

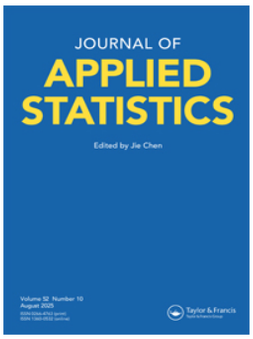
https://dspace.org/

Adapting and evaluating deeppseudo neural network for survival data with timevarying covariates

Albert Whata; Justine Nasejje; Najmeh Nakhaei Rad; Tshilidzi Mulaudzi; Ding-Geng Chen

TAYLOR & FRANCIS LTD
https://hdl.handle.net/10539/46688
WOS

Downloaded from a DSpace repository. https://dspace.org/



Journal of Applied Statistics

ISSN: 0266-4763 (Print) 1360-0532 (Online) Journal homepage: www.tandfonline.com/journals/cjas20

Adapting and evaluating deep-pseudo neural network for survival data with time-varying covariates

Albert Whata, Justine B. Nasejje, Najmeh Nakhaei Rad, Tshilidzi Mulaudzi & Ding-Geng Chen

To cite this article: Albert Whata, Justine B. Nasejje, Najmeh Nakhaei Rad, Tshilidzi Mulaudzi & Ding-Geng Chen (2025) Adapting and evaluating deep-pseudo neural network for survival data with time-varying covariates, Journal of Applied Statistics, 52:10, 1847-1870, DOI: 10.1080/02664763.2024.2444649

To link to this article: https://doi.org/10.1080/02664763.2024.2444649

Published online: 24 Dec 2024.

Submit your article to this journal

Article views: 223

View related articles

View Crossmark data

Full Terms & Conditions of access and use can be found at
https://www.tandfonline.com/action/journalInformation?journalCode=cjas20

JOURNAL OF APPLIED STATISTICS
2025, VOL. 52, NO. 10, 1847–1870
https://doi.org/10.1080/02664763.2024.2444649

Taylor & Francis
Taylor & Francis Group

Check for updates

Adapting and evaluating deep-pseudo neural network for survival data with time-varying covariates

Albert Whata[a], Justine B. Nasejje[b], Najmeh Nakhaei Rad[a,c], Tshilidzi Mulaudzi[d] and Ding-Geng Chen[a,e]

[a]Department of Statistics, University of Pretoria, Pretoria, South Africa; [b]School of Statistics and Actuarial Science, University of the Witwatersrand, Johannesburg, South Africa; [c]NRF Centre of Excellence in Mathematical and Statistical Sciences (CoE-MaSS), Johannesburg, South Africa; [d]Department of Mathematical and Computational Sciences, University of Venda, Limpopo, South Africa; [e]College of Health Solutions, Arizona State University, Phoenix, AZ, USA

ABSTRACT

The Extended Cox model provides an alternative to the proportional hazard Cox model for modelling data including time-varying covariates. Incorporating time-varying covariates is particularly beneficial when dealing with survival data, as it can improve the precision of survival function estimation. Deep learning methods, in particular, the Deep-pseudo survival neural network (DSNN) model have demonstrated a high potential for accurately predicting right-censored survival data when dealing with time-invariant variables. The DSNN's ability to discretise survival times makes it a natural choice for extending its application to scenarios involving time-varying covariates. This study adapts the DSNN to predict survival probabilities for data with time-varying covariates. To demonstrate this, we considered two scenarios: significant and non-significant time-varying covariates. For significant covariates, the Brier scores were below 0.25 at all considered specific time points, while, in the non-significant case, the Brier scores were above 0.25. The results illustrate that the DSNN performed comparably to the extended Cox, the Dynamic-DeepHit and mulitivariate joint models and on the simulated data. A real-world data application further confirms the predictive potential of the DSNN model in modelling survival data with time-varying covariates.

ARTICLE HISTORY
Received 31 October 2023
Accepted 6 December 2024

KEYWORDS
Deep neural networks; pseudo values; time-varying covariate; extended Cox model; Dynamic-DeepHit; multivariate joint model

2020 MATHEMATICS SUBJECT CLASSIFICATIONS
92B15; 92B20; 62R07

1. Introduction

In the estimation of survival probabilities, there has been extensive utilization of both semi-parametric and parametric models, including the Cox proportional hazard (CPH) models, generalized additive models, and the logistic regression model [27,30]. Although parametric assumptions are generally thought of as being inadequate for the analysis of large datasets, censored survival data has frequently been analyzed by the semi-parametric Cox

CONTACT Ding-Geng Chen dinchen@asu.edu Department of Statistics, University of Pretoria, Pretoria 0002, South Africa

© 2024 Informa UK Limited, trading as Taylor & Francis Group

Proportional-Hazards (CPH) regression model. This model evaluates the effects of covariates on survival times by estimating hazard functions. The hazard ratio is assumed to be constant over time for any two subjects [19]. The Cox regression imposes constraints on the model, ensuring that a particular feature's influence does not change over time [22].

