MOX-Report No. 75/2023

Scaling survival analysis in healthcare with federated survival forests: A comparative study on heart failure and breast cancer genomics

Archetti, A.; Ieva, F.; Matteucci, M.

MOX, Dipartimento di Matematica
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

mox-dmat@polimi.it https://mox.polimi.it

Scaling Survival Analysis in Healthcare with Federated Survival Forests: A Comparative Study on Heart Failure and Breast Cancer Genomics

Alberto Archetti[a,*], Francesca Ieva[b,c], Matteo Matteucci[a]

[a] *Department of Electronics, Information and Bioengineering (DEIB), Politecnico di Milano Via Ponzio 34, 20133, Milan, Italy*
[b] *Department of Mathematics, Politecnico di Milano Via Bonardi 9, 20133, Milan, Italy*
[c] *Health Data Science Center, Human Technopole Viale Rita Levi-Montalcini 1, 20157, Milan, Italy*

Abstract

Survival analysis is a fundamental tool in medicine, modeling the time until an event of interest occurs in a population. However, in real-world applications, survival data are often incomplete, censored, distributed, and confidential, especially in healthcare settings where privacy is critical. The scarcity of data can severely limit the scalability of survival models to distributed applications that rely on large data pools. Federated learning is a promising technique that enables machine learning models to be trained on multiple datasets without compromising user privacy, making it particularly well-suited for addressing the challenges of survival data and large-scale survival applications. Despite significant developments in federated learning for classification and regression, many directions remain unexplored in the context of survival analysis. In this work, we propose an extension of the Federated Survival Forest algorithm, called FedSurF++. This federated ensemble method constructs random survival forests in heterogeneous federations. Specifically, we investigate several new tree sampling methods from client forests and compare the results with state-of-the-art survival models based on neural networks. The key advantage of FedSurF++ is its ability to achieve comparable performance to existing methods while requiring only a single communication round to complete. The extensive empirical investigation results in a significant improvement from the algorithmic and privacy preservation perspectives, making the original FedSurF algorithm more efficient, robust, and private. We also present results on two real-world datasets – a heart failure dataset from the Lombardy HFData project and Fed-TCGA-BRCA from the Falmby suite – demonstrating the success of FedSurF++ in real-world healthcare studies. Our results underscore the potential of FedSurF++ to improve the scalability and effectiveness of survival analysis in distributed settings while preserving user privacy.

Keywords: survival analysis, federated learning, random survival forest, heart failure, breast cancer

1. Introduction

Survival analysis, or time-to-event analysis, is a branch of statistical machine learning that models the time until an event occurs in a population [1]. It is an essential tool for clinical trials, used to compare the survival rates of different treatments or groups of patients and to study the factors that influence disease onset or progression [2]. The goal of a survival model is to construct a survival function for a given subject in the population. The survival function

$$S(t) = P(T > t) \tag{1}$$

represents the probability that the subject will not experience, or *survive*, a given event by time t. Survival models use data to estimate the survival function, however, most healthcare applications involve data that are distributed across multiple devices, scarce, and confidential [3, 4]. Additionally, some data may have incomplete information about the subjects' survival time, a phenomenon called *censoring*. For example, when studying the survival rates of patients with a certain disease, some patients may drop out of the trial or be alive at the end of the trial, making their true survival time unknown. Censoring is a common challenge in survival analysis because it can bias results and reduce the statistical power of the analysis. Increasing the number of data samples for training could help, but this is often not feasible due to difficulties in data collection and confidentiality constraints.

To overcome these limitations, Federated Learning (FL) [5, 6] has emerged as a promising technique to improve the success of survival applications in large-scale real-world scenarios. FL allows multiple parties with private data sets to collaboratively train a machine learning model without sharing private data information. Private data remain on the storage device, ensuring confidentiality for agents in the federation. Federated models have better generalization performance than local models because they can leverage a large and representative data pool. FL has great potential in scenarios with small local privacy-protected datasets, such as clinics and hospitals, where each data sample is valuable and private.

Federated survival analysis aims to develop techniques for applying survival models in federated settings. Several survival studies have used federated learning to analyze clinical data from different domains, such as cancer genomics [3, 7, 8], stroke

[*] Corresponding author

detection [9], and COVID-19 survival [10]. These studies mostly use either non-parametric methods, such as Kaplan-Meier estimators [11], or semi-parametric Cox models [12, 3, 9, 13, 14, 15, 16, 17, 10, 18, 19]. However, the Cox model is based on the proportional hazard assumption, which may not be true in large federated datasets. In addition, Cox models have a linear relationship between covariates and survival ratios across subjects, which facilitates their interpretation but limits their modeling power. Some recent works have extended federated survival analysis to non-linear models based on neural networks [7, 20, 21]. This is an emerging line of research that enables survival analysis on large distributed datasets. However, most of the works use survival datasets only for benchmarking and do not address clinically relevant questions or conduct clinical trials [22].

This paper presents an extension of the Federated Survival Forest (FedSurF) algorithm [23], called FedSurF++, that applies Random Survival Forests (RSFs) [24] in a federated setting. RSFs are tree-based models that can handle censored data, missing values, and categorical variables. They also have lower computational complexity and higher interpretability than neural networks due to their tree-based nature. FedSurF++ exploits the advantages of RSFs and adapts them to the federated environment, where data are distributed across multiple clients and cannot be shared. The key idea of FedSurF++ is to train a local random survival forest on each client's private data and then build a federated ensemble of trees on the central server. The server selects the local client trees with a sampling method proportional to their performance metrics computed on a local validation split. This way, FedSurF++ can train a global RSF model from local RSFs with only one round of communication, reducing communication overhead and latency compared to iterative federated learning algorithms. As shown in [23], the final global model consists of the best-performing trees, increasing the model's expressiveness in heterogeneous scenarios.

With this work, our contribution is twofold. First, we extend and investigate new sampling techniques for tree selection based on standard survival metrics. The results show how including a metric evaluation step to select the best trees is generally more powerful than sampling trees at random. Also, while metric evaluation increases model performance in heterogeneous settings, the specific tree-sampling metric does not affect in a statistically significant way the effectiveness of the final model. Therefore, the simplest evaluation metric, such as concordance, is sufficient to obtain the best final model. Second, we apply FedSurF++ to two real-world cases. The first is an administrative study concerning hospitalizations of patients experiencing heart failure [25]. The dataset comes from the Lombardy HF-Data research project and is composed of 895 samples with 32 covariates split across 23 medical institutes. The second comes from the dataset suite called (Flamby) [8] and collects 38 finary features for each of the 1088 patients suffering from breast invasive carcinoma. Results on both datasets show how FedSurF++ remains competitive even in real-world scenarios from a survival modelization standpoint while requiring a single communication exchange between server and clients to terminate.

The rest of the paper is organized as follows. Section 2 provides in-depth background on survival analysis and federated learning. Section 3 reviews the current literature on federated survival analysis, highlighting the applied techniques and healthcare applications. Section 4 describes the FedSurF++ algorithm. Section 5 analyzes the empirical results obtained, both on simulated federations and on real-world datasets. Finally, Section 6 summarizes the work.

2. Background

In this section, we review the basics of survival analysis and federated learning. First, we define the problem of survival analysis and how it relates to statistical modeling and machine learning (Section 2.1). We then categorize and explain the state-of-the-art survival models based on neural networks (Section 2.2) and the most common survival evaluation metrics (Section 2.3). Finally, we introduce federated learning and the techniques for dealing with data heterogeneity in distributed federations (Section 2.4).

2.1. Survival Analysis

Survival analysis [1, 2] is a branch of statistical machine learning concerned with the analysis of time-to-event data, where the event of interest may be death, disease onset, hardware failure, or any other event. The goal of survival analysis is to model the relationship between survival time and predictors related to a particular subject, called features or covariates. The output of survival models is the probability of survival or the risk of experiencing the event over time. Specifically, the survival function

$$S(t) = P(T > t) \tag{2}$$

is the probability that an individual will not experience the event, i.e. *survive*, beyond time t. It is a non-increasing function, ranging from 1 at $t = 0$ to 0 for $t \to \infty$. The hazard function

$$h(t) = \lim_{\delta t \to 0} \frac{P(t \leq T < t + \delta t | T \geq t)}{\delta t} \tag{3}$$

is the instantaneous failure rate at time t, given that the individual has survived up to t. It is a non-negative function that can take any value greater than 0. The survival function and the hazard function are related as

$$S(t) = \exp(-H(t)) \tag{4}$$

where $H(t) = \int_0^t h(\tau)\, d\tau$ is the cumulative hazard function. Each of these functions can be estimated using statistical and machine learning techniques starting from a survival dataset. A survival dataset is a set of triplets

$$D = \{(\mathbf{x}_i, \delta_i, t_i)\}_{i=1}^N \tag{5}$$

such that

- $\mathbf{x}_i \in \mathbb{R}^d$ is a d-dimensional real-valued feature vector.
- $\delta_i \in \{0, 1\}$ is an event indicator set to 1 if the i-th subject experienced the event and set to 0 if the sample is censored instead.
- $t_i > 0$ is the event time if $\delta_i = 1$ or the censoring time if $\delta_i = 0$.

2.2. Survival Models

There are different types of survival models, depending on the assumptions made about the form of the survival or hazard function. Non-parametric models make no assumptions about the shape of the survival or hazard function and rely on empirical data aggregations without considering feature vectors. These models are easy to calculate and provide unbiased estimates of the survival function. These models are most useful for data exploration and visualization purposes. Examples of non-parametric models include Kaplan-Meier (KM) [26] and Nelson-Aalen [27, 28]. Specifically, the KM estimator calculates the cumulative probability of survival based on data by successively multiplying the probabilities of survival at each unique event time. In particular, for each unique event time t_j in the set $T_D = \{t_j : (\mathbf{x}_i, \delta_i, t_j) \in D\}$, KM counts the number of observed events d_j and the number of samples r_j that are still at risk. Then, the KM estimator calculates the survival function $\hat{S}(t)$ by cumulatively multiplying these survival probabilities for all time points preceding t, as

$$\hat{S}(t) = \prod_{j:t_j<t} \left(1 - \frac{d_j}{r_j}\right). \tag{6}$$

This way, the KM estimator encapsulates the intuition that the probability of surviving up to a given time is the cumulative product of the probabilities of surviving each preceding moment.

Semi-parametric models decompose hazard functions into a common baseline hazard $h_0(t)$ and a subject-related risk function $\phi(t, \mathbf{x})$. The resulting hazard function is calculated as

$$h(t|\mathbf{x}) = h_0(t) \cdot \phi(t, \mathbf{x}). \tag{7}$$

One of the most widely used semi-parametric models is the Cox proportional hazard model [29]. In this model, the risk function for an individual with a feature vector $\mathbf{x}$ is given by

$$\phi(\mathbf{x}) = \exp\left(\sum_{j=0}^{d} \beta_j \cdot x_j\right) \tag{8}$$

where β is the vector of regression coefficients measuring the effect of each element x_j of $\mathbf{x}$ on the hazard function. Cox models are based on the proportional hazard assumption. This assumption states that the hazard ratio between different subjects is constant over time, i.e., the effect of $\mathbf{x}$ on the hazard does not depend on t. Cox models are trained using the partial log-likelihood loss function, given by

$$L(\beta) = \sum_{i=1}^{N} \delta_j \left[\beta^T \mathbf{x}_i - \log\left(\sum_{j \in R_i} \exp\left(\beta^T \mathbf{x}_j\right)\right)\right] \tag{9}$$

where R_i is the set of subjects at risk at time t_i. For this reason, this function is *non-separable*, i.e., it requires access to all available samples in the dataset to be evaluated. Cox models have several advantages. First, they are easy to interpret and require little computation due to their linear nature. These models can also incorporate feature vectors without explicitly modeling the baseline hazard. However, they rely on the assumption of proportional hazards, which may not hold for large datasets. The lack of an explicit survival function may also be a disadvantage in studies where risk ratios are not relevant.

Non-linear models use flexible functions, such as neural networks or splines, to capture complex and non-linear relationships between survival times and feature variables. These models have the highest modeling power because they can approximate any survival or hazard function. However, they require large amounts of data and computational resources for training and optimization. They also suffer from poor interpretability and a high risk of overfitting, especially when data pools are small.

Among the nonlinear models, DeepSurv [30] is an extension of the Cox proportional hazards model. DeepSurv replaces the linear risk function, common in traditional survival models, with a single-output neural network. This allows for a more flexible representation of complex interactions between covariates, handling nonlinear relationships in high-dimensional data that are challenging for traditional models. DeepSurv relies on the same partial log-likelihood loss function as the Cox model, but the risk scores are predicted by a deep neural network rather than a linear function.

DeepHit [31], on the other hand, is a neural-based architecture that focuses on the analysis of multiple concurrent events in survival analysis. It does this through time discretization, where each interval is modeled as a multi-class classification problem using sigmoid activations. Each class is tailored to identify the occurrence of one of the concurrent events. This enables DeepHit to capture complex patterns in time-to-event data, particularly when there are multiple concurrent event types to consider.

Neural Multi-Task Logistic Regression (N-MTLR) [32] is a non-linear extension of the Multi-Task Logistic Regression (MTLR) model [33]. The MTLR model is a discrete-time survival model that calculates the likelihood of an event happening in each time interval using a separate logistic regression model for each time interval. Similarly to DeepHit, the classification task is tailored to identify whether an event occurred in a specific time interval. N-MTLR extends MTLR by incorporating a nonlinear predictor for each time bin, based on neural networks, mapping input covariates to a single output. These binned outputs are then combined using a softmax function to obtain survival estimates for each time bin.

