Discover Artificial Intelligence

RESEARCH Open Access

Deep learning models for the analysis of high-dimensional survival data with time-varying covariates while handling missing data

Sarah Ogutu[1*], Mohanad Mohammed[1] and Henry Mwambi[1]

*Correspondence:
Sarah Ogutu
ogutusarah@gmail.com
[1]School of Mathematics, Statistics & Computer Science, University of KwaZulu-Natal, Pietermaritzburg, South Africa

Abstract

Recent advances in deep learning have expanded the potential for predictive modeling in survival analysis, particularly in high-dimensional datasets with time-varying covariates. This paper applies deep learning approaches, DeepSurv, DeepHit, and Dynamic DeepHit, to model HIV incidence (time-to-event outcome) using high-dimensional longitudinal data, incorporating time-varying cytokine profiles alongside baseline covariates. We employ the time-dependent concordance index (C-index) and Brier scores to assess the models' predictive accuracy. We also address missing data using missForest, evaluating model performance on imputed and complete-case datasets. Different strategies for integrating cytokine profiles were explored: DeepSurv and DeepHit utilized derived variables, mean, and difference between the first and last measurements, while Dynamic DeepHit preserved the original time-varying nature of the cytokine data. Our findings demonstrate that retaining the dynamic nature of cytokine covariates, rather than relying on derived summary measures, underscores the robustness and suitability of Dynamic DeepHit as a clinical prediction model, particularly in scenarios where key variables evolve over time.

Keywords DeepSurv, DeepHit, Dynamic DeepHit, HIV incidence, Cytokine profiles, Time-varying covariate, MissForest

1 Introduction

Survival analysis plays a critical role in biomedical research, particularly in understanding the time to event for diseases such as HIV [1]. The emergence of high-throughput technologies has led to the generation of increasingly high-dimensional or ultra-high-dimensional clinical data [2]. As a result, effectively analyzing this data has become a substantial challenge. Traditional methods like the Cox Proportional Hazards (PH) model, its regularized extensions (e.g., Lasso-Cox, Ridge-Cox), and the time-dependent Cox have been widely used [3]. Although they can handle high-dimensional and time-varying covariates reasonably well, these approaches face limitations in scenarios involving complex non-linear interactions and intricate temporal dynamics [4]. Regularized Cox models assume linear or additive covariate effects, which may fail to

capture intricate biological interactions [5]. While the time-dependent Cox models can incorporate time-varying features, they require explicit specification of time-covariate interactions and often struggle with high-dimensional longitudinal data (e.g., cytokine trajectories with irregular measurements) [6, 7]. The standard Cox models require stratification or cause-specific hazards for competing events, which can be inefficient for multi-event outcomes [8, 9]. These limitations can result in suboptimal predictions in settings with complex biological interactions, where deep learning techniques could leverage their capacity to model such relationships more effectively [10].

Deep learning survival models, including DeepSurv, DeepHit, and Dynamic DeepHit, offer advanced methodologies that address these challenges by leveraging the flexibility of neural networks. These models can capture nonlinear interactions between covariates and time-to-event outcomes, and some can handle time-varying data, making them well-suited for the analysis of high dimensional data with time-dependent covariates [11]. DeepSurv extends the traditional Cox PH model by replacing the linear predictor with a neural network, making it more flexible in capturing intricate patterns in the data [12]. DeepHit, on the other hand, is designed to predict single or multiple competing risks and events by directly modeling the joint distribution of time-to-event and event type [13]. Dynamic DeepHit builds upon DeepHit by incorporating time-varying covariates, making it particularly useful for datasets with longitudinal information, such as cytokine profiles, where measurements change over time [14].

This paper explores the performance of DeepSurv, DeepHit, and Dynamic DeepHit applied to high-dimensional survival data with time-varying covariates alongside baseline covariates. The goal in the application is to predict HIV incidence, explicitly focusing on cytokine profiles as predictive features. Cytokine, which are signaling molecules that modulate immune responses, are considered key biomarkers of HIV risk [15, 16]. We incorporate the cytokine profile alongside baseline covariates into the DeepSurv and DeepHit using derived variables, specifically the mean of the longitudinal cytokine measurements (mean model) and the difference between the first and the last measurements (difference model). For Dynamic DeepHit, we utilize time-varying cytokine measurements to capture the evolving risk of HIV infection. Missing data is addressed using missForest, and model performance is evaluated using time-dependent concordance index (C-index) and Brier scores both, before and after imputation.

2 Deep learning survival models

2.1 DeepSurv

DeepSurv is a deep learning-based extension of the Cox PH model [17]. It employs a multi-layer perceptron, utilizing a feed-forward neural network to estimate an individual's risk of events such as death or infection [12]. The input data, denoted as $\mathbf{x}$, consists of observed covariates, which pass through fully connected, non-linear activation layers in the hidden network. These layers may vary in size and are followed by dropout layers to prevent overfitting [18]. The network's output is a single node that estimates the risk function $\hat{h}_\theta(x)$, based on the network's parameterized weights θ [12]. Like the Faraggi-Simon network [19], DeepSurv uses the negative log partial likelihood from the Cox PH model [20] as its loss function. The loss function of the network is the negative log partial likelihood $L(\theta)$ as shown in Eq. (1) with an additional regularization [12]:

$$l(\theta) := -\frac{1}{N_{E=1}} \sum_{i:E^i=1} \left(\hat{h}_\theta(x^i) - \log \sum_{r \in \Re(t^i)} e^{\hat{h}_\theta(x^r)} \right) + \lambda \|\theta\|_2^2, \tag{1}$$

where λ is the ℓ_2 regularization parameter and $N_{E=1}$ is the number of individuals with observable event and θ is a set of all parameters. t^i, E^i and x^i are time, event (1: an event observed; 0, otherwise) and covariates for the i^{th} observation respectively. The risk set $\Re(t) = \{i : t^i \geq t\}$ consists of those who are still at risk of the event at time *t*.

DeepSurv employs gradient descent optimization to determine the network's weights, as detailed in Eq. (1). Additionally, it utilizes modern deep learning techniques to optimize training, including input standardization, scaled exponential linear units (SELU) or Rectified Linear Unit (ReLU) [21] as the activation function, and the adaptive moment estimation (Adam) [22] optimizer for gradient descent with Nesterov momentum [23]. It also incorporates learning rate scheduling [24] and random or empirical hyper-parameter optimization search [25] to further refine the model.

2.2 DeepHit

DeepHit is a multi-task deep neural network designed to directly learn the distribution of survival times without assuming any underlying stochastic process [26]. This allows both the model's parameters and the form of the stochastic process to be determined by the covariates specific to the dataset used in survival analysis [13]. The network consists of two components: a shared subnetwork and cause-specific subnetworks. This architecture enables DeepHit to handle datasets with either a single risk or multiple competing risks effectively. A single softmax layer is used as the output layer, and a residual connection from the input covariates is maintained and fed into each cause-specific subnetwork [27]. The shared subnetwork and the k^{th} cause-specific subnetwork, for $k = 1, ..., K$ consist of L_S and $L_{C,k}$ fully connected layers, respectively.

