Survival Model Optimization via Federated Learning: A Study Combining Simulations and Experiments

Francesco Casadei
IRCCS Istituto delle Scienze Neurologiche di Bologna
Bologna, Italy
francesco.casadei@ausl.bologna.it

Luciana Carota
Department of Medical and Surgical Sciences University of Bologna
Bologna, Italy
luciana.carota@unibo.it

Gianluca Asti
Humanitas Clinical and Research Center IRCCS
Milan, Italy
gianluca.asti@humanitas.it

Saverio D'Amico
Humanitas Clinical and Research Center IRCCS
Milan, Italy
saverio.damico@humanitas.it

Davide Piscia
Centro Nacional de Análisis Genómico, C/Baldiri Reixac 4, 08028
Barcelona, Spain
davide.piscia@cnag.eu

Santiago Zazo
Information Processing and Telecommunications Center ETS Ingenieros de Telecomunicación, Universidad Politécnica de Madrid
Madrid, Spain
santiago.zazo@upm.es

Patricia A. Apellániz
Information Processing and Telecommunications Center ETS Ingenieros de Telecomunicación, Universidad Politécnica de Madrid
Madrid, Spain
patricia.alonsod@upm.es

Juan Parras
Information Processing and Telecommunications Center ETS Ingenieros de Telecomunicación, Universidad Politécnica de Madrid
Madrid, Spain
j.parras@upm.es

Claudia Sala
Department of Medical and Surgical Sciences University of Bologna
Bologna, Italy
claudia.sala3@unibo.it

Cesare Rollo
Computational Biomedicine Unit Department of Medical Sciences
Turin, Italy
cesare.rollo@unito.it

Nono S. C. Merleau
Max Planck Institute for Mathematics in the Sciences of Leipzig
Leipzig, Germany
nonosaha@mis.mpg.de

Piero Fariselli
Computational Biomedicine Unit Department of Medical Sciences
Turin, Italy
piero.fariselli@unito.it

Matteo Della Porta
Humanitas Clinical and Research Center IRCCS
Milan, Italy
matteo.della_porta@hunimed.eu

Tiziana Sanavia
Computational Biomedicine Unit Department of Medical Sciences
Turin, Italy
tiziana.sanavia@unito.it

Federico Alvarez Garcia
Information Processing and Telecommunications Center ETS Ingenieros de Telecomunicación, Universidad Politécnica de Madrid
Madrid, Spain
fag@gatv.ssr.upm.es

Gastone Castellani
Department of Medical and Surgical Sciences University of Bologna IRCCS Azienda Ospedaliero-Universitaria di Bologna
Bologna, Italy
gastone.castellani@unibo.it

Enrico Giampieri
Department of Medical and Surgical Sciences University of Bologna IRCCS Azienda Ospedaliero-Universitaria di Bologna
Bologna, Italy
enrico.giampieri@unibo.it

2024 IEEE International Conference on Big Data (BigData) | 979-8-3503-6248-0/24/$31.00 ©2024 IEEE | DOI: 10.1109/BigData62323.2024.10825368

***Abstract*—Federated Learning is an emerging, powerful approach that allows training an artificial intelligence model in a distributed setting. Two survival models, Cox and DeepSurv, have been trained in a federated setting, exploiting both code simulations and real experiments on the new platform, developed by the GenoMed4All consortium. Different scenarios have been tested by splitting a Myelodysplastic Syndrome dataset into three nodes and performing feature removal. A significant gain in model performance has been observed due to federated aggregation.**

***Index Terms*—Federated Learning, Survival Model, Myelodysplastic Syndromes, Neural Networks, GenoMed4All**

I. Introduction

In recent years, the rapid growth of Next Generation Sequencing (NGS) platforms led to a widespread availability of omics data (e.g. genomics, transcriptomics, proteomics and epigenomics), which are crucial for achieving clinical accuracy of predictive models and contributing to the development of precision medicine [1]. The integration of different types of omics data in model development improves performance [2], [3] and can be used to identify potential new biomarkers [4], increasing the predictive power [5]. Complex data in turn require sophisticated algorithms that are both flexible and able

to account for non-linear relationships, such as Deep Learning (DL) models [2], which are neural networks characterized by several hidden layers and a high number of nodes [6]. Therefore, these models have many parameters to set through a training procedure, which requires a large volume of data [7]. However, in healthcare applications, when dealing with rare diseases such as most oncohematological disorders, the availability of data is limited, especially for the development of prediction methods for the survival analysis, which have to deal with censored data that can heavily bias the results [8]–[10]. Therefore, new strategies to improve data availability are necessary, like synthetic data generation [10]. Survival models would benefit from increased data availability, as it could mitigate censoring phenomenon effects [9]. In addition, training DL models on heterogeneous datasets prevents overfitting issues or biases, making the trained model more robust and scalable [11]. However, a suitable training dataset requires the collection of data from different healthcare entities (e.g. hospitals, clinics and research institutions) with strict regulations on data privacy that limit the sharing of sensitive data [12].

Federated Learning (FL) provides a possible solution to the problem of data sharing. FL is a branch of Machine Learning (ML) consisting in a distributed training of an ML model [12]. This framework is based on sharing model weights and/or gradients rather than data itself [12], [13]. Different FL workflows are available [14]. One of the most adopted solutions is the horizontal FL with an aggregation server [14].

In this approach, a central server receives a common ML model architecture, called the global model, which is shared among the different institutions, called nodes. Each node has its local model, with local weights, and its own dataset, which cannot be shared with the other nodes. When the training phase starts, each node trains its local model on its data for a pre-established number of epochs. Then, local weights or gradients are aggregated (usually, they are averaged) in the central server, which sends them to each node, updating each local model. The training phase proceeds locally for several epochs, aggregating the weights. This process continues until a satisfactory model performance is reached. Fig. 1 shows the diagram of a basic FL workflow.

Federated aggregation refers to the process of aggregating model parameters from all nodes in each round of federated communication to build a global model [15]. Usually, trainable parameters that undergo aggregation are weight parameters or gradients of the neural networks. In a centralized FL scenario, the central server controls the aggregation, which collects and merges parameters from individual nodes.

The simplest federated aggregation strategy is called FedAvg [16] and basically consists of a weighted average of local model parameters:

$$W_{glob} = \sum_{k \in S_t} \frac{n_k}{N} \cdot w_k \tag{1}$$

where S_t is the set of nodes, n_k is the data volume of node k, N is the total amount of data, w_k are the model

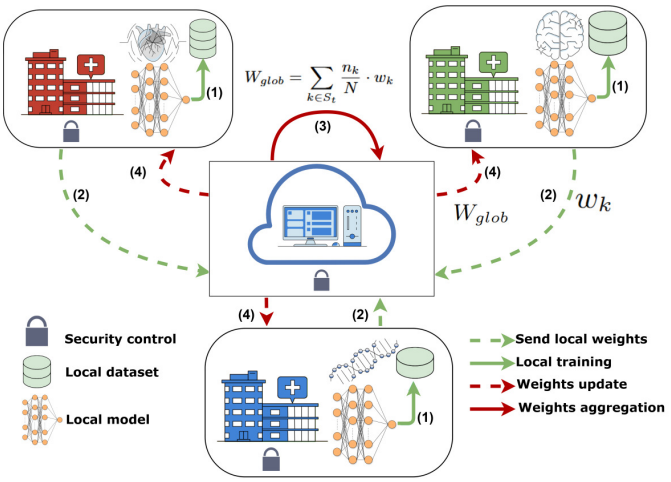


Fig. 1: Scheme of a horizontal FL setting with a central server and three local nodes, each with its own local data and local model instance. When training starts, local nodes train their model on its own data (1), shares weights to the central server (2) that aggregates them (3) and sends them back to the local nodes (4), whose models get updated.

parameters of node k and W_{glob} are the aggregated global model parameters. Other aggregation solutions can be used, spanning from small variations of FedAvg (e.g. using median instead of average) to more complicated algorithms (e.g. FedNova [17]).