Considering that each covariate must satisfy the Proportional Hazards (PH) assumption, this assumption is restrictive in real-world applications [22] and becomes questionable when dealing with a large number of covariates [42]. Over the years, efforts have been made to extend the Cox model to include predictors that do not satisfy the PH assumption. When the PH assumption of the Cox regression model is violated, the influence of the covariate is said to be time-varying. A covariate that changes over time throughout the follow-up period is said to be time-varying [41]. Investigators often model the covariate's baseline value and ignore the covariate's change with time. According to Kalbfleisch and Prentice [15] there are two types of time-varying covariates namely; internal and external. Internal covariates are those that can be affected by the survival status and they can be impacted by the failure process. The CD4 (cluster of differentiation 4) counts, procedural history, blood pressure, and the Karnofsky score are known examples. On the other hand, external covariates can directly affect the failure mechanism without being involved in the failure mechanisms. Examples include; an individual's age in a long-term follow-up study or the amount of pollution in the air as a factor that can trigger asthma attacks. There are alternative ways to adjust the Cox regression model to accommodate these types of covariates, namely; stratification and extending the Cox proportional hazards model [17], respectively.

The alternatives to parametric and semi-parametric models that have been used in practice include several traditional machine learning techniques such as the nearest neighbours, kernel methods, penalized least squares, neural networks (NN), and data partitioning techniques such as classification and regression trees (CART) and Random forests [6,13,26]. However, new developments in deep learning, which make use of advanced neural network designs, have gained popularity for estimating survival probability [20,22,38]. Deep neural networks (DNNs) have, in particular, been merged with discrete-time survival models to efficiently model survival data [2]. For instance, Nnet-survival is a discrete-time survival model specifically developed to integrate with NNs [9]. Fotso [8] proposed the neural multi-task logistic regression (N-MTLR) model as its base and a deep learning architecture as its core for modelling discrete-time covariates. As an alternative to the traditional methods of estimating the hazard function, Luck et al. [22] formulated a novel deep learning approach that directly models the survival function. This technique, which is based on the idea of multi-task learning, was created for predicting the survival times of graft patients. Lee et al. [21] developed a novel deep learning architecture designed to perform survival analysis with competing risks, incorporating longitudinal measurements.

In the DNN framework for survival data modelling, the follow-up time is divided into predetermined intervals. The conditional hazard/probability is then estimated for each time interval [9]. The DNNs' discrete-time models are more flexible than the Cox model and it does not depend on the PH assumption [42]. Bayesian and non-Bayesian DNN for survival data have been developed. However, their performance in predicting survival probabilities had until recently provided mixed results [23,29,36]. These mixed results arose from the architecture and the loss function used while building the neural network. Additionally, when using censored observations, outcomes are only partially

observed, making it challenging to directly use conventional deep learning algorithms [32]. In order to address this issue, Steingrimsson and Morrison [32] created deep learning algorithms that were specifically designed for potentially censored outcomes. These algorithms address censoring by using a censoring-unbiased transformation in place of the unobservable loss function generally employed when there is no censoring.

These techniques are then used to calculate both the restricted mean survival and the survival probabilities. To handle censored data, DNNs such as recurrent neural networks [10] convolutional neural networks [43] require cost functions. However, Zhao and Feng [42] and Feng and Zhao [7] avoided the problem of using censored data by using the jackknife pseudo observations in place of the observed survival times. The pseudo values are then used as the quantitative response variable in a regression analysis based on the DNN.

To build the neural networks introduced by Zhao and Feng [42] and Feng and Zhao [7], jackknife pseudo survival probabilities of the existing datasets are created first using the discrete-time survival framework, and then a new regression outcome is created. A DNN model built using the new outcome variable has the potential to correctly model time-varying predictors and time-varying effects. However, there have not been studies to examine how robust these neural networks are in situations where the survival data has complex time-varying predictors and time-varying effects.

In this article, we first adapt the Deep Neural Network (DNN) architecture introduced by Zhao and Feng [42] and its extension by Feng and Zhao [7], known as the Deep-Pseudo Survival Neural Network (DSNN) model, to accommodate time-varying covariates. We then compare the performance of this modified DSNN with existing models for time-varying covariates, including the extended Cox, Dynamic-DeepHit, and the multivariate joint models, for predicting survival probabilities in the presence of time-varying predictors.

The rest of the paper is structured as follows: Section 2 introduces the extended Cox model, Dynamic-Deephit, the multivariate joint model, and the DSNN. Section 3 is dedicated to an extensive simulation study. In Section 4, we analyze a real-life dataset to assess the practical effectiveness of deep neural networks. Finally, the paper concludes with Sections 5 and 6, where we provide our conclusions and engage in discussions.

2. Methodology

In this section, the Cox model and the extended Cox model are briefly presented. Next, the Dynamic-Deephit model, the multivariate joint model, the deep-pseudo survival neural network (DSNN) are introduced and finally, the estimation of pseudo-conditional probabilities are discussed.

2.1. Extended Cox model

The Cox Proportional Hazard (CPH) model is a technique that is frequently used to analyse survival data. However, when dealing with complex survival outcomes that contain intricate inter-dependencies between variables, the basic assumption that the outcome may be explained as a linear combination of covariates is too restrictive. Therefore, there is a need to develop models that can predict these intricate survival outcomes.

