Computers in Biology and Medicine 172 (2024) 108288

Contents lists available at ScienceDirect

Computers in Biology and Medicine

journal homepage: www.elsevier.com/locate/compbiomed

SYNDSURV: A simple framework for survival analysis with data distributed across multiple institutions

Cesare Rollo, Corrado Pancotti, Giovanni Birolo, Ivan Rossi, Tiziana Sanavia *, Piero Fariselli

University of Torino, Via Santena 19, Torino, 10126, Italy

ARTICLE INFO

Dataset link: https://github.com/compbiomed-unito/SYNDSURV

Keywords:
Survival analysis
Synthetic data
Federated learning
Differential privacy

ABSTRACT

Data sharing among different institutions represents one of the major challenges in developing distributed machine learning approaches, especially when data is sensitive, such as in medical applications. Federated learning is a possible solution, but requires fast communications and flawless security. Here, we propose SYNDSURV (SYNthetic Distributed SURVival), an alternative approach that simplifies the current state-of-the-art paradigm by allowing different centres to generate local simulated instances from real data and then gather them into a centralised hub, where an Artificial Intelligence (AI) model can learn in a standard way. The main advantage of this procedure is that it is model-agnostic, therefore prediction models can be directly applied in distributed applications without requiring particular adaptations as the current federated approaches do. To show the validity of our approach for medical applications, we tested it on a survival analysis task, offering a viable alternative to train AI models on distributed data. While federated learning has been mainly optimised for gradient-based approaches so far, our framework works with any predictive method, proving to be a comparable way of performing distributed learning without being too demanding towards each participating institute in terms of infrastructural requirements.

1. Introduction

The past few decades have witnessed a massive and growing increase in well-structured clinical data on a global scale [1,2]. The development of increasingly sophisticated Artificial Intelligence (AI) techniques capable of fully exploiting large, heterogeneous, and complex data will be of crucial importance for healthcare systems in the coming years [3,4], and Federated Learning (FL) initiatives are being developed in healthcare to collaboratively train predictive models without the need to centralise sensitive personal data [5–7]. However, a main constraint in using clinical data for predictive purposes is that, despite the relative global abundance, they are limited in individual centres, especially for rare diseases. Furthermore, sharing and aggregating local datasets is often impossible due to their strict privacy regulations. Therefore, it is essential to analyse the available clinical data in a manner that adheres to privacy regulations while still preserving high accuracy in the predictions. One of the main applications in healthcare is to predict when specific adverse events of interest to the patient occur. Survival analysis models address this task and several machine learning-based techniques have been proposed for prognosis predictions. However, making distributed versions of survival models is more complex than others for classification and regression models, since it requires ranking evaluations and dealing with "censored" instances. These instances are partially missing temporal data, which are common in real-world clinical scenarios, where patients can only sometimes be subject to an arbitrarily long follow-up period.

This work addresses data sharing challenges in critical contexts by proposing a distributed model tailored for survival analysis. The proposed framework generates simulated survival data based on real local datasets and aggregates them into a centralised hub for training. The primary aims for this work are: (1) to develop accurate predictive models that can improve patient outcomes and advance medical research without compromising data security; (2) to overcome current limitations of federated approaches by proposing a simple model-agnostic data sharing framework which can be easily implemented by the participating institutes.

1.1. Main contribution

This work shows that we can exploit a more feasible alternative to FL in the survival analysis context. To prove that, we applied our idea to the challenging task of time-to-event prediction in a privacy-preserving way. Specifically:

* Corresponding author.
E-mail addresses: cesare.rollo@unito.it (C. Rollo), corrado.pancotti@unito.it (C. Pancotti), giovanni.birolo@unito.it (G. Birolo), ivan.rossi@unito.it (I. Rossi), tiziana.sanavia@unito.it (T. Sanavia), piero.fariselli@unito.it (P. Fariselli).

https://doi.org/10.1016/j.compbiomed.2024.108288
Received 2 January 2024; Received in revised form 17 February 2024; Accepted 12 March 2024
Available online 15 March 2024

- We have developed a proof-of-concept scenario that allows us to build a global predictive model taking advantage of the aggregated information, and without using combinations of local losses.
- This approach does not require fast communication methods among the nodes, which is an issue for current federated approaches.
- Our framework is model-agnostic and not limited to gradient-based algorithms, making it a versatile solution since it avoids distributed optimisations.
- We have considered privacy and developed a way to enhance it by adding a controlled amount of noise to the synthetic samples in a Differential Privacy setting.
- We have shown that the performance for Survival Analysis is very close to what is attainable using real data in a centralised approach.

1.2. Related works

The significant efforts to protect patients' privacy through limitations on sharing sensitive clinical data can pose a challenge for researching rare diseases or developing effective machine learning-based predictive tools. Training data distributed across multiple institutions could enhance model performance and generalisation. Recently, this hurdle has led to the development of novel distributed learning approaches, where patients' data are not directly shared among the institutions [8–11]. In particular, FL, since its first introduction in 2017 [12], has been one of the most studied and promising methods to overcome limitations related to data privacy [13,14].

Among the different supervised learning tasks, Survival Analysis is intrinsically more complex to distribute compared to classification and regression models. Indeed, optimising Cox-like loss functions requires the knowledge of all individuals' sorted times, requiring a ranking evaluation (see Eq. (5)). This implies that optimising the function in "batches" (or per centre) may lead to a different solution, which may not be optimal compared to the one obtained using the loss function computed on the complete and centralised data.

Recent studies faced the problem of creating distributed survival models by building "surrogate" likelihood functions [15] or aggregating different CoxPH models trained in separated nodes [16,17]. In [18], a classical federation was built on a novel reformulation of the survival task in terms of pseudovalues, while in [19] the authors proposed a discrete-time extension of the CoxPH model that makes the loss separable and turns the survival task into a multiple classification problem. However, most of these methods (except [18]) are specifically tailored either to the linear CoxPH framework, which is additive in the input features, or more generally to the CoxPH loss function and its underlying assumptions. Thus, they cannot be easily extended to other more comprehensive and non-linear algorithms.

The field of synthetic data generation has recently gained much attention as a viable strategy for sharing information among different data holders that nevertheless avoids the dissemination of real patient data [20–23]. However, the generation of synthetic data for Survival Analysis tasks is not straightforward due to the presence of censored times, as outlined by Norcliffe et al. [24], who proposed a generative adversarial model tailored for survival predictions.

Finally, SYNDSURV idea has similarities with dataset distillation and one-shot FL techniques [25–27], since all these methods try to solve some FL issues by exploiting more local computational efforts rather than optimising the communication of algorithms.