This study aims to implement a federated version of survival models and to train them on a FL setting composed of three distinct nodes. FL training has been performed in two different ways: one is based on a simple Python code simulation without the use of specific FL libraries, while the other one exploits the FL platform developed in the context of the EU project GenoMed4All [18]. These two approaches are called FLsimul and FLreal, respectively.

II. Materials and Methods

A. Federated implementation of the GenoMed4All platform

The FL platform includes a Manager Node (MN) and multiple Worker Nodes (WNs). The WN is the safe environment within each institution to upload the data. The MN acts as an orchestrating platform for the FL framework. As shown in Fig. 2, on this main interface the WNs installed in other hospitals can be registered, uploading the models in pickle format and saving the dataset's path on each WN. When a training session starts, the selected pickle model is shared from the MN to the different WNs. Within each WN, a containerized Flower Environment [19], [20] is spawned, and the model is trained for a full epoch on the local dataset. When the epoch is completed, the model weights are shared back to the MN node for FedAvg aggregation, in order to create a more robust central model; specific training metrics are also shared back to the MN for monitoring through an MLFlow interface, Fig. 3. The aggregated weights of the central model are then sent once again to the various WNs involved in the training: this process is repeated until the number of training rounds specified by the user is met. Thanks to this paradigm, the dataset never

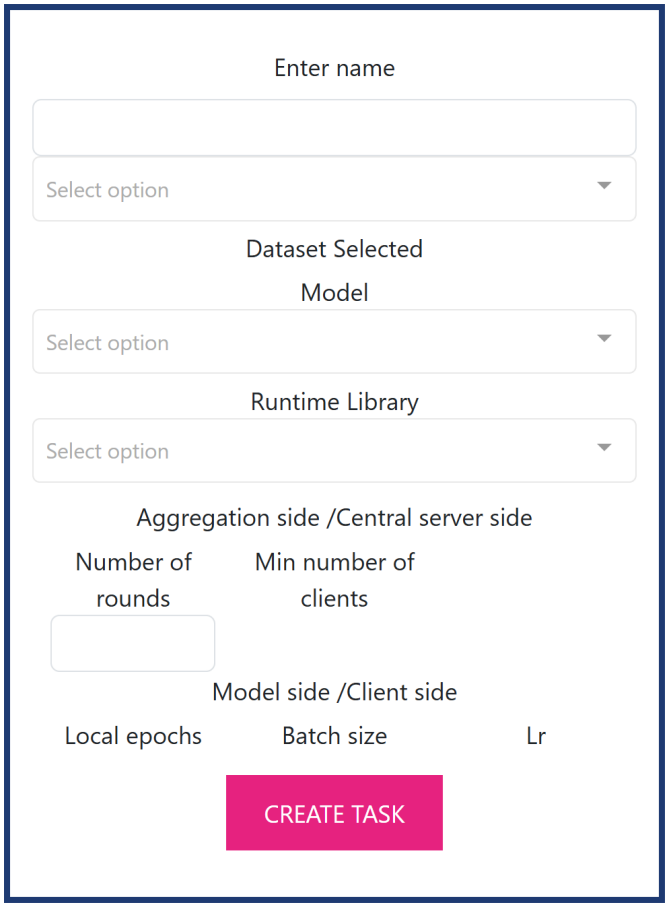


Fig. 2: Screenshot of the GenoMed4All platform dahsboard. From top to bottom: test name definition, dataset selection, model selection, choice of the runtime library and of the number of epochs.

leaves the safe environment of the WN, and only the model's weights are shared between the virtual machines. The MN is hosted at Humanitas Research Hospital, and the 3 WNs are located at Humanitas, Universität Leipzig, and the University of Bologna.

B. Model implementation with Flower

The FL implementation is based on Flower library [19], [20]. Flower is a FL framework developed by a research group at the University of Cambridge. It is a Python object-oriented library, compatible with the most common ML libraries, such as Scikit-learn [21], Tensorflow [22], Pytorch [23], etc. Its main classes allow the development of a FL model and its corresponding client and server instances. The FL model is characterized by some functions, i.e. $set\_weight()$, $get\_weight()$, $fit()$ and $evaluate()$, controlling the main FL actions: setting network parameters, retrieving weights from the network, training the network on the dataset and evaluating its performance on a test dataset. Several tutorials and a large documentation are provided on the Flower website [19]. The models under study have been implemented through the Pytorch library [23].

C. Models

1) Cox model: The Cox Proportional Hazards (PH) model is the most popular method to analyze survival data [24], and it is mathematically described as:

$$h(t|X_i) = h_0(t) \cdot exp(\beta_i \cdot X_i) \tag{2}$$

where $h_0(t)$ is the baseline hazard function and β_i are the regression coefficients. They can be optimized by the minimization of the partial log-likelihood:

$$log(PL(\beta)) = \sum_{i=1}^{N} \beta_i X_i - \sum_{i=1}^{N} log\Big(\sum_{j \in R(T_i)} exp(\beta_j X_j) \Big) \tag{3}$$

2) DeepSurv: DeepSurv provides a nonlinear model for survival analysis using a deep feedforward neural network, where the output is a linear activation function that estimates the log-risk function of the Cox model [25]. The cost function is a negative partial log likelihood, which is optimized at each training step:

$$l(\theta) = -\frac{1}{N_{E=1}} \sum_{i:E_i=1} \Big(\hat{h_\theta}(x_i) - log \sum_{j \in R(T_i)} exp(\hat{h_\theta}(x_j)) \Big) \tag{4}$$

where θ are the weights of the network, $N_{E=1}$ is the number of patients with an observed event and $\hat{h_\theta}(x_i)$ is the output of the network.

D. Application to a clinical dataset

1) Myelodysplastic Syndromes: Myelodysplastic Syndromes (MDS) are a group of hematological cancers characterized by the failure of the bone marrow stem cells to mature into normal functioning blood cells, with a risk of progression into acute myeloid leukaemia [26]. It occurs mainly in elderly people and the median age at diagnosis is about 65-70 years, while less than 10% of patients are under 50 years [27]. MDS are still diagnosed by examining blood and bone marrow, which shows blood cytopenias and hypercellular marrow with dysplasia, with or without excess of marrow immature cells [27]. International Prognostic Scoring System (IPSS) [28] and its revised version (IPSS-R) [29] can be used to assess disease risk. These prognostic systems are mainly based on clinical and hematological parameters, like the percentage of blasts in the bone marrow, the presence of cytogenetic abnormalities, the number and the extent of cytopenias [30]. The IPSS-R defines 5 risk groups based on a numbered point value for each prognostic variable that is summed to attain a total score ranging from 0 to 10 [31]. However, these scoring systems have some limitations, and they can fail in capturing crucial prognostic information [30].

