POLITECNICO DI TORINO
Repository ISTITUZIONALE

Federated Survival Forests

Original
Federated Survival Forests / Archetti, A., Matteucci, M.. - ELETTRONICO. - 0:(2023), pp. 1-9. (2023 International Joint Conference on Neural Networks Gold Coast (AU) 18-23 June 2023) [10.1109/IJCNN54540.2023.10190999].

Availability:
This version is available at: 11583/2980985 since: 2023-08-22T08:59:11Z

Publisher:
IEEE

Published
DOI:10.1109/IJCNN54540.2023.10190999

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright
IEEE postprint/Author's Accepted Manuscript

©2023 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collecting works, for resale or lists, or reuse of any copyrighted component of this work in other works.

(Article begins on next page)

16 November 2026

Federated Survival Forests

Alberto Archetti
DEIB
Politecnico di Milano
Milan, Italy
alberto.archetti@polito.it

Matteo Matteucci
DEIB
Politecnico di Milano
Milan, Italy
matteo.matteucci@polimi.it

***Abstract*—Survival analysis is a subfield of statistics concerned with modeling the occurrence time of a particular event of interest for a population. Survival analysis found widespread applications in healthcare, engineering, and social sciences. However, real-world applications involve survival datasets that are distributed, incomplete, censored, and confidential. In this context, federated learning can tremendously improve the performance of survival analysis applications. Federated learning provides a set of privacy-preserving techniques to jointly train machine learning models on multiple datasets without compromising user privacy, leading to a better generalization performance. However, despite the widespread development of federated learning in recent AI research, few studies focus on federated survival analysis. In this work, we present a novel federated algorithm for survival analysis based on one of the most successful survival models, the random survival forest. We call the proposed method Federated Survival Forest (FedSurF). With a single communication round, FedSurF obtains a discriminative power comparable to deep-learning-based federated models trained over hundreds of federated iterations. Moreover, FedSurF retains all the advantages of random forests, namely low computational cost and natural handling of missing values and incomplete datasets. These advantages are especially desirable in real-world federated environments with multiple small datasets stored on devices with low computational capabilities. Numerical experiments compare FedSurF with state-of-the-art survival models in federated networks, showing how FedSurF outperforms deep-learning-based federated algorithms in realistic environments with non-identically distributed data.**

***Index Terms*—deep learning, federated learning, random forest, survival analysis**

I. Introduction

Survival analysis [1], [2], also known as time-to-event analysis, is a subfield of statistics concerned with modeling the time until an event of interest occurs for a population of individuals. In particular, a survival model builds on statistical and machine learning techniques to predict a survival function $S(t)$. This function evaluates the probability of a subject not experiencing the event up to time t. Survival analysis found widespread success in healthcare, engineering, economics, and social science applications [2]. However, in real-world scenarios survival data are often distributed, inaccurate, and incomplete [3], [4]. In addition, survival data may contain considerable proportions of censored samples, i.e., data records for which an individual has not experienced the event yet. Even if survival models are suited to extract all the useful information from censored samples, high censorship percentages in datasets still pose severe challenges for survival models to succeed. One of the possible solutions is to increase the number of data samples available for training. However, most survival applications rely on confidential data, which is private and non-sharable due to security or regulatory constraints [3], [4].

In this context, Federated Learning (FL) [5], [6] can significantly improve the success of survival applications in real-world scenarios. FL is a machine learning field in which a set of clients, each holding a private dataset, collaborate to train a machine learning model under the coordination of a central server. The crucial difference between distributed learning and FL is that private data information never leaves the device in which it is collected and stored. FL operates by iteratively exchanging model parameters between the clients and the central server to build an average model with good performance for all the clients in the federation. In this way, FL allows multiple parties to collectively train a machine learning model without leaking private data information stored on local devices by design. Federated models exhibit better generalization performance than their local counterparts since they can train on a large and representative data pool.

FL can support survival analysis in overcoming the limitations of scarce, censored, and confidential survival data. To this end, the field of federated survival analysis investigates the techniques to integrate survival models into federated algorithms. Among the works related to federated survival analysis, most focus on Cox models [3], [7]–[16]. Despite being prominent in classical survival analysis, the Cox model is based on the proportionality assumption, which may not hold in large-scale federated datasets. Other works extend federated survival analysis to deep neural models [17]–[19]. While being extremely powerful, training these models requires several communication rounds, which may hinder convergence speed and bandwidth usage. To the best of our knowledge, ensemble learning for federated survival analysis is yet unexplored.

In this work, we propose a novel federated adaptation of one of the most successful survival models from the machine learning literature, the Random Survival Forest (RSF) [20]. We call the proposed algorithm Federated Survival Forest (FedSurF). In FedSurF, each client trains a survival forest on their data locally and sends a carefully chosen subset of trees composing the local forest to the central server. Then, the central server builds the ensemble of trees trained on the client devices. For each client, tree selection can occur randomly

with uniform probability or according to local evaluation. Specifically, we discuss a tree sampling technique based on the Integrated Brier Score (IBS) [21].

With FedSurF, our goal is to bring RSFs and their advantages with respect to the state-of-the-art survival models to the federated setting. In particular, RSFs have a lower computational demand than deep learning models and require less hyperparameter tuning, making them less inclined to overfitting. Secondly, RSFs naturally deal with missing values and categorical variables, making them able to extract most of the useful information from incomplete, inaccurate, and censored local datasets. Thirdly, RSFs can be interpreted more naturally due to their tree-based nature. Finally, concerning the federated training aspects, FedSurF requires a single communication round between the clients and the central server to build the final model. This quality makes FedSurF much more efficient than the traditional iterative algorithms to train federated deep learning models from an inter-node communication perspective.

FedSurF has been tested in federated scenarios alongside several state-of-the-art survival models. Numerical experiments demonstrate the efficacy of the proposed algorithm to solve survival problems with a high generalization power. Moreover, experiments in federations with non-identically distributed data show the resilience of FedSurF to realistic scenarios in which each client exhibits a different data distribution. In particular, FedSurF outperforms deep-learning models in most of these scenarios.

The rest of the paper is organized as follows. Section II provides a comprehensive overview of the background and current state of the art in federated learning and survival analysis. Section III describes our proposed algorithm, FedSurF. Section IV presents a thorough analysis of the experimental evidence supporting the efficacy of FedSurF in comparison to existing approaches. All experimental procedures are extensively described and the source code is made publicly available to promote reproducibility. Finally, Section V summarizes the work.

II. Background and Related Works

This section presents an overview of the relevant literature on survival analysis and federated learning, with a focus on the current state of research in the field of federated survival analysis.

