Bromberg J. An introduction to survival analysis: statistical methods for analysis of clinical trial data. J Consult Clin Psychol. 1989;57(4):536-544. [doi: 10.1037//0022-006x.57.4.536] [Medline: 2768615]
4. Prinja S, Gupta N, Verma R. Censoring in clinical trials: review of survival analysis techniques. Indian J Community Med. 2010;35(2):217-221. [FREE Full text] [doi: 10.4103/0970-0218.66859] [Medline: 20922095]
5. Díaz JSP, García ÁL. Comparison of machine learning models applied on anonymized data with different techniques. IEEE; 2023. Presented at: 2023 IEEE International Conference on Cyber Security and Resilience (CSR); 31 July 2023 - 02 August 2023;618-623; Venice, Italy. URL: https://ieeexplore.ieee.org/document/10224917 [doi: 10.1109/csr57506.2023.10224917]
6. Antman E. Data sharing in research: benefits and risks for clinicians. BMJ. 2014;348:g237. [doi: 10.1136/bmj.g237] [Medline: 24458978]
7. Kairouz P, McMahan HB, Avent B, Bellet A, Bennis M, Nitin BA, et al. Advances and open problems in federated learning. In: Foundations and Trends® in Machine Learning. Boston, Massachusetts. Now Foundations and Trends; 2021;1-210.