Nnet-Survival [34, 35], also known as Logistic Hazard, is a model that leverages the discrete formulation of survival problems to model discrete hazard functions. Again, the method involves breaking down the survival problem into a series of binary classification tasks, each representing the risk of event occurrence in a particular time interval. This allows for flexibility in modeling time-varying effects and interactions.

Piecewise-Constant Hazard (PC-Hazard) [35, 36], as the name suggests, estimates the hazard function as a piecewise constant. This model assumes that the hazard, or the risk of an event happening, remains constant within certain time intervals, but can change between intervals. This assumption simplifies the model to a series of regression problems, allowing it to take advantage of existing machine learning techniques. The neural component of the model maps covariates to a finite number of outputs, corresponding to the hazard in each interval. In this way, while the output of the neural network has a discrete size, the resulting survival function is still continuous and can be computed using Equation (4), resulting in a series of piecewise exponential functions.

Finally, Random Survival Forests (RSFs) [24] are a class of ensemble-based models that use survival trees to estimate the cumulative hazard function $H(t)$. They follow the same principle as random forests for classification and regression [37], where a large number of binary trees are grown using bootstrap samples of the data. The main difference lies in the node-splitting technique, which maximizes the hazard difference between the child nodes. The resulting hazard function is obtained by averaging the hazard functions of the terminal nodes across all trees, which are computed using the Nelson-Aalen estimators [27, 28] on the leaf samples.

2.3. Survival Metrics

The Concordance Index (C-Index), the Integrated Brier Score (IBS), and the Cumulative Area-Under-the-Curve (Cumulative AUC) are the most used metrics to evaluate survival models. The C-Index [38] is a measure of the agreement between the predicted and true survival outcomes for a pair of samples. The predicted outcome is the estimated survival probability or risk score for a given time point, and the true outcome is the actual survival time or event status (1 if the event has occurred and 0 otherwise). A pair of samples is comparable if at least one of them has experienced the event of interest. The C-Index is calculated as the ratio of concordant pairs to comparable pairs. A pair is concordant if the sample with the higher predicted outcome survives longer than the sample with the lower predicted outcome. The C-Index ranges from 0 to 1, where 0.5 is a random prediction and 1 is a perfect prediction. The C-index reflects the discriminative power of the model, i.e., its ability to rank samples according to their actual survival times.

The Brier Score [39] (BS) is a measure of the accuracy of the predicted survival probability for a sample at a given time. The Brier Score is calculated as the squared difference between the true survival status (1 if the event has occurred and 0 otherwise) and the predicted survival probability of the model at that time. The Brier Score ranges from 0 to 1, where 0 indicates a perfect prediction and 1 indicates a completely incorrect prediction. A random guessing model would have a BS of 0.25. Thus, the lower the BS, the better the model. The Brier score reflects also the calibration of the model, i.e., its ability to estimate the correct survival probabilities for each sample. The Integrated Brier Score (IBS) is a measure of the overall calibration of the model over time. The IBS is calculated as the average of the Brier scores over a series of time points. The IBS also ranges from 0 to 1, where 0 indicates a perfect prediction and 1 indicates a completely wrong prediction.

In survival analysis, the evaluation of the AUC for classification can be extended to time-varying outcomes [40]. In particular, the time-dependent AUC defines a time interval and compares the predicted survival probability at the beginning of the interval with the observed event status within the interval. Samples that are censored within or before the interval are considered negative cases. The Cumulative AUC is a summary measure that integrates the time-dependent AUC over time. The Cumulative AUC ranges from 0 to 1, with 1 indicating perfect prediction.

Survival metrics can account for the censoring distribution by applying the Inverse Probability of Censoring Weighting (IPCW) [38, 41]. The IPCW assigns a weight to each sample based on the inverse probability of being censored at a given time point. The weight reflects how representative the sample is of the underlying population at that time point. Samples with a higher probability of being censored are assigned higher weights, and vice versa. IPCW weighting can help to reduce the bias introduced by censored samples in the resulting metrics.

2.4. Federated Learning

Federated Learning (FL) [5, 6] is a distributed machine learning paradigm that allows multiple clients to collaboratively train a model without sharing their private data. In FL, data remain on the devices where they are generated, and only model updates are communicated to a central server that coordinates the learning process. This approach contrasts with traditional centralized machine learning techniques, where all local data are uploaded to a single server, as well as more classical decentralized approaches, which often assume that local data are identically distributed. Federated learning allows multiple actors to build a shared machine learning model without sharing data while addressing security and data access rights. In addition, a model trained on distributed heterogeneous data is representative of a large portion of the population. By processing data mostly at the edge, federated learning can reduce latency, power consumption, and communication costs compared to explicitly sharing data with a central server.

In a typical FL setting, there are K clients, each holding a local dataset D_k. The goal of a FL algorithm is to learn a set of model parameters w that minimize a global loss function L. This function is the weighted sum of local loss functions L_k computed by each client k on their own data D_k. Each contribution is weighted in proportion to the number of samples stored in each database D_k. Federated Averaging (FedAvg) [42] is the first algorithm proposed to optimize L. FedAvg works in rounds, where each round consists of three steps: a broadcast step, a training step, and an aggregation step. In the broadcast step, the server selects a subset of clients and sends them the current model parameters w. In the training step, each of these clients trains the model with parameters w on its local data for a small number of epochs, obtaining a new set of parameters w'_k. These parameters are then sent back to the server. Finally, in the aggregation step, the server updates the global parameters by taking a weighted average of the updates w'_k received from the clients and repeats the process until convergence.

While FedAvg achieves good performance in simulated settings, it faces several challenges when applied to real-world scenarios where heterogeneity is prevalent both in terms of computational resources and data distribution across clients [5]. For example, some clients may have slower computation time or connectivity, resulting in missed updates during federated training. For this purpose, several asynchronous frameworks have been developed that are reliable for stragglers [43]. In addition, data distributions among clients may not be independent and identically distributed (IID), leading to biased or inaccurate updates that can affect the global model quality. To address this issue, some works propose regularization techniques that reduce the discrepancy between local and global models [44, 45, 46, 47, 48].

As data heterogeneity is one of the key challenges in federated networks, test and simulation environments are essential for federated learning research, as they allow to evaluate and compare different algorithms under different realistic settings. Several benchmarking methods have been developed for this purpose. LEAF [49] provides a collection of heterogeneous datasets for standard machine learning tasks such as image classification and next character prediction. SGDE [50] generates synthetic datasets from privacy-preserving data generators that learn the characteristics of the client's data. Other works [51, 52] investigate data splitting techniques, based on the Dirichlet distribution, that adjust the degree of heterogeneity in federated classification datasets.

3. Related Work

Machine learning methods have significantly advanced healthcare applications, revolutionizing disease diagnosis, prognosis, and treatment [53, 54, 55, 56]. Nevertheless, the application of these methods often requires the use of extensive datasets, which necessitates a careful balance between data utilization and patient privacy preservation. To answer this issue, Federated Learning (FL) has emerged as a promising approach for large-scale healthcare applications [4]. Leveraging distributed data while maintaining data privacy, FL models have outperformed traditional statistical methods in predicting outcomes based on medical data [57, 58, 59].

In this context, federated survival analysis models the time to event of interest (such as death, disease, or failure) in a population, where data is distributed across multiple institutions. This field bridges privacy-preserving federated training with conventional survival analysis techniques. Federated survival analysis has proved to be effective, particularly in oncology, contributing to more robust and privacy-preserving predictive models [3, 11, 22, 15]. Beyond cancer research, other studies have also applied federated survival analysis to examine stroke events [9] and COVID-19 survival rates [10].

Much effort has been devoted to federated Cox models [12, 3, 9, 13, 14, 15, 16, 17, 10, 18, 19]. In fact, the Cox proportional hazard model is one of the most prominent models from classical survival analysis that is easy to interpret, fast to compute, and does not require explicit modeling of the baseline hazard function. However, as explained in Section 2, the partial log-likelihood is not separable. This is a serious problem in federations where data are confidential, as clients are unable to effectively compute hazard rates for the entire data. Many alternative formulations to the standard Cox model have been proposed. For example, the authors of [3] propose a discrete extension of the proportional Cox model to formulate survival analysis as a classification problem with a separable loss function. The method in [19] is also based on discretization but takes into account the effects of time-varying covariates. In [9], patient-level data from one client is combined with aggregated information from the other clients to construct a surrogate likelihood function that approximates the Cox partial likelihood function obtained using all available patient-level data. Cox models have been adapted for vertically partitioned data, where data samples from the same patients are stored in different institutions [14, 16, 18]. In particular, VERTICOX [14] is an algorithm based on the ADMM framework [60] that obtains global model parameters in a distributed manner by computing and exchanging intermediate statistics, achieving an accuracy similar to that of a centralized Cox model.

Federated implementations of classical survival models are not limited to Cox. In [11], the authors propose FAMHE, a federated system that allows privacy-preserving estimation of Kaplan-Meier models. Regarding nonlinear models, the authors of [20] propose a method to improve the performance of nonlinear federated survival models with differential privacy by adding a post-processing step that adjusts the magnitude of the average noisy parameter update and facilitates model convergence. In [7], weakly supervised attention modules are used to estimate discrete survival rates. FedPseudo [21] uses pseudo values as surrogate labels for federated deep learning models. Finally, FedSurF [23] leverages federated ensemble learning to construct random survival forests from distributed survival data.

Privacy is a crucial aspect of survival applications, as patient-level data are often sensitive and confidential. Several works have used differential privacy [61] to protect survival models against inference attacks [7, 20]. Alternatively, some works have relied on secure multi-party computation (SMPC), which allows computation on distributed data without revealing individuals' information. For example, SMPC has been applied to the Newton-Raphson algorithm to optimize the partial log-likelihood of distributed Cox models by computing intermediate statistics [16]. Another SMPC protocol, SecureFedYJ [62], allows the Yeo-Johnson transformation to be applied to vertically-partitioned data while preserving privacy. FAHME [11] uses multiparty homomorphic encryption to estimate distributed Kaplan-Meier models. Finally, in [18], a Cox model is designed for horizontal and vertical federated learning exploiting a privacy-preserving subspace projection technique that allows each local institution to obtain a secure approximation of the model parameters, survival curves, and statistics such as p-values.

Federated learning has been applied to genomic analysis for cancer survival and recurrence studies [3, 7, 8], using data from The Cancer Genome Atlas (TCGA) project. The TCGA Project is a large-scale database from the National Cancer Institute and the National Human Genome Research Institute that molecularly characterizes 33 types of cancer by collecting genomic, epigenomic, transcriptomic, and proteomic data that are publicly available to researchers. Among the works based on the TCGA project, FLamby (Federated Learning AMple Benchmark of Your cross-silo strategies) [8] is a collection of cross-silo federated learning datasets for healthcare applications. One of the datasets in the Flamby suite, Fed-TCGA-BRCA, consists of genomic and clinical data of breast cancer patients from 6 different hospitals. The dataset is naturally partitioned according to the geographic origin of the patients, with each patient assigned to the closest center. Finally, to evaluate the performance and compare the results of federated survival analysis methods, [63] provides two algorithms to split existing survival datasets into heterogeneous federations.

4. Method

In this section, we present the proposed extension of the Federated Survival Forest (FedSurF) algorithm [23], called FedSurF++. As the original algorithm, FedSurF++ relies on Random Survival Forests (RSF) [24] to build a tree-based ensemble model for survival analysis in a federated learning setting. Our approach builds upon prior works in federated ensemble learning [64, 65], where the central server merges base models from local ensembles on each client to create a global model. Specifically, the FedSurF++ algorithm constructs a RSF on the central server by aggregating the top-performing trees from local RSF models on each client, with an emphasis on the tree sampling strategy.

4.1. The FedSurF++ Algorithm

The FedSurF++ algorithm consists of three steps: *local training*, *tree assignment*, and *tree sampling*. The *local training* and *tree assignment* stages remain unchanged from the original FedSurF algorithm. In particular, during the *local training* step, each client k builds a local RSF M_k from the local data D_k. At this point, each local RSF model M_k is a set of T_k survival trees. These binary trees are built with a recursive node-splitting technique inspired by CART [66] that maximizes the survival difference between samples in child nodes. Tree leaves contain the Nelson-Aalen estimator [27] of the cumulative hazard resulting from their samples. Each client may tune the RSF hyperparameters to best fit their data distribution and hardware constraints, making local execution feasible and effective. For example, clients with high computational power can train forests with high cardinality, while clients with hardware limitations can lower the number of trees in their local models.

In the *tree assignment* stage, the server determines the number of trees each client is required to send on the server. To this end, the server iteratively increments a client counter $T'_k \leq T_k$ for a number of times equal to the number of desired trees T in the final ensemble M. Intuitively, the counter for client k cannot exceed the number of trees T_k in their local model M_k. At each iteration, a client counter T'_k is incremented with a probability proportional to $N_k = |D_k|$. This is to promote the selection of trees coming from clients that have larger datasets. This way, FedSurF++ promotes trees trained on larger data samples, which are likely to be more representative of the entire population. This procedure is inspired by the weighted updates of FedAvg [42] that assign a weight proportional to the local dataset cardinality when aggregating model parameters coming from different clients.