A. Survival Analysis

Survival analysis, or time-to-event analysis, is a subfield of statistics that models the occurrence time of an event of interest for a population. The main goal of a survival problem is to estimate the event occurrence probability as a function of time. More formally, the output of a survival model is the survival function

$$S(t|\mathbf{x}) = P(T > t|\mathbf{x})$$

which evaluates the probability of a particular subject not having experienced the event up to time t. The subject is characterized by a feature vector $\mathbf{x} \in \mathbb{R}^d$. The survival function is estimated from data using machine learning and statistical techniques starting from a survival dataset. A survival dataset D is a set of triplets $(\mathbf{x}_i, \delta_i, t_i)$, $i = 1, \ldots, N$. $\mathbf{x}_i$ is the d-dimensional feature vector characterizing the i-th sample. δ_i is the event occurrence indicator. If $\delta_i = 1$, the subject experienced the event at time t_i. Conversely, if $\delta_i = 0$, the subject did not experience the event and the sample is censored. $t_i = \min\{t_i^c, t_i^e\}$ is the minimum between the censor time t_i^c and the actual event occurrence time t_i^e for the i-th subject.

There are three types of survival models: non-parametric, semi-parametric, and parametric. Non-parametric models make no assumption about the underlying distribution of event times. These models are mostly used for data visualization, as they encode the summary statistics of survival data. Non-parametric survival models are Kaplan-Meier [22], Nelson-Aalen [23], [24], and Life-Table [25].

Semi-parametric models focus on modeling the instantaneous risk of experiencing the event over time. Instead of the survival function, these models predict the hazard function

$$h(t|\mathbf{x}) = \lim_{\Delta t \to 0} \frac{P(t \leq T < t + \Delta t | T \geq t, \mathbf{x})}{\Delta t}$$

or the cumulative hazard function $H(t|\mathbf{x}) = \int_0^t h(\tau|\mathbf{x})\, d\tau$. Nevertheless, the survival function of semi-parametric models can still be obtained as $S(t|\mathbf{x}) = e^{-H(t|\mathbf{x})}$. The hazard function helps semi-parametric models to decouple the time-varying risk $h_0(t)$ associated with the entire population, called baseline hazard, and the time-invariant risk $h(\mathbf{x})$ related to each subject, called risk function. The most notable example of a semi-parametric survival model is Cox Proportional Hazard (CoxPH) [26]. CoxPH estimates the hazard function as

$$h(t|\mathbf{x}) = h_0(t) e^{\langle \beta, \mathbf{x} \rangle},$$

where $h_0(t)$ is the baseline hazard and the risk function is $h(\mathbf{x}) = e^{\langle \beta, \mathbf{x} \rangle}$. $h_0(t)$ can be evaluated with the Nelson-Aalen estimator [23], [24]. The CoxPH model is based on the proportionality assumption, which states that the hazard ratio of two subjects is constant over time. DeepSurv [27] extends the CoxPH model allowing for non-linear dependencies between the input features and the output hazard. In DeepSurv, the risk function is $h(\mathbf{x}) = \phi_w(\mathbf{x})$, where ϕ_w is a single-output neural network with parameters w. The parameters of semi-parametric models can be optimized using the partial log-likelihood loss function, which can be evaluated using risk ratios only.

Parametric models assume that the event occurrence follows a parametric distribution. These models are often more expressive than semi-parametric models, as they are not limited by the proportional hazard assumption, which may not hold in most real-world scenarios. DeepHit [28] is a parametric model based on time discretization which models survival times directly using neural networks with sigmoid activations. N-MTLR [29] extends the Multi-Task Logistic Regression (MTLR) survival model [30] with a non-linear dependency between the input features and the predicted probabilities.

TABLE I
MACHINE LEARNING MODELS FOR SURVIVAL ANALYSIS.

Model	Linear	Proportional hazard	Differentiable	Continuous
CoxPH [26]	✓	✓	Partially	✓
DeepSurv [27]	×	✓	Partially	✓
DeepHit [28]	×	×	✓	×
N-MTLR [29]	×	×	✓	×
Nnet-Survival [31]	×	×	✓	×
PC-Hazard [32]	×	×	✓	✓
RSF [20]	×	×	×	✓

Nnet-Survival [31], [32], also known as Logistic Hazard, parametrizes discrete hazard functions with neural networks. PC-Hazard [32], [33] is based on a piecewise constant hazard function, so that the resulting survival function is a continuous piecewise exponential. PC-Hazard casts survival problems as Poisson regression problems, making regression models able to solve survival tasks. Random Survival Forest (RSF) [20] recursively builds a set of binary survival trees to estimate the cumulative hazard function. As in [34], binary trees are created with bootstrapping. At each node, the feature maximizing the hazard difference between child nodes is selected among a set of $d' \leq d$ features. The ensemble hazard function is the average of the hazard functions in the leaf nodes of the trees.

Table I summarizes the characteristics of the models described. Their comparison in federated settings is extensively discussed in Section IV.

B. Federated Learning

Federated Learning (FL) [5], [6] is a machine learning scenario in which a set of clients collaborate to train a model under the coordination of a central server. In FL, data never leave the device in which they are collected. A typical federated scenario is composed of K clients, each holding a private dataset $D_k, k = 1, \ldots, K$. Given a machine learning model with parameters w, the goal a FL algorithm is to minimize a global loss function $\mathcal{L}$ with respect to w, i.e.,

$$\min_w \mathcal{L}(w) = \min_w \sum_{k=1}^{K} \lambda_k \mathcal{L}_k(w).$$

Here, $\mathcal{L}_k$ is the loss function computed by client k using their private data D_k. λ_k is a set of parameters that weigh the contribution of each client to the global loss. Usually, the contribution is proportional to the number of samples $|D_k|$ in order to promote low-variance losses evaluated on more data.

The first algorithm proposed in the federated learning literature to optimize $\mathcal{L}(w)$ is Federated Averaging (FedAvg) [35]. FedAvg alternates a broadcast iteration and an aggregation iteration until convergence. During the broadcast iteration, the server selects subset of $K' \leq K$ clients and sends the current model parameters w. Then, these clients optimize the model on local data D_k and send the updated parameters back to the server. Usually, a small number of epochs suffices for local training. Finally, the server updates the global model parameters by averaging the updates received from the clients.

FedAvg performs consistently well in simulated environments, but real-world scenarios pose several challenges that hinder its generalization power and convergence speed. In particular, real-world scenarios present heterogeneous characteristics, both in terms of computational power and data distribution [5]. To this end, several works extend FedAvg to deal with federations presenting non-reliable communication channels and non-identically distributed data among clients [36]–[41].