The shared sub-network receives covariates $\mathbf{x}$ as input and outputs a vector $f_s(\mathbf{x})$, which captures the latent representation common to the *K* events. Each cause-specific sub-network then takes as input the pair $\mathbf{z} = (f_s(\mathbf{x}), \mathbf{x})$, and produces an output vector $f_{ck}(\mathbf{z})$ representing the probability of the first hitting time for a specific cause *k*. These outputs collectively form a joint probability distribution for the first hitting time and event. The output of the softmax layer is a joint probability distribution $\mathbf{y} = [y_{1,1}, ..., y_{1,t_{max}}, ...y_{K,1}, ..., y_{K,t_{max}}]$, where $y_{k,t}$ represents the estimated probability of an individual experiencing event *k* at time *t* given covariates $\mathbf{x}$. This architecture enables the network to learn potentially non-linear and even non-proportional relationships between covariates and risks.

To train DeepHit, the total loss function $\mathcal{L}_{Total} = \mathcal{L}_1 + \mathcal{L}_2$ is minimized. $\mathcal{L}_1$ represents the negative log-likelihood of the first hitting time and event [28], which is defined as (Eq. (2))

$$\begin{aligned} \mathcal{L}_1 = -\sum_{i=1}^{N} \Bigg[& \mathbb{1}(k^i \neq \varnothing) \cdot \log \left(y^i_{k^i, t^i} \right) \\ & + \mathbb{1}(k^i = \varnothing) \cdot \log \left(1 - \sum_{k=1}^{K} \hat{F}_k(t^i | x^i) \right) \Bigg], \end{aligned} \tag{2}$$

where $\mathbb{1}(.)$ is an indicator function, $\varnothing$ represents censored observations and N is the sample size. k^i, t^i, and x^i are event type, observed time and covariates for i^{th} observation respectively. $y^i_{k^i,t^i}$ is the predicted probability that the i^{ith} observation has an event type k^i at time t^i. The first term accounts for information from uncensored patients, while the second term addresses censoring bias by incorporating the knowledge that these patients were still alive at the time of censoring [29].

The Cumulative Incidence Function (CIF) represents the probability of a specific event $k^* \in \mathcal{K}$ occurring on or before time t^*, conditional on the covariates $\mathbf{x}^*$ [30], which is defined as (Eq. (3))

$$\begin{aligned} F_{k^*}(t^*|\mathbf{x}^*) &= P(t \le t^*, k = k^*|\mathbf{x} = \mathbf{x}^*) \\ &= \sum_{t^*=0}^{t^*} P(t = t^*, k = k^*|\mathbf{x} = \mathbf{x}^*), \end{aligned} \tag{3}$$

where $\mathcal{K}$ is a set of all possible events. However, since the true CIF $F_{k^*}(t^*|\mathbf{x}^*)$ is unknown, it is replaced with *estimated* CIF $\hat{F}_{k^*}(t^*|\mathbf{x}^*) = \sum_{m=0}^{t^*} y_{k^*,m}$. In order to fine-tune the network for each cause-specific estimated CIF, a ranking loss function is employed, which adapts the concept of concordance [31]. This ensures that an individual who dies at time *t* has a higher risk at that time compared to an individual who survives longer than *t*. $\mathcal{L}_2$ incorporates the estimated (cause-specific) cumulative incidence function (CIF) calculated at various time points. By incorporating $\mathcal{L}_2$ into the total loss function, incorrect ordering of pairs is penalized (with respect to event time).

2.3 Dynamic DeepHit

Dynamic DeepHit is a multi-task neural network that extends the DeepHit model by incorporating time-varying covariates, enabling it to capture dynamic changes in risk over time [14]. Given longitudinal data, the model is trained to estimate the joint distribution of the first hitting time and the event. This joint distribution is then used to estimate the cause-specific CIFs and the survival probabilities.The architecture consists of two parts: a shared subnetwork and a set of cause-specific subnetworks.

Shared subnetwork processes the history of longitudinal measurements to predict the next values of time-varying covariate. The history of longitudinal measurements for each subject *i* is described by $\mathcal{X}^i = (\mathbf{X}^i, \mathbf{M}^i, \Delta^i)$ where $\mathbf{X}^i$ is a sequence of covariate vectors $\left\{\mathbf{x}^i_1, ..., \mathbf{x}^\mathbf{i}{}_j\right\}$ and $\mathbf{x}^i_j$ is a vector of specific covariate $\left[x^i_{j,1}, ..., x^i_{j,d_x}\right]$, that includes both time-varying and static covariates measured at time t^i_j $(0 \le t^i_j \le t)$ for time stamps $j = 1, ..., J$ with number of covariates d_x not necessarily measured at regular time intervals. $\mathbf{M}^i = \left\{\mathbf{m}^i_1, ..., \mathbf{m}^i_j\right\}$ is a sequence of mask vectors indicating missing covariates where $\mathbf{m}^i_j = \left[m^i_{j,1}, ..., m^i_{j,d_x}\right]$ with $m^i_{j,d} = 1$ if $x^i_{j,d}$ (the d^{th} element of x^i_j) was not measured and $m^i_{j,d} = 0$ otherwise. $\Delta^i = \left\{\delta^i_1, \delta^i_2 ..., \delta^i_j\right\}$ with δ^i_j representing the time intervals between consecutive measurements for the subject *i*.The shared subnetwork comprises of

(i) **Recurrent neural network (RNN) structure** that inputs a tuple of $(\mathbf{x}_j, \mathbf{m}_j, \delta_j)$ and outputs $(\mathbf{y}_j, \mathbf{h}_j)$ for time stamps $j = 1, ..., J-1, J$ where $\mathbf{y}_j$ is the estimate of time-dependent covariate after δ_j has passed and $\mathbf{h}_j$ is the hidden state at time stamp *j*. $\mathbf{h}_j$ is derived in Eq. (4) by utilizing gated recurrent unit (GRU) RNN.

$$\begin{aligned}
\mathbf{z}_j &= \sigma(W_z \mathbf{h}_{j-1} + U_z[\mathbf{x}_j; \mathbf{m}_j; \delta_j] + \mathbf{b}_z), \\
\mathbf{r}_j &= \sigma(W_r \mathbf{h}_{j-1} + U_r[\mathbf{x}_j; \mathbf{m}_j; \delta_j] + \mathbf{b}_r), \\
\tilde{\mathbf{h}}_j &= \tanh(W_h(\mathbf{r}_j \odot \mathbf{h}_{j-1}) + U_h[\mathbf{x}_j; \mathbf{m}_j; \delta_j] + \mathbf{b}_h), \\
\mathbf{h}_j &= (1 - \mathbf{z}_j) \odot \mathbf{h}_{j-1} + \mathbf{z}_j \odot \tilde{\mathrm{h}}_j,
\end{aligned} \tag{4}$$

where the parameters of the shared subnetwork are given by $\mathbf{b}_z$, $\mathbf{b}_r$ and $\mathbf{b}_h$ as the bias vectors; $\mathbf{W}_z$, $\mathbf{W}_r$ and $\mathbf{W}_h$ as the hidden-state weights; $\mathbf{U}_z$, $\mathbf{U}_r$ and $\mathbf{U}_h$ as the inputs weights for update gates $\mathbf{z}_j$, reset gate $\mathbf{r}_j$ and candidate hidden state $\mathbf{h}_j$ respectively, processing concatenated inputs $[\mathbf{x}_j; \mathbf{m}_j; \delta_j]$. $\sigma(.)$ is the sigmoid function and $\odot$ is the element-wise multiplication.