Finally, in the *tree sampling* stage, each client samples T'_k trees to be shared with the server. We introduce three new sampling strategies that are proportional to the Concordance Index (C-Index), the Concordance Index with IPCW weighting (C-Index-IPCW), and the Cumulative Area-Under-the-Curve (Cumulative AUC). This is an extension with respect to the original FedSurF, which is limited to sample trees according to the inverse of the Integrated Brier Score (IBS). Each of these metrics is discussed in Section 2.3. Given one of these metrics, clients evaluate each local tree, obtaining a set of estimations $\{\text{Metric}_j\}_{j=1}^{T_k}$. At this point, each client selects T'_k trees with a probability proportional to the chosen metric. In order to differentiate between sampling strategies, we adopt different names for our algorithm. Specifically, if clients choose to use a uniform sampling strategy, the method is referred to as FedSurF. For each of the metric-based sampling strategies, instead, we denote the method as FedSurF-Metric, where Metric represents the chosen performance measure. Section 5 collects experiments comparing the uniform sampling strategy (FedSurF) and the strategies proportional to the C-Index (FedSurF-C), the C-Index-IPCW (FedSurF-C-IPCW), the inverse IBS (FedSurF-IBS), and the Cumulative AUC (FedSurF-AUC).

While FedSurF++ is a relatively straightforward extension of the original FedSurF algorithm, it allows us to delve deeper into how tree-sampling methods affect the corresponding metric in the final model. Our experimental findings suggest that using the least expensive evaluation metric can still produce a high-performing model. Consequently, trees can be sampled using the C-Index without IPCW weighting, as in FedSurF-C, to achieve the optimal ensemble model. This has several implications from the algorithmic perspective. First, it allows each client to evaluate concordance locally, without relying on the aggregated statistics of other clients. This way, the number of messages to be shared in a federation is reduced. Second, this simpler metric protects privacy, as cumulated statistics are not shared with the server or any other client in the federation.

In summary, FedSurF++ extends FedSurF with a simple yet natural operation – allowing the selection of the tree-sampling method – that results in important implications from the efficiency, communication, and privacy perspectives. The pseudocode of FedSurF++ is presented in Algorithm 1.

4.2. Computational Complexity

Deriving an accurate estimate of the computational complexity related to federated algorithms is a complex task, as many factors that arise in real-world scenarios are difficult to integrate into the analysis. However, we can derive a rough estimate of the computational complexity of FedSurF++ to assess its strengths and limitations from a scalability perspective.

The training time complexity of RSFs primarily involves the number of trees in the forest T, the number of samples $N = |D|$, the number of features F, and the depth of the trees (which can be, at most, $\log(N)$ in a balanced tree scenario). RSFs use log-rank tests [67] to determine the best split at each node. Log-rank tests compare the survival distributions of two groups to determine if they are statistically different, which is an $O(N)$ operation. As this operation is performed at each node, it

Algorithm 1 FedSurF++ Algorithm

```
function FedSurF-Client($D_k$)
    ▹ Local training
    Tune parameters of local RSF $M_k$ using cross-validation.
    Train local RSF $M_k$ on $D_k$.
    Send the number of local trees $T_k$ to the central server.
    ▹ Tree sampling
    for $j = 1$ to $T_k$ do
        Compute Metric$_j$ for tree $j \in M_k$.
    end for
    Receive $T'_k$, the number of trees to send back to the server.
    Select $T'_k$ trees using probabilities proportional to Metric$_j$.
    Send selected trees to the server.
end function

function FedSurF-Server($T$)
    ▹ Tree assignment
    Receive $T_k$ from each client $k$.
    Compute $T'_k$ for each client $k$ according to $T_k$ and $T$.
    Send $T'_k$ to each client $k$
    ▹ Model construction
    Receive $T'_k$ trees from each client $k$.
    Construct the final model $M$ by aggregating $T$ trees.
    Return: Random survival forest $M$.
end function
```

multiplies by the number of candidate features for splitting, $\sqrt{F}$. If the tree is fully expanded, the number of internal nodes in a binary tree is, at most, $N - 1$. Therefore, the overall complexity related to node splitting is $O\left(T \cdot N^2 \cdot \sqrt{F}\right)$. At each leaf node, the Nelson-Aalen estimator accounts for a cost of $O(N)$. Since there could be at most N leaves in a fully expanded tree, the overall leaf evaluation complexity is $O\left(T \cdot N^2\right)$.

By combining the complexity of node splitting and leaf computations, the overall training time complexity for RSFs is $O\left(T \cdot N^2 \cdot \sqrt{F}\right)$. However, in practice, trees are not usually fully grown, as they are pruned or have a maximum depth, and samples are bootstrapped in each tree. The complexity of FedSurF++ is comparable to a single RSF execution, as it requires a single communication round, once the forests are trained in parallel on the clients. To complete the analysis, Section 5.1.4 collects empirical time executions for RSFs and neural-based models.

5. Experiments

This section collects the experiments based on which we compare the performance of FedSurF++ with other neural-based models from the state of the art in federated survival analysis. We present two sets of experiments. The former focuses on simulated federations, and the latter focuses on real-world federations. In particular, Section 5.1 reports the experiments on federations with splits simulated by the label-skewed splitting algorithm [63]. Instead, Section 5.2 collects the experiments on Lombardy Heart Failure [25] and Fed-TCGA-BRCA [8], which are based on real-world data splits.

Table 1: Survival datasets for simulated federation studies.

Dataset	Samples	Censored	Covariates
WHAS500 [68]	461	38%	16
GBSG2 [69]	686	44%	8
METABRIC [70, 30]	1904	58%	8
NWTCO [71]	4028	14%	8
FLCHAIN [72]	7874	28%	10

5.1. Experiments on Simulated Federations

This section covers the experiments related to simulated uniform and heterogeneous data splits of existing survival datasets.

5.1.1. Datasets

The following survival datasets are commonly used to evaluate non-federated survival methods. From these, we conduct experiments on simulated federations. Table 1 summarizes the statistics of these datasets.

- The Worcester Heart Attack Study (WHAS500) dataset [68] contains data on 461 patients who experienced acute myocardial infarction. The data were collected during the first hospitalization and included 16 covariates. The outcome of interest is survival time after the event.
- The German Breast Cancer Study Group (GBSG2) dataset [69] examines the effects of hormone treatment for breast cancer in 686 women. The outcome of interest is time to cancer recurrence. The dataset includes 8 covariates, such as age, menopausal status, tumor grade and size, and hormone levels.
- The Molecular Taxonomy of Breast Cancer International Consortium (METABRIC) dataset [70, 30] is a Canada-UK project that provides survival data for 1904 patients with breast cancer. The dataset comprises clinical attributes, gene expression profiles, copy number variations, and single nucleotide polymorphisms as covariates.
- The National Wilm's Tumor Study (NWTCO) dataset [71] consists of 4028 observations on 8 covariates for patients with Wilm's tumor, a rare type of kidney cancer that primarily affects children. The covariates include histology status, disease stage, and other factors. The outcome of interest is time to relapse.
- The FLCHAIN survival dataset [73] contains subjects from a study concerning mortality rates of serum-free light chain (FLC). The original data come from the residents of Olmsted County, Minnesota, with more than 50 years. The dataset has 7874 samples and 10 features.

5.1.2. Simulated Federations

The datasets from Section 5.1.1 are used to simulate federated datasets by assigning each sample to a particular client in the federation. First, each dataset is split into a training set and a test set by randomly selecting 70% of the total samples for training and the remaining 30% for testing.

Then, each sample in the training set is assigned to one of the clients in the federation. We assume that there are 5, 10, or 20 cooperating institutions in each simulated experiment, i.e., $K = 5, 10$, or 20 depending on the considered setting. Sample assignment is performed by the label-skewed splitting algorithm [63]. This algorithm is based on the Dirichlet distribution as in [51, 52]. The aim is to create unbalanced distributions of times and events across clients. In fact, it is important to test federated algorithms in heterogeneous contexts, as non-identical data distributions can affect the performance and convergence speed of federated algorithms. The label-skewed splitting algorithm has a hyperparameter α that controls the degree of heterogeneity among clients. Smaller values of α result in more heterogeneous distributions. We set $\alpha \to \infty$ to simulate a federation with uniformly split data and $\alpha = 5$ to simulate a federation with heterogeneous data distribution. The label distributions for each client are shown in Figure 1. We plot the Kaplan-Meier estimator for each client in the federation. By inspecting the estimators, federations with $\alpha \to \infty$ exhibit similar label distributions across clients, while for $\alpha = 5$ distributions differ.

In addition, 30% of the local samples on each client are reserved for validation and hyperparameter tuning. During our simulations, clients are assumed to be always available and communication packets are never lost.

5.1.3. Baseline Models

FedSurF++ is compared with the six state-of-the-art survival models described in Section 2.2. The first is the Cox proportional hazards model (CoxPH) [29], which uses the Nelson-Aalen estimator [27] to estimate the baseline hazard. A nonlinear extension of the Cox model, DeepSurv [30], is also included. The other models are discretized survival models based on neural networks: DeepHit [31], Neural Multi-Task Logistic Regression (N-MTLR) [32], and Nnet-Survival [34]. Finally, we consider Piecewise-Constant Hazard (PC-Hazard) [35], which is a non-proportional hazard neural-based model that provides a continuous estimation of the survival function.

The neural network architectures of DeepSurv, DeepHit, N-MTLR, Nnet-Survival, and PC-Hazard consist of two fully connected layers of 32 neurons each with ReLU activation functions. We also add a dropout layer with a probability of 10% to prevent overfitting. The output of DeepSurv is a scalar obtained by a linear transformation while the other models have 10 outputs corresponding to different discretization instants. Specifically, DeepHit, N-MTLR, and Nnet-Survival produce 10 survival probabilities, while PC-Hazard produces 10 discrete hazard values that are converted to survival probabilities using Equation 4.

5.1.4. Training

Each model is evaluated in three settings: *Global*, *Local*, and *Federated*. In the *Global* setting, it is assumed that the entire survival dataset is centralized in a single node, and a single model is trained. This setting does not require federated learning and serves as an empirical upper bound to assess the performance loss due to data distribution.

The *Local* setting involves clients training their models only on their local data without participating in the federation. The average performance of the local models is reported as an empirical lower bound, which is targeted for improvement using federated learning. It is expected that joining a federated learning algorithm would benefit the clients in terms of model performance.

In the *Federated* setting, multiple clients collaborate in a federated learning procedure. To achieve the most effective baseline model training, we performed a comparative analysis between the widely used Federated Averaging algorithm (FedAvg) [42], and FedProx [46], an alternative algorithm aimed at enhancing generalization in heterogeneous federations. Our study employed the five datasets listed in Table 1, examining three distinct client configurations ($K = 5, 10, 20$). We ran both FedAvg and FedProx training on each of the six baseline models (CoxPH, DeepSurv, DeepHit, N-MTLR, Nnet-Survival, and PC-Hazard), culminating in a total of 90 direct FedAvg versus FedProx comparisons based on the dataset, model, and client number. Each of these 90 pairings was repeated five times, followed by a t-test analysis. Notably, only a fraction (5.6%) displayed a statistically significant performance variation in concordance index between FedAvg and FedProx. Consequently, we chose to utilize the standard FedAvg algorithm for training each neural model in the subsequent experiments. Furthermore, we assumed that each proportional hazard model, i.e., Cox and DeepSurv, has access to a global Kaplan-Meier estimate of the survival data.

Federated averaging is implemented using the Flower library [74] for Python and run for 150 rounds, allowing each client to execute 2 local epochs for each round. The best model parameters are selected based on the highest concordance index on the validation set of each client. RSFs are implemented with scikit-survival [40], and neural-based models are implemented with PyCox [75]. The Adam optimizer with a learning rate of 0.01 is employed to train the neural-based models.

Figure 2 collects the average execution time of each model for centralized training. Results show that RSFs have a comparable execution time to neural-base models.

5.1.5. RSF Parameters

We optimized the RSF parameter configuration for each dataset adopting a cross-validation approach. The most impactful parameters we discovered were the number of estimators T and the maximum tree depth d. Upon conducting a grid search, we determined the optimal T values within the range of 100 to 4000 for each dataset, analyzed at intervals of 20. We found a point of diminishing returns for each dataset beyond which increasing the number of trees did not significantly improve the results. Then, we fixed the number of trees beyond this identified threshold.