The hazard function in the CPH model is dependent on time t and a vector of covariates $\mathbf{X}_i$, $i = 1, \ldots, n$ through:

$$h\left(t, \mathbf{X}_i\right) = h_0\left(t\right) \exp\left(g\left(\mathbf{X}_i\right)\right), \tag{1}$$

where $h_0(t)$, is the baseline hazard function, and $g(\mathbf{X}_i) = \boldsymbol{\beta}^\top \mathbf{X}_i$ in which $\boldsymbol{\beta}^\top = (\beta_1, \beta_2, \ldots, \beta_p)$ are the regression coefficients. The proportional hazards assumption requires a time-fixed covariate vector, $\mathbf{X}_i$. The coefficients of the CPH model can be estimated by maximising the partial likelihood [4]. The overall partial likelihood is given as:

$$L = \prod_{i=1}^{n} \left[\frac{\exp\left(g\left(\mathbf{X}_i\right)\right)}{\sum_{l \in \mathbb{R}_i} \exp\left(g\left(\mathbf{X}_l\right)\right)} \right]^{\delta_i}$$

where δ_i is the censoring indicator. Subsequently, the log partial-likelihood is:

$$l = \log\left(\prod_{i=1}^{n} \left[\frac{\exp\left(g\left(\mathbf{X}_i\right)\right)}{\sum_{l \in \mathbb{R}_i} \exp\left(g\left(\mathbf{X}_l\right)\right)} \right]^{\delta_i} \right) = \sum_{i=1}^{n} \delta_i \left(g\left(\mathbf{X}_i\right) - \log\left[\sum_{l \in \mathbb{R}_i} \exp\left(g\left(\mathbf{X}_l\right)\right) \right] \right).$$

The CPH model's restrictive assumption makes it very difficult to apply in real-life scenarios especially when the assumption is violated. This assumption is often violated in the presence of time-varying covariates. Therefore, Equation (1) is expanded to incorporate time-varying covariates, as discussed in previous works by [33,34]. Extending the CPH model to accommodate time-dependent variables is one of the model's strengths [33].

$$h\left(t, \mathbf{Y}_i(t)\right) = h_0 \exp\left(f\left(\mathbf{Y}_i\left(t\right)\right)\right), \tag{2}$$

where $f(\mathbf{Y}_i(t)) = \boldsymbol{\beta}^\top \mathbf{X}_i + \boldsymbol{\gamma}^\top \mathbf{X}_i(t)$ and $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ are the vectors of coefficients for the time-fixed and time-varying covariates, respectively. It is clear from Equation (2) that when $\boldsymbol{\gamma} = 0$, we have the original CPH model, and therefore, the PH assumption holds. The inference on the extended model is similar to the CPH model. The primary distinction between these two approaches lies in the fact that the time-dependent covariate values undergo changes at the risk set. In addition, the procedure for fitting the extended model is much more computationally intensive and complex. This complexity arises due to the intricacies of the risk sets used in constructing the likelihood function, as highlighted by Ata and Sözer [1].

2.2. Dynamic-DeepHit

Dynamic-DeepHit is a novel deep learning architecture designed to perform survival analysis with competing risks, incorporating longitudinal measurements. The model consists of shared and event-specific subnetworks that learn from the temporal patterns in the data [21]. The aim is to estimate the joint distribution of the first hitting time and competing events based on time-varying covariates. This estimate enables computation of cause-specific cumulative incidence functions (CIFs) and survival probabilities. Let the longitudinal history for each subject i be denoted as $X_i = \{x_{i1}, \ldots, x_{iJ_i}\}$, with missing

data indicators $M_i = \{m_{i1}, \ldots, m_{ij_i}\}$ and time intervals between successive measurements $\Delta_i = \{\delta_{i1}, \ldots, \delta_{ij_i}\}$, where δ_{ij} represents the elapsed time between two adjacent measurements, $t_{ij+1} - t_{ij}$. The training set is represented as a tuple $D = \{(X_i, M_i, \Delta_i, \tau_i, k_i)\}_{i=1}^{N}$, where τ_i is the event time and k_i indicates the type of event.

The architecture comprises two subnetworks: a shared subnetwork handling the longitudinal history, and cause-specific subnetworks estimating the joint distribution of time-to-event and competing risks. The shared subnetwork includes a Gated Recurrent Unit (GRU)-based Recurrent Neural Network (RNN) that processes longitudinal data at irregular intervals with missing observations. For each time step j, the GRU updates the hidden state h_j based on the input (x_j, m_j, δ_j), computed as follows:

$$\begin{aligned} z_j &= \sigma(W_z h_{j-1} + U_z[x_j, m_j, \delta_j] + b_z), \\ r_j &= \sigma(W_r h_{j-1} + U_r[x_j, m_j, \delta_j] + b_r), \\ \tilde{h}_j &= \tanh(W_h(r_j \odot h_{j-1}) + U_h[x_j, m_j, \delta_j] + b_h), \\ h_j &= (1 - z_j) \odot h_{j-1} + z_j \odot \tilde{h}_j, \end{aligned} \tag{3}$$