To test the efficacy of federated algorithms in heterogeneous federations, common benchmarking practices have been developed. The most widely spread heterogeneous dataset collection for FL is LEAF [42]. This library includes several heterogeneous datasets for standard machine learning tasks, such as image classification and next-character prediction. SGDE [43] provides a framework to build synthetic datasets from privacy-preserving data generators trained on the clients of the federation. With SGDE, each generator embodies the characteristic features of each client, producing inherently heterogeneous datasets. Finally, other works [44], [45] study data splitting techniques based on the Dirichlet distribution. These techniques assign data samples from a non-federated classification dataset to each client of a federation with a controllable level of heterogeneity.

C. Federated Survival Analysis

Federated survival analysis refers to the integration of survival models into federated algorithms. Most works [3], [7]–[16] focus on the federated adaptation of the CoxPH model. In particular, the Cox partial log-likelihood loss function requires access to the entire dataset in order to be evaluated. This *non-separability* poses a significant challenge in federated applications. Alternative formulations to the standard partial log-likelihood have been proposed in some works [3], [16], relying on survival model discretization from [32]. Others have explored one-shot evaluation of a surrogate likelihood [8] or distributed adaptation of the Newton-Raphson optimization method [7], [10].

Moving beyond Cox models, in [46], the authors propose a secure Kaplan-Meier estimation procedure for federated genomics analysis. Other works [17]–[19] focus on general parametric models, including deep neural networks. In [19], for example, the survival problem is formulated as a regression problem using pseudo-values as target labels. In [17], on the other hand, discrete survival rates are evaluated with weakly-supervised attention modules.

Privacy is a crucial aspect of distributed healthcare applications, and several works adopt differential privacy [47] to protect survival models against inference attacks [17], [18]. In particular, the authors of [18] tackle the tradeoff between privacy level and model performance by adding a post-processing step to regulate the parameter update and improve the convergence of the differentially-private model. Other works [12], [48] rely on secure multiparty computation to prevent data leakage in a distributed network, while homomorphic encryption [46]

and bootstrapping with dimensionality reduction [15] are less commonly used approaches for privacy preservation.

Regarding the target application, several works focus on genomics analysis for cancer investigation [3], [17], [49], relying on the Cancer Genome Atlas (TCGA) project. FLamby [49] provides a suite of federated datasets for healthcare settings, including a survival dataset for breast cancer analysis called Fed-TCGA-BRCA. The Surveillance Epidemiology and End Results (SEER) database has also been used in several works [10], [13]. Other studies examine stroke detection [8], larynx cancer [11], and COVID-19 survival rates [14]. Finally, concerning performance evaluation and result comparability in federated survival settings, [50] provides several techniques to split existing survival datasets among the clients of a federation with a controllable level of heterogeneity.

To the best of our knowledge, FedSurF is the first ensemble learning technique for federated survival analysis, providing a parametric model that does not rely on the proportionality assumption while requiring only a single communication round to terminate.

III. Federated Survival Forests

In this section, we present the proposed method for adapting the Random Survival Forest (RSF) algorithm to the federated learning setting, referred to as Federated Survival Forest (FedSurF). Our approach is inspired by previous works in federated ensemble learning [51], [52], which involve building an ensemble model on the central server by merging base models from local ensembles on each client. Specifically, FedSurF involves building a RSF on the central server by aggregating the top-performing trees from local RSF models on each client. To this end, two techniques for tree selection are evaluated: a uniform probability selection method and a metric-based method for promoting the best-performing trees.

The FedSurF algorithm is structured in three stages: local training, tree assignment, and tree sampling. During the local training stage, each client k independently runs the RSF algorithm using their local dataset D_k, resulting in a local model M_k. Each M_k comprises an ensemble of survival trees $\{T_1, \ldots, T_{N_k}\}$. To optimize the performance of M_k specifically for D_k, the hyperparameters of each M_k can be fine-tuned through cross-validation using a train-validation split on the local dataset D_k. This allows for the selection of optimal values for the number of local trees N_k and the tree-building parameters (maximum tree depth, minimum samples per split, minimum samples per leaf, and maximum number of leaves).

Once each client has trained a local RSF model, the tree assignment stage involves the central server determining the number of trees that each client must transmit in order to achieve the desired total of N_S trees. To facilitate this process, the server must be notified of the number of trees N_k in each local model M_k. This message has a negligible impact on the communication channel with respect to messages containing model parameters, as it contains a single integer. Upon receiving the number of available trees N_k on each client, the server performs N_S iterations incrementing a tree counter $N'_k \leq N_k$ for a randomly selected client with probability proportional to the dataset cardinality $|D_k|$. This non-uniform probability promotes the inclusion of trees from clients with larger datasets, similar to the weighted contribution of FedAvg [35]. After the tree assignment stage, each client is required to send N'_k trees to the server, such that the total number of trees on the server is equal to N_S.

Algorithm 1: Federated Survival Forest (FedSurF)

Input: K clients, D_k, N_S, sampling strategy
Output: Ensemble of survival trees M_S
Initialize (server): $N'_k \leftarrow 0$
Initialize (client k): $M_k \leftarrow \emptyset$, $M'_k \leftarrow \emptyset$
for $k = 1$ to K in parallel **do**
  $M_k \leftarrow \text{RandomSurvivalForest}(D_k)$
  $N_k \leftarrow |M_k|$
end for
for $i = 1$ to N_S **do**
  $k \leftarrow \text{Sample}(1, \{n\}_{n=1}^{K}, \{|D_k|\}_{k=1}^{K}, \text{True})$
  $N'_k \leftarrow N'_k + 1$
end for
for $k = 1$ to K in parallel **do**
  if sampling strategy requires IBS **then**
    $M'_k \leftarrow \text{Sample}(N'_k, M_k, \{1/\text{IBS}_j\}_{j=1}^{N_k}, \text{False})$
  else
    $M'_k \leftarrow \text{Sample}(N'_k, M_k, \{1/N_k\}_{j=1}^{N_k}, \text{False})$
  end if
  Send M'_k to the server
end for
$M_S \leftarrow \bigcup_{k=1}^{K} M'_k$
return M_S

In the final stage of FedSurF, clients are required to select N'_k trees from their local models M_k to send to the central server. This selection process can be performed using either a uniform sampling strategy or a strategy that weights the selection probability of each tree based on a validation metric. To this end, we propose the use of the inverse of the Integrated Brier Score (IBS) as a validation metric. The IBS is a widely-used measure of the accuracy of survival models and is further discussed in Section IV-E. Clients can calculate the IBS score of each local tree, denoted as IBS_j, where $j = 1, \ldots, N_k$, and select N'_k trees with a probability proportional to $1/\mathit{IBS}_j$. This method is referred to as FedSurF-IBS. If clients instead choose to use a uniform sampling strategy, the method is simply referred to as FedSurF. Finally, the server constructs the ensemble model M_S by aggregating the trees received from each client k.