(ii) **Attention mechanism** helps to unravel the temporal significance of past measurements in making risk predictions. Specifically, the temporal attention mechanism [32] applied to the hidden states allows the network to focus on the most relevant parts of previous longitudinal measurements, guiding the model in deciding which historical data points to prioritize. It outputs weighted sum of past hidden states $\mathbf{c}$ (Eq. (5))

$$\mathbf{c} = \sum_{j=1}^{J-1} a_j \mathbf{h}_j \tag{5}$$

where $a_j = \frac{exp(e_j)}{\sum_{\ell=1}^{J-1} exp(e_\ell)}$ is the importance of the j^{th} value with $e_j = f_a(\mathbf{h}_j, \mathbf{x}_J, \mathbf{m}_J)$ as the score importance of the j^{th} measurement by referencing on the last measurement $(\mathbf{x}_J, \mathbf{m}_J)$. We set $f_a(.)$ as a two-layer feed-forward network that takes the hidden state at time stamp *j*, $\mathbf{h}_j$, and the tuple of $(\mathbf{x}_J, \mathbf{m}_J)$ as the input and outputs of a scalar e_j for $j = 1, ..., J-1, J$.

Cause specific subnetwork estimates the joint distribution of the first hitting time and event by utilizing feed-forward network comprising of fully connected layers. This subnetwork inputs the vector $\mathbf{c}$ and last measurement $(\mathbf{x}_J, \mathbf{m}_J)$ and outputs a vector $f_{ck}(\mathbf{c}, \mathbf{x}_J, \mathbf{m}_J)$.

Dynamic DeepHit uses a softmax layer to aggregate the outputs of the cause-specific subnetworks, $f_{c1}(.), ..., f_{cK}(.)$, and convert them into a valid probability distribution. This allows the network to estimate the joint distribution of the first hitting time and the event. Specifically, for a subject with covariates $\mathcal{X}^*$ each output node represents the estimated probability of experiencing event *k* at time *t*. The CIF [30] for cause $k^* \in \mathcal{K}$ on or before time t^* conditional on the history of longitudinal measurements $\mathcal{X}^*$ is recorded up to $t^*_{j^*}$, where $\mathcal{K}$ is the set of all possible events.

The true CIF $F_{k^*}(t^*|\mathcal{X}^*)$ is not known, hence the *estimated* CIF is utilized and given by (Eq. (6))

$$\hat{F}_{k^*}(t^*|\mathcal{X}^*) = \sum_{t \leq t^*} o^*_{k^*,t}, \tag{6}$$

where $o^*_{k^*}$ is the estimated probability of experiencing event $k^* t$ at time t conditional on $\mathcal{X}^*$.

To train Dynamic DeepHit, the total loss function $\mathcal{L}_{Total} = \mathcal{L}_1 + \mathcal{L}_2 + \mathcal{L}_3$ is mininmized. $\mathcal{L}_1$ represents the negative log-likelihood of the first hitting time and the

corresponding event, accounting for right-censoring [31]. This is adapted to a survival analysis context, where the history of longitudinal measurements and *k* risks are available. $\mathcal{L}_2$ (ranking loss) incorporates the *estimated* CIFs at different time points to refine the network's accuracy for each cause-specific CIF. To achieve this, a ranking loss function is used, which adapts the concept of concordance [31]. Since subjects' longitudinal measurements can start at any point in their life or disease progression [33], their risks are compared based on the time elapsed since their last measurement that is for subject *i*, we focus on $\tau^i = t^i - t^i_j$. A pair (*i*, *r*) is considered *acceptable pair* for event *k* if subject *i* experiences event *k* at time t^i_j while subject *r* remains event-free until t^i_j (i.e the event time of *r* is later than event time of $i, t^r_j > t^i_j$). The estimated CIF satisfies concordance if $\hat{F}_k(t^r_j + t^i_j|\mathcal{X}^r) > \hat{F}_k(t^i_j + t^i_j|\mathcal{X}^i)$. The ranking loss is then defined among valid pairs of subjects with different measurement histories. $\mathcal{L}_3$ (prediction loss) incorporates an auxiliary task that predicts $\mathbf{y_j}$ the step-ahead covariates $\mathbf{x_{j+1}}$, to regularize the shared subnetwork, ensuring that the hidden representations preserve information for future predictions and takes into account missing measurements.

3 Performance evaluation

3.1 Time-dependent concordance index (C^{td}-index)

The time-dependent C-index is a metric used in survival analysis to assess concordance between predicted survival times and actual event times [34]. Unlike the standard C-index, which provides a single performance value [31], the time-dependent version accounts for the chronological progression of events, reflecting changes in model discrimination over time [35]. By comparing predicted survival probabilities at various time points with actual outcomes, it evaluates whether individuals who experience an event (e.g., death or failure) have a lower predicted probability of survival than those who do not [36]. This approach ensures an accurate representation of model performance throughout the study period.

For the setting with longitudinal measurements, a cause specific time-dependent C-index ($C^{td}_k(t, \Delta t)$) is adapted which is an extension of the C-index [37]. More specifically, $C^{td}_k(t, \Delta t)$ incorporates both the prediction and evaluation times to account for potential changes in risk over time [38]. The time dependent C-index ranges from 0 to 1, and a larger value indicates better performance.