The hidden state at time step h_j represents the GRU's internal memory and carries information from previous time steps. The hidden state from the previous time step is denoted as h_{j-1}. The update gate z_j, along with the weight matrices W_z, U_z, and the bias vector b_z, parametrizes the transformation of the previous hidden state and the input variables. The sigmoid activation function, σ, ensures that the output values range between 0 and 1. The update gate z_j controls how much of the previous hidden state h_{j-1} is retained in the current hidden state h_j. The reset gate, denoted as r_j, determines how much of the previous hidden state should be forgotten or reset, modulating the influence of the previous memory on the current step. The reset gate is calculated using the weight matrices W_r, U_r, and the bias vector b_r, which determine the effect of the previous hidden state and the input on the reset gate. The candidate hidden state $\tilde{h}_j$ is proposed at time step j and is computed as shown in Equation (3). In addition, the weight matrices W_h, U_h, and the bias b_h transform the input and the previous hidden state. The *tanh* function (hyperbolic tangent) ensures the candidate hidden state is bounded between -1 and 1. The symbol $\odot$ represents element-wise multiplication of the reset gate r_j and the previous hidden state h_{j-1}. The final hidden state h_j is a combination of the previous hidden state h_{j-1} and the candidate hidden state $\tilde{h}_j$, modulated by the update gate z_j. Here, $(1 - z_j)$ decides how much of the old hidden state h_{j-1} is carried forward, while z_j decides how much of the new candidate hidden state $\tilde{h}_j$ should be included in the final hidden state. The temporal attention mechanism assigns a weight a_j to each hidden state h_j to emphasize relevant parts of the history for risk prediction. The context vector c is then calculated as a weighted sum of the hidden states:

$$c = \sum_{j=1}^{J-1} a_j h_j, \quad a_j = \frac{\exp(e_j)}{\sum_{j'} \exp(e_{j'})}, \quad e_j = f_a(h_j, x_J, m_J), \tag{4}$$

where f_a is a two-layer feedforward network scoring the importance of hidden states.

The cause-specific subnetworks take the context vector c and the final measurement (x_J, m_J) as inputs to output the joint probability of event occurrence at time τ, $o^*_{k,\tau} =$

$\hat{P}(T = \tau, k = k \mid X^*)$. The final estimated CIF is computed as:

$$\hat{F}_{k^*}(\tau^* \mid X^*) = \sum_{t_j^* < \tau \leq \tau^*} \frac{o_{k,\tau}^*}{1 - \sum_{k \neq \emptyset} \sum_{n \leq t_j^*} o_{k,n}^*},$$

where $o_{k,\tau}^*$ is the probability output of the cause-specific subnetwork. While the Dynamic-DeepHit model was originally designed for survival outcomes with competing risks, it has been adapted in this paper to focus on a single event (e.g. death) while retaining the shared subnetwork based on a Gated Recurrent Unit (GRU) recurrent neural network (RNN), as shown in Equation (3). This retains the core elements necessary for processing longitudinal data. The GRU-based RNN forms the backbone of the shared subnetwork, effectively handling irregular time intervals and missing data. A temporal attention mechanism (Equation (4)) highlights the most relevant hidden states in the time-varying covariates. The output layer directly predicts the probability of an event occurring at each time step, with the survival function estimated from these predictions. The model's loss function includes a negative log-likelihood for survival time, a ranking loss to ensure proper ordering of survival probabilities, and a prediction loss for handling time-varying covariates. Therefore, the overall loss function is given by the sum of the following loss functions [21]; the Log-Likelihood Loss; the Ranking Loss; and the Prediction Loss as described below.

The log-likelihood loss is defined to maximize the likelihood of the observed event times and causes. For an individual i, the log-likelihood loss is defined as:

$$\mathcal{L}_1 = -\sum_{i=1}^{N} \left[\mathbf{1}(k^i \neq \emptyset) \log \left(\frac{o_{k^i,\tau^i}^i}{1 - \sum_{k \neq \emptyset} \sum_{n \leq t_{j^i}^i} o_{k^i,n}^i} \right) + \mathbf{1}(k^i = \emptyset) \cdot \log \left(1 - \sum_{k \neq \emptyset} \hat{F}_k(\tau^i | X^i) \right) \right].$$

The first term captures the information provided by uncensored subjects. The second term follows from the knowledge that they are alive at the censoring time, and so the first hitting time of each event $k \in K$ occurs after the given censoring time. The $\emptyset$ represents a null class or state, indicating the absence of an event or default state in classification problems. The indicator functions $\mathbf{1}(k^i \neq \emptyset)$ and $\mathbf{1}(k^i = \emptyset)$ distinguish between cases where the observation belongs to a meaningful class versus the null state.

The ranking loss is designed to improve the discriminative performance of the model by ensuring that the predicted CIFs respect the observed ordering of event times. It is defined as:

$$\mathcal{L}_2 = \sum_{k=1}^{K} \alpha_k \sum_{i \neq j} A_{kij} \cdot \eta(\hat{F}_k(s^i + t_{j^i}^i \mid X^i), \hat{F}_k(s^i + t_{j^j}^i \mid X^j)).$$

Here, the indicator $A_{kij} = 1(k_i = k, s_i < s_j)$ identifies acceptable pairs (i, j) for event k, where $k_i = k$ denotes the event for individual i and s_i and s_j represent their respective event times. The hyperparameter $\alpha_k \geq 0$ controls the trade-off between ranking losses associated with the k-th event. For simplicity, it is assumed that $\alpha_k = \alpha$ for all $k = 1, \ldots, K$, implying equal treatment of all events. The loss function is defined as

$\eta(a,b) = \exp\left(-\frac{a+b}{\sigma}\right)$, where σ is a scaling parameter that adjusts the magnitude of the penalty. Incorporating this ranking loss into the total loss function L_2 penalizes incorrect pairwise orderings and encourages correct ordering with respect to each competing event.