2. Background

2.1. Federated learning

In a FL setting, data are stored in separated data centres and the goal is to train a global predictive model by sharing only its parameters with the centres. Specifically, each participating institution separately updates the parameters by performing a fixed number of training iterations using only its own data and then sends the updates back to a central server, which computes a weighted aggregation. This procedure is repeated for N communication rounds until convergence. Although this method has proved to be effective in several predictive tasks, FL still has some practical and theoretical limitations. Most FL existing strategies are optimised for standard gradient descent algorithms (like neural network-based models) for weights updating. This approach rules out the use of other algorithms that could be more suitable or perform better in some applications, thus posing a hard constraint on the development of optimal distributed solutions. Furthermore, issues related to non-independent and non-identically distributed (non-IID) data among the participating institutions, communication overheads during multiple training iterations and security threats are still open challenges, as outlined in [28,29].

One-shot federated learning

One-Shot FL [30] emerges as a significant variation of traditional FL, aimed at addressing some limitations. In one-shot FL, the fundamental shift lies in the reduction of communication rounds between the participating institutions and the central server. Unlike conventional FL, where multiple rounds of parameter updates and aggregations are necessary, it aims to achieve model convergence with just a single round of communication. This approach requires that each participating institution performs extensive local training on its data, and then the locally updated model parameters are sent to the central server only once. The server then aggregates these parameters to form a global model. The advantage here is the drastic reduction in communication overhead, making one-shot FL particularly beneficial in scenarios where communication costs are high or where network bandwidth is limited. However, its effectiveness depends on local training capabilities and the nature of distributed datasets, as there is no iterative process to refine the model based on aggregated global feedback.

Recently, Distilled One-Shot FL has also emerged as a promising approach [27]. In this setting, the process begins with each client performing dataset distillation on their local data, starting from a common initialised model. This involves condensing the data into a more compact yet representative version that captures the essential features and patterns of the original dataset [31]. The learned distilled datasets are then sent to the central server that aggregates them and performs the global training. This approach of applying distillation to the datasets, rather than the models, significantly reduces the size of the data that need to be communicated, thereby enhancing efficiency while still preserving privacy.

2.2. Survival analysis

The goal of Survival Analysis (SA), also referred to as *time-to-event* prediction, is to predict the time of occurrence of an event of interest, which can be an adverse event for patients in a clinical setting. In particular, SA techniques seek to infer the probability distribution of the time of that event for each patient. The *survival function* $S(t)$ is defined as:

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

where $P(T > t)$ is the probability that the event of interest occurs after time t, while $F(t)$ is the *cumulative incidence function* (CIF). A function strictly connected to the survival function is the hazard $h(t)$, defined as:

$$h(t) = \lim_{\Delta t \to 0} \frac{P\left(t \leq T < t + \Delta t \mid T \geq t\right)}{\Delta t} \geq 0 \tag{2}$$

This distribution represents the approximate probability that an event occurs in the time interval $[t, t + \Delta[$, under the condition that an

individual would remain event-free up to time t. Hence, it can be shown that:

$$S(t) = \exp\left[-\int_0^t h(u)du\right] = \exp[-H(t)] \tag{3}$$

where $H(t)$ is the cumulative hazard. Survival models aim to infer these distributions for each individual, conditioning them on the covariates of each patient.

From a practical point of view, to obtain time-to-event data, patients need to be followed in long observational studies, which typically last years. However, patients frequently drop out before any event occurs. In this case, it is only possible to know that the event did not occur before a specific dropping time. This partial knowledge is known in SA as right censoring, and it needs to be handled by any survival model. Survival data is composed of tuples $(\boldsymbol{x}_i, t_i, \delta_i)$, where $\boldsymbol{x}_i$ are the patient's covariates, δ_i is a binary event indicator and t_i is the time of the last follow-up visit, which coincides with the event time for uncensored patients, i.e. when $\delta_i = 1$.

COXPH

Cox proportional hazards model (CoxPH) is the most widely used technique to learn from censored survival data. It assumes that the hazard of the ith patient having an event at time t can be modelled as:

$$h\left(t \mid x_{i1}, \ldots, x_{id}\right) = h_0(t)\exp\left(\sum_{j=1}^{d} x_{ij}\beta_j\right) \tag{4}$$

where $\beta \in \mathbb{R}^d$ are the coefficients associated with each of the d features, while $h_0(t)$ is the baseline hazard function, common to all individuals in the group and dependent only on time. The second term of factorisation can be viewed as the relative risk of experiencing the event of interest, and it is considered to be constant over time as a direct implication of the proportional hazard assumption. The coefficients β are chosen through maximisation of the *partial log-likelihood*:

$$\begin{aligned}\widehat{\beta} &= \arg\max_{\beta} \log PL(\beta) = \\ &= \arg\max_{\beta} \sum_{i=1}^{n} \delta_i \left[\mathbf{x}_i^\top \beta - \log\left(\sum_{j\in\mathcal{R}_i} \exp\left(\mathbf{x}_j^\top \beta\right)\right)\right]\end{aligned} \tag{5}$$

where $\mathcal{R}_i = \left\{j \mid t_j > t_i\right\}$ is the risk set of the ith patient, which contains all the subjects that have not experienced an event or been censored at time t. The baseline hazard $h_0(t)$ can be calculated *a posteriori* through non-parametric methods such as Nelson–Aalen or Breslow estimators [32,33].

AFT-XGBOOST

The Accelerated Failure Time (AFT) model is another well-known method for survival analysis. Since CoxPH returns an estimate for the risk associated to each patient and predicts the baseline function in a separated step (usually through non-parametric approaches like the Breslow's estimator [32]), it does not yield directly a useable prediction of the time-to-event $\hat{t}$.

With the AFT model, it is possible to predict unknown labels using only the fitted parameters and a feature vector. This is done by modelling the survival time T as:

$$\ln T = \langle \beta, \mathbf{x} \rangle + \sigma Z \tag{6}$$

where Z is a random variable of a known probability distribution and σ is a scale parameter. Then, XGBoost method can be naturally embedded in this formulation by replacing the linear term with the output from the decision tree ensemble $\mathcal{T}(\mathbf{x})$ [34]:

$$\ln T = \mathcal{T}(\mathbf{x}) + \sigma Z \tag{7}$$

Thus, the model is optimised by maximising a revised likelihood function $\mathcal{L}_{AFT}(t, \mathcal{T}(\mathbf{x}))$, which properly takes into consideration censored and uncensored labels. The functional form of the likelihood will depend on the chosen distribution for the random variable Z. The most commonly used are logistic, normal, and extreme distributions.