MDS pathophysiology is a multistep process involving both gene mutations and cytogenetic changes [27]. The most frequent mutations occur in three chromatin-related genes: *DNMT3A*, *TET2*, and *ASXL1*. Other commonly mutated genes are *SF3B1*, *SRSF2*, *U2AF1*, and *ZRSR2*, but also *RUNX1*,

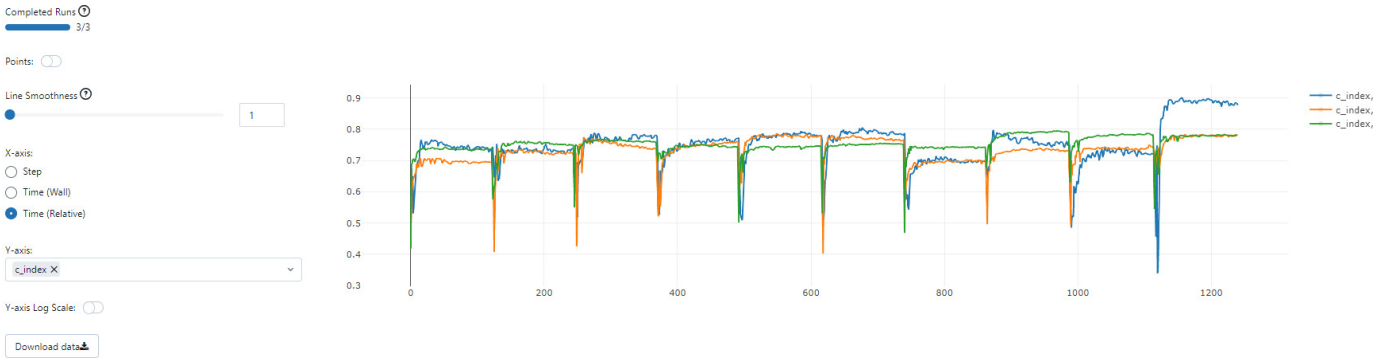


Fig. 3: Screenshot of the MLflow-based user interface. The C-index trends for the three federated institutions during the training phase are displayed.

TP53, *STAG2* [30]. In addition, complex karyotypes can be observed in MDS patients, like del(5q), del(11q), del(9q) or del(12p) [26]. Treatment of MDS patients has improved in the past few years, ranging from symptomatic treatment of cytopenia (e.g. by transfusions) to allogeneic stem-cell transplantation [27].

2) Dataset: The entire cohort of MDS patients is formed by joining two different datasets: 1) a retrospective cohort of 2043 patients collected by GenoMed4All [18], [31] and Synthema [32] consortia; 2) a publicly available dataset of 2384 patients by the International Working Group for the study of Prognosis in MDS [33], [34]. Inclusion criteria were age $\geq$ 18 years, diagnosis of MDS by WHO 2016 [35]–[37] criteria, and information available on demographics, clinical features, mutational screening/chromosomal abnormalities, treatment, and survival. Humanitas Ethics Committee approved the study (ClinicalTrials.gov Identifier: NCT04889729). Written informed consent was obtained from each participant [33]. Each patient is characterized by 64 features, which can be grouped into point mutations, karyotype abnormalities and clinical data. In addition, time-to-event, censoring status and prognostic indexes (IPSS and IPSS-R) are provided.

The MDS dataset has been divided among the 3 nodes, named Node1, Node2 and Node3, with 2656, 1328 and 443 patients, respectively.

3) FL scenarios: Five different scenarios have been designed to be tested in both an isolated and a federated framework:

a Best scenario: all 64 features are present.
b Random scenario: random removal of a specific percentage of mutation variables:
  – 10% of mutation variables are removed
  – 30% of mutation variables are removed
  – 60% of mutation variables are removed
c Worst scenario: a list of 12 karyotype variables and 16 mutations are removed.

The total number of available features for each scenario is shown in Tab. I.

The design of the scenarios described above is not arbitrary, but it has been chosen to represent realistic situations, taking into account which features are often not available. These scenarios allow us to test how FL affects the performance of the models trained on datasets of different sizes. Moreover, they will enable us to study how the number of features affects model performance and how FL can influence it.

4) Training and performance evaluation: Federated models have been trained through a 10-fold cross-validation (CV) procedure, each lasting 100 epochs, which are sufficient to allow the model performance to reach stability. The 10-fold CV approach has been chosen to have a balance between training, validation and testing data. Every 5 epochs, federated aggregation is applied. Each local dataset has been split into training, validation and test sets (72%, 8% and 20% of the total amount of local data, respectively). Each FL experiment has been performed in both the FLsimul and the FLreal settings, using the 3 nodes through the GenoMed4All platform.

Model performance has been assessed using the Harrell's Concordance index (C-index), which is based on the idea that an individual observed to have an earlier time-to-event compared to other individuals would ideally be assigned a higher predicted risk score from the candidate survival model, and it can deal with censored data [38], [39]. Let's consider a pair of patients, i and j, whose time-to-event variables are denoted as T_i and T_j, respectively. Let us introduce a status variable δ_i: if $\delta_i = 1$, the patient had the event, while $\delta_i = 0$ otherwise. The survival model can predict a risk score γ_i for each patient, based on their covariates. If the survival model is good, a higher risk score is expected to be assigned to patients with shorter time-to-event. The C-index is computed as follows:

$$C - index = \frac{\sum_{i \neq j} 1_{T_j < T_i} \cdot 1_{\gamma_j > \gamma_i} \cdot \delta_j}{\sum_{i \neq j} 1_{T_j < T_i} \cdot \delta_j} \tag{5}$$

Basically, the C-index is the ratio between concordant pairs and the sum of concordant and discordant pairs. A C-index lower than 0.5 means that it is better to conclude the opposite of what the estimated risk score states. If the C-index is near 1, it means that the model performs well in estimating risk scores.

TABLE I: NUMBER OF AVAILABLE FEATURES FOR EACH FEDERATED SCENARIO.

Scenario	Best	Random 10%	Random 30%	Random 60%	Worst
Features	64	62	59	54	36

TABLE II: COX AND DEEPSURV PERFORMANCE FOR ALL THE SCENARIOS IN THE CENTRALIZED TRAINING SETTING (I.E. ALL DATA MERGED).

Model	Best	Random 10%	Random 30%	Random 60%	Worst
Cox	0.737± 0.002	0.738± 0.001	0.737± 0.002	0.740± 0.002	0.737± 0.002
DeepSurv	0.728± 0.004	0.728± 0.004	0.726± 0.004	0.727± 0.004	0.729± 0.003

III. Results

A. Centralized training

Both Cox and DeepSurv have been trained in a centralized framework, where the three local datasets have been joined into a unique one with 4427 samples. Ideally, the models' performance on the centralized dataset should be the best. Centralized C-indexes for all the scenarios are reported in Tab. II.