The prediction loss is introduced to handle missing measurements in the covariates. It penalizes the model if it fails to predict the auxiliary tasks derived from the covariates. The prediction loss is defined as:

$$\mathcal{L}_3 = \beta \cdot \sum_{i=1}^{N} \sum_{j=0}^{J-1} \sum_{d \in \mathcal{I}} (1 - m^i_{j+1,d}) \cdot \zeta(x^i_{j+1,d}, y^i_{j,d}),$$

where $m^i_{j+1,\delta}$ is a missing indicator, $\beta \geq 0$ is a hyperparameter, and the continuous and binary covariates are represented by $\zeta(a,b) = |a-b|^2$ or $\zeta(a,b) = -a\log b - (1-a)\log(1-b)$. $\mathcal{I}$ is selected as a set of time-varying covariates that the network's formalization is based on.

Training is conducted by minimizing a total loss function $L_{total} = L_1 + L_2 + L_3$, where L_1 is the negative log-likelihood of the joint distribution of the first hitting time and event, L_2 is a ranking loss to ensure proper event ordering, and L_3 is a prediction loss regularizing the network to predict time-varying covariates accurately. These components collectively improve the prediction of survival probabilities over time. The architecture's input layer processes time-varying covariates and missing data indicators, while the output layer predicts survival probabilities at each time point, making the model suitable for survival analysis with longitudinal data.

2.3. Multivariate joint model

The multivariate joint model (MVJM) is employed to simultaneously analyze multiple longitudinal outcomes and time-to-event data. This approach provides a comprehensive framework that captures the correlations between these processes, allowing for more accurate and individualized risk predictions. The MVJMs are particularly advantageous because they account for the correlation between different longitudinal outcomes and the time-to-event process. Integrating both submodels allows us to simultaneously model the progression of multiple outcomes and their relationship with the time-to-event process. An MVJM consists of two submodels:

(1) a multivariate longitudinal data model, which tracks the evolution of multiple correlated outcomes over time, and
(2) a time-to-event data model, which focuses on the occurrence of an event such as survival or failure.

Let $\mathbf{Y}_{ik}(t_{ijk})$ denote the j-th observed value of the k-th longitudinal outcome for subject i, measured at time t_{ijk}, for $i = 1,\ldots,N$, $k = 1,\ldots,K$, and $j = 1,\ldots,n_{ik}$. There are numerous modelling approaches for multivariate longitudinal data [12]. A generalized linear mixed model (GLMM) is a common approach, where measurements for different outcomes can be recorded at different times between patients and outcomes, and is given

by:

$$h_k\left(\mathbb{E}\left[\mathbf{Y}_{ik}(t_{ijk})\right]\right) = \mu_{ik}(t_{ijk}), \tag{5}$$

where $h_k(\cdot)$ denotes a known one-to-one link function for the k-th outcome, and $\mu_{ik}(\cdot)$ is the linear predictor:

$$\mu_{ik}(t_{ijk}) = \mathbf{X}_{ik}^{(1)}(t_{ijk})\beta_k^{(1)} + \mathbf{Z}_{ik}(t_{ijk})b_{ik}, \tag{6}$$

where $\mathbf{X}_{ik}^{(1)}(t_{ijk})$ and $\mathbf{Z}_{ik}(t_{ijk})$ are row-vectors of covariates for subject i associated with fixed and random effects respectively, which can vary by outcome; $\beta_k^{(1)}$ is a vector of fixed effects parameters for the k-th outcome; and b_{ik} is a vector of subject-specific random effects for the k-th outcome. We denote the stacked vector of subject-specific random effects for all K outcomes by $\mathbf{b}_i = (b_{i1}^T, b_{i2}^T, \ldots, b_{iK}^T)^T$.

Cox's proportional hazards (PH) semiparametric model [12] has been a common choice for the time-to-event submodel when modelling continuous event times. For a single failure-time per subject, the hazard function for subject i at time t is given by

$$\lambda_i(t) = \lambda_0(t)\exp\left(\mathbf{X}_i^{(2)}(t)\beta^{(2)} + \mathbf{W}_i(t)\right), \tag{7}$$

where $\mathbf{X}_i^{(2)}(t)$ is a (possibly time-varying) row-vector of external covariates; $\lambda_0(t)$ denotes the baseline hazard function; $\beta^{(2)}$ is a vector of fixed effects parameters; and $\mathbf{W}_i(t)$ is a latent process that captures the association structure between the measurement and event processes. When the PH assumption is not met, alternative modelling frameworks can be exploited, such as accelerated failure time models, which can be written directly in terms of the survival function, $S_i(t)$, as:

$$S_i(t) = S_0\left(\exp\left(\mathbf{X}_i^{(2)}(t)\beta^{(2)} + \mathbf{W}_i(t)\right)t\right), \tag{8}$$

where $S_0(\cdot)$ is the baseline survival function that depends on the parametric family used for modelling, and all other parameters are defined as in the PH model (3). Discrete event times can also be jointly modelled with longitudinal data, particularly for selection models, which is applicable to situations of interval-censored continuous event times and pre-defined measurement schedules. Models here might include logistic or probit regression models [12].

2.4. Deep survival neural network

In this study, we propose a DSNN (Figure 1) that is based on time-varying covariates to model survival probabilities. Figure 1 shows that the DSNN model takes into account both covariates (represented by Z_i) and time-related information (represented by the dummy indicators $d(t_0), \ldots, d(t_J)$) as inputs. These inputs are then processed through the neural network layers, which include dense layers, dropout layers, and activation functions.