3.2 Time-dependent Brier score BS^{td}

The time-dependent Brier score is a crucial metric for evaluating the accuracy of survival predictions over time [39]. Unlike static measures, the time-dependent Brier score evaluates model performance at multiple time intervals, making it sensitive to how well the model captures changes in survival risk throughout the study period. By definition *BS*(*t*) is the mean squared error of the difference between a model based survival function *S*(*t*; *x*) and event status $I(T > t)$ [40]. The time-dependent Brier score is calculated as the mean squared error between the predicted survival function $\hat{S}(t; x^i)$ and the observed event status $I(T^i > t)$ at a given time point *t*, detailed in Eq. (7)

$$\widehat{BS}(t) = \frac{1}{n}\sum_{i=1}^{n} \widehat{w}^i(t)\left\{I(T^i > t) - \widehat{S}(t; x^i)\right\}^2, \tag{7}$$

where $\widehat{w}^i(t)$ is the inverse probability of censoring weights (IPCW) [41], given by (Eq. (8))

$$\widehat{w}^i(t) = \frac{(1 - I(T^i > t))\gamma^i}{\widehat{G}(T^i-)} + \frac{I(T^i > t)}{\widehat{G}(t)}, \tag{8}$$

where $I(.)$ is an indicator function, γ^i indicates the censoring status of subject i (1 if event is observed, 0 if event is censored) and $\widehat{G}(t)$ is the estimated survival function of censoring at time t while $\widehat{G}(T^i-)$ is the estimated probability of uncensored subjects. The score typically ranges between 0 and 1, but values may exceed 1 when inverse probability of censoring weights (IPCW) are applied in heavily censored datasets [42].

4 Handling missing data: missForest

MissForest is an imputation technique that utilizes the Random Forest algorithm to estimate missing values. Adapted from Breiman's Random Forest approach [43], it is well-suited for data containing a mix of variable types and can capture complex, nonlinear relationships without relying on parametric assumptions. The imputation process is iterative: a Random Forest model is first built using available data, then used to predict missing values, with predictions refined over successive iterations [44]. MissForest is accessible as an *R* package named *missForest* [45]. After imputation, the performance of continuous variables is evaluated using the normalized root mean squared error (NRMSE) [46]. For categorical variables, the proportion of falsely classified entries (PFC) over the missing categorical values, Δ_F is used as a performance metric. In both cases, a value close to 0 indicates good performance, while a value around 1 signifies poor performance.

5 Real data application

5.1 Dataset

The dataset for this study was obtained from the Centre for the AIDS Programme of Research in South Africa (CAPRISA 004 trial), a double-blind, randomized, two-arm study comparing placebo and Tenofovir groups. The trial aimed to evaluate the effectiveness of Tenofovir gel, a vaginal microbicide, and involved enrolling HIV-negative, sexually active women aged 18 to 40 in South Africa [47]. The trial spanned 30 months, with an 18-month recruitment phase and a 12-month follow-up period. The dataset included longitudinal measurements of 48 cytokines and 46 baseline characteristics from 812 participants, 96 of whom contracted HIV. Cytokine levels were measured from stored plasma samples and cervicovaginal lavage specimens in both the case and control groups. The primary endpoint was time-to-HIV infection (incidence), measured from study enrollment until positive HIV test or censoring (end of follow-up/loss to contact). Events were confirmed via laboratory testing, with censoring applied for participants remaining HIV-negative at trial completion.

After pre-processing, 24 baseline variables and all 48 cytokines were retained for analysis. There were 699 participants included in the complete case analysis before imputation and 812 participants after imputation. All data preparation and imputation were conducted using R (version 4.4.1) [48] while statistical analyses both training and testing of the deep learning models were done in Python [49]. The dataset was divided into two distinct subsets: 80% for training and 20% for testing, applied separately to both the

complete case (train set: N=560, test set: N=139) and imputed (train set: N=650, test set: N=162) datasets. To prevent data leakage, the imputed dataset's training set excluded all observations present in the complete-case testing set. Consequently, the imputed dataset's testing set comprised all observations from the complete-case testing set, along with additional (imputed) data points. The training set was used to build the predictive model, while the testing set assessed its accuracy. The data was split using a fixed seed value of 2408 to maintain consistency and reproducibility.

5.2 Results

Survival models using DeepSurv, DeepHit, and Dynamic DeepHit were built with the training set (80% of data) and prediction done on the test set (20% of data), applied to both the complete case and imputed datasets. To ensure reproducibility of all the results, a fixed seed value of 2408 was applied. Missing data were handled using the missForest method, which employed 500 trees and 100 iterations. For each DeepSurv and DeepHit model, two versions were fitted: one based on the mean of individual cytokine measurements combined with baseline covariates (mean model), and another based on the difference between the first and last recorded cytokine measurements (difference model), also incorporating baseline covariates. In the Dynamic DeepHit model, a single version was fitted, which included the original time-varying cytokine values alongside baseline covariates. Cytokine profile measurements were collected longitudinally for each subject from the time of enrollment until either the time of HIV incidence or the end of the trial (for those who remained HIV-negative). As such, all available cytokine measurements up to the point of the event (HIV incidence) or censoring were included in the prediction process.

The models were fine-tuned using specific hyperparameters, which were optimized through an empirical search to enhance performance. To guide this tuning process and mitigate overfitting, 20% of the training data was set aside as a validation set. This validation set was used to monitor model performance during training and to select the hyperparameter configurations that yielded the best generalization. For the DeepSurv model, a dropout rate of 0.1 was applied to reduce overfitting, while the learning rate was set at 0.01 to control the model's convergence speed. The architecture included two layers with 32 nodes each, utilizing the ReLU activation function for non-linearity. The model was trained with a batch size of 256 and optimized using the Adam optimizer over 2000 epochs. Similarly, the DeepHit model used the same architecture, with two layers of 32 nodes and a dropout rate of 0.1. It employed a learning rate of 0.01, a batch size of 256, and was also trained for 2000 epochs. Key additional parameters included alpha (1.0) and sigma (0.1), which influenced the loss function and regularization, respectively, alongside the Adam optimizer.

For the Dynamic DeepHit model, more complex parameters were set. The batch size was 32, with a burn-in period of 3000 iterations followed by 5,000 training iterations. A keep probability of 0.6 was used to manage dropout during training, and the learning rate at 0.01. The architecture included two RNN layers (LSTM-Long Short-Term Memory) with 100 hidden units and two fully connected layers with ReLU activations. The model also incorporated attention mechanisms and conditional survival layers, each with two layers. Regularization terms for weights were set at 0.00001 for both input and

Table 1 The performance results for the deep learning survival models using the 20% test data (*td*; time-dependent)

	Model	Before Imputation (N=139)		After Imputation (N=162)	
		$\mathbf{C^{td}}$-index	$\mathbf{BS^{td}}$	$\mathbf{C^{td}}$-index	$\mathbf{BS^{td}}$
DeepSurv	Mean	0.6175	0.1318	0.6870	0.1141
	Difference	0.6246	0.1206	0.7022	0.1038
DeepHit	Mean	0.6318	0.1245	0.6976	0.1223
	Difference	0.6714	0.1101	0.7135	0.0979

Table 2 The performance results for Dynamic DeepHit model by time-dependent C-index over different evaluation and prediction time using the 20% test data