B. Best scenario

Cox and DeepSurv training curves in the best scenario are shown in Fig. 4 and Fig. 5, respectively. Training curves show performance behaviour in terms of C-index for 100 epochs of training. Centralized, isolated learning and FLsimul, Fig. 4a, are coloured in green, blue and orange, respectively. Each node is represented with a different line style. As expected, the third node suffers from a lower performance in the isolated training, due to its small dataset size. The aggregating weights allows it to increase its performance, making it similar to the other two. However, after federated averaging, Node3 performance drops rapidly. Nevertheless, its average performance is higher than the isolated one. Node1 and Node2 do not show significant differences between isolated and federated training. FLreal results are shown in Fig. 4b. Here, Node1, Node2 and Node3 are coloured in orange, blue and green, respectively. Observing DeepSurv in FLsimul, Fig. 5a, it is possible to notice that, after federated aggregation, Node3 continues to benefit from parameters' sharing, as its performance remains above a mean C-index of 0.76. The FLreal results are shown in Fig. 5b. Again, Node2 and Node3 have a similar performance in isolated and federated settings. The final c-indexes for the three training approaches are reported in Tab. III and Tab. IV for Cox and DeepSurv, respectively.

C. Random scenario

Random scenarios are characterized by missing data at different percentages (10%, 30% and 60%). A general observation could be that isolated training performance shows larger oscillations with respect to the best scenarios for both Cox and DeepSurv, Fig. 6a and Fig. 7a, respectively. Confidence intervals are larger. In Fig. 6a and Fig. 7a, Node3's C-indexes are lower than those of Node1 and Node2. In addition, the Node3 C-index confidence intervals have greater ranges. The same observation is valid for random 30% (Fig. 8a and Fig. 9a) and 60% (Fig. 10a and Fig. 11a). FLreal experiments mitigate these huge variations.

The Cox model of Node3 shows a significant performance drop after federated averaging. Furthermore, in a random 10% scenario, the DeepSurv model of Node2 encounters high oscillations (Fig. 7b). In random 30% and 60% scenarios, the DeepSurv model of Node1 shows large in-width confidence intervals, indicating the need for further investigation (Fig. 9b and Fig. 11b, respectively).

D. Worst scenario

Finally, the performance of Cox and DeepSurv trained on the worst scenario nodes is shown in Fig. 12 and Fig. 13, respectively. Isolated training, Fig. 12a and Fig. 13a, is characterized by poor performance for Node3, whose mean C-index is 0.62 and 0.64 for Cox and DeepSurv models, respectively. However, as in the previous cases, federated aggregation improves the C-index. Moreover, DeepSurv of Node2 benefits from weights averaging, as its mean C-index increases from 0.69 up to 0.76 in both FLsimul and FLreal, Fig. 13b. No significant improvement is observed for Node1.

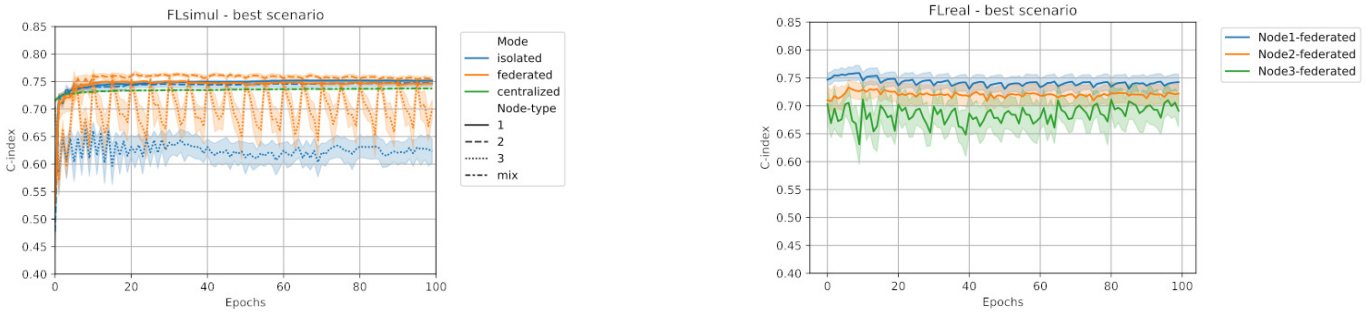
IV. Discussion

The performance of Cox and DeepSurv models trained on different scenarios in isolated, FLsimul and FLreal settings is reported in Tab. III and Tab. IV, respectively. Each C-index value has been computed as the mean of the 10 C-indexes at epoch number 100 for each CV fold. The uncertainty has been estimated with the 95% confidence interval.

In general, Node3, corresponding to the dataset of the University of Bologna, is the one that always benefits from federated aggregation of the neural network weights. Due to its smaller dataset size (443 samples), its local models cannot reach an average performance higher than 0.66. After federated aggregation, a significant improvement allows reaching an average C-index of 0.76 for DeepSurv.

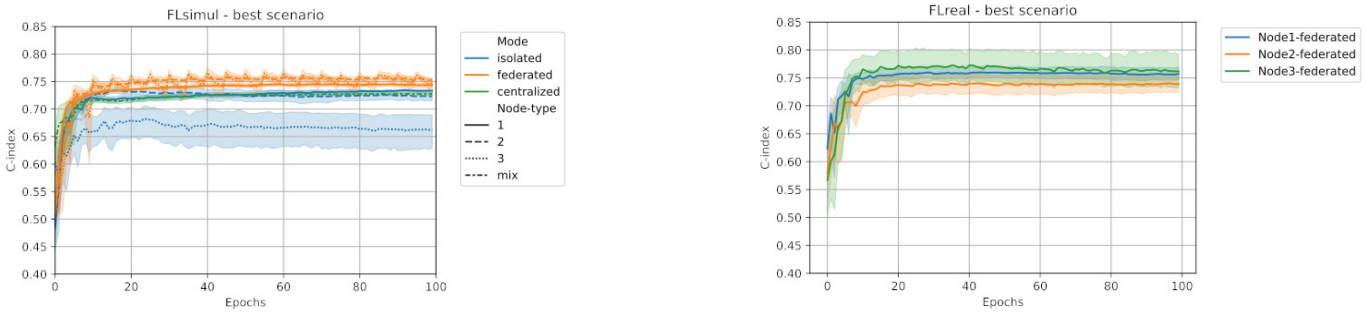
However, Cox model shows a rapid drop in its performance, reaching an average C-index of about 0.69. It is interesting to note that, in most federated settings, DeepSurv of Node3 has the best performance with respect to the other clients. Furthermore, it can also be noticed that Node3 is the one that shows larger in-width confidence intervals, meaning that, despite weights aggregation, its model performance is prone to oscillations.

Larger confidence intervals, as observed for Node3, indicate variability in performance, that implies to obtain high and low C-indexes depending on different input datasets during the CV procedure. This behaviour is detectable for small datasets after isolated learning, where local empirical distributions may greatly vary from the global distribution [40]. Generally, the difference between empirical and true distribution exponentially decreases with the sample size, but for small samples



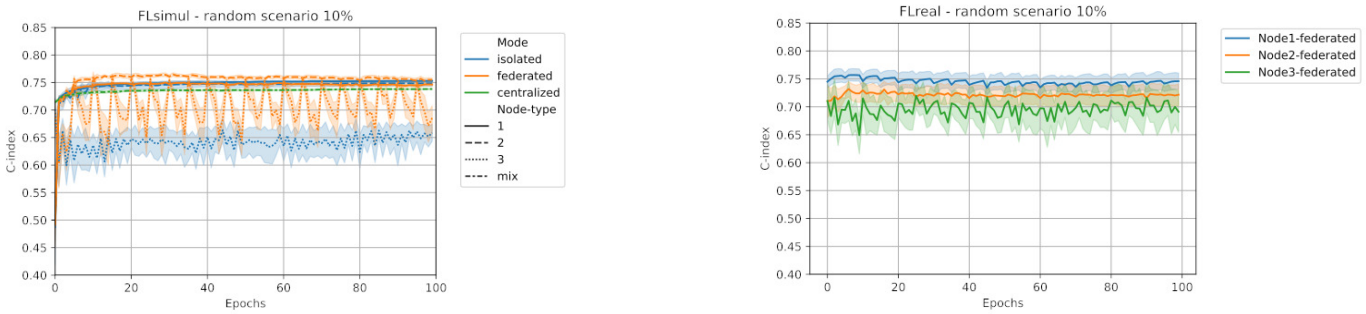
(a) (b)

Fig. 4: Cox training curves for the three nodes for best scenario. (a) Curves for centralized (green), isolated (blue) and FLsimul learning (orange) are displayed. The linestyle defines the node type (Node1, Node2, Node3 or centralized). (b) FLreal learning C-index curves for Node1 (orange), Node2 (blue) and Node3 (green).