To use the DSNN, the first step involves generating survival probabilities for both time-varying and time-invariant covariates. These survival probabilities are then replaced with pseudo-values, which serve as substitutes for the actual survival times when implementing standard DNN models for regression problems with quantitative outcomes. The calculation of pseudo-values is based on the survival times, censoring times, and censoring indicators of the subjects in the study [7].

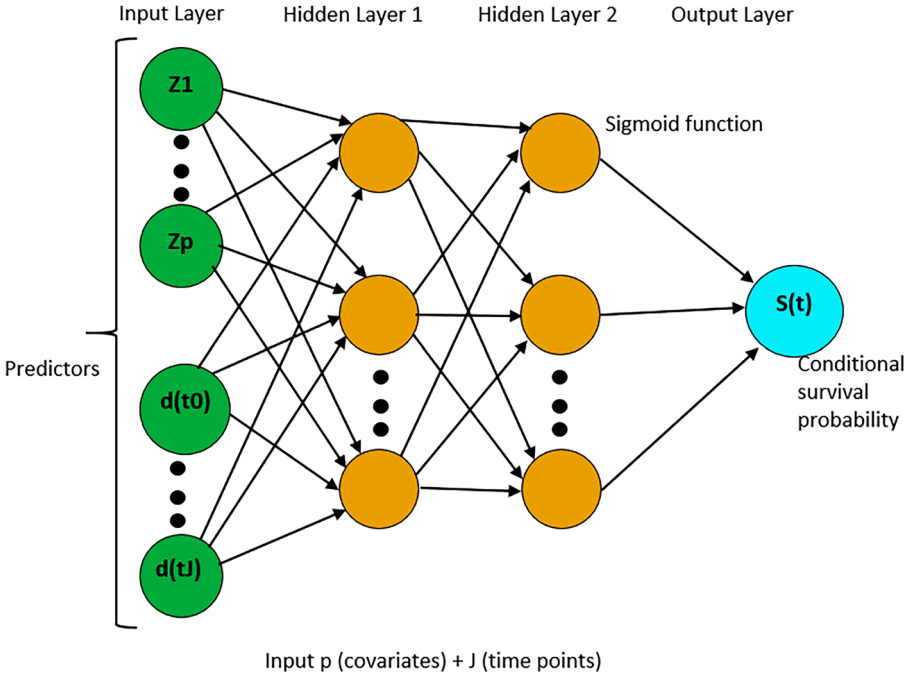


Figure 1. The figure represents the architecture of the Deep-pseudo Survival Neural Network (DSNN) model. In this model, Z_i represents the covariates, which may include time-varying covariates. In addition, the variables $d(t_0), \ldots, d(t_J)$ are dummy indicators of time, where $t_0, \ldots, t_J$ represent J specific time-points or intervals.

Specifically, we let T_i^* represent the survival time for subject i (where $i = 1, \ldots, n$), C_i be the censoring time, T_i be the observed survival time ($T_i = \min(T_i^*, C_i)$), and δ_i be the censoring indicator ($\delta_i = I(T_i^*, C_i)$). It's assumed that censoring times and survival times are independent. The covariates are denoted as $\mathbf{X}_i = (x_{i1}, \ldots, x_{id})$, with some of the covariates being time-varying. We observe that employing a generalized linear model (GLM) with a logit link to model the survival probability at a specific time t as a binary outcome ($I(T_i > t)$) is impractical for all subjects due to the presence of censoring. Therefore, to estimate survival probabilities in the presence of censoring, the Kaplan-Meier (KM) estimator is employed at each time point. For both censored and uncensored subjects, a pseudo survival probability is calculated using the jackknife method. Specifically, the pseudo value for the ith subject is calculated as:

$$\hat{S}_i(t) = n\hat{S}(t) - (n-1)\hat{S}^{-i}(t). \tag{9}$$

where, $\hat{S}(t)$ represents the Kaplan-Meier estimator of the survival function $S(t)$ for all n subjects, while $\hat{S}_i(t)$ denotes the Kaplan-Meier estimate achieved by removing the ith subject from the sample, resulting in a sample size of $n-1$.

Where, $\hat{S}(t)$ is the KM estimator of $S(t)$ for all n subjects, and $\hat{S}_i(t)$ is the KM estimate using a sample size of $n-1$ obtained after removing the ith subject. Regardless of censoring and in standard regression analysis, the KM estimator $\hat{S}(t)$ is used as the continuous response variable.

In the discrete-time survival analysis framework, we apply the pseudo-observations method in this study. The framework entails dividing the follow-up times into J intervals. Thereafter, the pseudo conditional survival probability, which represents the likelihood of surviving the interval given prior survival, is calculated for each interval. Individuals who are still at risk within a specific time window $(t_j, t_j + 1)$ are referred to as R_j. the pseudo conditional survival probability $\hat{S}_{ij}(t_{j+1} \mid R_j)$ is estimated as follows:

$$\hat{S}_{ij}(t_{j+1} \mid R_j) = R_j \hat{S}(t_{j+1} \mid R_j) - (R_j - 1)\hat{S}^{-i}(t_{j+1} \mid R_j) \tag{10}$$

All individuals are considered at risk during the initial interval $(0, t_1)$ (i.e. $R_0 = n$). It's important to note that the pseudo probability can be either 0 or 1 in the absence of censored data. However, in the presence of censoring, it can be a real value, potentially exceeding 1 or falling below 0. The choice of time intervals is crucial, with recommendations to use as few as five evenly spaced intervals across the event time scale [42]. The computation of pseudo conditional probabilities as described in the equations is facilitated using the R function 'getPseudoConditional' (https://github.com/lilizhaoUM/DNNSurv).