The pseudocode for FedSurF is presented in Algorithm 1. The function $\text{Sample}(N, S, P, R)$ is utilized to extract N elements from the set S, with each element being selected with probability proportional to P. If the value of R is set to True, sampling is performed with replacement.

TABLE II
SURVIVAL DATASETS INCLUDED IN THE EXPERIMENTS.

Dataset	Samples	Censored	Numerical features	Categorical features
GBSG2 [53]	686	44%	5	3
METABRIC [27]	1904	58%	5	3
AIDS [54]	2839	62%	1	3
FLCHAIN [55]	7874	28%	6	4
SUPPORT [56]	9105	68%	24	11

IV. EXPERIMENTS

This section presents a comprehensive set of experiments to empirically demonstrate the efficacy of FedSurF in solving survival problems with a high generalization power and resilience to non-identically distributed data. The source code is available at https://github.com/archettialberto/federated_survival_forests.

A. Datasets

In this study, we conduct experiments on five survival datasets: the German Breast Cancer Study Group 2 (GBSG2) [53], the Molecular Taxonomy of Breast Cancer International Consortium (METABRIC) [27], the Australian AIDS survival dataset (AIDS) [54], the assay of serum-free light chain dataset (FLCHAIN) [55], and the Study to Understand Prognoses Preferences Outcomes and Risks of Treatment (SUPPORT) [56]. The relevant summary statistics of these datasets are presented in Table II.

B. Techniques for Federated Simulation

In the course of our experiments, we begin by partitioning a non-federated survival dataset D into a training dataset and a test dataset, where the former constitutes 80% of the total samples and the latter constitutes 20% of the total samples. The test dataset is not further modified and is used exclusively for the final evaluation of each model. Subsequently, we assign each sample in the training dataset to one of the clients in a federation. Each federation is composed of 10 clients ($K = 10$). To determine the client to which each sample is assigned, we employ two techniques. The first technique assigns each sample in the training dataset to one of the available clients with equal probability. This technique results in uniform data distributions among the federation clients, facilitating convergence for federated algorithms. The first row of Figure 1 illustrates the Kaplan-Meier estimators $\hat{S}_k(t)$ of the client data under the uniform assignment technique.

The second splitting technique follows the label-skewed splitting algorithm described in [50]. In this case, data samples are assigned to the federation clients to make the resulting label distributions heterogeneous. This technique is based on the Dirichlet distribution, inspired by similar practices employed in federated datasets for classification [44], [45]. The goal is to simulate realistic federated environments in which data distributions are non-uniform. To this end, we apply the label-skewed splitting algorithm from [50] with $K = 10$ and $\alpha = 8.0$, imposing a minimum number of 25 samples per dataset. The second row of Figure 1 shows the Kaplan-Meier estimators $\hat{S}_k(t)$ for each client dataset D_k under label-skewed splitting.

On top of the data assignment procedures, an 80%-20% partition is applied to each client dataset in order to extract a local validation split for parameter tuning and early stopping. Finally, each client is assumed to be permanently available and the communication channel is assumed to be fully reliable.

C. Baseline Models

In the experimental evaluation, we test the performance of the proposed FedSurF and FedSurF-IBS algorithms with respect to the state-of-the-art survival models from Section II-A: Cox Proportional Hazard (CoxPH) [26], DeepSurv [27], DeepHit [28], Neural Multi-Task Logistic Regression (N-MTLR) [29], Nnet-Survival [31], and Piecewise-Constant Hazard (PC-Hazard) [32]. Hyperparameters for FedSurF and FedSurF-IBS are optimized using cross-validation. For CoxPH, the number of parameters is equal to the number of features in the input space. The neural network architectures of DeepSurv, DeepHit, N-MTLR, Nnet-Survival, and PC-Hazard are composed of two dense layers with 32 neurons each, followed by a ReLU activation function and a Dropout layer to prevent overfitting. The last dense layer of DeepSurv produces a scalar output compliant with the proportional hazard assumption while the other models had 10 outputs, one for each of the discretization instants.

D. Training

In the experiments, we trained each model both in a local setting and in a federated setting. In the local setting, clients train and validate their models on local data only. Local evaluation is useful to understand whether joining a federated learning algorithm is beneficial for a client.

The implementation of RSF is based on the scikit-survival Python library [57], while the other differentiable models rely on PyCox [58]. RSF hyperparameter tuning is conducted with cross-validation. Differentiable models are trained using the Adam optimizer with a learning rate of 0.01 and early stopping for 500 epochs at most. For FedSurF and FedSurF-IBS, federated training is performed according to the algorithm outlined in Section III. Conversely, FedAvg [35] is applied to all the other models. Concerning proportional hazard models (CoxPH and DeepSurv), we assume each client has access to the same global Nelson-Aalen estimation of the baseline hazard. To address the issue of convergence in heterogeneous federations, we ran FedAvg for 500 rounds with a single local epoch per round and selected the parameters with the best validation metrics for the final evaluation on the test set.

E. Evaluation

In our evaluation, we utilized the Concordance Index (C-Index) and the Integrated Brier Score (IBS) to assess the performance of our models on the test set. The C-Index [59] measures the proportion of comparable samples for which the

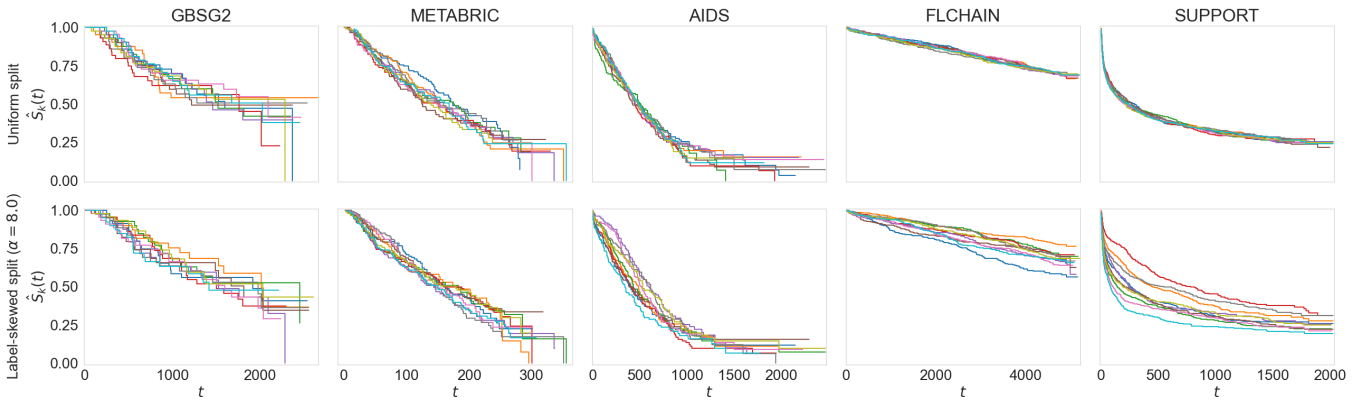

Fig. 1. Kaplan-Meier estimators $\hat{S}_k(t)$ for each client dataset D_k under uniform data assignment (first row) and label-skewed assignment [50] with $\alpha = 8.0$ (second row). Each line corresponds to a single client k.