3. Methodology

3.1. Model architecture

With the proposed framework, named *SYNDSURV* (SYNthetic Distributed SURVival), we address the task of developing a common survival model in a decentralised scenario, where data are stored in separated centres (e.g. healthcare institutions) and cannot be shared due to privacy limitations. The core idea behind our contribution is to build a unique survival model using an arbitrarily large set of aggregated synthetic data, that are independently generated at each node level. The pseudo-code of the whole procedure is shown in Algorithm 1.

Algorithm 1: Pseudo-code of the proposed framework for distributed survival analysis.

Input: Datasets $(\mathbf{X}_1, t_1, \delta_1), (\mathbf{X}_2, t_2, \delta_2), ..., (\mathbf{X}_C, t_C, \delta_C)$ from C nodes

Output: Trained global survival model $\mathcal{M}$

for *each node* c **do**
- $\left((\mathbf{X}_c^{\text{tr}}, t_c^{\text{tr}}, \delta_c^{\text{tr}}), (\mathbf{X}_c^{\text{tt}}, t_c^{\text{tt}}, \delta_c^{\text{tt}})\right) = \text{train\_test\_split}(\mathbf{X}_c, t_c, \delta_c)$
- $\widetilde{\mathbf{X}}_c = \text{Synthetic\_generator}(\mathbf{X}_c^{\text{tr}})$
- $\mathcal{M}_c = \text{FitAFTModel}(\mathbf{X}_c^{\text{tr}})$
  `# fit AFT-survival model to original data`
- $\widetilde{t}_c = \mathcal{M}_c(\widetilde{\mathbf{X}}_c^{\text{tr}})$
  `# predict event times for the synthetic data`
- $\widetilde{\delta}_c = \begin{cases} 0 & \text{if } \widetilde{t}_c > \max\left(t_c^{\text{tr}}\right) \\ 1 & \text{otherwise} \end{cases}$
  `# set synthetic event indicators`
- $\text{Send\_To\_Aggregator}(\widetilde{\mathbf{X}}_c, \widetilde{t}_c, \widetilde{\delta}_c)$

$(\widetilde{\mathbf{X}}, \widetilde{t}, \widetilde{\delta}) = \text{AggregateData}()$

$\mathcal{M} = \text{FitModel}(\widetilde{\mathbf{X}}, \widetilde{t}, \widetilde{\delta})$ `# train global survival model`

$\text{Evaluate}\left(\mathcal{M},\ (\mathbf{X}^{\text{tt}}, t^{\text{tt}}, \delta^{\text{tt}})\right)$

`# evaluate M on aggregated test data`

The global model $\mathcal{M}$ has been tested twice, using two different time-to-event predictive methods (CoxPH and DeepHit). The performance was assessed in terms of two survival-specific metrics (C-Index and time-dependent AUC), which were compared to the upper bound (centralised scenario) and the lower bound (isolated nodes) scores. A schematic representation of the framework is presented in Fig. 1.

3.2. Data synthesizer

To generate the synthetic data, we have employed the *DataSynthesizer* tool [35], which is a Python implementation of the *PrivBayes* algorithm [36]. It is a generative model that can be represented as a directed acyclic graph. In the graph, each node X_i represents a feature, and each edge from X_j to X_i corresponds to $\Pr(X_i|X_j)$, the probability of X_i given X_j.

In summary, the algorithm operates in two steps. First, it creates a k-degree Bayesian Network $\mathcal{N}$ (where k defines the maximum size of any parent node set Π_i in $\mathcal{N}$) that approximates the real distribution of the features set $\mathcal{A}$, such that $\Pr_{\mathcal{N}}[\mathcal{A}] \approx \Pr[\mathcal{A}]$. In particular, under the

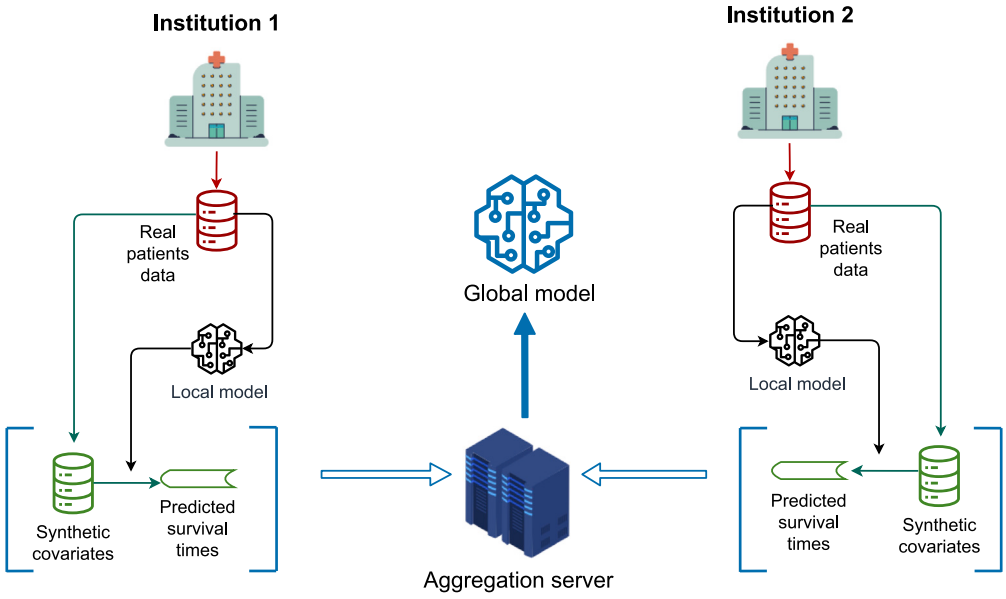

Fig. 1. Schematic representation of the SYNDSURV framework. In this example, there are two participating institutions that cooperate to create a global model. Each one trains a local AFT survival algorithm using only its own patients' data. Then, synthetic samples are generated through a Bayesian generative network built with patient covariates at single institution level, and local models are used to predict survival times for synthetic patients. Finally, all synthetic data are transferred from the data centres to a central aggregation server that trains the global model. Its performance is evaluated on an external test set composed of real patients from each institution.

assumption that any X_i and any $X_j \notin \Pi_i$ are conditionally independent given Π_i, we have:

$$\Pr[\mathcal{A}] = \Pr\left[X_1, X_2, \ldots, X_d\right] = \prod_{i=1}^{d} \Pr\left[X_i \mid \Pi_i\right] \tag{8}$$

Thus, the construction of $\Pr_{\mathcal{N}}[\mathcal{A}]$ can be viewed as an optimisation problem, where the objective is to minimise the *KL-divergence* $D_{KL}\left(\Pr[\mathcal{A}] \parallel \Pr_{\mathcal{N}}[\mathcal{A}]\right)$, by choosing the optimal parent node set Π_i related to each feature X_i. If k is a small value, this procedure becomes computationally feasible.