		Evaluation time points(months)									
		Before imputation (N=139)					After imputation (N=162)				
		Three	Six	Nine	Twelve	Fifteen	Three	Six	Nine	Twelve	Fifteen
Prediction time points (months)	Three	0.6851	0.6851	0.8326	0.5692	0.5692	0.8432	0.8213	0.8350	0.8098	0.7990
	Six	0.6851	0.8316	0.5692	0.5736	0.6190	0.8331	0.8298	0.8000	0.7990	0.7990
	Nine	0.9764	0.7248	0.7248	0.7544	0.7544	0.8276	0.8001	0.7968	0.7968	0.7968
	Twelve	0.7847	0.7843	0.8121	0.8121	0.8121	0.8501	0.8501	0.8501	0.8501	0.8501
	Fifteen	0.7958	0.8173	0.8173	0.8173	0.8173	0.8551	0.8551	0.8551	0.8544	0.8544

Table 3 The performance results for Dynamic DeepHit model by time-dependent Brier Score over different evaluation and prediction time using the 20% test data

		Evaluation time points(months)									
		Before imputation (N=139)					After imputation (N=162)				
		Three	Six	Nine	Twelve	Fifteen	Three	Six	Nine	Twelve	Fifteen
Prediction time points(months)	Three	0.0013	0.0032	0.0056	0.0135	0.0167	0.0081	0.0098	0.0171	0.0267	0.0523
	Six	0.0016	0.0059	0.0124	0.0163	0.0376	0.0091	0.0165	0.0279	0.0467	0.0574
	Nine	0.0064	0.0109	0.0143	0.0259	0.0326	0.0148	0.0264	0.0419	0.0579	0.0693
	Twelve	0.0111	0.0155	0.0255	0.0322	0.0410	0.0299	0.0427	0.0573	0.0682	0.0766
	Fifteen	0.0189	0.0252	0.0317	0.0433	0.0415	0.0429	0.0569	0.0691	0.0751	0.0777

output. Other key parameters included alpha (1.0), beta (0.1), and gamma (1.0) to balance the trade-offs between loss components.

The performance of the three models DeepSurv, DeepHit, and Dynamic DeepHit was assessed using the time-dependent C-index and time-dependent Brier score, with results in Tables 1, 2, and 3. These metrics were applied to both the complete case and imputed datasets to evaluate model accuracy. For the Dynamic DeepHit model, the metrics were specifically calculated at various prediction and evaluation time points to capture its performance across different time stamps.

The results presented in Table 1 show the time-dependent C-index and Brier scores for DeepSurv and DeepHit models using both mean-based and difference-based covariate approaches, before and after imputation. Before imputation, the DeepSurv models yielded C-index values of 0.6175 (mean-based) and 0.6246 (difference-based), with corresponding Brier scores of 0.1318 and 0.1206, respectively. After imputation, the C-index values were 0.6870 (mean-based) and 0.7022 (difference-based), while Brier scores were 0.1141 and 0.1038, respectively. These results reflect how model performance metrics varied across different covariate representations and data preprocessing strategies.

Table 1 also presents the performance metrics for the DeepHit models using both mean-based and difference-based covariate approaches, before and after imputation. Prior to imputation, the mean-based and difference-based models produced C-index values of 0.6318 and 0.6714, with Brier scores of 0.1245 and 0.1101, respectively. Following imputation, the C-index values were 0.6976 (mean-based) and 0.7135 (difference-based), while Brier scores were 0.1223 and 0.0979, respectively. These results illustrate how the DeepHit model's performance metrics varied across different covariate representations.

The performance of the Dynamic DeepHit model, as measured by the time-dependent C-index in Table 2, shows distinct trends across various prediction and evaluation time points, both before and after imputation. Before imputation, the C-index for earlier prediction times (three and six months) was generally moderate. However, as the prediction time increased, the model performance varied. After imputation, the model consistently improved across all evaluation and prediction time points. For instance, at the prediction time point of three months, the C-index improved significantly to 0.8432 at three months of evaluation and remained relatively high across subsequent evaluations, reaching 0.8350 at nine months. Similarly, the twelve-month prediction time showed a notable increase, with the C-index reaching 0.8501 across several evaluation time points. Even for the longest evaluation time (15 months), the C-index remained higher, consistently exceeding 0.85 for various prediction time points. These results provide a detailed view of model performance under different temporal settings, reflecting the behavior of the Dynamic DeepHit model under both complete-case and imputed data scenarios.

Similarly, evaluation using the time-dependent Brier Score depicted in Table 3, illustrated notable trends across various prediction and evaluation time points before and after imputation. Before imputation, the Brier scores were relatively low for shorter prediction times, with values such as 0.0013 at three months of evaluation. However, as the evaluation time extended, Brier scores generally increased, indicating a decline in predictive accuracy, with the score reaching 0.0167 at fifteen months. The scores remained below 0.05 throughout the whole evaluation period, suggesting good initial performance but a gradual increase in prediction error over time. After imputation, the Brier scores consistently increased across evaluation times. For example, the score at the prediction time point of three months rose to 0.0081 and continued increasing to 0.0523 at fifteen months. Across both scenarios, Brier scores remained relatively low throughout the evaluation periods, providing a quantitative summary of prediction error over time under different data conditions.

Figure 1 illustrates two survival curves (panels a and b) generated by the DeepSurv model for ten randomly sampled individuals at risk of HIV infection over a 30-month period. Each colored line represents a different individual from the test set of the imputed data. In the mean model (panel a), the survival probability decreases gradually and smoothly over time, with the curves diverging slowly. This indicates a moderate variation in predicted survival times among individuals. In contrast, the difference model (panel b) displays more pronounced separation between the curves, suggesting a greater variability in survival predictions. For instance, according to the mean model, individual 0 has approximately an 89% chance of surviving up to 15 months, while individual 8 shows a higher survival probability of about 96% at the same time point. However, in the difference model, the survival probability for individual 0 at the 15-month mark drops to roughly 87%, highlighting the impact of the difference-based approach