prediction and true outcome are in agreement. A sample pair is considered comparable if at least one of the two samples is not censored. The C-Index reflects the discriminatory power of the model, indicating the fraction of samples that the model has ordered correctly according to actual time outcomes. A random guessing model would have a C-Index of 0.5, while a model with perfect knowledge would have a C-Index of 1.0.

The Brier score [21] is a calibration metric that measures the squared difference between the true survival status and the predicted survival probability of the model at a given time. The IBS integrates the Brier score over time to condense the calibration level of a model into a single value. A random guessing model would have an IBS of 0.25, while a model with perfect knowledge would have an IBS of 0.0. Thus, the lower the IBS, the better the model.

To account for the censoring distribution, we applied Inverse Probability of Censoring Weighting (IPCW) [59], [60] during metrics estimation to ensure unbiased results.

F. Results

The performance of the compared survival models is summarized in Table III and Table IV for federations built with uniform data splitting, and in Table V and Table VI for federations built with label-skewed splitting. In the following, we provide a detailed analysis of the results obtained.

Is it beneficial for a client to join a federated learning procedure? All the results agree that on average the best performance is consistently achieved under the *Federated* columns. This indicates that, for a given survival model, it is generally advantageous to join a federation in terms of both discrimination and calibration.

Is FedSurF performing better than the alternatives when data is uniformly distributed? To answer this question, we refer to Table III and Table IV. From the results, it can be seen that FedSurF does not consistently outperform other state-of-the-art models. However, its performance is comparable to that of the other models in terms of both discrimination and calibration. Upon closer examination, we observe that FedSurF performs particularly well for smaller datasets (GBSG2 and METABRIC), while for larger datasets (AIDS, FLCHAIN, and SUPPORT), the results are comparable to those of deep learning models. This suggests that the model choice between ensemble-based and neural-based is not particularly relevant when data is uniformly distributed. However, we highlight that FedSurF and FedSurF-IBS obtain the recorded performance in just a single communication round, which may be a relevant advantage in federations with communication channel constraints.

Is IBS sampling relevant when data is uniformly distributed? When data is uniformly distributed FedSurF-IBS does not consistently outperform FedSurF. This can be seen by comparing the results in Table III and Table IV. This finding suggests that the use of IBS sampling is not necessary for achieving optimal performance in scenarios with identically distributed data. This outcome is expected, as IBS sampling is particularly useful to select the best trees from the best clients. In the case of IID data, where the local data distribution is roughly the same, the quality of the trees generated by each client should be similar. Under these conditions, IBS does not provide an advantage for achieving optimal performance.

Is FedSurF performing better than the alternatives when data is heterogeneous? To comment on this, we refer to Table V and Table VI. The results show that FedSurF-IBS obtains the best or very close to the best metrics on all datasets except for SUPPORT. Specifically, it is the model with the best C-Index on GBSG2, METABRIC, and AIDS. Additionally, the IBS is also very low and within the standard deviation of the slightly better models. From these results, it can be concluded that FedSurF with IBS sampling is a well-performing algorithm in heterogeneous federations when considering the C-Index and IBS metrics. It demonstrates a robust performance across different label-skewed datasets, with a consistent improvement over the other models.

Is IBS sampling relevant when data is heterogeneous? The results for the heterogeneous datasets, as presented in Table V and Table VI, show that for all datasets except for

TABLE III

C-INDEX FOR SURVIVAL MODELS EVALUATED ON UNIFORMLY SPLIT DATASETS. EACH METRIC IS SCALED BY A FACTOR OF 100 FOR READABILITY. WE REPORT THE MEAN AND STANDARD DEVIATION OVER 5 RUNS. THE BEST RESULTS (↑) ARE HIGHLIGHTED IN BOLD.

Dataset	GBSG2		METABRIC		AIDS		FLCHAIN		SUPPORT	
Model	*Local*	*Federated*	*Local*	*Federated*	*Local*	*Federated*	*Local*	*Federated*	*Local*	*Federated*
CoxPH	61.2±4.1	68.7±8.5	58.8±1.9	62.0±1.7	51.9±0.7	53.8±1.6	92.3±1.6	93.4±0.4	79.9±0.2	82.5±0.3
DeepSurv	60.7±4.0	69.3±7.5	59.7±1.7	**63.0±2.7**	52.5±0.2	53.7±1.3	93.5±0.4	94.2±0.2	79.9±0.4	83.1±0.4
DeepHit	59.8±5.3	68.5±7.5	57.3±1.3	61.1±3.0	52.6±0.5	53.8±1.6	93.1±0.3	93.8±0.4	81.7±0.4	83.5±0.6
N-MTLR	61.5±4.4	69.2±7.1	58.3±1.6	62.2±2.5	53.1±0.8	**54.2±1.6**	93.6±0.3	**94.2±0.3**	80.6±0.4	83.0±0.6
Nnet-Survival	60.5±7.9	66.7±7.7	49.4±3.9	60.4±4.1	50.8±1.2	54.0±1.6	90.7±1.3	94.1±0.4	80.9±0.3	**83.8±0.6**
PC-Hazard	60.3±8.2	67.3±7.4	50.2±3.9	59.8±3.8	51.0±1.3	53.3±1.8	89.8±0.8	94.1±0.3	80.7±0.3	81.9±1.2
FedSurF	66.5±5.7	**72.3±4.7**	59.9±2.1	62.2±1.9	52.4±0.6	54.1±1.3	93.4±0.3	93.5±0.3	80.4±0.4	81.0±0.4
FedSurF-IBS	–	71.8±5.4	–	62.4±2.0	–	53.8±1.3	–	93.5±0.3	–	80.7±0.6

TABLE IV

IBS FOR SURVIVAL MODELS EVALUATED ON UNIFORMLY SPLIT DATASETS. EACH METRIC IS SCALED BY A FACTOR OF 100 FOR READABILITY. WE REPORT THE MEAN AND STANDARD DEVIATION OVER 5 RUNS. THE BEST RESULTS (↓) ARE HIGHLIGHTED IN BOLD.