The second step is to generate synthetic data from the network $\mathcal{N}$. This is done by sampling each X_i from its conditional distribution $\Pr[X_i|\Pi_i]$ independently, without considering any attribute not in $\Pi_i \cup \{X_i\}$, which is possible thanks to Eq. (8).

Moreover, the generative process can also be combined with privacy-preserving approaches like Differential Privacy, which guarantees that the output of an algorithm is statistically indistinguishable on a pair of *neighbouring* databases (i.e. a pair of databases that differ by only one tuple). This concept is formalised with the following definition [37]:

Definition 3.1 (*ϵ-Differential Privacy*)**.** A randomised algorithm G satisfies ϵ-differential privacy if, for any two datasets D_1 and D_2 that differ only in one tuple x, and for any possible output O of G, we have:

$$\Pr\left[G\left(D_1\right) = O\right] \leq e^{\epsilon} \cdot \Pr\left[G\left(D_2\right) = O\right]. \tag{9}$$

The parameter ϵ is called privacy budget, and it controls how much the distribution of outputs can depend on data from an individual x. In practice, differential privacy is obtained by introducing a Laplacian noise $\mathrm{Lap}(\frac{2(d-k)}{n\epsilon})$ into each joint distribution $\Pr\left[X_i, \Pi_i\right]$, from which the differential private conditional distributions $\Pr^*\left[X_i \mid \Pi_i\right]$ are then derived. Here, d is the number of features and n is the number of samples in the dataset. It is worth highlighting that higher values of ϵ correspond to a lower amount of noise injected into the synthetic data, and hence to a lower level of privacy. Usually ϵ is set in a range of values that goes from 0.1 to 10 [36,38].

Once completed, this whole procedure leads to the generation of a new synthetic dataset $\widetilde{\mathbf{X}}_c$ for each node c, with the desired sample size, which depends on how far the sampling process is carried out.

3.3. Synthetic labels generation

The generated patients, in order to be used for training a common survival model, need to be paired with an estimated time-to-event label. Indeed, this information could not fit properly into the Bayesian Network due to the presence of the censored data. To overcome this issue, the proposed solution is to define, for each node, a survival regression with an accelerated failure time (AFT) model using XGBoost.

An advantage of applying an AFT model is that, unlike traditional Cox-PH, it yields directly a useable prediction of the time to the event $\hat{t}$, thus avoiding its estimation from the survival function. In addition, the AFT framework provides a better fit when the proportional hazard assumption does not hold and, combined with a tree-based Gradient Boosting method like XGBoost, it also considers nonlinear relationships among covariates, enriching the model's capabilities.

For each node, a separated AFT model $\mathcal{M}_c$ is trained by using only the real data of the node c and then applied to predict the time-to-event $\tilde{t}_c$ related to the synthetic samples $\widetilde{\mathbf{X}}_c$. In addition, in order to generate realistic survival data, the synthetic patients with a predicted time greater than the maximum observed event time of that node ($\widetilde{t_{c,i}} > t_{max,c}$) were considered right-censored at the time $t_{max,c}$, by setting $\widetilde{\delta_{c,i}} = 0$. This truncation avoids having long-tailed distributions for the synthetic times.

3.4. Centralised learning and validation strategy

The synthetic tuples $\left\{(\widetilde{\mathbf{X}_c}, \widetilde{t_c}, \widetilde{\delta_c})\right\}_{c=1,\ldots,C}$ are then shared with an aggregation server in order to be jointly used to build a common survival model $\mathcal{M}$. This allows the exploration of any desired survival algorithm in a centralised-like scenario, thus avoiding all the possible limitations related to model-specific federation processes.

In this work, to prove the effectiveness of the architecture, we applied both CoxPH model and DeepHit, which is a more complex neural network-based model that avoids the proportional hazard and linearity assumptions. In the CoxPH model we included an ElasticNet penalty to prevent overfitting [39], treating the penalty strength α and the l_1 *ratio* as tunable hyperparameters.

DeepHit is a recently developed survival model, which has been shown to outperform standard CoxPH in both single-event and

competing-risk survival predictions [40]. DeepHit discretises the survival times and trains a neural network to learn the estimated joint distribution of survival time and event, while capturing the right-censored nature inherent in survival data. The neural network returns a softmax layer that outputs an estimated probability for each discrete time interval, and it is trained by using as loss function a convex combination of a negative log-likelihood and a ranking loss. Both ElasticNet-CoxPH and DeepHit hyperparameters were optimised by applying on the internal validation set the tree-structured Parzen estimator algorithm [41] (implemented within the *Optuna* optimisation framework [42]).

The final model, despite being trained on the synthetic dataset, needs to be reliable also when applied to real patients. Therefore, for all simulations, we tested the performance with an external test set of the real data, which had never been used by the models $\mathcal{M}_c$ for the synthetic labels' prediction, nor by the centralised synthetic model $\mathcal{M}$. Moreover, in order to analyse the effect of the synthetic dataset's size on the final performance, we evaluated different sets of generated patients, ranging from 10^1 to 3×10^3 samples per node.

As in the case of FL strategies, the performance naturally has an *upper bound* and a *lower bound*. The *upper bound* represents the centralised scenario on the real data, i.e. when patients' data can be directly shared to a central node without restrictions. In this case, it is the performance of the model trained on the aggregated real data from all nodes. In contrast, the *lower bound* indicates the worst-case scenario, i.e. when the model is trained for each isolated node using only its own data. Therefore, it is the average performance across all nodes. The expected behaviour of our method in order to demonstrate its effectiveness is that its performance on the test set has to lie somewhere within these two limits. The whole procedure, including the synthetic patients' generation from the training data, has been repeated 10 times by changing the original random train–test split.