The DSNN is employed to directly learn the survival time distribution and explore variable-risk relationships over time without relying on underlying stochastic process assumptions. DSNN exemplifies models that naturally extend traditional parametric models while providing strong interpretability. Such models are becoming increasingly popular in the field of survival data analysis [40].

3. Simulation study

Simulation studies provide an important statistical tool for evaluating survival methods. Therefore, there is a need to use appropriate data-generating processes to effectively evaluate the survival models in question. In this paper, we aim to evaluate the robustness of DSNN models in analysing survival datasets in the presence of time-varying covariates via a simulation study.

3.1. Simulation setup

We simulate time-varying covariates from the Weibull distribution. The Weibull distribution is often or widely selected or preferred by researchers, scientists, or practitioners for generating survival times due to its flexibility and versatility in modelling a wide range of survival data patterns.

Consider the Weibull distribution with parameters (λ, α), its baseline cumulative hazard function is defined as:

$$H_0(t) = \lambda t^{\alpha}, \quad \lambda > 0, \quad \alpha > 0. \tag{11}$$

Given a time-invariant vector of covariates $\mathbf{X}$ with coefficient vector $\boldsymbol{\beta}$, the cumulative hazard is invertible, therefore:

$$t = \left(\frac{-\log(u)}{\lambda \exp\left(\boldsymbol{\beta}^{\top}\mathbf{X}\right)} \right)^{\frac{1}{\alpha}},$$

where $U \in \mathcal{U}(0, 1)$.

In this simulation study, the data is simulated according to a study by Yao et al. [37]. The authors simulated time-independent and time-dependent covariates with fixed coefficients. Observations from N subjects are generated independently with p covariates $\mathbf{X}_1, \ldots, \mathbf{X}_p$. We set $p = 20$. Eight of these covariates are time-invariant: $\mathbf{X}_1, \mathbf{X}_{11} \sim \text{Bern}(0.5)$, $\mathbf{X}_2, \mathbf{X}_7, \mathbf{X}_{10} \sim \text{Unif}(0, 1)$, $\mathbf{X}_8 \sim \text{Unif}(1, 2)$, $\mathbf{X}_9$ follows a categorical distribution with possible outcomes $\{1, 2, 3, 4, 5\}$ with equal probability, $\mathbf{X}_{12}$ follows a categorical distribution with possible outcomes $\{0, 1, 2\}$ with equal probability. The others are time-varying, whose values are obtained at m randomly generated time points, different for each subject. In the simulations, we set $m = 11$. At each of these preset time points, for some time-varying covariates, the value is randomly re-simulated from its distribution: $\mathbf{X}_3, \mathbf{X}_{19} \sim \text{Bern}(0.5), \mathbf{X}_4 \sim \text{Unif}(0, 1), \mathbf{X}_5$ and $\mathbf{X}_{14}$ both follow a categorical distribution with possible outcomes $\{1, 2, 3, 4, 5\}$ with equal probability; for other time-varying covariates, the value is re-simulated following particular patterns: $\mathbf{X}_6$, whose initial value is randomly generated from $\{0, 1, 2\}$ with equal probability, which will choose to stay at the original value or move one level up but the largest value can only be 2; the changing pattern of $\mathbf{X}_{13}$ is always $0 \to 1$; the changing pattern of $\mathbf{X}_{16}$ is either $0 \to 1$ or $1 \to 2$; the changing pattern of $\mathbf{X}_{18}$ is $0 \to 1 \to 2$; values of $\mathbf{X}_{15}, \mathbf{X}_{17}$ and $\mathbf{X}_{20}$ is a linear function of the left-truncated time point of the interval with slope and intercept follows Unif $(0, 1)$. After simulating the time-varying covariates' values at each m time point, the true survival time T is then computed under different model setups and the right-censoring time C is generated independently. The details about this simulation process are presented in a study by Yao et al. [37].

Figure 2 shows the coefficients of the 20 covariates after 100 repeated training of the extended cox model. The 'stars' represent the mean value for each and the 'plus' indicates the true value of the coefficients. These coefficients were simulated to be very close to zero such that they have no effect on the survival outcome.

Figure 3 shows the coefficients of the 20 covariates that were simulated to be highly significant in influencing the survival outcome. The mean values after training the extended Cox model over 100 times are displayed and the true coefficient values. These values are far from zero and therefore significantly influence the survival outcome.

The characteristics of four time-varying continuous covariates are presented in Figure 4. The covariates, $\mathbf{X}_4, \mathbf{X}_{15}, \mathbf{X}_{17}$, and $\mathbf{X}_{20}$ were simulated to be time-dependent and therefore expected to have variations over time. The trajectories of these covariates among event and censored individuals are presented in Figure 5. From Figure 4, we can observe that covariates $\mathbf{X}_{20}, \mathbf{X}_{15}, \mathbf{X}_{17}$ are increasing while $\mathbf{X}_4$ is decreasing with time.