8. Bonawitz K, Kairouz P, McMahan B, Ramage D. Federated learning and privacy: building privacy-preserving systems for machine learning and data science on decentralized data. Queueing. 2021;19(5):87-114. [FREE Full text] [doi: 10.1145/3494834.3500240]
9. McMahan B, Ramage D. Federated learning: collaborative machine learning without centralized training data. Google Research. 2017. URL: https://blog.research.google/2017/04/federated-learning-collaborative.html [accessed 2024-02-13]
10. McMahan B, Moore E, Ramage D, Hampson S, Arcas BAY. Communication-efficient learning of deep networks from decentralized data. PMLR. 2017;54:1273-1282. Singh A, Zhu J, editors.
11. Torkzadehmahani R, Nasirigerdeh R, Blumenthal DB, Kacprowski T, List M, Matschinske J, et al. Privacy-preserving artificial intelligence techniques in biomedicine. Methods Inf Med. 2022;61(S 01):e12-e27. [FREE Full text] [doi: 10.1055/s-0041-1740630] [Medline: 35062032]
12. Brauneck A, Schmalhorst L, Majdabadi MMK, Bakhtiari M, Völker U, Saak CC, et al. Federated machine learning in data-protection-compliant research. Nat Mach Intell. 2023;5(1):2-4. Springer Science and Business Media LLC. [doi: 10.1038/s42256-022-00601-5]
13. Yang A, Ma Z, Zhang C, Han Y, Hu Z, Zhang W, et al. Review on application progress of federated learning model and security hazard protection. Digit Commun Netw. 2023;9(1):146-158. [FREE Full text] [doi: 10.1016/j.dcan.2022.11.006]
14. Asad M, Moustafa A, Ito T. Federated learning versus classical machine learning: a convergence comparison. ArXiv. Preprint posted online on 22 Jul 2021. [FREE Full text] [doi: 10.22541/au.162074596.66890690/v1]
15. Sheller MJ, Edwards B, Reina GA, Martin J, Pati S, Kotrotsou A, et al. Federated learning in medicine: facilitating multi-institutional collaborations without sharing patient data. Sci Rep. 2020;10(1):12598. [FREE Full text] [doi: 10.1038/s41598-020-69250-1] [Medline: 32724046]
16. Zolotareva O, Nasirigerdeh R, Matschinske J, Torkzadehmahani R, Bakhtiari M, Frisch T, et al. Flimma: a federated and privacy-aware tool for differential gene expression analysis. Genome Biol. 2021;22(1):338. [FREE Full text] [doi: 10.1186/s13059-021-02553-2] [Medline: 34906207]
17. Nasirigerdeh R, Torkzadehmahani R, Matschinske J, Frisch T, List M, Späth J, et al. sPLINK: a hybrid federated tool as a robust alternative to meta-analysis in genome-wide association studies. Genome Biol. 2022;23(1):32. [FREE Full text] [doi: 10.1186/s13059-021-02562-1] [Medline: 35073941]
18. Lu CL, Wang S, Ji Z, Wu Y, Xiong L, Jiang X, et al. WebDISCO: a web service for distributed cox model learning without patient-level data sharing. J Am Med Inform Assoc. 2015;22(6):1212-1219. [FREE Full text] [doi: 10.1093/jamia/ocv083] [Medline: 26159465]
19. Andreux M, Manoel A, Menuet R, Saillard C, Simpson C. Federated survival analysis with discrete-time cox models. ArXiv. Preprint posted online on 16 Jun 2020. [FREE Full text]
20. Froelicher D, Troncoso-Pastoriza JR, Raisaro JL, Cuendet MA, Sousa JS, Cho H, et al. Truly privacy-preserving federated analytics for precision medicine with multiparty homomorphic encryption. Nat Commun. 2021;12(1):5910. [FREE Full text] [doi: 10.1038/s41467-021-25972-y] [Medline: 34635645]
21. Späth J, Matschinske J, Kamanu FK, Murphy SA, Zolotareva O, Bakhtiari M, et al. Privacy-aware multi-institutional time-to-event studies. PLOS Digit Health. 2022;1(9):e0000101. [FREE Full text] [doi: 10.1371/journal.pdig.0000101] [Medline: 36812603]
22. Pölsterl S. scikit-survival: a library for time-to-event analysis built on top of scikit-learn. J Mach Learn Res. 2020;21(1):8747-8752. [FREE Full text]
23. Matschinske J, Späth J, Bakhtiari M, Probul N, Majdabadi MMK, Nasirigerdeh R, et al. The FeatureCloud platform for federated learning in biomedicine: unified approach. J Med Internet Res. 2023;25:e42621. [FREE Full text] [doi: 10.2196/42621] [Medline: 37436815]
24. Pölsterl S, Navab N, Katouzian A. Fast training of support vector machines for survival analysis. In: Machine Learning and Knowledge Discovery in Databases. Cham, Switzerland. Springer International Publishing; 2015;243-259.
25. FeatureCloud AI Developer API (1.1.0). FeatureCloud. URL: https://featurecloud.ai/assets/api/redoc-static.html [accessed 2024-01-13]
26. Cramer R, Damgard IB, Nielsen JB. Secure Multiparty Computation and Secret Sharing. Cambridge, England. Cambridge University Press; 2015.
27. Desmedt C, Piette F, Loi S, Wang Y, Lallemand F, Haibe-Kains B, et al. Strong time dependence of the 76-gene prognostic signature for node-negative breast cancer patients in the TRANSBIG multicenter independent validation series. Clin Cancer Res. 2007;13(11):3207-3214. [FREE Full text] [doi: 10.1158/1078-0432.CCR-06-2765] [Medline: 17545524]
28. Schumacher M, Bastert G, Bojar H, Hübner K, Olschewski M, Sauerbrei W, et al. Randomized 2 x 2 trial evaluating hormonal treatment and the duration of chemotherapy in node-positive breast cancer patients. German Breast Cancer Study Group. J Clin Oncol. 1994;12(10):2086-2093. [doi: 10.1200/JCO.1994.12.10.2086] [Medline: 7931478]
29. Hosmer DW, Lemeshow S, May S. Applied Survival Analysis: Regression Modeling of Time to Event Data: Second Edition. New York, NY. John Wiley and Sons Inc; 2008.
30. Solé C, Guilly S, Da Silva K, Llopis M, Le-Chatelier E, Huelin P, et al. Alterations in gut microbiome in cirrhosis as assessed by quantitative metagenomics: relationship with acute-on-chronic liver failure and prognosis. Gastroenterology. 2021;160(1):206.e13-218.e13. [FREE Full text] [doi: 10.1053/j.gastro.2020.08.054] [Medline: 32941879]