(a) (b)

Fig. 5: DeepSurv training curves for the three nodes for best scenario. (a) Curves for centralized (green), isolated (blue) and FLsimul learning (orange) are displayed. The linestyle defines the node type (Node1, Node2, Node3 or centralized). (b) FLreal learning C-index curves for Node1 (orange), Node2 (blue) and Node3 (green).



(a) (b)

Fig. 6: Cox Training curves for the three nodes for random 10% scenario. (a) Curves for centralized (green), isolated (blue) and FLsimul learning (orange) are displayed. The linestyle defines the node type (Node1, Node2, Node3 or centralized). (b) FLreal learning C-index curves for Node1 (orange), Node2 (blue) and Node3 (green).

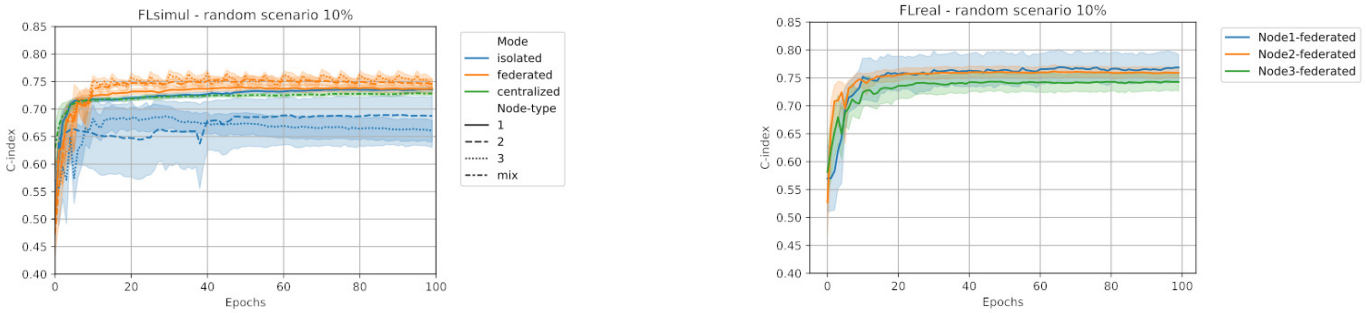
the difference can be substantial if the distribution differs from a Normal one [41]. The federated aggregation reduces that variability because the weights' averaging allows sharing the properties of other large datasets, improving also the model reliability.

Interestingly, DeepSurv of Node2 suffers from feature missingness in all three random scenarios. Moreover, despite the lower number of available features, it has a better performance in the worst scenario, where it reaches an average C-index of 0.69.

Similarly, Node1, which has the larger volume of the dataset, performs increasingly on the three random settings. However, the difference is not significant as it is within confidence intervals. In addition, Node1 maintains the same average performance (C-index of 0.75 and 0.73 for Cox and DeepSurv, respectively) among all the isolated training scenarios.

Depending on the number of available features, both random and worst scenarios require more training epochs to reach higher C-index values. The choice of 5 as the frequency of federated aggregation came out after trying several values (5, 10 and 20) and choosing the larger than allowed to reach an optimal performance.

Another crucial outcome of these experiments is the com-



(a) (b)

Fig. 7: DeepSurv training curves for the three nodes for random 10% scenario. (a) Curves for centralized (green), isolated (blue) and FLsimul learning (orange) are displayed. The linestyle defines the node type (Node1, Node2, Node3 or centralized). (b) FLreal learning C-index curves for Node1 (orange), Node2 (blue) and Node3 (green).

TABLE III: COX MODEL PERFORMANCE FOR ALL THE SCENARIOS IN ISOLATED, FLSIMUL AND FLREAL TRAINING SETTING.

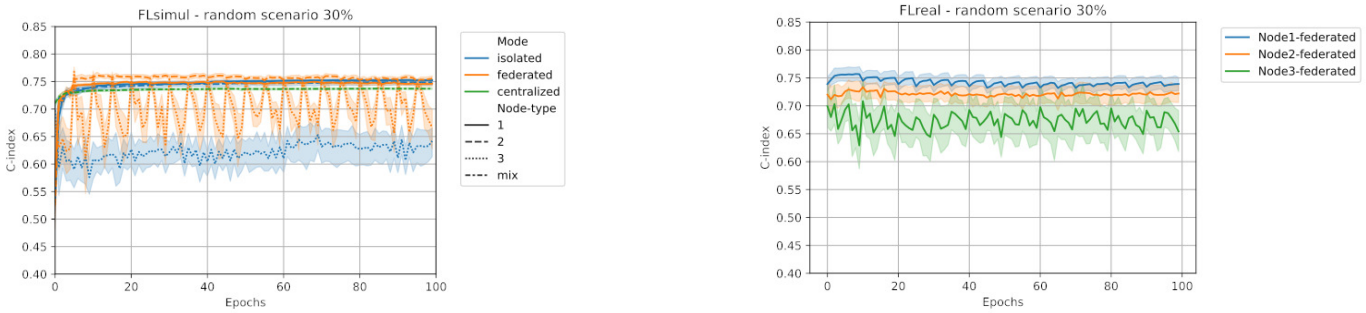
Scenario	Nodes	Samples	C-index isolated	C-index FLsimul	C-index FLreal
Best	Node1	2656	0.751 ± 0.003	0.746 ± 0.004	0.72 ± 0.02
	Node2	1328	0.747 ± 0.008	0.756 ± 0.005	0.74 ± 0.02
	Node3	443	0.62 ± 0.03	0.70 ± 0.02	0.69 ± 0.04
Random (10%)	Node1	2656	0.753 ± 0.002	0.746 ± 0.003	0.72 ± 0.02
	Node2	1328	0.748 ± 0.007	0.755 ± 0.004	0.75 ± 0.02
	Node3	443	0.66 ± 0.02	0.69 ± 0.03	0.69 ± 0.05
Random (30%)	Node1	2656	0.752 ± 0.002	0.746 ± 0.005	0.72 ± 0.02
	Node2	1328	0.748 ± 0.007	0.755 ± 0.007	0.74 ± 0.02
	Node3	443	0.64 ± 0.03	0.67 ± 0.05	0.65 ± 0.05
Random (60%)	Node1	2656	0.74 ± 0.02	0.747 ± 0.004	0.72 ± 0.03
	Node2	1328	0.744 ± 0.008	0.759 ± 0.007	0.74 ± 0.02
	Node3	443	0.64 ± 0.05	0.68 ± 0.04	0.67 ± 0.04
Worst	Node1	2656	0.74 ± 0.03	0.742 ± 0.006	0.72 ± 0.02
	Node2	1328	0.754 ± 0.006	0.767 ± 0.006	0.75 ± 0.02
	Node3	443	0.62 ± 0.05	0.65 ± 0.07	0.69 ± 0.02

TABLE IV: DEEPSURV MODEL PERFORMANCE FOR ALL THE SCENARIOS IN ISOLATED, FLSIMUL AND FLREAL TRAINING SETTING.