3.5. Metrics

Concordance index

Harrell's concordance index is the most frequently used evaluation metric for survival models [43]. It is a measure of the rank correlation between predicted risk scores $\hat{r}$ and observed time points t. It is defined as the ratio of correctly ordered (*concordant*) pairs to *comparable* pairs. Two samples i and j are comparable if the sample with lower observed time t experienced an event, i.e., if $t_j > t_i$ and $\delta_i = 1$, where $\delta_i = 1$ is the binary event indicator. A comparable pair (i, j) is concordant if the estimated risk $\hat{r}$ by a survival model is higher for subjects with lower survival time, i.e. $\hat{r_i} > \hat{r_j} \wedge t_j > t_i$, otherwise the pair is discordant. However, Harrell's estimate is based on the proportional hazard assumption, considering constant risks over time. Therefore, in this study, a time-dependent formulation of the concordance index (from Antolini et al. [44]) was used to evaluate the models. It takes into consideration the whole predicted survival function, and it is defined as:

$$C^{\text{td}} = \text{P}\left\{\hat{S}\left(T_i \mid \mathbf{x}_i\right) < \hat{S}\left(T_i \mid \mathbf{x}_j\right) \mid T_i < T_j, D_i = 1\right\}. \tag{10}$$

For proportional hazards models, this metric is equivalent to the regular Harrell's concordance index.

Time-dependent AUC

The receiver operating characteristic (ROC) curve and the area under the ROC curve (AUC) can be extended to the survival data by defining sensitivity and specificity as time-dependent measures [45]. It is possible to define as *cumulative cases* the individuals who experienced an event up to time t ($t_i \le t$), and as *dynamic controls* those individuals with $t_j > t$. Hence, the associated cumulative/dynamic AUC can be expressed as:

$$\widehat{\text{AUC}}(t) = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} I\left(t_j > t\right) I\left(t_i \le t\right) \omega_i I\left(\hat{f}\left(\mathbf{x}_j, t\right) \le \hat{f}\left(\mathbf{x}_i, t\right)\right)}{\left(\sum_{i=1}^{n} I\left(t_i > t\right)\right)\left(\sum_{i=1}^{n} I\left(t_i \le t\right)\omega_i\right)} \tag{11}$$

Table 1
Summary of survival datasets.

	Size	# Features	% Censored	C	Nodes size
FLChain	6524	7	70%	32	153
METABRIC	1904	9	42%	8	178
SUPPORT	9105	20	32%	32	213
GBSG+Rotterdam	3668	8	50%	16	172

where $\hat{f}\left(\mathbf{x}_i, t\right)$ is the predicted risk of a patient with covariates $\mathbf{x}_i$ at time t and ω_i are the inverse probability of censoring weights (IPCW), directly estimated from the censoring distribution of the training set. This metric quantifies how well a model can distinguish patients who experience an event in a given time ($t_i \le t$) from patients who fail after this time ($t_j > t$) and it is mostly relevant when the aim is to predict the occurrence of an event in a period of time up to t rather than at a specific time point. In our study, we have considered the integral of the time-dependent AUC over the range of observed event times.

4. Results

4.1. Survival datasets

To validate our framework, we considered four publicly available survival datasets, commonly used to assess the performance of the survival models: the Assay of Serum Free Light Chain [46] (FLCHAIN), the Molecular Taxonomy of Breast Cancer International Consortium [47] (METABRIC), the Study to Understand Prognoses Preferences Outcomes and Risks of Treatment [48] (SUPPORT) and the Rotterdam tumour bank and German Breast Cancer Study Group [49,50] (GBSG+ Rotterdam). FLChain and METABRIC datasets were preprocessed according to Kvamme et al. [51], while for GBSG+Rotterdam the same preprocessing schema of Royston and Altman [52] was applied, with the only difference that the data from the two studies were merged to create a single dataset. As regards SUPPORT, we used the same procedure of Chapfuwa et al. [53] for data cleaning and imputation, followed by a final feature selection to decrease the data dimensionality, as it is summarised in Figure S1 (see Supplementary Materials).

Each dataset was randomly split into training and test sets (75%–25%), repeating the partitioning 10 times by changing the random seeds. Then, the samples of the training sets were further distributed among C fictitious centres to simulate a decentralised scenario. The number of nodes C for each dataset was selected so that the performance gap between upper and lower bounds (in at least one evaluation metric) was large enough to justify an advantage in applying a decentralised approach. All the created fictitious centres included $\approx 2 \times 10^2$ samples, which is also a reasonable number of patients for real-world clinical trials of medium size. Summary statistics for the preprocessed survival datasets are reported in Table 1.

4.2. Performance comparison

As mentioned above, to properly evaluate the performance of the synthetic model $\mathcal{M}$ in the test data, it must be compared to the upper bound (centralised scenario) and the lower bound (isolated nodes). From a preliminary comparison of the performance between CoxPH and DeepHit in the centralised scenario, as shown in Table 2a and b, it clearly results that DeepHit outperforms or at least equals the standard CoxPH method for both metrics, which is expected due to the greater flexibility and capacity of the model. Conversely, in the isolated nodes the same consideration does not hold and CoxPH performs better than DeepHit in 2 out of 4 datasets for both metrics. This is probably due to the reduced size of the fictitious data, which mostly affects DeepHit. It is also worth noticing that the performance gap between upper and lower bounds is not negligible for each dataset and for both metrics. This confirms the necessity of the proposed approach.

Table 2
C-Index (a) and Time-dependent AUC (b) metrics on the test set in the centralised scenario (upper bound) and in the case of isolated nodes (lower bound). The reported scores represent the mean values over the 10 different test sets; error of the means are written in brackets. Best scores are highlighted in bold.

(a)

	Lower bound		Upper bound	
	COXPH	DEEPHIT	COXPH	DEEPHIT
FLChain	0.766(0.003)	**0.779**(0.003)	0.789(0.003)	0.789(0.003)
METABRIC	**0.612**(0.004)	0.604(0.005)	0.630(0.005)	**0.674**(0.005)
SUPPORT	0.776(0.001)	**0.785**(0.001)	0.802(0.001)	**0.843**(0.001)
GBSG+Rotter.	**0.641**(0.003)	0.604(0.005)	0.671(0.003)	**0.680**(0.003)

(b)

	Lower bound		Upper bound	
	COXPH	DEEPHIT	COXPH	DEEPHIT
FLChain	**0.790**(0.004)	0.743(0.004)	0.812(0.004)	**0.813**(0.004)
METABRIC	**0.659**(0.006)	0.631(0.004)	0.688(0.006)	**0.695**(0.005)
SUPPORT	0.866(0.001)	**0.867**(0.001)	0.894(0.002)	**0.920**(0.001)
GBSG+Rott.	0.678(0.005)	0.678(0.005)	0.716(0.005)	**0.727**(0.004)