As for d, our investigation considered trees of unrestricted depth and trees with a fixed depth of 1. Meanwhile, we retained the default values for other parameters such as the minimum number of samples needed to split an internal node s and the minimum number of samples required for a leaf node l. Specifically, for the scikit-survival implementation we used, s and l were fixed at 6 and 3, respectively. The maximum number of

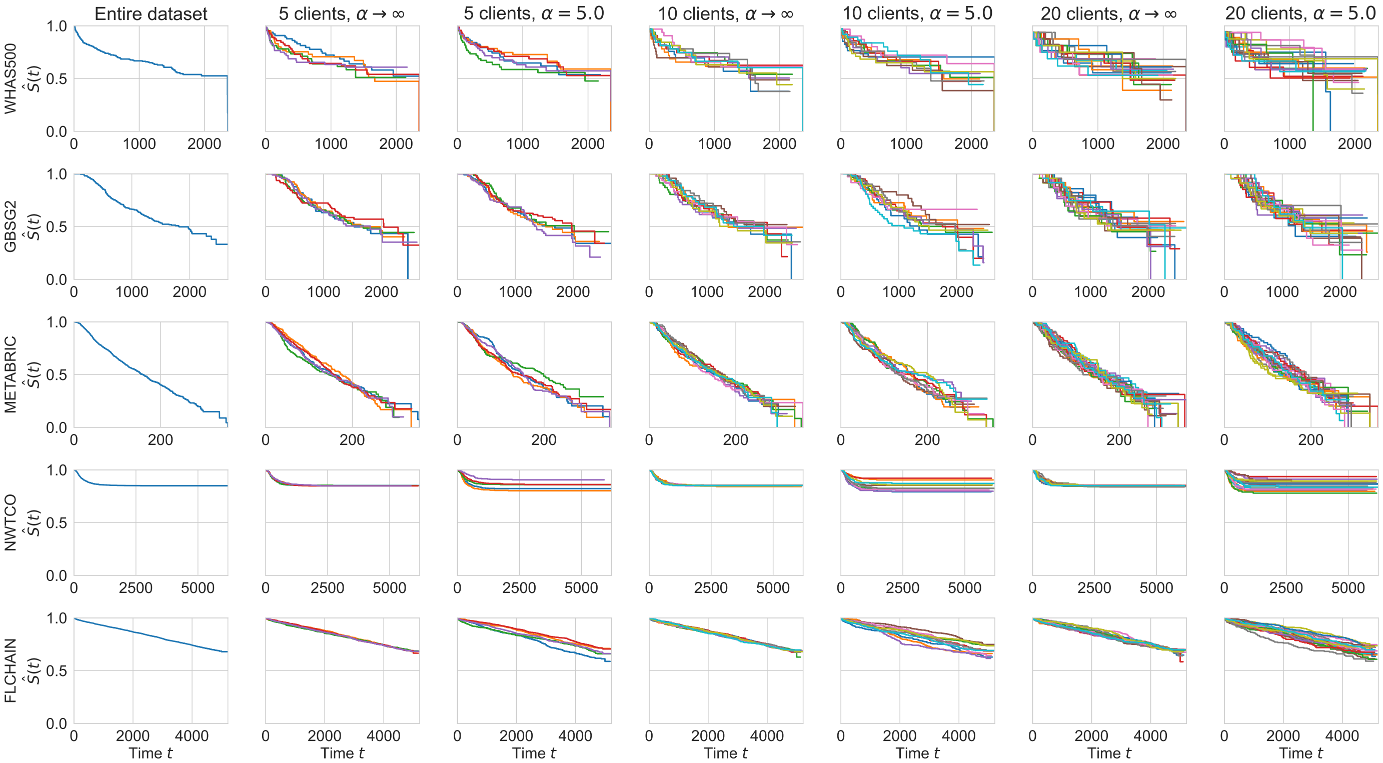

Figure 1: Kaplan-Meier estimators $\hat{S}(t)$ for datasets of simulated federations. The first row shows KM estimators for the entire dataset, while the second, third, and fourth rows depict KM curves for 5, 10, and 20 clients, respectively.

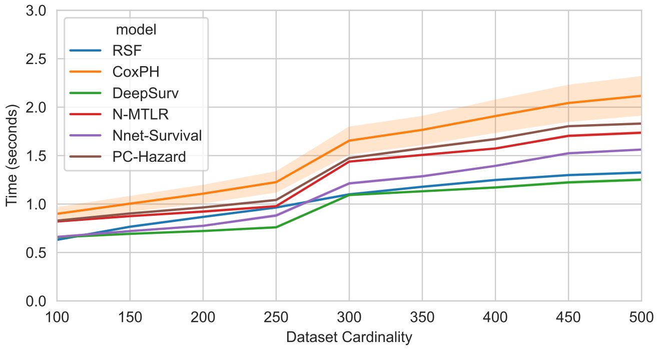

Figure 2: Execution time for each model on several cuts of the GBSG2 dataset. For neural models, time refers to 300 epochs. Results are averaged over 100 runs.

Table 2: RSF parameters for each dataset.

Dataset	T	d
WHAS500 [68]	400	1
GBSG2 [69]	700	1
METABRIC [70, 30]	500	∞
NWTCO [71]	600	1
FLCHAIN [72]	200	∞
LombardyHF [72]	1000	∞
Fed-TCGA-BRCA [72]	1000	∞

features retained for each tree was set to the square root of the total number of features in the dataset. Moreover, we did not impose any constraints on the maximum number of leaf nodes per tree. The parameters determined through this analysis are reported in Table 2. These values were then applied across all clients in our experiments.

5.1.6. *Evaluation*

The Concordance Index (C-Index-IPCW), the Integrated Brier Score (IBS), and the Cumulative Area-Under-the-Curve (Cumulative AUC) are the metrics used to evaluate our survival models on test splits. We account for the censoring distribution by applying the Inverse Probability of Censoring Weighting (IPCW) [38, 41], as described in Section 2.3.

5.1.7. *Simulated Federations Results*

Uniformly Split Data Across 10 Clients. The performance metrics of survival models across five datasets – WHAS500, GBSG2, METABRIC, NWTCO, and FLCHAIN – are illustrated in Tables 3, 4, and 5. Label-skewed splitting [63] with $\alpha \to \infty$ was employed for data assignment to simulate federations with uniform data distribution. This analysis centers around federations comprising 10 clients. The metrics presented include the Concordance Index with IPCW weighting (C-Index-IPCW), Integrated Brier Score (IBS), and Cumulative AUC. The mean values across 20 runs are reported. Metrics for the *Local*, *Federated*, and *Global* settings are given for each dataset.

Table 3: Concordance Index with IPCW weighting (C-Index-IPCW) [38, 41] for survival models evaluated on simulated federations ($K = 10$, $\alpha \to \infty$). Each C-Index-IPCW is scaled by a factor of 100 for better readability. We report the mean computed over 20 runs. The best results (↑) are highlighted in bold. Values marked with * do not exhibit statistically significant differences with FedSurF-C according to Dunn's test with 0.05 significance.

	WHAS500			GBSG2			METABRIC			NWTCO			FLCHAIN		
Model	*Loc.*	*Fed.*	*Glob.*	*Loc.*	*Fed.*	*Glob.*	*Loc.*	*Fed.*	*Glob.*	*Loc.*	*Fed.*	*Glob.*	*Loc.*	*Fed.*	*Glob.*
CoxPH	65.0	74.5	77.8	56.9	61.2	63.7	57.8	62.0*	64.6	63.5	65.9	53.4	90.7	91.7	94.2
DeepSurv	67.8	77.4	73.3	58.3	**65.9***	65.9	60.3	**64.5***	64.8	65.2	**69.7***	55.0	93.3	**94.2**	94.3
DeepHit	66.8	76.8	75.2	56.2	63.5*	64.3	57.4	62.4*	62.0	52.6	69.1*	71.6	92.7	93.7*	94.1
N-MTLR	66.8	76.2	74.4	58.0	65.5*	63.9	59.6	63.8*	64.4	65.2	70.2*	71.6	93.5	**94.2**	94.1
Nnet-Survival	65.0	74.6	75.9	54.9	60.8	63.5	50.3	58.5	62.7	44.3	66.7	70.2	87.8	93.9*	94.2
PC-Hazard	65.1	74.7	75.5	54.9	60.6	63.6	50.4	58.9	62.9	44.8	66.5	70.2	88.1	93.8*	94.2
FedSurF	73.0	79.2*	78.6	61.8	65.4*	64.2	60.7	63.8*	64.0	67.7	69.1*	68.1	93.6	93.8*	93.9
FedSurF-C	–	79.5*	–	–	65.3*	–	–	63.9*	–	–	69.1*	–	–	93.9*	–
FedSurF-C-IPCW	–	79.4*	–	–	65.6*	–	–	63.8*	–	–	69.1*	–	–	93.8*	–
FedSurF-IBS	–	79.3*	–	–	65.6*	–	–	63.9*	–	–	69.0*	–	–	93.9*	–
FedSurF-AUC	–	**79.8***	–	–	65.4*	–	–	64.0*	–	–	69.1*	–	–	93.8*	–

Table 4: Integrated Brier Score (IBS) [39] for survival models evaluated on simulated federations ($K = 10$, $\alpha \to \infty$). Each IBS is scaled by a factor of 100 for better readability. We report the mean computed over 3 runs. The best results (↓) are highlighted in bold. Values marked with * do not exhibit statistically significant differences with FedSurF-C according to Dunn's test with 0.05 significance.

	WHAS500			GBSG2			METABRIC			NWTCO			FLCHAIN		
Model	*Loc.*	*Fed.*	*Glob.*	*Loc.*	*Fed.*	*Glob.*	*Loc.*	*Fed.*	*Glob.*	*Loc.*	*Fed.*	*Glob.*	*Loc.*	*Fed.*	*Glob.*
CoxPH	18.5	**16.5***	14.8	19.1	18.0*	16.8	17.3	16.4*	15.8	11.3	11.1*	11.8	8.4	6.7	4.1
DeepSurv	19.4	**16.5***	21.5	19.1	**16.6***	16.8	17.2	**15.8***	16.2	11.3	**10.4**	11.8	5.7	4.2*	4.2
DeepHit	19.6	17.0*	16.5	20.1	18.2*	17.9	17.7	17.1*	16.3	14.7	10.7*	10.4	6.4	4.5*	4.3
N-MTLR	19.4	17.5*	20.4	19.4	16.8*	17.9	17.2	15.9*	16.1	11.7	**10.4**	10.2	4.9	**4.1***	4.4
Nnet-Survival	22.4	18.6	17.0	21.9	18.7	17.2	22.3	18.8	16.4	16.5	10.8*	10.1	7.7	4.5*	4.2
PC-Hazard	22.1	18.5	17.3	21.6	19.0	17.2	22.3	22.8	16.5	16.2	10.8*	10.1	7.6	4.6	4.2
FedSurF	18.1	17.4*	17.5	18.7	18.0*	18.2	17.3	16.2*	16.2	11.1	11.0*	11.0	4.7	4.5*	4.1
FedSurF-C	–	17.0*	–	–	17.8*	–	–	16.1*	–	–	11.0*	–	–	4.4*	–
FedSurF-C-IPCW	–	17.0*	–	–	17.8*	–	–	16.1*	–	–	11.0*	–	–	4.4*	–
FedSurF-IBS	–	17.0*	–	–	17.9*	–	–	16.1*	–	–	11.0*	–	–	4.3*	–
FedSurF-AUC	–	17.0*	–	–	17.8*	–	–	16.1*	–	–	11.0*	–	–	4.4*	–

Table 5: AUC for survival models evaluated on simulated federations ($K = 10$, $\alpha \to \infty$). Each C-Index-IPCW is scaled by a factor of 100 for better readability. We report the mean computed over 3 runs. The best results (↑) are highlighted in bold. Values marked with * do not exhibit statistically significant differences with FedSurF-C according to Dunn's test with 0.05 significance.

	WHAS500			GBSG2			METABRIC			NWTCO			FLCHAIN		
Model	*Loc.*	*Fed.*	*Glob.*	*Loc.*	*Fed.*	*Glob.*	*Loc.*	*Fed.*	*Glob.*	*Loc.*	*Fed.*	*Glob.*	*Loc.*	*Fed.*	*Glob.*
CoxPH	66.4	75.0	79.3	61.8	69.5	74.4	60.1	65.8	69.0	65.2	67.2*	54.0	92.3	93.2	95.6
DeepSurv	68.0	77.4*	73.0	63.0	74.8*	73.7	63.2	69.0*	68.9	66.1	70.9*	55.1	94.7	95.6*	95.9
DeepHit	65.7	76.0	74.5	58.9	71.5	70.4	59.8	69.0*	69.8	52.7	73.0	74.6	93.9	95.5*	95.4
N-MTLR	67.4	75.9	73.4	62.3	73.4*	72.1	63.3	**70.7***	72.3	67.6	**71.9**	74.6	94.7	95.8*	95.7
Nnet-Survival	62.2	72.6	75.7	57.4	68.2	72.7	48.9	61.7	69.5	36.7	68.5*	72.1	88.9	95.4*	95.9
PC-Hazard	62.3	72.1	74.5	57.7	67.7	73.0	48.8	60.5	69.5	37.2	67.4*	71.8	88.9	94.9	95.3
FedSurF	73.6	79.7*	80.0	68.4	74.9*	73.7	64.8	69.9*	71.0	68.2	69.8*	68.5	94.9	95.6*	96.1
FedSurF-C	–	79.9*	–	–	74.9*	–	–	70.1*	–	–	69.9*	–	–	95.6*	–
FedSurF-C-IPCW	–	79.8*	–	–	**75.4***	–	–	70.2*	–	–	69.9*	–	–	95.6*	–
FedSurF-IBS	–	79.8*	–	–	75.1*	–	–	70.3*	–	–	70.2*	–	–	**95.7***	–
FedSurF-AUC	–	**80.3***	–	–	75.0*	–	–	70.5*	–	–	70.7*	–	–	95.6*	–

The Kruskal-Wallis test, followed by a pairwise Dunn's test at a significance level of 0.05, was conducted to assess statistical differences in the results. We focus on FedSurF-C for its efficient evaluation metric that does not necessitate IPCW weights. Any results not showing a statistically significant difference with the FedSurF-C performance are marked with an asterisk (*).