31. Wen C, Zheng Z, Shao T, Liu L, Xie Z, Le Chatelier E, et al. Quantitative metagenomics reveals unique gut microbiome biomarkers in ankylosing spondylitis. Genome Biol. 2017;18(1):142. [FREE Full text] [doi: 10.1186/s13059-017-1271-6] [Medline: 28750650]
32. Oñate FP, Le Chatelier E, Almeida M, Cervino ACL, Gauthier F, Magoulès F, et al. MSPminer: abundance-based reconstitution of microbial pan-genomes from shotgun metagenomic data. Bioinformatics. 2019;35(9):1544-1552. [FREE Full text] [doi: 10.1093/bioinformatics/bty830] [Medline: 30252023]
33. Harrell FE, Califf RM, Pryor DB, Lee KL, Rosati RA. Evaluating the yield of medical tests. JAMA. 1982;247(18):2543-2546. [Medline: 7069920]
34. Dwork C. Differential privacy. In: Automata, Languages and Programming. Berlin, Heidelberg. Springer; 2006;1-12.
35. Yeom S, Giacomelli I, Fredrikson M, Jha S. Privacy risk in machine learning: analyzing the connection to overfitting. IEEE; 2018. Presented at: 2018 IEEE 31st Computer Security Foundations Symposium (CSF); July 09-12, 2018;268-282; Oxford, UK. URL: https://ieeexplore.ieee.org/abstract/document/8429311/ [doi: 10.1109/csf.2018.00027]
36. Hsu J, Gaboardi M, Haeberlen A, Khanna S, Narayan A, Pierce BC, et al. Differential privacy: an economic method for choosing epsilon. IEEE; 2014. Presented at: 2014 IEEE 27th Computer Security Foundations Symposium; July 19-22, 2014;398-410; Vienna, Austria. URL: https://ieeexplore.ieee.org/abstract/document/6957125/ [doi: 10.1109/csf.2014.35]
37. Lundberg SM, Lee SI. A unified approach to interpreting model predictions. USA. Curran Associates Inc; 2017. Presented at: Proceedings of the 31st International Conference on Neural Information Processing Systems Red Hook; 2017;4768-4777; NY, USA. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf
38. Smith A, Lambert PC, Rutherford MJ. Generating high-fidelity synthetic time-to-event datasets to improve data transparency and accessibility. BMC Med Res Methodol. 2022;22(1):176. [FREE Full text] [doi: 10.1186/s12874-022-01654-1] [Medline: 35739465]
39. Merkel D. Docker: Lightweight Linux Containers for Consistent Development and Deployment. Linux J Houston, TX. Belltown Media; 2014. URL: https://www.seltzer.com/margo/teaching/CS508.19/papers/merkel14.pdf [accessed 2024-03-06]
40. Kaplan EL, Meier P. Nonparametric estimation from incomplete observations. J Am Stat Assoc. 1958;53(282):457-481. [doi: 10.1080/01621459.1958.10501452]
41. Aalen O. Nonparametric inference for a family of counting processes. Ann Statist. 1978;6(4):701-726. [doi: 10.1214/aos/1176344247]
42. Cox D. Regression models and life-tables. J R Stat Soc. 1972;34(2):187-202. [FREE Full text] [doi: 10.1007/978-1-4612-4380-9_37]
43. Ishwaran H, Kogalur UB, Blackstone EH, Lauer MS. Random survival forests. Ann Appl Stat. 2008;2(3):841-860. [doi: 10.1214/08-aoas169]
44. Hauschild AC, Lemanczyk M, Matschinske J, Frisch T, Zolotareva O, Holzinger A, et al. Federated random forests can improve local performance of predictive models for various healthcare applications. Bioinformatics. 2022;38(8):2278-2286. [FREE Full text] [doi: 10.1093/bioinformatics/btac065] [Medline: 35139148]
45. Shamir A. How to share a secret. Commun ACM. 1979;22(11):612-613. [FREE Full text] [doi: 10.1145/359168.359176]
46. Kairouz P, Brendan MH, Avent B, Bellet A, Bennis M, Bhagoji AN, et al. Advances and open problems in federated learning. ArXiv. Preprint posted online on 9 Mar 2021. [FREE Full text] [doi: 10.1561/9781680837896]
47. Liu P, Xu X, Wang W. Threats, attacks and defenses to federated learning: issues, taxonomy and perspectives. Cybersecurity. 2022;5(1):1-19. [FREE Full text] [doi: 10.1186/s42400-021-00105-6]
48. Normalization app. FeatureCloud. 2022. URL: https://featurecloud.ai/app/normalization [accessed 2024-01-13]
49. Cross validation app. FeatureCloud. 2022. URL: https://featurecloud.ai/app/cross-validation [accessed 2024-01-13]
50. Späth J. julianspaeth / federated-survival-svm. GitHub. URL: https://github.com/julianspaeth/federated-survival-svm [accessed 2024-03-21]
51. Späth J. FeatureCloud / fc-survival-svm. GitHub. URL: https://github.com/FeatureCloud/fc-survival-svm [accessed 2024-03-21]