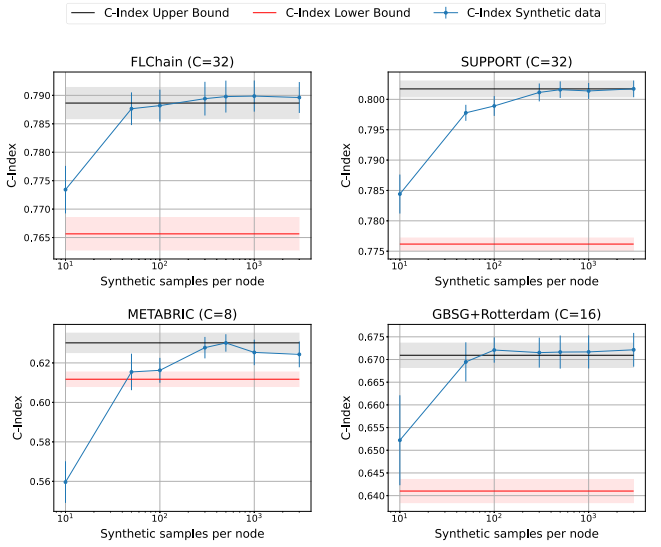

Fig. 2. Test set C-Index vs. number of generated patients per node, for the CoxPH model trained on the synthetic data. Reported values represent the mean over 10 different test sets; error bars indicate the errors of the mean.

Hence, we present the results of the global models, trained only with the synthetic data and evaluated on the external test sets of real patients. C-index and time-dependent AUC were considered for a variable number of synthetic patients generated per node, and, for each, the mean and standard deviation values over 10-times repeated simulations were computed. The C-Indices for the CoxPH model are reported in Fig. 2, while those related to DeepHit are shown in Fig. 3. In both figures, each panel corresponds to a separated survival dataset; red and black lines represent the lower and upper bounds, respectively. The time-dependent AUCs are shown in Fig. S3–S5 in the Supplementary Materials.

For CoxPH, the synthetic model reaches the upper limit of the C-Index in all four datasets, even with less than 10^3 generated patients per node; when considering the time-dependent AUCs, this happens in 3 out of 4 datasets, even though in the remaining one (METABRIC) the average scores fall clearly within the desired range. Considering DeepHit, in all simulations the average performance of the synthetic model exceeds the lower bound as the number of patients generated per node is above 10^2, and most of the time it is closer to the upper limit when more than 10^3 patients are generated.

These results show that the survival models trained on the synthetic data generated through the proposed method are capable of

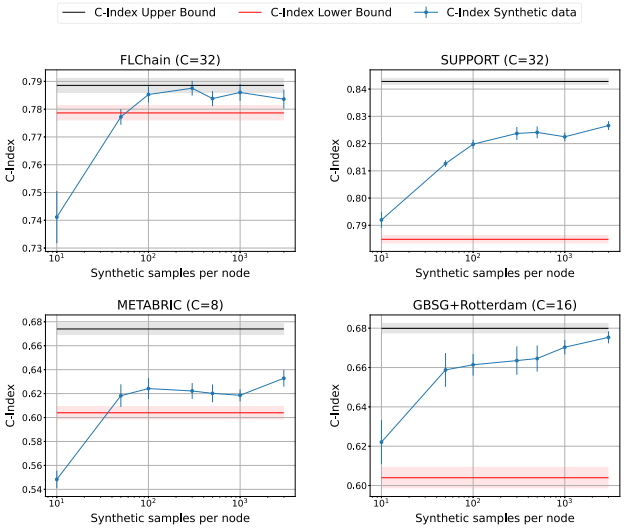

Fig. 3. Test set C-Index vs. number of generated patients per node, for the DeepHit model trained on the synthetic data. Reported values represent the mean over 10 different test sets; error bars indicate the errors of the mean.

generalising well to real patients' data, achieving a performance always comparable to that of the centralised scenario or considerably higher than the lower bound. Moreover, such achievements are not model-dependent, since the same considerations apply for both CoxPH and DeepHit models, which are different in terms of baseline assumptions, optimisation function, and complexity.

4.3. Differential privacy

Finally, we also tested the goodness of the proposed framework in a Differential Privacy setting. This is done by adding to the Bayesian Network (from which synthetic covariates are sampled) a certain amount of noise parameterised by the privacy budget ϵ, as explained in 3.2. In our experiments, we set $\epsilon = 0.1$, which produces a high level of noise at the expense of the synthetic data quality and hence of the model's performance. The results for DeepHit on the datasets considered in this scenario are presented in Fig. 4.

Due to the added noise, both C-Indices and time-dependent AUCs are slightly deteriorated on the test set with respect to the scenario without differential privacy, but they are higher than the lower bound in all four datasets when more than 5×10^2 patients per node are generated. This achievement shows that the framework *SYNDSURV* is robust to the presence of high levels of noise in the synthetic data. Moreover, by controlling the privacy budget ϵ, it is possible to choose the desired trade-off between model performance and privacy requirements.

5. Conclusion

We proposed a distributed survival analysis framework to address the task of creating a common survival model that indirectly uses time-to-event data from multiple participating institutions, without sharing patients' sensitive data. This is achieved by generating synthetic patients at the level of the single data centres through a two-steps procedure that properly handles survival labels: (1) the synthetic patient's covariates are sampled from a Bayesian generative network; (2) a survival AFT model trained on the original data is used to predict the survival times of the synthetic patients. Thus, the synthetic samples from each institution are shared with a common aggregation server to train the desired global model.

As demonstrated by simulations performed over four distinct survival datasets (split among separated fictitious nodes), the global

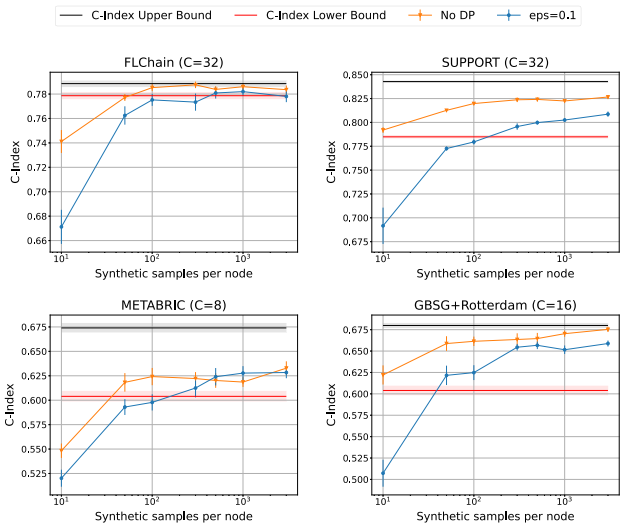

Fig. 4. Test set C-Index vs. number of generated patients per node, for the DeepHit model trained on the synthetic data, using a differential privacy setting with $\epsilon = 0.1$. Reported values represent the mean over 10 different test sets; error bars indicate the errors of the mean.

model's performance on a test set composed of real patients is often comparable to the centralised scenario, where data are shared without limitations, or at least significantly greater than the lower bound (isolated nodes scenario). These results prove the applicability of the method as a valid alternative to the current FL approaches used to learn common predictive models. A main advantage of this approach is that it is flexible to the choice of any desired algorithm for designing the global model, since it does not depend on the algorithm structure as it happens for the FL methods. In addition, it is also suitable to properly handle time-to-event data in presence of censoring. Finally, we also demonstrated that our approach is robust to the presence of noise into the synthetic data, hence it can be naturally embedded in differential privacy settings, which can be necessary when controlling the privacy budget.