The results indicate that federated learning, as compared to local training, is advantageous for all clients on average from a performance perspective. In fact, all tables demonstrate superior performance in the *Federated* setting compared to the *Local* setting. Furthermore, the *Federated* performance closely resembles the *Global* performance, signifying that the performance gap between distributed and centralized learning is minimal while offering the added advantage of user privacy preservation in the *Federated* setting.

Table 3, which pertains to the C-Index IPCW, reveals that DeepSurv consistently performs well across datasets among the baseline models. FedSurF achieves comparable performance, particularly in WHAS500, where it surpasses all baselines. However, no clear winner emerges among the sampling techniques. In fact, sampling based on any of the proposed metrics yields better results on average than uniform sampling, but the difference is not statistically significant.

Regarding Table 4 and IBS, survival forests do not outperform neural baselines. Nevertheless, their performance is comparable with no statistically significant difference, particularly in datasets with more samples (METABRIC, NWTCO, and FLCHAIN).

Lastly, Table 8 presents the Cumulative AUC. Here, survival forests exhibit exceptional performance, where the best average AUC is achieved by one of the FedSurF variations, or within a non-statistically significant difference. The only difference is the NWTCO dataset, where DeepHit and N-MTLR exhibit better AUC than FedSurF models.

In summary, when data are uniformly split, FedSurF effectively enhances model performance compared to local models. The FedSurF variations consistently achieve robust performance across diverse evaluation metrics and datasets. However, any sampling strategy produces results close to the best.

Label-skewed Data Across 10 Clients. Adopting the same experimental methodology, Tables 6, 7, and 8 illustrate performance metrics for survival models assessed in federations handling heterogeneous data. The data allocation was conducted utilizing label-skewed splitting[63] with a parameter value of $\alpha = 5$. This evaluation focuses on federations consisting of 10 clients.

In reference to Table 6, variations of the FedSurF algorithm demonstrate similar or superior concordance compared to the neural models, particularly in the context of smaller datasets (WHAS500 and GBSG2). The FLCHAIN dataset is the only exception where FedSurF variants did not perform at par with the top model, albeit the discrepancy was only by a small margin of a few percentage points.

Analyzing from the perspective of the IBS as presented in Table 7, the FedSurF variations tend to rank towards the lower end of the spectrum. In the case of the NWTCO and FLCHAIN datasets, the N-MTLR model exhibits better IBS. However, for the remaining datasets, while FedSurF's average IBS is generally higher, the gap between it and the best-performing model is not statistically significant.

Lastly, Table 8 showcases a promising trend in AUC, with FedSurF models either outperforming or matching the average of other models. NWTCO emerges as the sole exception, where N-MTLR outpaces all other alternatives.

From these results, it is evident that FedSurF variations perform comparably or even surpass neural baselines. With heterogeneously distributed data, FedSurF models typically exhibit better performance than neural models compared to when the data are uniformly split. This suggests that our algorithm is more resilient to federations comprising clients with different data distributions and dataset cardinalities.

Moreover, any FedSurF variation attains roughly the same performance with a slight disadvantage for FedSurF with uniform sampling. Consequently, we recommend employing FedSurF-C, as it is remarkably simple to compute without necessitating integration or IPCW weighting for evaluation. FedSurF++ can thus be considered a viable alternative to neural-based architectures for large-scale survival analysis, as it attains comparable performance with just a single model exchange round.

Federations with Varied Numbers of Clients. To conclude our analysis of simulated federations, we evaluated federations that consisted of a varying number of clients. Specifically, we assessed federations with 5, 10, and 20 clients. The results are summarized in Figure 3. For simplicity, we chose to present only the C-Index-IPCW metrics and focused on the FedSurF-C variation among the FedSurF models.

These results highlight the robustness of FedSurF-C across diverse client configurations. Notably, the performance of FedSurF-C remains consistent regardless of whether the number of clients increased or decreased. This consistency is not matched by neural baselines. For instance, the performance of Nnet-survival and PCH tends to decline as the number of clients increases. Conversely, proportional hazard models (CoxPH and DeepSurv) do not exhibit a performance trend that is proportional to the number of clients. In fact, their results do markedly vary when the number of clients is modified, displaying high variance and thus proving less reliable than the alternatives.

To summarize, although FedSurF-C may not outperform neural models in all configurations, it does display the most consistent concordance when varying the number of clients.

5.2. *Experiments on Real-World Federations*

This section covers the experiments related to real-world heterogeneous datasets, Lombardy Heart Failure [25] (Section 5.2.1) and Fed-TCGA-BRCA [8] (Section 5.2.2).

Experiments on real federated data follow the same procedures and hyperparameters as in simulated federations. Therefore, neural networks have the same structure and training follows the same number of local epochs and rounds of federated averaging. 30% of the local data sets are selected for validation. The same metrics described in Section 2.3 are used to evaluate the methods. The only difference is that the data are already

Table 6: Concordance Index with IPCW weighting (C-Index-IPCW) [38, 41] for survival models evaluated on simulated federations ($K = 10$, $\alpha = 5$). Each C-Index-IPCW is scaled by a factor of 100 for better readability. We report the mean computed over 3 runs. The best results (↑) are highlighted in bold. Values marked with * do not exhibit statistically significant differences with FedSurF-C according to Dunn's test with 0.05 significance.

	WHAS500			GBSG2			METABRIC			NWTCO			FLCHAIN		
Model	*Loc.*	*Fed.*	*Glob.*	*Loc.*	*Fed.*	*Glob.*	*Loc.*	*Fed.*	*Glob.*	*Loc.*	*Fed.*	*Glob.*	*Loc.*	*Fed.*	*Glob.*
CoxPH	65.0	74.3	77.8	56.7	61.8	63.9	57.3	61.6	64.5	63.2	62.7	52.7	88.7	89.7	94.2
DeepSurv	68.8	77.3	73.4	58.0	65.1*	65.7	60.6	**64.2***	64.8	65.4	67.3*	56.4	93.1	89.6*	94.3
DeepHit	67.4	77.1	75.2	57.4	63.0	64.2	57.2	61.3	62.0	52.1	68.8*	71.6	92.5	93.6	94.1
N-MTLR	68.5	76.6	74.6	58.2	64.8*	63.7	60.0	63.8*	64.5	64.9	**70.3***	71.8	93.5	**94.2**	94.1
Nnet-Survival	65.2	74.6	76.1	55.1	61.4	63.5	50.1	58.6	62.7	45.8	66.1	70.3	87.6	93.7*	94.2
PC-Hazard	66.0	75.1	75.6	55.1	61.1	63.6	50.3	58.4	62.8	45.8	66.1	70.3	86.7	93.8*	94.2
FedSurF	73.0	78.4*	78.5	61.9	65.2*	64.2	60.7	63.8*	64.0	67.7	69.2*	68.0	93.6	93.8*	93.8
FedSurF-C	–	**79.3***	–	–	65.1*	–	–	63.8*	–	–	69.1*	–	–	93.8*	–
FedSurF-C-IPCW	–	79.1*	–	–	**65.4***	–	–	63.8*	–	–	69.1*	–	–	93.8*	–
FedSurF-IBS	–	79.0*	–	–	**65.4***	–	–	63.8*	–	–	69.1*	–	–	93.8*	–
FedSurF-AUC	–	79.1*	–	–	**65.4***	–	–	63.7*	–	–	69.2*	–	–	93.8*	–

Table 7: Integrated Brier Score (IBS) [39] for survival models evaluated on simulated federations ($K = 10$, $\alpha = 5$). Each IBS is scaled by a factor of 100 for better readability. We report the mean computed over 3 runs. The best results (↓) are highlighted in bold. Values marked with * do not exhibit statistically significant differences with FedSurF-C according to Dunn's test with 0.05 significance.

	WHAS500			GBSG2			METABRIC			NWTCO			FLCHAIN		
Model	*Loc.*	*Fed.*	*Glob.*	*Loc.*	*Fed.*	*Glob.*	*Loc.*	*Fed.*	*Glob.*	*Loc.*	*Fed.*	*Glob.*	*Loc.*	*Fed.*	*Glob.*
CoxPH	18.6	**16.4***	14.8	19.1	17.8*	16.8	17.4	16.4*	15.8	11.4	12.5	11.9	8.5	7.7	4.1
DeepSurv	19.1	16.5*	21.5	19.1	**16.9***	16.8	17.1	**16.0***	16.2	11.2	10.6*	11.8	5.7	6.1*	4.2
DeepHit	19.6	16.7*	16.5	20.1	18.3	17.9	17.8	17.4	16.3	14.9	10.9*	10.4	6.5	4.5*	4.3
N-MTLR	19.4	17.4*	20.2	19.4	17.1*	18.0	17.3	**16.0***	16.1	11.7	**10.3**	10.1	4.9	**4.2**	4.4
Nnet-Survival	22.1	20.6	16.9	21.7	18.7	17.2	22.5	23.3	16.4	16.2	10.9*	10.1	7.9	4.5*	4.2
PC-Hazard	22.1	18.7	17.2	21.5	18.7	17.2	22.4	18.8	16.5	16.0	10.9*	10.1	8.1	4.5*	4.2
FedSurF	18.2	17.5*	17.5	18.7	18.0*	18.2	17.4	16.2*	16.2	11.3	11.1*	11.0	4.8	4.6	4.1
FedSurF-C	–	17.0*	–	–	17.8*	–	–	16.2*	–	–	11.0*	–	–	4.4*	–
FedSurF-C-IPCW	–	17.1*	–	–	17.8*	–	–	16.2*	–	–	11.0*	–	–	4.4*	–
FedSurF-IBS	–	17.1*	–	–	17.9*	–	–	16.2*	–	–	11.0*	–	–	4.3*	–
FedSurF-AUC	–	17.1*	–	–	17.8*	–	–	16.2*	–	–	11.0*	–	–	4.4*	–

Table 8: Cumulative AUC [40] for survival models evaluated on simulated federations ($K = 10$, $\alpha = 5$). Each Cumulative AUC is scaled by a factor of 100 for better readability. We report the mean computed over 3 runs. The best results (↑) are highlighted in bold. Values marked with * do not exhibit statistically significant differences with FedSurF-C according to Dunn's test with 0.05 significance.

	WHAS500			GBSG2			METABRIC			NWTCO			FLCHAIN		
Model	*Loc.*	*Fed.*	*Glob.*	*Loc.*	*Fed.*	*Glob.*	*Loc.*	*Fed.*	*Glob.*	*Loc.*	*Fed.*	*Glob.*	*Loc.*	*Fed.*	*Glob.*
CoxPH	65.7	75.8	79.4	61.2	70.3	74.6	59.5	65.4	68.9	64.1	64.4*	53.2	90.3	91.5	95.6
DeepSurv	69.1	77.5*	73.2	62.9	73.9*	73.4	63.8	68.9*	68.9	66.7	67.8*	56.7	94.3	91.3*	95.9
DeepHit	66.6	76.0	74.5	60.7	71.3	70.6	60.3	66.4	69.7	52.3	72.9	74.5	93.7	95.3	95.4
N-MTLR	68.4	76.4	73.8	62.6	72.7*	72.0	64.1	**71.0***	72.3	67.8	**73.5**	74.6	94.6	**95.6***	95.8
Nnet-Survival	62.9	72.5	75.9	57.8	68.6	72.7	48.7	60.3	69.4	37.7	67.5*	72.4	88.6	95.2	95.8
PC-Hazard	63.1	72.4	74.7	57.9	68.8	72.9	48.5	61.1	69.3	37.9	67.2*	71.9	87.2	94.8	95.3
FedSurF	73.5	78.8*	79.9	69.0	74.9*	73.6	64.9	70.1*	70.9	68.3	69.9*	68.5	94.8	95.5*	96.1
FedSurF-C	–	**79.8***	–	–	74.8*	–	–	70.2*	–	–	69.8*	–	–	**95.6***	–
FedSurF-C-IPCW	–	79.3*	–	–	**75.2***	–	–	70.2*	–	–	69.9*	–	–	95.5*	–
FedSurF-IBS	–	79.4*	–	–	75.1*	–	–	70.2*	–	–	70.3*	–	–	**95.6***	–
FedSurF-AUC	–	**79.8***	–	–	**75.2***	–	–	70.3*	–	–	70.8*	–	–	95.5*	–