Scenario	Nodes	Samples	C-index isolated	C-index FLsimul	C-index FLreal
Best	Node1	2656	0.733 ± 0.003	0.744 ± 0.005	0.74 ± 0.02
	Node2	1328	0.72 ± 0.01	0.752 ± 0.006	0.76 ± 0.01
	Node3	443	0.66 ± 0.04	0.75 ± 0.01	0.76 ± 0.04
Random (10%)	Node1	2656	0.735 ± 0.006	0.74 ± 0.01	0.76 ± 0.01
	Node2	1328	0.69 ± 0.06	0.75 ± 0.01	0.77 ± 0.03
	Node3	443	0.66 ± 0.03	0.75 ± 0.01	0.74 ± 0.02
Random (30%)	Node1	2656	0.71 ± 0.06	0.741 ± 0.006	0.74 ± 0.02
	Node2	1328	0.66 ± 0.08	0.718 ± 0.08	0.76 ± 0.02
	Node3	443	0.69 ± 0.02	0.72 ± 0.06	0.76 ± 0.04
Random (60%)	Node1	2656	0.734 ± 0.007	0.74 ± 0.05	0.74 ± 0.02
	Node2	1328	0.71 ± 0.01	0.71 ± 0.07	0.76 ± 0.02
	Node3	443	0.64 ± 0.06	0.70 ± 0.07	0.76 ± 0.04
Worst	Node1	2656	0.726 ± 0.09	0.740 ± 0.004	0.73 ± 0.02
	Node2	1328	0.69 ± 0.07	0.754 ± 0.009	0.76 ± 0.01
	Node3	443	0.64 ± 0.03	0.75 ± 0.01	0.75 ± 0.04

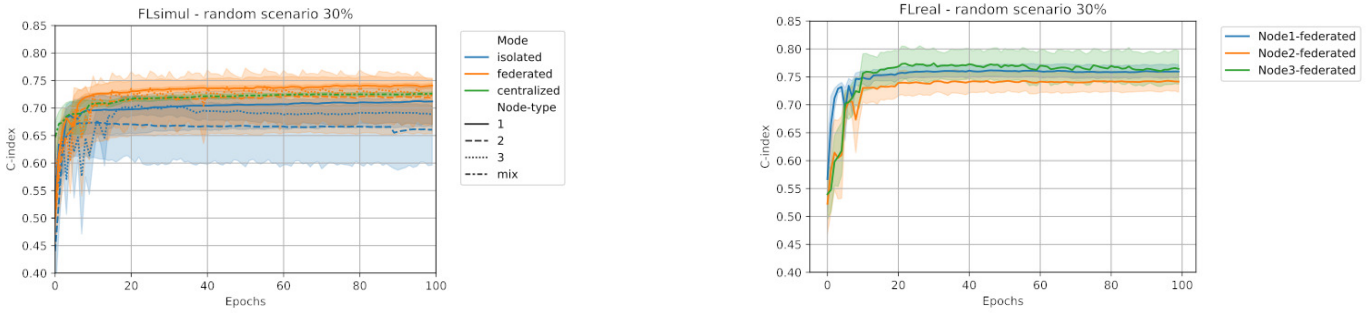
parison between results from FLsimul and FLreal tests, performed with the GenoMed4All platform. Both Cox and DeepSurv models have been initialized with the same hyperparameters in both FLsimul and FLreal settings, except for initial weights. Training phases have the same number of epochs and the same federated aggregation frequency. From the C-index estimates in Tab. III and Tab. IV, it is possible to assess that the two approaches agree. Differences fall within confidence intervals and it can be justified with different initialization weights. This result makes the GenoMed4All platform reliable from the point of view of the obtained outcomes. Therefore, it is possible to use this platform for FL experiments without resorting to control simulations, knowing that the obtained results are reliable. Moreover, other FL experiments involving different models and tasks (e.g. clustering) can be done. Finally, it could facilitate the application of this approach to different kinds of diseases, like other oncohematological conditions or neurodegenerative disorders, which commonly belong to rare pathologies.

While the present study provides valuable insights into FL of survival models in healthcare applications, it is necessary to mention some limitations. Firstly, the use of only three clients



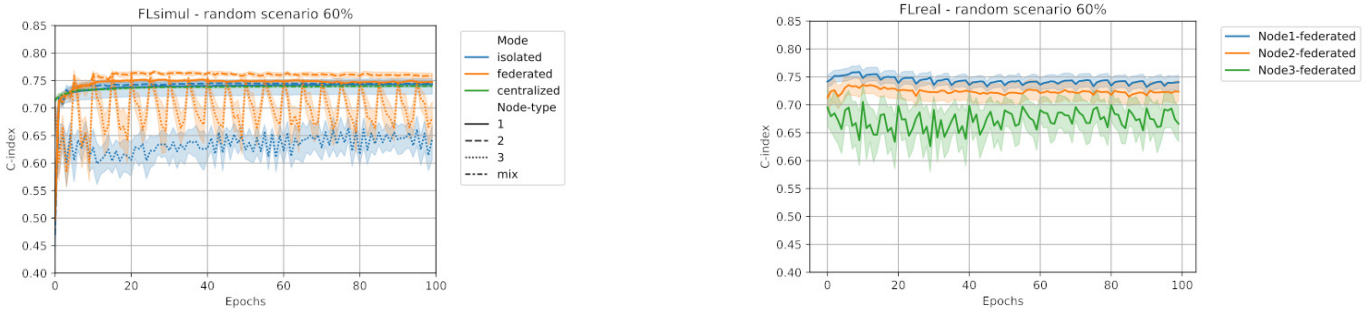
(a) (b)

Fig. 8: Cox training curves for the three nodes for random 30% scenario. (a) Curves for centralized (green), isolated (blue) and FLsimul learning (orange) are displayed. The linestyle defines the node type (Node1, Node2, Node3 or centralized). (b) FLreal learning C-index curves for Node1 (orange), Node2 (blue) and Node3 (green).



(a) (b)

Fig. 9: DeepSurv training curves for the three nodes for random 30% scenario. (a) Curves for centralized (green), isolated (blue) and FLsimul learning (orange) are displayed. The linestyle defines the node type (Node1, Node2, Node3 or centralized). (b) FLreal learning C-index curves for Node1 (orange), Node2 (blue) and Node3 (green).