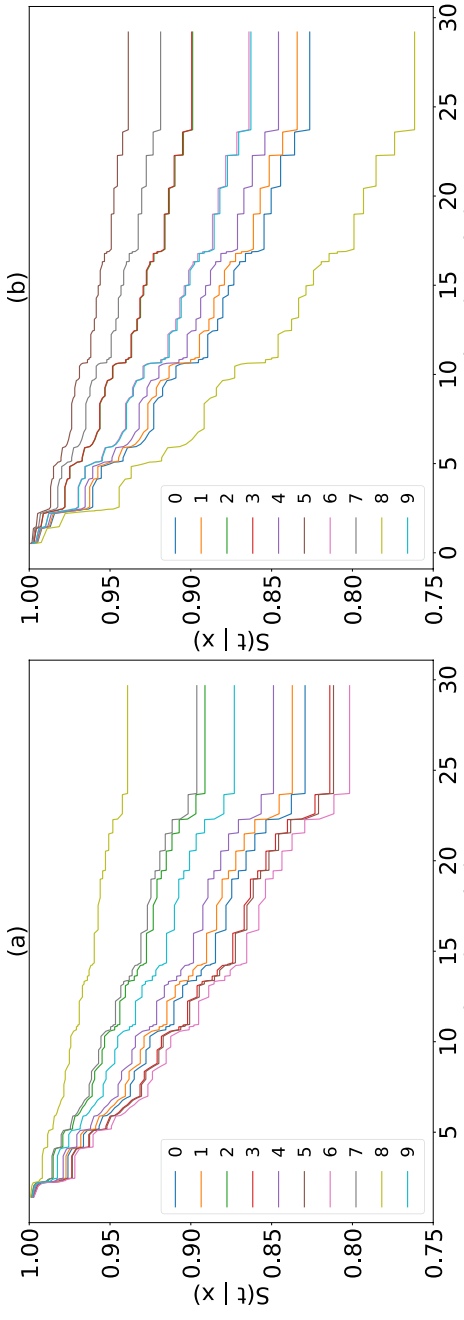

Fig. 1 The graph of estimated survival lifetimes for 10 sampled individuals at risk of HIV infection from the 20% test data (N=162) over the study period using the output of the DeepSurv models (with imputed datasets). The left panel (a) is for the mean model and right panel (b) is for the difference model. The graphs represents the possible survival patterns in the data.

on survival predictions. The curves flatten at $t = 24$ months and beyond likely due to some individuals receiving preventative treatments, which helps reduce their risk of HIV infection.

Figure 2 shows the predicted probabilities of HIV risk generated by the DeepHit model for each 10 randomly sampled individuals in the test set of the imputed data over a 30-month period. The DeepHit model provides a probability vector for each individual, indicating their risk of HIV infection at each time point. During the first 20 months, the risk of HIV infection remains relatively low, but it increases thereafter. This increase is more pronounced in the mean model (panel a) compared to the difference model (panel b). In the mean model, there are several noticeable spikes in risk throughout the evaluation period, indicating periodic increases in predicted probabilities for certain individuals. The lower risk observed in the DeepHit model during the first few months aligns with the predicted survival curves from the DeepSurv model, where survival probabilities are initially high and gradually decrease with increase in the evaluation period.

6 Discussion

Traditional hazard-based models like the Cox PH model are designed to assess the impact of variables on survival outcomes, rather than to make predictions [50]. When predicting outcomes like HIV incidences alternative approaches such as machine learning or deep learning methods may be considered for additional flexibility and improved performance. While traditional statistics focus on explanation, machine learning prioritizes prediction [51]. The Cox PH models are established statistical tools for providing interpretable results but may struggle with complex, high-dimensional data, non-linear relationships, and covariates that violate PH assumptions, such as time-varying covariates [7, 52–55]. In contrast, deep learning models, such as recurrent neural networks (RNNs) and other architectures, excel in handling these complexities [56]. They model non-linear relationships and high-dimensional data without requiring explicit feature engineering or prior assumptions leading to superior prediction performance [57]. Deep learning's adaptability to large datasets and its capability to learn from nuanced patterns in the data provides it with a significant advantage in HIV incidence prediction [58, 59].

In this paper, we applied DeepSurv, DeepHit, and Dynamic DeepHit to a real HIV dataset and successfully evaluated their ability to make accurate dynamic survival predictions. DeepSurv, a deep feed-forward neural network, estimates each individual's impact on their hazard rates, integrating deep learning with traditional survival analysis [12]. By incorporating non-linear relationships, it extends the Cox PH model to handle complex survival data more effectively. DeepSurv has been widely used in clinical studies to improve survival predictions and treatment recommendations, demonstrating state-of-the-art performance [12]. Its application shows great potential in advancing personalized medicine and enhancing treatment outcomes across diverse patient populations [60, 61]. DeepHit on the other hand is designed to directly learn the distribution of survival times without assuming any underlying stochastic process [26] while Dynamic DeepHit incorporates time-varying covariates, enabling it to capture dynamic changes in risk over time [14].

Our results from the data analysis highlight the importance of incorporating time-varying covariates into survival models, mainly when dealing with high-dimensional biomarker data. While DeepSurv and DeepHit offered practical approaches to

modeling static summary statistics of cytokine profiles, they did not fully capture the evolving nature of the cytokine data over time. In contrast, the Dynamic DeepHit model is uniquely positioned to accommodate the dynamic structure of longitudinal biomarker data. Integrating the full-time profiles allowed for a more nuanced interpretation of how temporal changes in biomarkers influenced HIV incidence over time. This does not necessarily imply superior performance in predictive metrics but instead highlights the model's robustness and suitability in contexts where dynamic patterns are critical.

The use of missForest for imputation proved to be a pivotal step in enriching the datasets, enabling the construction of more informative and potentially more generalizable models. While direct comparison between imputed and non-imputed datasets is inherently challenging due to differences in sample size and composition, imputation significantly enhanced usable data. This enrichment facilitated training on a broader and more representative population sample, providing valuable insights into the underlying patterns. A stringent data-splitting strategy was applied to ensure fairness and prevent data leakage during evaluation. Specifically, the training set for the imputed dataset excluded all observations present in the complete-case testing set. Conversely, the testing set for the imputed dataset incorporated both the complete-case testing observations and additional imputed data points. This approach minimized overlap and supported a more balanced framework for assessing the models built on imputed versus non-imputed datasets.

Although hyperparameters were optimized through empirical search, this method presents several challenges. The process is computationally intensive, requiring exhaustive exploration of predefined parameter ranges without adaptive refinement. Additionally, identifying the optimal configuration is not guaranteed, as the initially selected bounds constrain the search. Overall, the enriched dataset and careful data-splitting methodology underscore the importance of thoughtful handling of missing data. Coupled with the challenges of hyperparameter optimization, these considerations highlight the intricate balance required to develop robust and meaningful survival models tailored to complex, high-dimensional biomarker datasets.

7 Conclusions

Deep learning models offer transformative possibilities in survival analysis for high-dimensional datasets, particularly in scenarios involving time-varying covariates. Our findings emphasize the value of Dynamic DeepHit as a robust approach for capturing the dynamic nature of cytokine profiles over time, providing deeper insights into their influence on HIV incidence. While imputation techniques like missForest enriched the datasets and enabled the development of more informative and generalizable models, it is essential to attribute the improved robustness to the added information rather than imputation alone. This study highlights the necessity of dynamic modeling and thoughtful data handling strategies, paving the way for future research to advance survival analysis methodologies in clinical and epidemiological settings.

8 Supplementary information

The GitHub repositories of the model developers (DeepSurv, DeepHit, Dynamic DeepHit) were instrumental in creating the analysis code utilized in this paper.