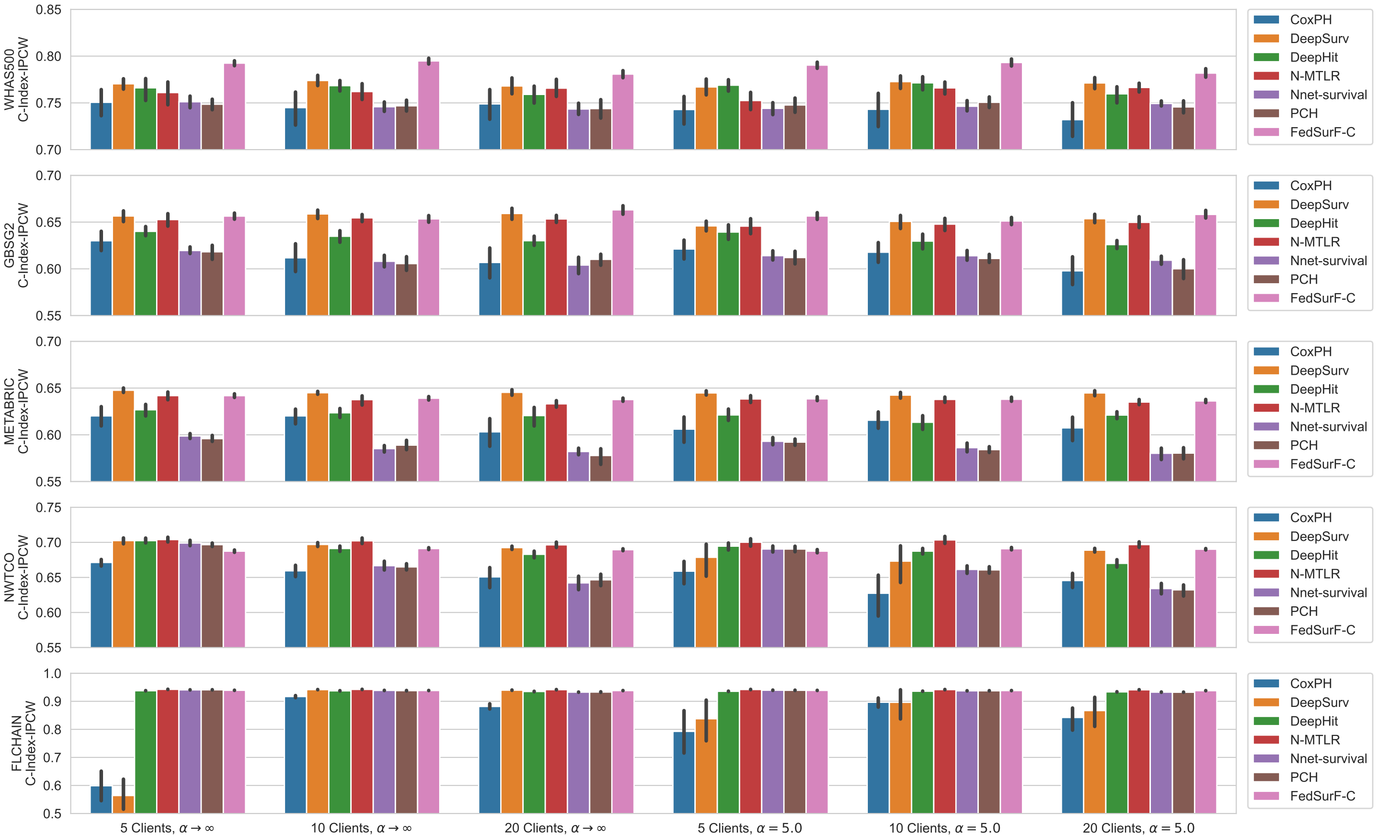

Figure 3: C-Index-IPCW metrics for several federation configurations. Specifically, the number of clients K can be 5, 10, or 20 and the label splitting parameter α either tends to infinity (federation with uniformly split data) or equals 5 (federation with heterogeneous data distributions). Each row corresponds to a survival dataset and each bar to a different survival model. Results are averaged over 20 runs.

distributed across multiple clients, so label-skewed splitting [63] is not applied to create federations.

5.2.1. The Lombardy Heart Failure Dataset

The Lombardy Heart Failure administrative dataset [25] was derived from the HFData research project (RF-2009-1483329), which aimed to examine heart failure cases in Lombardy between 2000 and 2012. Lombardy, one of Italy's largest and most populous regions, has a population of approximately 10 million individuals, accounting for 16.5% of the nation's total population. The dataset was provided by the Regione Lombardia – Healthcare Division and pertains to non-pediatric residents who were hospitalized for heart failure between January 2006 and December 2012. Hospital discharge charts (HDC) were employed to gather information about patients' hospitalization, including the discharge date, length of stay, and comorbidity conditions. Additionally, information about pharmaceutical purchases was obtained from the Anatomical Therapeutic Chemical (ATC) codes. The dataset offers a detailed view of patients' clinical histories of hospitalizations. A description of the preliminary preprocessing and collection can be found in [25].

The initial dataset includes 339,690 samples with 48 features, including several hospitalizations and pharmaceutical prescriptions per patient. For our case study, we focused solely on hospitalizations, reducing the original data to 22,418 samples. To ensure that each patient had a follow-up of at least 5 years and to create a comparable cohort for our algorithms, we eliminated new hospitalizations between 2008 and 2012. This left us with data from 2006 to 2007, with updated time labels to match each patient's actual outcome. Administrative censoring was employed for patients who survived until the end of 2012. Additionally, we removed features related to pharmacological prescriptions, resulting in a dataset with 32 covariates.

Finally, the dataset was split into the federation of medical structures where hospitalization occurred. We excluded medical structures with fewer than 10 events or fewer than 20 total samples in their local data, leaving us with 895 samples distributed across 23 clients. Figure 4 shows the Kaplan-Meier estimators of the entire dataset and the ones related to each client. From these plots, client distributions exhibit significantly different patterns.

5.2.2. The Fed-TCGA-BRCA Dataset

The Fed-TCGA-BRCA survival dataset [8] is a federated dataset for survival analysis based on clinical data from The Cancer Genome Atlas (TCGA) project. TCGA is a large-scale initiative that aims to characterize the genomic changes in various types of cancer. BRCA stands for breast invasive carcinoma, which is one of the cancer types studied by TCGA. The Fed-TCGA-BRCA survival dataset contains survival outcomes for 1088 patients with BRCA and 38 binary features for each patient.

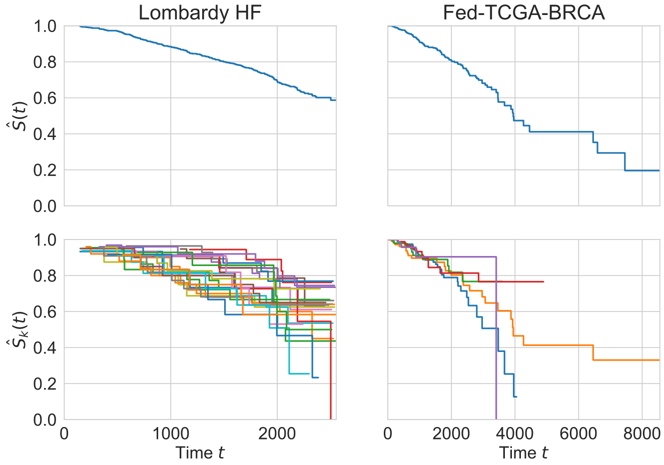

Figure 4: Kaplan-Meier estimators $\hat{S}(t)$ for datasets of real-world federations. The first row shows KM estimators for the entire dataset, while the second row depicts a KM curve for each client k.

The dataset is distributed among six regions (Northeast, South, West, Midwest, Europe, and Canada) based on the tissue sort site (TSS) of each patient. Each of these regions contains 248, 156, 164, 129, 129, and 40 samples respectively. In particular, among the 40 samples from Canada, only a single entry exhibits an event. If the Canada dataset would be split into training and validation, one of the splits would contain no events. For that split, it would not be possible to evaluate the concordance index, as there would be no comparable pairs of subjects. For this reason, we excluded the client corresponding to Canada for training and retained the other five regions. Figure 4 shows the Kaplan-Meier estimators of the entire dataset and the ones related to each client.

5.2.3. Results on Lombardy HF and Fed-TCGA-BRCA

Table 9, Table 10, and Table 11 display performance metrics for neural and ensemble-based survival models evaluated on two real-world federations: Lombardy Heart Failure and Fed-TCGA-BRCA. The metrics include Concordance Index with IPCW weighting (C-Index-IPCW), Integrated Brier Score (IBS), and Cumulative AUC. The reported results represent the mean and standard deviation computed over 20 runs. The Kruskal-Wallis and Dunn's tests with a significance level of 0.05 are conducted to assess statistical differences in the results. Any results not showing a statistically significant difference with FedSurF-C are marked with an asterisk (*).

Experiments on real-world federations corroborate the trend that participating in a federation yields superior results compared to training solely on local data. In fact, for most models, the *Federated* results surpass the *Local* results, occasionally even slightly exceeding the performance of *Global* models.

Table 9 (C-Index-IPCW) indicates that FedSurF-IBS achieves the best performance for Lombardy HF, while DeepSurv exceeds the other models for Fed-TCGA-BRCA. However, for the latter case, no statistically significant difference exists between DeepSurv and FedSurF-C. Consequently, FedSurF variations have a strong discriminative power on real-world data, accurately identifying patients at risk in most cases. Again, the specific metric for sampling trees does not significantly impact the final outcome.

Table 10 (IBS) identifies DeepSurv as the most calibrated model for Lombardy HF, demonstrating its exceptional performance as a survival model when communication constraints are not a concern. In contrast, FedSurF-C exhibits better results by a noticeable margin in the Fed-TCGA-BRCA dataset compared to all neural alternatives.

Finally, Table 11 (Cumulative AUC) showcases the best model performance in terms of discriminative ability. Concerning Fed-TCGA-BRCA, DeepSurv outperforms any other model by a fair margin. Instead, for Lombardy HF, the FedSurF variations outperform neural models, with the only exception of DeepSurv, where the performance difference is not statistically significant.

In summary, FedSurF++ proves to be a valuable alternative to neural-network-based models concerning the most common survival metrics, C-Index-IPCW, IBS, and Cumulative AUC. DeepSurv is a robust alternative from a performance standpoint, but its training procedure needs iterative averaging of model parameters, resulting in substantial bandwidth usage. FedSurF, conversely, may occasionally exhibit lower – yet still comparable – performance metrics while requiring only a single tree exchange round to generate the final model. Regarding the sampling strategy, experiments on real-world data confirm that the metrics considered during sampling do not affect the final results to a significant degree. Therefore, FedSurF-C is the most straightforward choice, relying solely on the local evaluation of the concordance index.

6. Conclusion

In this paper, we presented an extension of the Federated Survival Forest (FedSurF++) algorithm, which applies Random Survival Forests (RSFs) to a federated learning setting. The FedSurF++ algorithm builds upon the original FedSurF by introducing new tree sampling strategies, including concordance index, IPCW concordance index, integrated Brier score, and cumulative AUC. These strategies enable the selection of the best-performing trees from local RSF models, consequently improving the performance of the global RSF model.

Our experimental results on synthetic and real-world clinical trial datasets, covering heart failure and breast cancer genomics, demonstrate the effectiveness of FedSurF++ in various federations. The algorithm outperforms local models and achieves performance metrics comparable to global models, showcasing its robustness across diverse evaluation metrics and datasets. In particular, the FedSurF++ family consistently attains strong performance across different evaluation metrics and datasets. Moreover, our findings reveal that any sampling strategy, except for uniform sampling, yields results close to the best.

While DeepSurv exhibits strong performance, its training procedure requires iterative averaging of model parameters, leading to heavy bandwidth usage. In contrast, FedSurF++ demands only a single tree exchange round to produce the final model,

Table 9: Concordance Index with IPCW weighting (C-Index-IPCW) [38, 41] for survival models evaluated on real-world federations. Each C-Index-IPCW is scaled by a factor of 100 for better readability. We report the mean and the standard deviation computed over 20 runs. The best results (↑) are highlighted in bold. Values marked with * do not exhibit statistically significant differences with FedSurF-C according to Dunn's test with 0.05 significance.

	Lombardy Heart Failure			Fed-TCGA-BRCA		
Model	*Local*	*Federated*	*Global*	*Local*	*Federated*	*Global*
CoxPH	58.4 ± 1.4	69.2 ± 2.4	71.4 ± 0.4	60.3 ± 5.5	75.4 ± 4.6*	77.0 ± 0.9
DeepSurv	59.9 ± 1.9	71.7 ± 0.9	70.6 ± 0.8	61.8 ± 4.4	**78.9 ± 2.1***	72.3 ± 2.5
DeepHit	57.6 ± 1.4	69.1 ± 2.0	71.4 ± 0.9	59.3 ± 4.2	76.4 ± 5.7*	79.9 ± 2.1
N-MTLR	57.7 ± 1.9	70.9 ± 1.2	69.6 ± 0.8	60.6 ± 4.5	77.7 ± 3.4*	75.0 ± 3.2
Nnet-Survival	50.9 ± 1.2	69.2 ± 1.6	71.7 ± 0.5	55.9 ± 3.4	71.4 ± 3.0	77.4 ± 2.4
PC-Hazard	50.8 ± 1.2	69.2 ± 1.3	71.4 ± 0.4	56.9 ± 2.8	68.5 ± 7.4	76.3 ± 2.4
FedSurF	61.7 ± 0.9	73.6 ± 0.9*	72.7 ± 0.1	66.7 ± 3.1	76.3 ± 2.3*	72.3 ± 0.7
FedSurF-C	–	73.5 ± 0.7*	–	–	77.2 ± 2.1*	–
FedSurF-C-IPCW	–	73.6 ± 0.6*	–	–	76.7 ± 2.1*	–
FedSurF-IBS	–	**73.7 ± 0.8***	–	–	76.9 ± 1.8*	–
FedSurF-AUC	–	73.6 ± 0.5*	–	–	77.1 ± 1.9*	–

Table 10: Integrated Brier Score (IBS) [39] for survival models evaluated on real-world federations. Each IBS is scaled by a factor of 100 for better readability. We report the mean and the standard deviation computed over 20 runs. The best results (↓) are highlighted in bold. Values marked with * do not exhibit statistically significant differences with FedSurF-C according to Dunn's test with 0.05 significance.