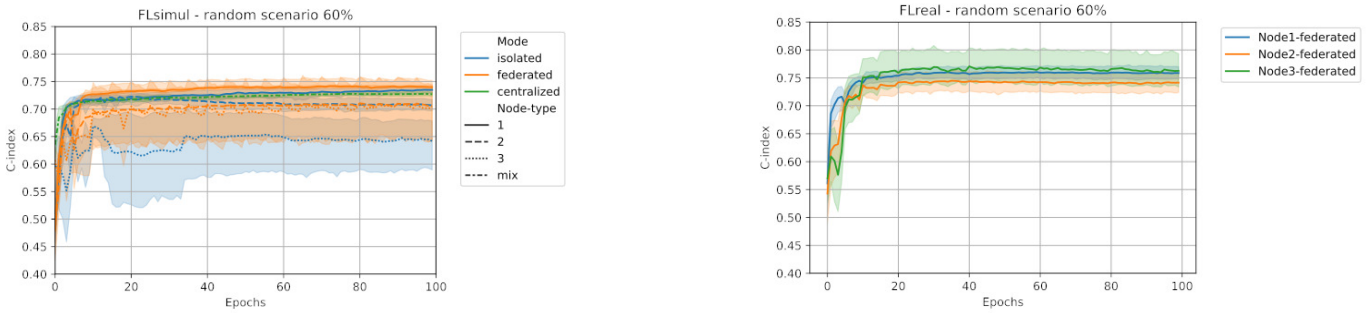


(a) (b)

Fig. 10: Cox training curves for the three nodes for random 60% scenario. (a) Curves for centralized (green), isolated (blue) and FLsimul learning (orange) are displayed. The linestyle defines the node type (Node1, Node2, Node3 or centralized). (b) FLreal learning C-index curves for Node1 (orange), Node2 (blue) and Node3 (green).

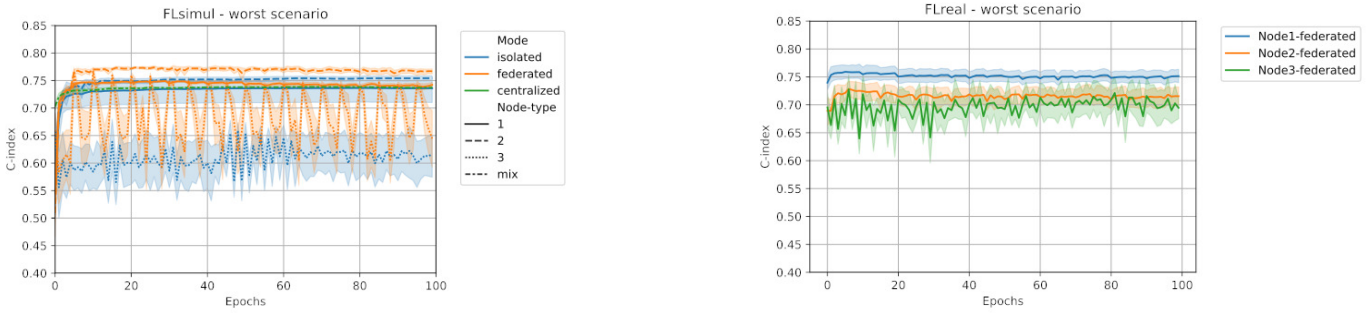
does not allow us to fully simulate the potential of a robust FL scenario. This is indeed a choice made by the GenoMed4All consortium, so at the moment there are only three WNs. Secondly, the study involves relatively small datasets. Testing the infrastructure and model scalability to large-scale datasets would provide more insights into performance and algorithm efficiency in dealing with higher levels of data heterogeneity and data imbalances. In addition, the trade-off between privacy of sensitive medical data and model performance should be investigated, in particular for what concerns the aggregation process. All these limitations should be addressed in further analyses.

Finally, the platform is powerful and secure, but still has room for improvement: working with pickle files can be difficult and cumbersome for newcomers. Furthermore, during the training phase, the different weights of the models must be collected from the various WNs: this procedure can be faster or slower according to the amount of data available, the virtual machine specifications and the network connectivity of the node. The platform has currently been validated on the MDS use case, but in the near future it will be tested on other different hematological diseases and data types. Last but not least, the platform currently implements the FedAvg algorithm, but in the future we plan to apply other different



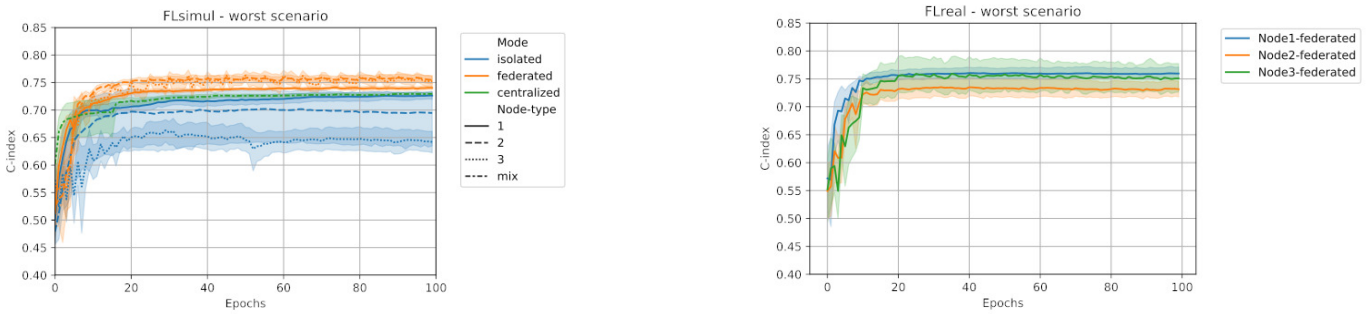
(a) (b)

Fig. 11: DeepSurv training curves for the three nodes for random 60% scenario. (a) Curves for centralized (green), isolated (blue) and FLsimul learning (orange) are displayed. The linestyle defines the node type (Node1, Node2, Node3 or centralized). (b) FLreal learning C-index curves for Node1 (orange), Node2 (blue) and Node3 (green).



(a) (b)

Fig. 12: Cox training curves for the three nodes for worst scenario. (a) Curves for centralized (green), isolated (blue) and FLsimul learning (orange) are displayed. The linestyle defines the node type (Node1, Node2, Node3 or centralized). (b) FLreal learning C-index curves for Node1 (orange), Node2 (blue) and Node3 (green).



(a) (b)

Fig. 13: DeepSurv training curves for the three nodes for worst scenario. (a) Curves for centralized (green), isolated (blue) and FLsimul learning (orange) are displayed. The linestyle defines the node type (Node1, Node2, Node3 or centralized). (b) FLreal learning C-index curves for Node1 (orange), Node2 (blue) and Node3 (green).

types of weights aggregation methods available, considering the quantity and distribution of the data across nodes.

V. Conclusion

Federated Learning is a powerful approach for the decentralized training of ML models. Sharing and aggregating model parameters instead of data leads to an improvement in the overall performance, especially when models are trained on small-sized dataset. This study shows a federated implementation of two standard survival models, Cox and DeepSurv. The federated training procedure has been performed both through Python code simulations (FLsimul) and exploiting the GenoMed4All platform (FLreal). Experimental outcomes confirm the benefits of weights aggregation, especially for Node3, which is the client characterized by the smallest dataset (443 samples). Moreover, FL improves model performance when the dataset is affected by feature missingness. Finally, the experiments assessed the reliability of the GenoMed4All platform, as its results agree with those of the simulated ones. This finding allows the possibility of using this process for other different tasks, such as federated embedding and clustering, extending the application to other rare diseases.

REFERENCES

[1] N. Rieke, et al., "The future of digital health with federated learning," *npj Digital Medicine*, vol. 3, September 2020.