5.1. Limitations and future works

Despite its advantages, the proposed framework may have some limitations. A possible limitation is that the performance of the global model depends on the quality of the synthetic samples generated by the local data centres. If the synthetic samples are not representative of the true data, the global model may not perform well.

Another limitation is that the proposed method assumes that the censoring mechanism is the same across all local data centres, which may only sometimes be the case in practice.

Finally, generating synthetic samples at local data centres might be challenging, especially for datasets with many features. However, current federated approaches may suffer from the same problem.

Future developments include further experiments to test if our framework can address more troublesome real-world scenarios, where nodes are highly unbalanced or data are non-IID across the participating centres. Likewise, different strategies for synthetic data generation should be investigated, including Autoencoders (Decoders), Generative Adversarial Networks (GANs) or other generative Bayesian network techniques.

Ethical statement

No ethical approval or informed consent was required for the current analysis.

CRediT authorship contribution statement

Cesare Rollo: Writing – original draft, Formal analysis, Conceptualization. **Corrado Pancotti:** Formal analysis. **Giovanni Birolo:** Formal analysis. **Ivan Rossi:** Conceptualization. **Tiziana Sanavia:** Writing – review & editing, Supervision, Formal analysis. **Piero Fariselli:** Writing – review & editing, Supervision, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

All the public datasets used in this manuscript are available at the GitHub repository https://github.com/compbiomed-unito/SYNDSURV.

Funding and acknowledgements

This study was funded by the European Union's Horizon 2020 project GenoMed4All (Grant Agreement ID: 101017549) and we thank the European Union's Horizon 2020 project Brainteaser (Grant Agreement ID: 101017598).

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.compbiomed.2024.108288.

References

[1] S. Dash, S.K. Shakyawar, M. Sharma, S. Kaushik, Big data in healthcare: management, analysis and future prospects, J. Big Data 6 (1) (2019) 1–25.
[2] S. Zillner, S. Neururer, Big data in the health sector, in: New Horizons for a Data-Driven Economy, Springer, 2016, pp. 179–194.
[3] A. Panesar, Machine Learning and AI for Healthcare, Springer, 2019.
[4] F. Jiang, Y. Jiang, H. Zhi, Y. Dong, H. Li, S. Ma, Y. Wang, Q. Dong, H. Shen, Y. Wang, Artificial intelligence in healthcare: past, present and future, Stroke Vasc. Neurol. 2 (4) (2017).
[5] N. Rieke, J. Hancox, W. Li, et al., The future of digital health with federated learning, NPJ Digit. Med. 3 (2020) 119.
[6] F. Cremonesi, V. Planat, V. Kalokyri, H. Kondylakis, T. Sanavia, V. Miguel Mateos Resinas, B. Singh, S. Uribe, The need for multimodal health data modeling: A practical approach for a federated-learning healthcare platform, J. Biomed. Inform. 141 (2023) 104338.
[7] S. D'Amico, L. Dall'Olio, C. Rollo, P. Alonso, I.n. Prada-Luengo, D. Dall'Olio, C. Sala, M. Bersanelli, E. Sauta, M. Bicchieri, P. Morandini, T. Tommasini, V. Savevski, L.-P. Zhao, U. Platzbecker, M. Diez-Campelo, V. Santini, P. Fenaux, T. Haferlach, A. Krogh, S. Zazo, P. Fariselli, T. Sanavia, M.G. Della Porta, G. Castellani, Multi-modal analysis and federated learning approach for classification and personalized prognostic assessment in myeloid neoplasms, Blood 140 (Supplement 1) (2022) 9828–9830.
[8] M. Gawali, C. Arvind, S. Suryavanshi, H. Madaan, A. Gaikwad, K. Bhanu Prakash, V. Kulkarni, A. Pant, Comparison of privacy-preserving distributed deep learning methods in healthcare, in: Annual Conference on Medical Image Understanding and Analysis, Springer, 2021, pp. 457–471.
[9] F. Zerka, S. Barakat, S. Walsh, M. Bogowicz, R.T. Leijenaar, A. Jochems, B. Miraglio, D. Townend, P. Lambin, Systematic review of privacy-preserving distributed machine learning from federated databases in health care, JCO Clin. Cancer Inform. 4 (2020) 184–200.
[10] T. Pereira, J. Morgado, F. Silva, M.M. Pelter, V.R. Dias, R. Barros, C. Freitas, E. Negrão, B. Flor de Lima, M. Correia da Silva, et al., Sharing biomedical data: Strengthening ai development in healthcare, in: Healthcare, Vol. 9, MDPI, 2021, p. 827.
[11] M. Kirienko, M. Sollini, G. Ninatti, D. Loiacono, E. Giacomello, N. Gozzi, F. Amigoni, L. Mainardi, P.L. Lanzi, A. Chiti, Distributed learning: a reliable privacy-preserving strategy to change multicenter collaborations using AI, Eur. J. Nucl. Med. Mol. Imaging 48 (12) (2021) 3791–3804.
[12] 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.

[13] 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, Sci. Rep. 10 (1) (2020) 1–12.

[14] 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 Digit. Med. 3 (1) (2020) 1–7.

[15] 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, et al., Learning from local to global: An efficient distributed algorithm for modeling time-to-event data, J. Am. Med. Inf. Assoc. 27 (7) (2020) 1028–1036.

[16] 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, J. Am. Med. Inf. Assoc. 22 (6) (2015) 1212–1219.

[17] 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, J. Biomed. Inform. 134 (2022) 104176.

[18] M.M. Rahman, S. Purushotham, FedPseudo: Pseudo value-based deep learning models for federated survival analysis, 2022, arXiv preprint arXiv:2207.05247.

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

[20] J. Yoon, L.N. Drumright, M. Van Der Schaar, Anonymization through data synthesis using generative adversarial networks (ads-gan), IEEE J. Biomed. Health Inform. 24 (8) (2020) 2378–2388.