	Lombardy Heart Failure			Fed-TCGA-BRCA		
Model	*Local*	*Federated*	*Global*	*Local*	*Federated*	*Global*
CoxPH	13.6 ± 0.1	13.0 ± 0.3*	12.4 ± 0.1	28.2 ± 1.9	24.8 ± 2.1*	24.7 ± 0.4
DeepSurv	14.0 ± 0.3	**12.3 ± 0.3**	13.0 ± 0.2	28.8 ± 1.2	25.7 ± 1.6	32.3 ± 4.3
DeepHit	17.5 ± 0.3	14.3 ± 3.6*	12.5 ± 0.1	31.2 ± 0.4	28.2 ± 2.4	27.3 ± 2.2
N-MTLR	15.3 ± 0.3	12.5 ± 0.3	13.2 ± 0.3	29.5 ± 2.1	26.6 ± 4.1	36.2 ± 5.6
Nnet-Survival	20.2 ± 0.4	12.8 ± 0.4*	12.5 ± 0.1	47.3 ± 1.1	37.0 ± 2.8	26.5 ± 1.2
PC-Hazard	20.0 ± 0.5	12.8 ± 0.3*	12.6 ± 0.1	46.7 ± 1.2	43.7 ± 14.9	26.3 ± 0.8
FedSurF	14.2 ± 0.1	13.3 ± 0.1*	12.1 ± 0.0	26.5 ± 0.7	23.2 ± 0.4*	24.2 ± 0.3
FedSurF-C	–	13.1 ± 0.1*	–	–	**22.9 ± 0.3***	–
FedSurF-C-IPCW	–	13.1 ± 0.1*	–	–	23.1 ± 0.4*	–
FedSurF-IBS	–	13.2 ± 0.0*	–	–	23.1 ± 0.3*	–
FedSurF-AUC	–	13.1 ± 0.0*	–	–	23.0 ± 0.3*	–

Table 11: Cumulative AUC [40] for survival models evaluated on real-world federations. Each Cumulative AUC is scaled by a factor of 100 for better readability. We report the mean and the standard deviation computed over 20 runs. The best results (↑) are highlighted in bold. Values marked with * do not exhibit statistically significant differences with FedSurF-C according to Dunn's test with 0.05 significance.

	Lombardy Heart Failure			Fed-TCGA-BRCA		
Model	*Local*	*Federated*	*Global*	*Local*	*Federated*	*Global*
CoxPH	58.6 ± 1.4	70.5 ± 2.7	73.6 ± 0.3	62.7 ± 6.1	79.7 ± 3.6	77.6 ± 0.5
DeepSurv	60.5 ± 2.0	73.5 ± 1.1*	74.2 ± 1.1	63.4 ± 4.9	**80.7 ± 2.0**	71.0 ± 4.3
DeepHit	57.6 ± 1.2	70.4 ± 2.4	72.4 ± 1.1	59.0 ± 5.0	75.0 ± 5.5*	77.2 ± 2.1
N-MTLR	57.4 ± 2.2	71.9 ± 1.1	69.9 ± 1.3	61.3 ± 5.8	77.2 ± 5.0*	72.7 ± 4.0
Nnet-Survival	51.2 ± 1.3	70.3 ± 1.9	73.5 ± 0.8	55.7 ± 3.2	71.1 ± 2.4*	75.7 ± 3.0
PC-Hazard	51.1 ± 1.2	70.2 ± 1.9	72.9 ± 0.7	55.3 ± 2.3	67.0 ± 8.1*	76.0 ± 2.9
FedSurF	60.3 ± 0.9	**74.9 ± 1.3***	73.4 ± 0.2	64.5 ± 3.0	72.9 ± 2.1*	72.1 ± 0.5
FedSurF-C	–	74.8 ± 1.0*	–	–	73.8 ± 1.6*	–
FedSurF-C-IPCW	–	**74.9 ± 1.1***	–	–	73.0 ± 1.9*	–
FedSurF-IBS	–	74.8 ± 1.2*	–	–	73.9 ± 1.7*	–
FedSurF-AUC	–	**74.9 ± 1.0***	–	–	73.7 ± 1.4*	–

minimizing the communication overhead. In conclusion, the FedSurF++ algorithm proves to be a valuable alternative to neural-network-based models for large-scale survival analysis on confidential clinical data as it achieves comparable performance while preserving data privacy and offering an efficient solution in terms of communication.

Acknowledgment

This project has been supported by AI-SPRINT: AI in Secure Privacy-pReserving computINg conTinuum (European Union H2020 grant agreement No. 101016577) and FAIR: Future Artificial Intelligence Research (NextGenerationEU, PNRR-PE-AI scheme, M4C2, investment 1.3, line on Artificial Intelligence).

Declaration of Generative AI and AI-Assisted Technologies in the Writing Process

During the preparation of this work, the authors used ChatGPT (May 24, 2023 version) in order to improve readability and language. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

References

[1] J. P. Klein, M. L. Moeschberger, Survival analysis: techniques for censored and truncated data, Vol. 1230, Springer, 2003.

[2] P. Wang, Y. Li, C. K. Reddy, Machine learning for survival analysis: A survey, ACM Computing Surveys (CSUR) 51 (6) (2019) 1–36.

[3] M. Andreux, A. Manoel, R. Menuet, C. Saillard, C. Simpson, Federated survival analysis with discrete-time cox models, arXiv preprint arXiv:2006.08997.

[4] N. Rieke, J. Hancox, W. Li, F. Milletari, H. R. Roth, S. Albarqouni, S. Bakas, M. N. Galtier, B. A. Landman, K. Maier-Hein, et al., The future of digital health with federated learning, NPJ digital medicine 3 (1) (2020) 1–7.

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

[6] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, et al., Advances and open problems in federated learning, Foundations and Trends in Machine Learning 14 (1–2) (2021) 1–210.

[7] M. Y. Lu, R. J. Chen, D. Kong, J. Lipkova, R. Singh, D. F. Williamson, T. Y. Chen, F. Mahmood, Federated learning for computational pathology on gigapixel whole slide images, Medical Image Analysis 76 (2022) 102298. doi:10.1016/j.media.2021.102298.
URL `https://linkinghub.elsevier.com/retrieve/pii/S1361841521003431`

[8] J. Ogier du Terrail, S.-S. Ayed, E. Cyffers, F. Grimberg, C. He, R. Loeb, P. Mangold, T. Marchand, O. Marfoq, E. Mushtaq, B. Muzellec, C. Philippenko, S. Silva, M. Teleńczuk, S. Albarqouni, S. Avestimehr, A. Bellet, A. Dieuleveut, M. Jaggi, S. P. Karimireddy, M. Lorenzi, G. Neglia, M. Tommasi, M. Andreux, Flamby: Datasets and benchmarks for cross-silo federated learning in realistic healthcare settings, in: S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, A. Oh (Eds.), Advances in Neural Information Processing Systems, Vol. 35, Curran Associates, Inc., 2022, pp. 5315–5334.
URL `https://proceedings.neurips.cc/paper_files/paper/2022/file/232eee8ef411a0a316efa298d7be3c2b-Paper-Datasets_and_Benchmarks.pdf`

[9] R. Duan, C. Luo, M. J. Schuemie, J. Tong, C. J. Liang, H. H. Chang, M. R. Boland, J. Bian, H. Xu, J. H. Holmes, C. B. Forrest, S. C. Morton, J. A. Berlin, J. H. Moore, K. B. Mahoney, Y. Chen, Learning from local to global: An efficient distributed algorithm for modeling time-to-event data, Journal of the American Medical Informatics Association 27 (7) (2020) 1028–1036. doi:10.1093/jamia/ocaa044.
URL `https://doi.org/10.1093/jamia/ocaa044`

[10] X. Wang, H. G. Zhang, X. Xiong, C. Hong, G. M. Weber, G. A. Brat, C.-L. Bonzel, Y. Luo, R. Duan, N. P. Palmer, et al., Survmaximin: robust federated approach to transporting survival risk prediction models, Journal of biomedical informatics 134 (2022) 104176.

[11] D. Froelicher, J. R. Troncoso-Pastoriza, J. L. Raisaro, M. A. Cuendet, J. S. Sousa, H. Cho, B. Berger, J. Fellay, J.-P. Hubaux, Truly privacy-preserving federated analytics for precision medicine with multiparty homomorphic encryption, Nature Communications 12 (1) (2021) 5910. doi:10.1038/s41467-021-25972-y.
URL `https://www.nature.com/articles/s41467-021-25972-y`

[12] C.-L. Lu, S. Wang, Z. Ji, Y. Wu, L. Xiong, X. Jiang, L. 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) (2015) 1212–1219.

[13] S. Banerjee, G. N. Sofack, T. Papakonstantinou, D. Avraam, P. Burton, D. Zöller, T. R. P. Bishop, dsSurvival: Privacy preserving survival models for federated individual patient meta-analysis in DataSHIELD, BMC Research Notes 15 (1) (2022) 197. doi:10.1186/s13104-022-06085-1.
URL `https://bmcresnotes.biomedcentral.com/articles/10.1186/s13104-022-06085-1`

[14] W. Dai, X. Jiang, L. Bonomi, Y. Li, H. Xiong, L. Ohno-Machado, VERTICOX: Vertically Distributed Cox Proportional Hazards Model Using the Alternating Direction Method of Multipliers, IEEE Transactions on Knowledge and Data Engineering 34 (2) (2022) 996–1010. doi:10.1109/TKDE.2020.2989301.
URL `https://ieeexplore.ieee.org/document/9076318/`

[15] C. R. Hansen, G. Price, M. Field, N. Sarup, R. Zukauskaite, J. Johansen, J. G. Eriksen, F. Aly, A. McPartlin, L. Holloway, D. Thwaites, C. Brink, Larynx cancer survival model developed through open-source federated learning, Radiotherapy and Oncology 176 (2022) 179–186. doi:10.1016/j.radonc.2022.09.023.
URL `https://linkinghub.elsevier.com/retrieve/pii/S0167814022044930`

[16] B. Kamphorst, T. Rooijakkers, T. Veugen, M. Cellamare, D. Knoors, Accurate training of the Cox proportional hazards model on vertically-partitioned data while preserving privacy, BMC Medical Informatics and Decision Making 22 (1) (2022) 49. doi:10.1186/s12911-022-01771-3.
URL `https://bmcmedinformdecismak.biomedcentral.com/articles/10.1186/s12911-022-01771-3`

[17] C. Masciocchi, B. Gottardelli, M. Savino, L. Boldrini, A. Martino, C. Mazzarella, M. Massaccesi, V. Valentini, A. Damiani, Federated Cox Proportional Hazards Model with multicentric privacy-preserving LASSO feature selection for survival analysis from the perspective of personalized medicine, in: 2022 IEEE 35th International Symposium on Computer-Based Medical Systems (CBMS), IEEE, Shenzen, China, 2022, pp. 25–31. doi:10.1109/CBMS55023.2022.00012.
URL `https://ieeexplore.ieee.org/document/9867090/`

[18] A. Imakura, R. Tsunoda, R. Kagawa, K. Yamagata, T. Sakurai, DC-COX: Data collaboration Cox proportional hazards model for privacy-preserving survival analysis on multiple parties, Journal of Biomedical Informatics 137 (2023) 104264. doi:10.1016/j.jbi.2022.104264.
URL `https://linkinghub.elsevier.com/retrieve/pii/S1532046422002696`

[19] D. K. Zhang, F. Toni, M. Williams, A federated cox model with non-proportional hazards, in: Multimodal AI in Healthcare, Springer, 2023, pp. 171–185.

[20] S. Rahimian, R. Kerkouche, I. Kurth, M. Fritz, Practical challenges in differentially-private federated survival analysis of medical data, in: Conference on Health, Inference, and Learning, PMLR, 2022, pp. 411–425.

[21] M. M. Rahman, S. Purushotham, Fedpseudo: Pseudo value-based deep learning models for federated survival analysis, arXiv preprint arXiv:2207.05247.

[22] A. Chowdhury, H. Kassem, N. Padoy, R. Umeton, A. Karargyris, A review of medical federated learning: Applications in oncology and cancer re-

search, in: International MICCAI Brainlesion Workshop, Springer, 2021, pp. 3–24.

[23] A. Archetti, M. Matteucci, Federated Survival Forests, in: 2023 International Joint Conference on Neural Networks (IJCNN2023), IEEE (in press), 2023.

[24] H. Ishwaran, U. B. Kogalur, E. H. Blackstone, M. S. Lauer, Random survival forests, The annals of applied statistics 2 (3) (2008) 841–860.

[25] C. Mazzali, A. M. Paganoni, F. Ieva, C. Masella, M. Maistrello, O. Agostoni, S. Scalvini, M. Frigerio, Methodological issues on the use of administrative data in healthcare research: the case of heart failure hospitalizations in lombardy region, 2000 to 2012, BMC Health Services Research 16.

[26] E. L. Kaplan, P. Meier, Nonparametric estimation from incomplete observations, Journal of the American statistical association 53 (282) (1958) 457–481.

[27] W. Nelson, Theory and applications of hazard plotting for censored failure data, Technometrics 14 (4) (1972) 945–966.

[28] O. Aalen, Nonparametric inference for a family of counting processes, The Annals of Statistics (1978) 701–726.