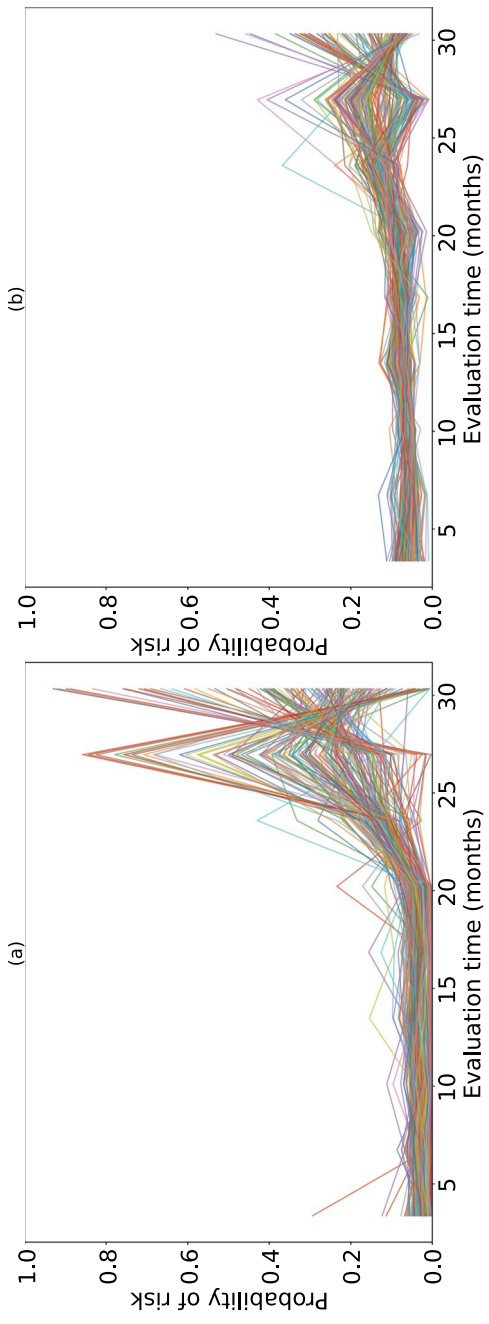

Fig. 2 Hazard built byDeepHit models (with imputed datasets) showing the probabilities of HIV risk for each individual at every time point within the 30 -month evaluation period using the 20% test data (N=162). The left panel (a) is for the mean model and right panel (b) is for the difference model.

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1007/s44163-025-00429-z.

Supplementary Material 1_Analysis code

Acknowledgements

The authors express their sincere gratitude to CAPRISA for generously providing access to the dataset, which was essential to the progress of this research.

Author Contributions

Sarah Ogutu: Conceptualization, Writing– review and editing, Writing– original draft, Visualization, Validation, Software, Resources, Project administration, Methodology, Investigation, Formal analysis, Data curation. Mohanad Mohammed: Writing– review and editing, Visualization, Validation, Supervision, Software, Formal analysis. Henry Mwambi: Writing– review and editing, Validation, Supervision, Methodology. All authors approved the final manuscript submitted for publication.

Funding

This research was funded in whole or in part by Science for Africa Foundation to the Sub-Saharan Africa Consortium for Advanced Biostatistics (SSACAB II) programme [Grant number DEL-22-009] with support from Wellcome Trust and the UK Foreign, Commonwealth & Development Office and is part of the EDCPT2 programme supported by the European Union. For purposes of open access, the author has applied a CC BY public copyright license to any Author Accepted Manuscript version arising from this submission.

Data Availability

Researchers seeking access to data from completed CAPRISA studies are required to fill out a data request form, which can be found on their website CAPRISA Studies.

Code Availability

The python scripts files that produced the results of the analysis are available here https://github.com/SarahOgutu/Survival-Deep-Learning-models.The PDF version of the python scripts is available in the Supplementary Material 1_Analysis code.

Declarations

Ethical approval, consent to participate, and consent to publish

Not applicable.

Competing interests

The author(s) declared no potential Conflict of interest with respect to the research, authorship, and/or publication of this article.

Received: 27 January 2025 / Accepted: 9 July 2025

Published online: 24 July 2025

References

1. Clark TG, Bradburn MJ, Love SB, et al. Survival analysis part i: basic concepts and first analyses. Br J Cancer. 2003;89(2):232–8.
2. D'Argenio V. The high-throughput analyses era: are we ready for the data struggle? High-throughput. 2018;7(1):8.
3. Lin DY, Wei LJ. The robust inference for the cox proportional hazards model. J Am Stat Assoc. 1989;84(408):1074–8.
4. Spooner A, Chen E, Sowmya A, et al. A comparison of machine learning methods for survival analysis of high-dimensional clinical data for dementia prediction. Sci Rep. 2020;10(1):20410.
5. Sinnott JA, Cai T. Inference for survival prediction under the regularized cox model. Biostatistics. 2016;17(4):692–707.
6. Therneau T, Crowson C, Atkinson E. Using time dependent covariates and time dependent coefficients in the cox model. Surviv Vignettes. 2017;2(3):1–25.
7. Fisher LD, Lin DY. Time-dependent covariates in the cox proportional-hazards regression model. Annu Rev Public Health. 1999;20(1):145–57.
8. Lunn M, McNeil D. Applying cox regression to competing risks. Biometrics. 1995;51:524–32.
9. Cooper H, Wells S, Mehta S. Are competing-risk models superior to standard cox models for predicting cardiovascular risk in older adults? analysis of a whole-of-country primary prevention cohort aged greater than 65 years. Int J Epidemiol. 2022;51(2):604–14.
10. Park SY, Park JE, Kim H, et al. Review of statistical methods for evaluating the performance of survival or other time-to-event prediction models (from conventional to deep learning approaches). Korean J Radiol. 2021;22(10):1697.
11. Salerno S, Li Y. High-dimensional survival analysis: methods and applications. Ann Rev Stat Appl. 2023;10(1):25–49.
12. Katzman JL, Shaham U, Cloninger A, et al. Deepsurv: personalized treatment recommender system using a cox proportional hazards deep neural network. BMC Med Res Methodol. 2018;18:1–12.
13. Lee C, Zame W, Yoon J et al. Deephit: a deep learning approach to survival analysis with competing risks. In: Proceedings of the AAAI conference on artificial intelligence, vol 32.
14. Lee C, Yoon J, Van Der Schaar M. Dynamic-deephit: a deep learning approach for dynamic survival analysis with competing risks based on longitudinal data. IEEE Trans Biomed Eng. 2019;67(1):122–33.