Abbreviations

BRCA: breast cancer data set
CV: cross-validation
DP: differential privacy
FL: federated learning
GBSG2: German Breast Cancer Study Group 2 data set
GDPR: General Data Protection Regulation
HE: homomorphic encryption
ML: machine learning
MSP: Metagenomic Species Pangenome
PET: privacy-enhancing technique

RSA: Rivest–Shamir–Adleman
SHAP: Shapley additive explanations
SMPC: secure multiparty computation
SVM: support vector machine
WHAS500: Worcester Heart Attack Study data set

Edited by K El Emam, B Malin; submitted 30.03.23; peer-reviewed by N Mungoli, S Nagavally, R Gorantla, D Gopukumar, X Jiang, Y Huang; comments to author 02.07.23; revised version received 06.08.23; accepted 10.02.24; published 29.03.24

Please cite as:
Späth J, Sewald Z, Probul N, Berland M, Almeida M, Pons N, Le Chatelier E, Ginès P, Solé C, Juanola A, Pauling J, Baumbach J Privacy-Preserving Federated Survival Support Vector Machines for Cross-Institutional Time-To-Event Analysis: Algorithm Development and Validation
JMIR AI 2024;3:e47652
URL: https://ai.jmir.org/2024/1/e47652
doi: 10.2196/47652
PMID:

B Conference Record

Presenter

1. **International Society for Computational Biology (ISMB)**, **Tutorial VT3: Federated Learning in Biomedicine**
   July 10-14, 2022. Madison, USA & Virtual.

Participant

1. **The Federated & Decentralized Machine Learning Conference**, June 16-17, 2021. Virtual.
2. **OpenMined Privacy Conference**, September 26-27, 2020. Virtual.
3. **The Federated and Distributed Machine Learning Conference**, June 18-19, 2020. Virtual.

C Scholarship Record

DAAD IFI — Internationale Forschungsaufenthalte für Informatikerinnen & Informatiker (2022)

From January 2022 to July 2022, I received the **DAAD scholarship IFI — international research exchange for informaticians** for my research exchange to the **Brigham and Women's Hospital/Harvard Medical School** in the laboratory of Joseph Loscalzo, M.D., Ph.D., at the Department of Medicine.

Eidesstattliche Versicherung

Hiermit erkläre ich an Eides statt, dass ich die vorliegende Dissertationsschrift selbst verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Ort, Datum Unterschrift