[29] D. R. Cox, Regression models and life-tables, Journal of the Royal Statistical Society. Series B (Methodological) 34 (2) (1972) 187–220.
URL `http://www.jstor.org/stable/2985181`

[30] J. L. Katzman, U. Shaham, A. Cloninger, J. Bates, T. Jiang, Y. Kluger, Deepsurv: personalized treatment recommender system using a cox proportional hazards deep neural network, BMC medical research methodology 18 (1) (2018) 1–12.

[31] C. Lee, W. Zame, J. Yoon, M. Van Der Schaar, Deephit: A deep learning approach to survival analysis with competing risks, in: Proceedings of the AAAI conference on artificial intelligence, Vol. 32, 2018.

[32] S. Fotso, Deep neural networks for survival analysis based on a multi-task framework, arXiv preprint arXiv:1801.05512.

[33] C.-N. Yu, R. Greiner, H.-C. Lin, V. Baracos, Learning patient-specific cancer survival distributions as a sequence of dependent regressors, Advances in neural information processing systems 24.

[34] M. F. Gensheimer, B. Narasimhan, A scalable discrete-time survival model for neural networks, PeerJ 7 (2019) e6257.

[35] H. Kvamme, Ø. Borgan, Continuous and discrete-time survival prediction with neural networks, Lifetime Data Analysis 27 (4) (2021) 710–736.

[36] A. Bender, D. Rügamer, F. Scheipl, B. Bischl, A general machine learning framework for survival analysis, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2021, pp. 158–173.

[37] L. Breiman, J. H. Friedman, R. A. Olshen, C. J. Stone, Classification and regression trees, Routledge, 2017.

[38] H. Uno, T. Cai, M. J. Pencina, R. B. D'Agostino, L.-J. Wei, On the c-statistics for evaluating overall adequacy of risk prediction procedures with censored survival data, Statistics in medicine 30 (10) (2011) 1105–1117.

[39] E. Graf, C. Schmoor, W. Sauerbrei, M. Schumacher, Assessment and comparison of prognostic classification schemes for survival data, Statistics in medicine 18 (17-18) (1999) 2529–2545.

[40] S. Pölsterl, scikit-survival: A library for time-to-event analysis built on top of scikit-learn, Journal of Machine Learning Research 21 (212) (2020) 1–6.
URL `http://jmlr.org/papers/v21/20-729.html`

[41] J. M. Robins, A. Rotnitzky, Recovery of information and adjustment for dependent censoring using surrogate markers, in: AIDS epidemiology, Springer, 1992, pp. 297–331.

[42] B. McMahan, E. Moore, D. Ramage, S. Hampson, B. A. y Arcas, Communication-efficient learning of deep networks from decentralized data, in: Artificial intelligence and statistics, PMLR, 2017, pp. 1273–1282.

[43] Y. Chen, Y. Ning, M. Slawski, H. Rangwala, Asynchronous online federated learning for edge devices with non-iid data, in: 2020 IEEE International Conference on Big Data (Big Data), 2020, pp. 15–24. doi:10.1109/BigData50022.2020.9378161.

[44] S. Reddi, Z. Charles, M. Zaheer, Z. Garrett, K. Rush, J. Konečný, S. Kumar, H. B. McMahan, Adaptive federated optimization, arXiv preprint arXiv:2003.00295.

[45] J. Wang, Z. Charles, Z. Xu, G. Joshi, H. B. McMahan, M. Al-Shedivat, G. Andrew, S. Avestimehr, K. Daly, D. Data, et al., A field guide to federated optimization, arXiv preprint arXiv:2107.06917.

[46] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, V. Smith, Federated optimization in heterogeneous networks, Proceedings of Machine Learning and Systems 2 (2020) 429–450.

[47] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, A. T. Suresh, Scaffold: Stochastic controlled averaging for federated learning, in: International Conference on Machine Learning, PMLR, 2020, pp. 5132–5143.

[48] D. A. E. Acar, Y. Zhao, R. M. Navarro, M. Mattina, P. N. Whatmough, V. Saligrama, Federated learning based on dynamic regularization, arXiv preprint arXiv:2111.04263.

[49] S. Caldas, S. M. K. Duddu, P. Wu, T. Li, J. Konečný, H. B. McMahan, V. Smith, A. Talwalkar, Leaf: A benchmark for federated settings, arXiv preprint arXiv:1812.01097.

[50] E. Lomurno, A. Archetti, L. Cazzella, S. Samele, L. Di Perna, M. Matteucci, SGDE: Secure generative data exchange for cross-silo federated learning, in: AIPR 2022, International Conference on Artificial Intelligence and Pattern Recognition, 2022.

[51] T.-M. H. Hsu, H. Qi, M. Brown, Measuring the effects of non-identical data distribution for federated visual classification, arXiv preprint arXiv:1909.06335.

[52] Q. Li, Y. Diao, Q. Chen, B. He, Federated learning on non-iid data silos: An experimental study, in: 2022 IEEE 38th International Conference on Data Engineering (ICDE), IEEE, 2022, pp. 965–978.

[53] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. Van Der Laak, B. Van Ginneken, C. I. Sánchez, A survey on deep learning in medical image analysis, Medical image analysis 42 (2017) 60–88.

[54] J. D. Frizzell, L. Liang, P. J. Schulte, C. W. Yancy, P. A. Heidenreich, A. F. Hernandez, D. L. Bhatt, G. C. Fonarow, W. K. Laskey, Prediction of 30-day all-cause readmissions in patients hospitalized for heart failure: comparison of machine learning and other statistical approaches, JAMA cardiology 2 (2) (2017) 204–209.

[55] T. Yue, H. Wang, Deep learning for genomics: A concise overview, arXiv preprint arXiv:1802.00810.

[56] F. Piccialli, V. Di Somma, F. Giampaolo, S. Cuomo, G. Fortino, A survey on deep learning in medicine: Why, how and when?, Information Fusion 66 (2021) 111–137.

[57] J. Xu, B. S. Glicksberg, C. Su, P. Walker, J. Bian, F. Wang, Federated learning for healthcare informatics, Journal of Healthcare Informatics Research 5 (2021) 1–19.

[58] M. J. Sheller, B. Edwards, G. A. Reina, J. Martin, S. Pati, A. Kotrotsou, M. Milchenko, W. Xu, D. Marcus, R. R. Colen, et al., Federated learning in medicine: facilitating multi-institutional collaborations without sharing patient data, Scientific reports 10 (1) (2020) 1–12.

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

[60] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, Distributed optimization and statistical learning via the alternating direction method of multipliers, Found. Trends Mach. Learn. 3 (1) (2011) 1–122. doi:10.1561/2200000016.
URL `https://doi.org/10.1561/2200000016`

[61] C. Dwork, Differential privacy: A survey of results, in: International conference on theory and applications of models of computation, Springer, 2008, pp. 1–19.

[62] T. Marchand, B. Muzellec, C. Beguier, J. O. d. Terrail, M. Andreux, SecureFedYJ: a safe feature Gaussianization protocol for Federated Learning, arXiv:2210.01639 [cs] (Oct. 2022).
URL `http://arxiv.org/abs/2210.01639`

[63] A. Archetti, E. Lomurno, F. Lattari, A. Martin, M. Matteucci, Heterogeneous datasets for federated survival analysis simulation, in: Companion of the 2023 ACM/SPEC International Conference on Performance Engineering, ICPE '23 Companion, Association for Computing Machinery, New York, NY, USA, 2023, p. 173–180. doi:10.1145/3578245.3584935.
URL `https://doi.org/10.1145/3578245.3584935`

[64] A.-C. Hauschild, M. Lemanczyk, J. Matschinske, T. Frisch, O. Zolotareva, A. Holzinger, J. Baumbach, D. Heider, Federated random forests can improve local performance of predictive models for various healthcare applications, Bioinformatics 38 (8) (2022) 2278–2286.

[65] M. Gencturk, A. A. Sinaci, N. K. Cicekli, Bofrf: A novel boosting-based federated random forest algorithm on horizontally partitioned data, IEEE Access 10 (2022) 89835–89851.

[66] L. Breiman, J. Friedman, R. Olshen, C. Stone, Classification and regression trees.

[67] J. M. Bland, D. G. Altman, The logrank test, Bmj 328 (7447) (2004) 1073.

[68] D. W. Hosmer, S. Lemeshow, S. May, Applied Survival Analysis: Regression Modeling of Time-to-Event Data, Wiley Series in Probability and Statistics, John Wiley & Sons, Inc., Hoboken, NJ, USA, 2008. doi:10.1002/9780470258019.
URL http://doi.wiley.com/10.1002/9780470258019

[69] M. Schumacher, G. Bastert, H. Bojar, K. Hübner, M. Olschewski, W. Sauerbrei, C. Schmoor, C. Beyerle, R. Neumann, H. Rauschecker, Randomized 2 x 2 trial evaluating hormonal treatment and the duration of chemotherapy in node-positive breast cancer patients. german breast cancer study group., Journal of Clinical Oncology 12 (10) (1994) 2086–2093.

[70] B. Pereira, S.-F. Chin, O. M. Rueda, H.-K. M. Vollan, E. Provenzano, H. A. Bardwell, M. Pugh, L. Jones, R. Russell, S.-J. Sammut, D. W. Y. Tsui, B. Liu, S.-J. Dawson, J. Abraham, H. Northen, J. F. Peden, A. Mukherjee, G. Turashvili, A. R. Green, S. McKinney, A. Oloumi, S. Shah, N. Rosenfeld, L. Murphy, D. R. Bentley, I. O. Ellis, A. Purushotham, S. E. Pinder, A.-L. Børresen-Dale, H. M. Earl, P. D. Pharoah, M. T. Ross, S. Aparicio, C. Caldas, The somatic mutation profiles of 2,433 breast cancers refine their genomic and transcriptomic landscapes, Nature Communications 7 (1) (2016) 11479. doi:10.1038/ncomms11479.
URL https://www.nature.com/articles/ncomms11479

[71] N. E. Breslow, N. Chatterjee, Design and Analysis of Two-Phase Studies with Binary Outcome Applied to Wilms Tumour Prognosis, Journal of the Royal Statistical Society Series C: Applied Statistics 48 (4) (1999) 457–468. doi:10.1111/1467-9876.00165.
URL https://academic.oup.com/jrsssc/article/48/4/457/6990670

[72] T. Therneau, T. Lumley, E. Atkinson, C. Crowson, R package: survival (Jan. 9, 2023).
URL https://stat.ethz.ch/R-manual/R-devel/library/survival/html/00Index.html

[73] A. Dispenzieri, J. A. Katzmann, R. A. Kyle, D. R. Larson, T. M. Therneau, C. L. Colby, R. J. Clark, G. P. Mead, S. Kumar, L. J. Melton, S. V. Rajkumar, Use of Nonclonal Serum Immunoglobulin Free Light Chains to Predict Overall Survival in the General Population, Mayo Clinic Proceedings 87 (6) (2012) 517–523. doi:10.1016/j.mayocp.2012.03.009.
URL https://linkinghub.elsevier.com/retrieve/pii/S0025619612003886

[74] D. J. Beutel, T. Topal, A. Mathur, X. Qiu, J. Fernandez-Marques, Y. Gao, L. Sani, H. L. Kwing, T. Parcollet, P. P. d. Gusmão, N. D. Lane, Flower: A friendly federated learning research framework, arXiv preprint arXiv:2007.14390.

[75] H. Kvamme, Ø. Borgan, I. Scheel, Time-to-event prediction with neural networks and cox regression, arXiv preprint arXiv:1907.00825.

MOX Technical Reports, last issues

Dipartimento di Matematica
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

74/2023 Pidò, S.; Pinoli, P.; Crovari, P.; Ieva, F.; Garzotto, F.; Ceri, S.
Ask Your Data—Supporting Data Science Processes by Combining AutoML and Conversational Interfaces

71/2023 Conni, G.; Piccardo, S.; Perotto, S.; Porta, G.M.; Icardi, M.
HiPhome: HIgh order Projection-based HOMogEnisation for advection diffusion reaction problems

70/2023 Ragni, A.; Ippolito, D.; Masci, C.
Assessing the Impact of Hybrid Teaching on Students' Academic Performance via Multilevel Propensity Score-based techniques

69/2023 Ferro, N.; Micheletti, S.; Parolini, N.; Perotto, S.; Verani, M.; Antonietti, P. F.
Level set-fitted polytopal meshes with application to structural topology optimization

68/2023 Vitullo, P.; Colombo, A.; Franco, N.R.; Manzoni, A.; Zunino, P.
Nonlinear model order reduction for problems with microstructure using mesh informed neural networks

67/2023 Conti, P.; Guo, M.; Manzoni, A.; Hesthaven, J.S.
Multi-fidelity surrogate modeling using long short-term memory networks

66/2023 Fresca, S.; Gobat, G.; Fedeli, P.; Frangi, A.; Manzoni, A.
Deep learning-based reduced order models for the real-time simulation of the nonlinear dynamics of microstructures

65/2023 Gatti, F.; de Falco, C.; Perotto, S.; Formaggia, L.; Pastor, M.
A scalable well-balanced numerical scheme for the modelling of two-phase shallow granular landslide consolidation

Gatti, F.; de Falco, C.; Perotto, S.; Formaggia, L.; Pastor, M.
A scalable well-balanced numerical scheme for the modeling of two-phase shallow granular landslide consolidation

64/2023 Heltai, L.; Zunino, P.
Reduced Lagrange multiplier approach for non-matching coupling of mixed-dimensional domains