15. Dembic Z. The cytokines of the immune system: the role of cytokines in disease related to immune response. New York: Academic Press; 2015.
16. Ye Q, Shao WX, Xu XJ, et al. The clinical application value of cytokines in treating infectious diseases. PLoS ONE. 2014;9(6): e98745.
17. Lin H, Zelterman D. Modeling survival data: extending the cox model; 2002.
18. Srivastava N, Hinton G, Krizhevsky A, et al. Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res. 2014;15(1):1929–58.
19. Faraggi D, Simon R. A neural network model for survival data. Stat Med. 1995;14(1):73–82.
20. Christensen E. Multivariate survival analysis using cox's regression model. Hepatology. 1987;7(6):1346–58.
21. Klambauer G, Unterthiner T, Mayr A, et al. Self-normalizing neural networks. In: Advances in neural information processing systems 2017;30.
22. Okewu E, Misra S, Lius FS. Parameter tuning using adaptive moment estimation in deep learning neural networks. In: Computational science and its applications—ICCSA 2020: 20th international conference, Cagliari, Italy, July 1–4, 2020, Proceedings, Part VI 20. Springer, pp. 261–272.
23. Nesterov Y. Gradient methods for minimizing composite functions. Math Program. 2013;140(1):125–61.
24. Iiduka H. Appropriate learning rates of adaptive learning rate optimization algorithms for training deep neural networks. IEEE Trans Cybernet. 2021;52(12):13250–61.
25. Bergstra J, Bengio Y. Random search for hyper-parameter optimization. J Mach Learn Res. 2012;13(2):281–305.
26. Collobert R, Weston J. A unified architecture for natural language processing: deep neural networks with multitask learning. In: Proceedings of the 25th international conference on Machine learning. pp. 160–167.
27. He K, Zhang X, Ren S et al. Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778.
28. Lee M, Whitmore G. Proportional hazards and threshold regression: their theoretical and practical connections. Lifetime Data Anal. 2010;16:196–214.
29. Lawless JF. Statistical models and methods for lifetime data. New York: Wiley; 2011.
30. Fine JP, Gray RJ. A proportional hazards model for the subdistribution of a competing risk. J Am Stat Assoc. 1999;94(446):496–509.
31. Harrell FE, Califf RM, Pryor DB, et al. Evaluating the yield of medical tests. JAMA. 1982;247(18):2543–6.
32. Bahdanau D. Neural machine translation by jointly learning to align and translate. arXiv:1409.0473 2014.
33. Ranganath R, Perotte A, Elhadad N et al. Deep survival analysis. In Machine Learning for Healthcare Conference. PMLR, pp. 101–114.
34. Antolini L, Boracchi P, Biganzoli E. A time-dependent discrimination index for survival data. Stat Med. 2005;24(24):3927–44.
35. Wang J, Jiang X, Ning J. Evaluating dynamic and predictive discrimination for recurrent event models: use of a time-dependent c-index. Biostatistics. 2023;25(4):1140–55.
36. Longato E, Vettoretti M, Di Camillo B. A practical perspective on the concordance index for the evaluation and selection of prognostic time-to-event models. J Biomed Inform. 2020;108: 103496.
37. Gerds TA, Kattan MW, Schumacher M, et al. Estimating a time-dependent concordance index for survival prediction models with covariate dependent censoring. Stat Med. 2013;32(13):2173–84.
38. Rizopoulos D, Molenberghs G, Lesaffre EM. Dynamic predictions with time-dependent covariates in survival analysis using joint modeling and landmarking. Biom J. 2017;59(6):1261–76.
39. Hao L, Kim J, Kwon S, et al. Deep learning-based survival analysis for high-dimensional survival data. Mathematics. 2021;9(11):1244.
40. Graf E, Schmoor C, Sauerbrei W, et al. Assessment and comparison of prognostic classification schemes for survival data. Stat Med. 1999;18(17–18):2529–45.
41. Gerds TA, Schumacher M. Consistent estimation of the expected brier score in general survival models with right-censored event times. Biom J. 2006;48(6):1029–40.
42. Kvamme H, Borgan Ø. The brier score under administrative censoring: Problems and a solution. J Mach Learn Res. 2023;24(2):1–26.
43. Breiman L. Random forests. Mach Learn. 2001;45:5–32.
44. Stekhoven DJ, Bühlmann P. Missforest-non-parametric missing value imputation for mixed-type data. Bioinformatics. 2012;28(1):112–8.
45. Stekhoven DJ, Stekhoven M. Package 'missforest'. R package version. 2013;1:21.
46. Oba S, Ma S, Takemasa I, et al. A Bayesian missing value estimation method for gene expression profile data. Bioinformatics. 2003;19(16):2088–96.
47. Abdool Karim Q, Abdool Karim SS, Frohlich JA, et al. Effectiveness and safety of tenofovir gel, an antiretroviral microbicide, for the prevention of hiv infection in women. Science. 2010;329(5996):1168–74.
48. R Core Team. The r project for statistical computing. https://www.r-project.org/, 2024. Accessed: 2024-08-06.
49. Python team. Python programming language. https://www.python.org/downloads/, 2024. Accessed: 2024-10-08.
50. Fox J, Weisberg S. Cox proportional-hazards regression for survival data. An R and S-PLUS companion to applied regression 2002; 2002.
51. Kim DW, Lee S, Kwon S, et al. Deep learning-based survival prediction of oral cancer patients. Sci Rep. 2019;9(1):6994.
52. Jiang N, Wu Y, Li C. Limitations of using cox proportional hazards model in cardiovascular research. Cardiovasc Diabetol. 2024;23(1):219.
53. Tian L, Zucker D, Wei L. On the cox model with time-varying regression coefficients. J Am Stat Assoc. 2005;100(469):172–83.
54. Zhang Z, Reinikainen J, Adeleke KA, et al. Time-varying covariates and coefficients in cox regression models. Ann Transl Med. 2018;6(7):121.
55. Bellera CA, MacGrogan G, Debled M, et al. Variables with time-varying effects and the cox model: some statistical concepts illustrated with a prognostic factor study in breast cancer. BMC Med Res Methodol. 2010;10:1–12.
56. Sargent DJ. Comparison of artificial neural networks with other statistical approaches: results from medical data sets. Cancer Interdiscipl Int J Am Cancer Soc. 2001;91(S8):1636–42.

57. Xiang A, Lapuerta P, Ryutov A, et al. Comparison of the performance of neural network methods and cox regression for censored survival data. Comput Stat Data Anal. 2000;34(2):243–57.
58. Xu Y, Hosny A, Zeleznik R, et al. Deep learning predicts lung cancer treatment response from serial medical imaging. Clin Cancer Res. 2019;25(11):3266–75.
59. Belete DM, Huchaiah MD. A deep learning approaches for modeling and predicting of hiv test results using edhs dataset. In: Future opportunities and tools for emerging challenges for HIV/AIDS control. IntechOpen; 2023.
60. Lei J, Xu X, Xu J, et al. The predictive value of modified-Deepsurv in overall survivals of patients with lung cancer. iScience. 2023;26(11): 108200.
61. Moradmand H, Aghamiri S, Ghaderi R, et al. The role of deep learning-based survival model in improving survival prediction of patients with glioblastoma. Cancer Med. 2021;10(20):7048–59.

Publisher's Note