Dataset	GBSG2		METABRIC		AIDS		FLCHAIN		SUPPORT	
Model	*Local*	*Federated*	*Local*	*Federated*	*Local*	*Federated*	*Local*	*Federated*	*Local*	*Federated*
CoxPH	20.2±1.1	18.5±1.6	17.9±1.9	17.4±2.3	15.1±1.5	14.7±1.5	5.6±0.4	5.4±0.2	13.3±0.2	12.7±0.1
DeepSurv	21.2±1.2	**17.9±1.2**	17.8±1.5	**16.6±1.5**	15.2±1.2	14.7±1.5	5.6±0.6	4.2±0.1	13.9±0.2	12.2±0.2
DeepHit	21.8±2.0	19.5±1.1	18.7±1.7	17.6±1.6	16.0±1.4	14.8±1.5	5.8±0.2	4.5±0.1	13.7±0.6	12.7±0.4
N-MTLR	21.3±1.6	18.3±1.0	18.1±1.5	17.1±1.7	14.9±1.4	14.7±1.4	4.8±0.1	**4.2±0.1**	14.5±0.3	**11.9±0.4**
Nnet-Survival	23.0±2.1	19.5±1.0	23.9±2.7	17.7±1.6	15.4±1.5	15.0±1.5	6.5±0.4	4.5±0.2	12.9±0.3	12.3±0.4
PC-Hazard	22.4±1.9	19.6±0.8	23.7±2.4	18.0±1.6	15.4±1.6	15.1±1.5	6.8±0.4	4.5±0.2	13.0±0.3	12.4±0.4
FedSurF	19.3±1.2	18.5±1.2	17.6±1.3	16.8±1.2	14.8±1.5	14.7±1.5	4.6±0.1	4.5±0.1	18.2±0.5	18.1±0.5
FedSurF-IBS	–	18.4±1.2	–	16.7±1.3	–	**14.7±1.5**	–	4.4±0.1	–	15.9±0.5

TABLE V

C-INDEX FOR SURVIVAL MODELS EVALUATED ON LABEL-SKEWED DATASETS. EACH METRIC IS SCALED BY A FACTOR OF 100 FOR READABILITY. WE REPORT THE MEAN AND STANDARD DEVIATION OVER 5 RUNS. THE BEST RESULTS (↑) ARE HIGHLIGHTED IN BOLD.

Dataset	GBSG2		METABRIC		AIDS		FLCHAIN		SUPPORT	
Model	*Local*	*Federated*	*Local*	*Federated*	*Local*	*Federated*	*Local*	*Federated*	*Local*	*Federated*
CoxPH	60.3±2.8	68.4±7.8	57.3±0.9	60.6±3.8	52.2±1.4	54.2±1.1	91.5±2.8	77.6±22.5	80.0±0.2	82.3±0.3
DeepSurv	59.8±2.2	67.5±8.5	58.7±1.8	62.1±3.7	53.0±0.6	54.5±1.6	92.3±3.1	81.9±18.6	80.0±0.5	80.7±2.2
DeepHit	60.9±5.3	68.5±7.0	56.2±1.0	60.4±2.9	53.2±1.1	53.9±0.9	93.2±0.3	93.7±0.1	81.0±0.2	82.9±0.6
N-MTLR	61.8±8.9	69.3±7.6	59.0±1.9	61.7±3.3	53.0±0.7	53.6±2.1	93.6±0.2	**94.0±0.2**	80.3±0.3	81.1±0.4
Nnet-Survival	59.5±8.5	68.2±7.7	49.1±3.6	59.5±3.7	51.3±1.5	54.2±2.3	90.6±1.2	93.9±0.3	80.7±0.5	**83.5±0.7**
PC-Hazard	60.9±8.0	68.9±7.3	49.4±3.8	59.2±3.6	50.9±0.6	53.6±2.6	89.9±1.1	93.9±0.3	80.6±0.5	80.6±0.8
FedSurF	66.8±5.4	71.9±5.0	59.9±1.4	61.8±2.9	53.2±0.8	55.0±0.9	93.4±0.3	93.5±0.3	80.3±0.4	81.3±0.4
FedSurF-IBS	–	**72.4±5.4**	–	**62.1±2.4**	–	**55.1±0.6**	–	93.5±0.2	–	80.6±0.7

TABLE VI

IBS FOR SURVIVAL MODELS EVALUATED ON LABEL-SKEWED DATASETS. EACH METRIC IS SCALED BY A FACTOR OF 100 FOR READABILITY. WE REPORT THE MEAN AND STANDARD DEVIATION OVER 5 RUNS. THE BEST RESULTS (↓) ARE HIGHLIGHTED IN BOLD.

Dataset	GBSG2		METABRIC		AIDS		FLCHAIN		SUPPORT	
Model	*Local*	*Federated*	*Local*	*Federated*	*Local*	*Federated*	*Local*	*Federated*	*Local*	*Federated*
CoxPH	20.1±1.2	18.5±1.4	18.3±1.6	17.3±2.3	15.1±1.5	14.7±1.5	5.7±0.6	8.5±4.0	13.5±0.2	12.8±0.2
DeepSurv	20.6±1.5	**18.2±1.0**	18.7±1.6	**16.8±1.7**	15.2±1.1	14.7±1.5	5.5±0.6	9.0±4.4	14.0±0.3	12.6±0.5
DeepHit	21.7±2.0	19.2±0.7	18.6±1.7	17.7±1.5	16.0±1.3	14.7±1.5	5.7±0.2	4.5±0.1	14.4±0.6	12.8±0.4
N-MTLR	20.8±0.9	18.7±1.5	18.2±1.7	17.4±1.9	15.0±1.4	14.7±1.4	4.8±0.1	4.5±0.2	14.7±0.3	13.6±0.7
Nnet-Survival	23.6±2.5	19.1±1.5	23.9±2.4	17.9±1.4	15.4±1.6	14.9±1.6	6.6±0.5	4.5±0.2	13.0±0.4	**12.6±0.6**
PC-Hazard	22.6±2.1	19.0±1.5	23.9±2.5	18.2±1.6	15.4±1.5	15.0±1.6	6.8±0.4	4.5±0.2	13.1±0.4	12.7±0.5
FedSurF	19.3±1.2	18.5±1.3	17.8±1.2	16.9±1.3	14.9±1.5	14.7±1.5	4.7±0.1	4.5±0.1	18.5±0.6	18.0±0.6
FedSurF-IBS	–	18.4±1.1	–	16.9±1.2	–	**14.7±1.5**	–	**4.4±0.1**	–	15.8±0.5