[2] Z. Huang, et al., "SALMON: Survival Analysis Learning With Multi-Omics Neural Networks on Breast Cancer," *Frontiers in genetics*, vol. 10, March 2019.

[3] L. Tong, J. Mitchel, K. Chatlin, and M.D. Wang, "Deep learning based feature-level integration of multi-omics data for breast cancer patients survival analysis," *BMC medical informatics and decision making*, vol. 20, September 2020.

[4] N. K. Mishra, S. Southekal, and C. Guda, "Survival Analysis of Multi-Omics Data Identifies Potential Prognostic Markers of Pancreatic Ductal Adenocarcinoma," *Frontiers in genetics*, vol. 10, July 2019.

[5] L. Zhao, et al., "DeepOmix: A scalable and interpretable multi-omics deep learning framework and application in cancer survival analysis," *Computational and structural biotechnology journal*, vol. 19, pp. 2719—2725, May 2021.

[6] Y.LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436—444, May 2015.

[7] A. Shrestha, and A. Mahmood, "Review of Deep Learning Algorithms and Architectures," *IEEE Access*, vol. 7, pp. 53040–53065, April 2019.

[8] M. Andreux, A. Manoel, R. Menuet, C. Saillard, and C. Simpson, "Federated survival analysis with discrete-time cox models," June 2020, arXiv:2006.08997 [Online]. Available: http://doi.org/10.48550/arXiv.2006.08997.

[9] A. Archetti, F. Ieva, and M. Matteucci, "Scaling survival analysis in healthcare with federated survival forests: A comparative study on heart failure and breast cancer genomics," *Future Generation Computer Systems*, vol. 149, pp. 343–358, December 2023.

[10] C. Rollo, et al., "SYNDSURV: A simple framework for survival analysis with data distributed across multiple institutions," *Computers in Biology and Medicine*, vol. 172, April 2024.

[11] M. Shah, and N. Sureja, "A Comprehensive Review of Bias in Deep Learning Models: Methods, Impacts and Future Directions," *Arch Computat Methods Eng*, May 2024.

[12] M. J. Sheller, et al., "Federated learning in medicine: facilitating multi-institutional collaborations without sharing patient data," *Scientific reports*, vol. 10, July 2020.

[13] J. Xu, et al., "Federated Learning for Healthcare Informatics," *Journal of Health-care Informatics Research*, vol. 5, pp. 1—19, November 2021.

[14] M. Aledhari, R. Razzak, R. M. Parizi, and F. Saees, "Federated Learning: A Survey on Enabling Technologies, Protocols, and Applications," *IEEE Access*, vol. 8, pp. 140699–140725, July 2020.

[15] P. Qi, et al., "Model aggregation techniques in federated learning: A comprehensive survey," *Future Generation Computer Systems*, vol. 150, pp. 272—293, January 2024.

[16] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. Arcas, "Communication-efficient learning of deep networks from decentralized data," *Artificial Intelligence and Statistics*, vol. 54, pp. 1273–1282, April 2017.

[17] J. Wang, Q. Liu, H. Liang, G. Joshi, and V. Poor, "Tackling the Objective Inconsistency Problem in Heterogeneous Federated Optimization," *Advances in neural information processing systems*, vol. 33, pp. 7611–7623, July 2020.

[18] F. Cremonesi, et al., "The need for multimodal health data modeling: A practical approach for a federated-learning healthcare platform," *Journal of Biomedical Informatics*, vol. 141, May 2023.

[19] "Flower A Friendly Federated Learning Framework". [Online]. Available: http://flower.ai/.

[20] D. Beutel, et al. "Flower: A Friendly Federated Learning Research Framework," 2020, arXiv:2007.14390. [Online]. Available: http://doi.org/10.48550/arXiv:2007.14390.

[21] "Scikit-learn Machine Learning in Python". [Online]. Available: http://scikit-learn.org/stable/.

[22] "Tensorflow An end-to-end platform for machine learning". [Online]. Available: http://www.tensorflow.org/.

[23] "Pytorch". [Online]. Available: http://pytorch.org/.

[24] D. R. Cox, "Regression Models and Life-Tables," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 34, pp. 187–220, 1972.

[25] J. L. Katzman, et al., "DeepSurv: personalized treatment recommender system using a Cox proportional hazards deep neural network," *BMC Medical Research Methodology*, vol. 18, pp. 1—12, February 2018.

[26] A. M. Sekeres, and J. Taylor, "Diagnosis and Treatment of Myelodysplastic Syndromes A Review," *Journal of American Medical Association*, vol. 328, pp. 872–880, September 2022.

[27] L. Adès, R. Itzykson, and P. Fenaux, "Myelodysplastic syndromes," *The Lancet*, vol. 383, pp. 2239–2252, June 2014.

[28] P. Greenberg, et al., "International scoring system for evaluating prognosis in myelodysplastic syndromes," *Blood*, vol. 89, pp. 2079–2088, March 1997.

[29] P. Greenberg, et al., "Revised international prognostic scoring system for myelodysplastic syndromes," *Blood*, vol. 120, pp. 2454–2465, September 2012.

[30] C. Chiereghin, et al., "The Genetics of Myelodysplastic Syndromes: Clinical Relevance," *Genes*, vol. 12, July 2021.

[31] "GenoMed4All: Genomics for next generation healthcare". [Online]. Available: http://www.genomed4all.eu.

[32] "Synthema: Synthetic Haematological Data". [Online]. Available: http://www.synthema.eu.

[33] S. D'Amico, et al., "MOSAIC: An Artificial Intelligence–Based Framework for Multimodal Analysis, Classification, and Personalized Prognostic Assessment in Rare Cancers," *JCO Clinical Cancer Informatics*, vol. 8, June 2024.

[34] E. Bernard, et al., "Molecular International Prognostic Scoring System for Myelodysplastic Syndromes," *NEJM Evidence*, vol. 1, June 2022.

[35] D. A. Arber, et al., "The 2016 revision to the World Health Organization classification of myeloid neoplasms and acute leukemia," *Blood*, vol. 127, pp. 2391—2405, May 2016.

[36] L. Malcovati, et al., "Prognostic Factors and Life Expectancy in Myelodysplastic Syndromes Classified According to WHO Criteria: A Basis for Clinical Decision Making," *Journal of Clinical Oncology*, vol. 23, pp. 7594—7603, October 2005.

[37] M.G. Della Porta, et al., "Minimal morphological criteria for defining bone marrow dysplasia: a basis for clinical implementation of WHO classification of myelodysplastic syndromes," *Leukemia*, vol. 29, pp. 66–75, January 2015.

[38] F. E. Harrell, R.M. Califf, D.B. Pryor, K.L. Lee, and R.A. Rosati, "Evaluating the Yield of Medical Tests," *JAMA*, vol. 247, pp. 2543–2546, May 1982.

[39] M. Schmid, M.N. Wright, and A. Ziegler, "On the use of Harrell's C for clinical risk prediction via random survival forests," *Expert Systems with Applications*, vol. 63, pp. 450–459, November 2016.

[40] M. Kamp, J. Fischer, and J. Vreeken, "Federated Learning from Small Datasets," 2021, arXiv,2110.03469. [Online]. Available: http://arxiv.org/abs/2110.03469.

[41] S. G. Kwak, J. H. Kim, "Central limit theorem: the cornerstone of modern statistics," *Korean J Anesthesiol*, vol. 70, pp. 144–156, April 2017.