[21] J. Jordon, J. Yoon, M. Van Der Schaar, PATE-GAN: Generating synthetic data with differential privacy guarantees, in: International Conference on Learning Representations, 2019.

[22] J. de Benedetti, N. Oues, Z. Wang, P. Myles, A. Tucker, Practical lessons from generating synthetic healthcare data with Bayesian networks, in: ECML PKDD 2020 Workshops: Workshops of the European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD 2020): SoGood 2020, PDFL 2020, MLCS 2020, NFMCP 2020, DINA 2020, EDML 2020, XKDD 2020 and INRA 2020, Ghent, Belgium, September 14–18, 2020, Proceedings, Springer, 2020, pp. 38–47.

[23] Z. Wang, P. Myles, A. Tucker, Generating and evaluating cross-sectional synthetic electronic healthcare data: Preserving data utility and patient privacy, Comput. Intell. 37 (2) (2021) 819–851.

[24] A. Norcliffe, B. Cebere, F. Imrie, P. Lio, M. van der Schaar, SurvivalGAN: Generating time-to-event data for survival analysis, in: International Conference on Artificial Intelligence and Statistics, PMLR, 2023, pp. 10279–10304.

[25] R. Song, D. Liu, D.Z. Chen, A. Festag, C. Trinitis, M. Schulz, A. Knoll, Federated learning via decentralized dataset distillation in resource-constrained edge environments, in: 2023 International Joint Conference on Neural Networks, IJCNN, IEEE, 2023, pp. 1–10.

[26] Y. Xiong, R. Wang, M. Cheng, F. Yu, C.-J. Hsieh, Feddm: Iterative distribution matching for communication-efficient federated learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 16323–16332.

[27] Y. Zhou, G. Pu, X. Ma, X. Li, D. Wu, Distilled one-shot federated learning, 2020, arXiv preprint arXiv:2009.07999.

[28] P.M. Mammen, Federated learning: opportunities and challenges, 2021, arXiv preprint arXiv:2101.05428.

[29] 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, Found. Trends® Mach. Learn. 14 (1–2) (2021) 1–210.

[30] N. Guha, A. Talwalkar, V. Smith, One-shot federated learning, 2019, arXiv preprint arXiv:1902.11175.

[31] T. Wang, J.-Y. Zhu, A. Torralba, A.A. Efros, Dataset distillation, 2018, arXiv preprint arXiv:1811.10959.

[32] D. Lin, On the Breslow estimator, Lifetime Data Anal. 13 (4) (2007) 471–480.

[33] E. Colosimo, F.v. Ferreira, M. Oliveira, C. Sousa, Empirical comparisons between Kaplan-Meier and Nelson-Aalen survival function estimators, J. Stat. Comput. Simul. 72 (4) (2002) 299–308.

[34] A. Barnwal, H. Cho, T. Hocking, Survival regression with accelerated failure time model in XGBoost, J. Comput. Graph. Statist. (just-accepted) (2022) 1–25.

[35] H. Ping, J. Stoyanovich, B. Howe, Datasynthesizer: Privacy-preserving synthetic datasets, in: Proceedings of the 29th International Conference on Scientific and Statistical Database Management, 2017, pp. 1–5.

[36] J. Zhang, G. Cormode, C.M. Procopiuc, D. Srivastava, X. Xiao, Privbayes: Private data release via Bayesian networks, ACM Trans. Database Syst. 42 (4) (2017) 1–41.

[37] C. Dwork, F. McSherry, K. Nissim, A. Smith, Calibrating noise to sensitivity in private data analysis, in: Theory of Cryptography Conference, Springer, 2006, pp. 265–284.

[38] J. Lee, C. Clifton, How much is enough? Choosing ε for differential privacy, in: International Conference on Information Security, Springer, 2011, pp. 325–340.

[39] N. Simon, J. Friedman, T. Hastie, R. Tibshirani, Regularization paths for Cox's proportional hazards model via coordinate descent, J. Stat. Softw. 39 (5) (2011) 1.

[40] 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.

[41] J. Bergstra, R. Bardenet, Y. Bengio, B. Kégl, Algorithms for hyper-parameter optimization, Adv. Neural Inf. Process. Syst. 24 (2011).

[42] T. Akiba, S. Sano, T. Yanase, T. Ohta, M. Koyama, Optuna: A next-generation hyperparameter optimization framework, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 2623–2631.

[43] F.E. Harrell, R.M. Califf, D.B. Pryor, K.L. Lee, R.A. Rosati, Evaluating the yield of medical tests, JAMA 247 (18) (1982) 2543–2546.

[44] L. Antolini, P. Boracchi, E. Biganzoli, A time-dependent discrimination index for survival data, Statist. Med. 24 (24) (2005) 3927–3944.

[45] H. Hung, C.-T. Chiang, Estimation methods for time-dependent AUC models with survival data, Canad. J. Statist. 38 (1) (2010) 8–26.

[46] T. Therneau, et al., A package for survival analysis in S, 2015, R package version 2.

[47] C. Curtis, S.P. Shah, S.-F. Chin, G. Turashvili, O.M. Rueda, M.J. Dunning, D. Speed, A.G. Lynch, S. Samarajiwa, Y. Yuan, et al., The genomic and transcriptomic architecture of 2,000 breast tumours reveals novel subgroups, Nature 486 (7403) (2012) 346–352.

[48] W.A. Knaus, F.E. Harrell, J. Lynn, L. Goldman, R.S. Phillips, A.F. Connors, N.V. Dawson, W.J. Fulkerson, R.M. Califf, N. Desbiens, et al., The SUPPORT prognostic model: Objective estimates of survival for seriously ill hospitalized adults, Ann. Intern. Med. 122 (3) (1995) 191–203.

[49] J.A. Foekens, H.A. Peters, M.P. Look, H. Portengen, M. Schmitt, M.D. Kramer, N. Brünner, F. Jänicke, M.E.M.-v. Gelder, S.C. Henzen-Logmans, et al., The urokinase system of plasminogen activation and prognosis in 2780 breast cancer patients, Cancer Res. 60 (3) (2000) 636–643.

[50] 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, J. Clin. Oncol. 12 (10) (1994) 2086–2093.

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

[52] P. Royston, D.G. Altman, External validation of a cox prognostic model: principles and methods, BMC Med. Res. Methodol. 13 (1) (2013) 1–15.

[53] P. Chapfuwa, C. Li, N. Mehta, L. Carin, R. Henao, Survival cluster analysis, in: Proceedings of the ACM Conference on Health, Inference, and Learning, 2020, pp. 60–68.