SUPPORT, FedSurF-IBS outperforms FedSurF. This indicates that, in contrast to federations with uniformly split data, IBS sampling is relevant and beneficial in federations where data is heterogeneous. Indeed, it allows to select the best trees from the best clients and improve the overall performance of the ensemble model. This highlights the importance of considering the characteristics of the data distribution when choosing the appropriate sampling method for FedSurF. Concerning the results of the SUPPORT dataset, despite an extensive hyper-parameter search, survival forests exhibit a higher Brier Score than neural models in 10-client federations. We impute the lower IBS to the tree-based nature of the models, which may model the specific survival rates of SUPPORT suboptimally with respect to neural-based methods. This claim is confirmed by the *Local* results, which highlight how, on average, tree-based models have lower performance than neural models, even without considering federated learning. Nevertheless, applying IBS sampling in the *Federated* experiments lowers the IBS by more than 0.02, yielding a significant improvement over uniform sampling.

V. Conclusion

This work presents Federated Survival Forest (FedSurF), an extension of the Random Survival Forest (RSF) algorithm to the federated learning setting. Our proposed algorithm is based on sampling the best local trees from the best clients in the federation to build a RSF on the server. We demonstrate the effectiveness of FedSurF through extensive experiments in two different federated settings: one with uniformly split data and another with heterogeneous data. Our results indicate that FedSurF performs comparably well to state-of-the-art models when data is uniformly distributed while obtaining a noticeable advantage in heterogeneous federations. Furthermore, FedSurF relies on a single communication round between clients and server, rather than multiple iterative updates, making it an efficient and practical solution in federations with strict bandwidth requirements.

Acknowledgment

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

References

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

[2] P. Wang, Y. Li, and C. K. Reddy, "Machine learning for survival analysis: A survey," *ACM Computing Surveys (CSUR)*, vol. 51, no. 6, pp. 1–36, 2019.

[3] M. Andreux, A. Manoel, R. Menuet, C. Saillard, and C. Simpson, "Federated survival analysis with discrete-time cox models," *arXiv preprint arXiv:2006.08997*, 2020.

[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*, vol. 3, no. 1, pp. 1–7, 2020.

[5] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, 2020.

[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*, vol. 14, no. 1–2, pp. 1–210, 2021.

[7] C.-L. Lu, S. Wang, Z. Ji, Y. Wu, L. Xiong, X. Jiang, and L. Ohno-Machado, "Webdisco: a web service for distributed cox model learning without patient-level data sharing," *Journal of the American Medical Informatics Association*, vol. 22, no. 6, pp. 1212–1219, 2015.

[8] 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, and Y. Chen, "Learning from local to global: An efficient distributed algorithm for modeling time-to-event data," *Journal of the American Medical Informatics Association*, vol. 27, no. 7, pp. 1028–1036, 07 2020. [Online]. Available: https://doi.org/10.1093/jamia/ocaa044

[9] S. Banerjee, G. N. Sofack, T. Papakonstantinou, D. Avraam, P. Burton, D. Zöller, and T. R. P. Bishop, "dsSurvival: Privacy preserving survival models for federated individual patient meta-analysis in DataSHIELD," *BMC Research Notes*, vol. 15, no. 1, p. 197, Dec. 2022. [Online]. Available: https://bmcresnotes.biomedcentral.com/articles/10.1186/s13104-022-06085-1

[10] W. Dai, X. Jiang, L. Bonomi, Y. Li, H. Xiong, and L. Ohno-Machado, "VERTICOX: Vertically Distributed Cox Proportional Hazards Model Using the Alternating Direction Method of Multipliers," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 2, pp. 996–1010, Feb. 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9076318/

[11] C. R. Hansen, G. Price, M. Field, N. Sarup, R. Zukauskaite, J. Johansen, J. G. Eriksen, F. Aly, A. McPartlin, L. Holloway, D. Thwaites, and C. Brink, "Larynx cancer survival model developed through open-source federated learning," *Radiotherapy and Oncology*, vol. 176, pp. 179–186, Nov. 2022. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0167814022044930

[12] B. Kamphorst, T. Rooijakkers, T. Veugen, M. Cellamare, and D. Knoors, "Accurate training of the Cox proportional hazards model on vertically-partitioned data while preserving privacy," *BMC Medical Informatics and Decision Making*, vol. 22, no. 1, p. 49, Dec. 2022. [Online]. Available: https://bmcmedinformdecismak.biomedcentral.com/articles/10.1186/s12911-022-01771-3

[13] C. Masciocchi, B. Gottardelli, M. Savino, L. Boldrini, A. Martino, C. Mazzarella, M. Massaccesi, V. Valentini, and 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)*. Shenzen, China: IEEE, Jul. 2022, pp. 25–31. [Online]. Available: https://ieeexplore.ieee.org/document/9867090/

[14] 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*, vol. 134, p. 104176, 2022.

[15] A. Imakura, R. Tsunoda, R. Kagawa, K. Yamagata, and T. Sakurai, "DC-COX: Data collaboration Cox proportional hazards model for privacy-preserving survival analysis on multiple parties," *Journal of Biomedical Informatics*, vol. 137, p. 104264, Jan. 2023. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1532046422002696

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

[17] M. Y. Lu, R. J. Chen, D. Kong, J. Lipkova, R. Singh, D. F. Williamson, T. Y. Chen, and F. Mahmood, "Federated learning for computational pathology on gigapixel whole slide images," *Medical Image Analysis*, vol. 76, p. 102298, Feb. 2022. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1361841521003431

[18] S. Rahimian, R. Kerkouche, I. Kurth, and 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.

[19] M. M. Rahman and S. Purushotham, "Fedpseudo: Pseudo value-based deep learning models for federated survival analysis," *arXiv preprint arXiv:2207.05247*, 2022.

[20] H. Ishwaran, U. B. Kogalur, E. H. Blackstone, and M. S. Lauer, "Random survival forests," *The annals of applied statistics*, vol. 2, no. 3, pp. 841–860, 2008.

[21] E. Graf, C. Schmoor, W. Sauerbrei, and M. Schumacher, "Assessment and comparison of prognostic classification schemes for survival data," *Statistics in medicine*, vol. 18, no. 17-18, pp. 2529–2545, 1999.

[22] E. L. Kaplan and P. Meier, "Nonparametric estimation from incomplete observations," *Journal of the American statistical association*, vol. 53, no. 282, pp. 457–481, 1958.

[23] W. Nelson, "Theory and applications of hazard plotting for censored failure data," *Technometrics*, vol. 14, no. 4, pp. 945–966, 1972.

[24] O. Aalen, "Nonparametric inference for a family of counting processes," *The Annals of Statistics*, pp. 701–726, 1978.

[25] S. J. Cutler and F. Ederer, "Maximum utilization of the life table method in analyzing survival," *Journal of chronic diseases*, vol. 8, no. 6, pp. 699–712, 1958.

[26] D. R. Cox, "Regression models and life-tables," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 34, no. 2, pp. 187–220, 1972. [Online]. Available: http://www.jstor.org/stable/2985181

[27] J. L. Katzman, U. Shaham, A. Cloninger, J. Bates, T. Jiang, and Y. Kluger, "Deepsurv: personalized treatment recommender system using a cox proportional hazards deep neural network," *BMC medical research methodology*, vol. 18, no. 1, pp. 1–12, 2018.

[28] C. Lee, W. Zame, J. Yoon, and 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, no. 1, 2018.

[29] S. Fotso, "Deep neural networks for survival analysis based on a multi-task framework," *arXiv preprint arXiv:1801.05512*, 2018.

[30] C.-N. Yu, R. Greiner, H.-C. Lin, and V. Baracos, "Learning patient-specific cancer survival distributions as a sequence of dependent regressors," *Advances in neural information processing systems*, vol. 24, 2011.

[31] M. F. Gensheimer and B. Narasimhan, "A scalable discrete-time survival model for neural networks," *PeerJ*, vol. 7, p. e6257, 2019.

[32] H. Kvamme and Ø. Borgan, "Continuous and discrete-time survival prediction with neural networks," *Lifetime Data Analysis*, vol. 27, no. 4, pp. 710–736, 2021.

[33] A. Bender, D. Rügamer, F. Scheipl, and 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.

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

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

[36] S. Reddi, Z. Charles, M. Zaheer, Z. Garrett, K. Rush, J. Konečnỳ, S. Kumar, and H. B. McMahan, "Adaptive federated optimization," *arXiv preprint arXiv:2003.00295*, 2020.

[37] 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*, 2021.

[38] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Proceedings of Machine Learning and Systems*, vol. 2, pp. 429–450, 2020.

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

[40] D. A. E. Acar, Y. Zhao, R. M. Navarro, M. Mattina, P. N. Whatmough, and V. Saligrama, "Federated learning based on dynamic regularization," *arXiv preprint arXiv:2111.04263*, 2021.

[41] D. Caldarola, B. Caputo, and M. Ciccone, "Improving generalization in federated learning by seeking flat minima," in *European Conference on Computer Vision*. Springer, 2022, pp. 654–672.

[42] S. Caldas, S. M. K. Duddu, P. Wu, T. Li, J. Konečnỳ, H. B. McMahan, V. Smith, and A. Talwalkar, "Leaf: A benchmark for federated settings," *arXiv preprint arXiv:1812.01097*, 2018.

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

[44] T.-M. H. Hsu, H. Qi, and M. Brown, "Measuring the effects of non-identical data distribution for federated visual classification," *arXiv preprint arXiv:1909.06335*, 2019.

[45] Q. Li, Y. Diao, Q. Chen, and 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.

[46] D. Froelicher, J. R. Troncoso-Pastoriza, J. L. Raisaro, M. A. Cuendet, J. S. Sousa, H. Cho, B. Berger, J. Fellay, and J.-P. Hubaux, "Truly privacy-preserving federated analytics for precision medicine with multiparty homomorphic encryption," *Nature Communications*, vol. 12, no. 1, p. 5910, Oct. 2021. [Online]. Available: https://www.nature.com/articles/s41467-021-25972-y

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

[48] T. Marchand, B. Muzellec, C. Beguier, J. O. d. Terrail, and M. Andreux, "SecureFedYJ: a safe feature Gaussianization protocol for Federated Learning," Oct. 2022, arXiv:2210.01639 [cs]. [Online]. Available: http://arxiv.org/abs/2210.01639

[49] J. O. d. 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, and M. Andreux, "FLamby: Datasets and Benchmarks for Cross-Silo Federated Learning in Realistic Healthcare Settings," Oct. 2022, arXiv:2210.04620 [cs]. [Online]. Available: http://arxiv.org/abs/2210.04620

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

[51] A.-C. Hauschild, M. Lemanczyk, J. Matschinske, T. Frisch, O. Zolotareva, A. Holzinger, J. Baumbach, and D. Heider, "Federated random forests can improve local performance of predictive models for various healthcare applications," *Bioinformatics*, vol. 38, no. 8, pp. 2278–2286, 2022.

[52] M. Gencturk, A. A. Sinaci, and N. K. Cicekli, "Bofrf: A novel boosting-based federated random forest algorithm on horizontally partitioned data," *IEEE Access*, vol. 10, pp. 89 835–89 851, 2022.

[53] M. Schumacher, G. Bastert, H. Bojar, K. Hübner, M. Olschewski, W. Sauerbrei, C. Schmoor, C. Beyerle, R. Neumann, and 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*, vol. 12, no. 10, pp. 2086–2093, 1994.

[54] B. Ripley, B. Venables, D. M. Bates, K. Hornik, A. Gebhardt, and D. Firth, "R package: Mass," Jul. 27, 2022. [Online]. Available: https://stat.ethz.ch/R-manual/R-devel/library/MASS/html/00Index.html

[55] T. Therneau, T. Lumley, E. Atkinson, and C. Crowson, "R package: survival," Jan. 9, 2023. [Online]. Available: https://stat.ethz.ch/R-manual/R-devel/library/survival/html/00Index.html

[56] Vanderbilt University Department of Biostatistics, "Vanderbilt biostatistics datasets," Dec. 1, 2022. [Online]. Available: http://hbiostat.org/data

[57] S. Pölsterl, "scikit-survival: A library for time-to-event analysis built on top of scikit-learn," *Journal of Machine Learning Research*, vol. 21, no. 212, pp. 1–6, 2020. [Online]. Available: http://jmlr.org/papers/v21/20-729.html

[58] H. Kvamme, Ø. Borgan, and I. Scheel, "Time-to-event prediction with neural networks and cox regression," *arXiv preprint arXiv:1907.00825*, 2019.

[59] H. Uno, T. Cai, M. J. Pencina, R. B. D'Agostino, and L.-J. Wei, "On the c-statistics for evaluating overall adequacy of risk prediction procedures with censored survival data," *Statistics in medicine*, vol. 30, no. 10, pp. 1105–1117, 2011.

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