3.2. Evaluating model performance

3.2.1. Concordance index

To evaluate the DSNN model in the presence of a time-varying covariate, we used the time-dependent concordance index [18]. The concordance index (C-index), is particularly used for model evaluation. It measures how well a model predicts the ordering of sample event times [18,42]. The C-index can be defined as the fraction of all pairs of subjects whose predicted survival times are correctly ordered among all subjects that can actually be ordered [31]. It can also be defined as a measure of the percentage of all pairs of patients for whom we can successfully obtain ordered survival times. Therefore, for a randomly chosen pair of

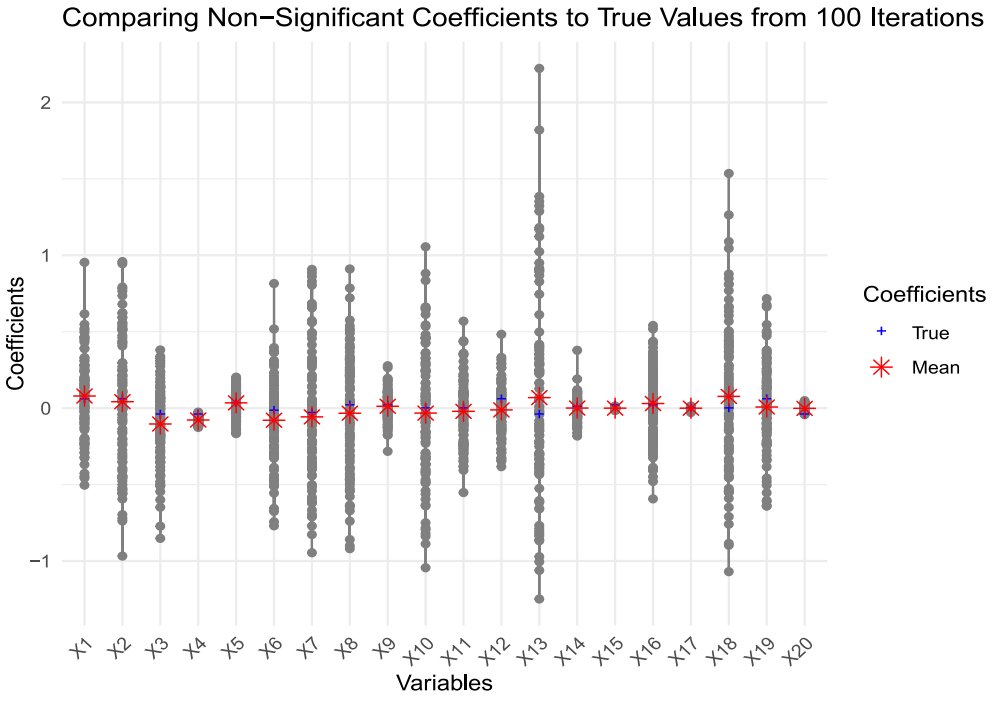


Figure 2. All 20 covariates simulated with non-significant coefficients.

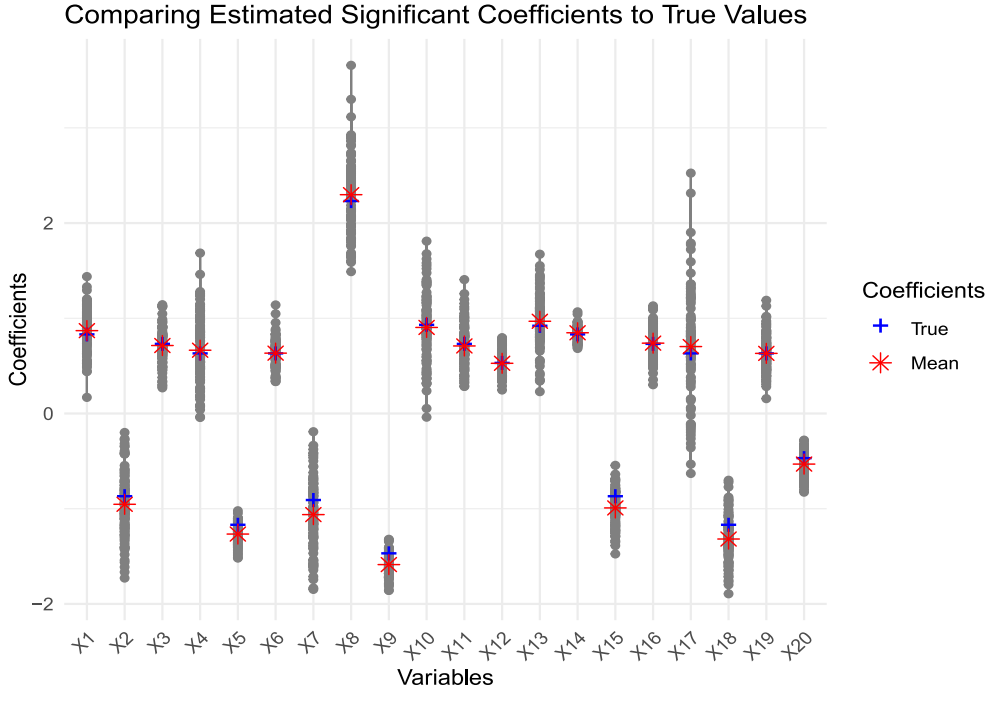


Figure 3. All 20 covariates simulated with significant coefficients.

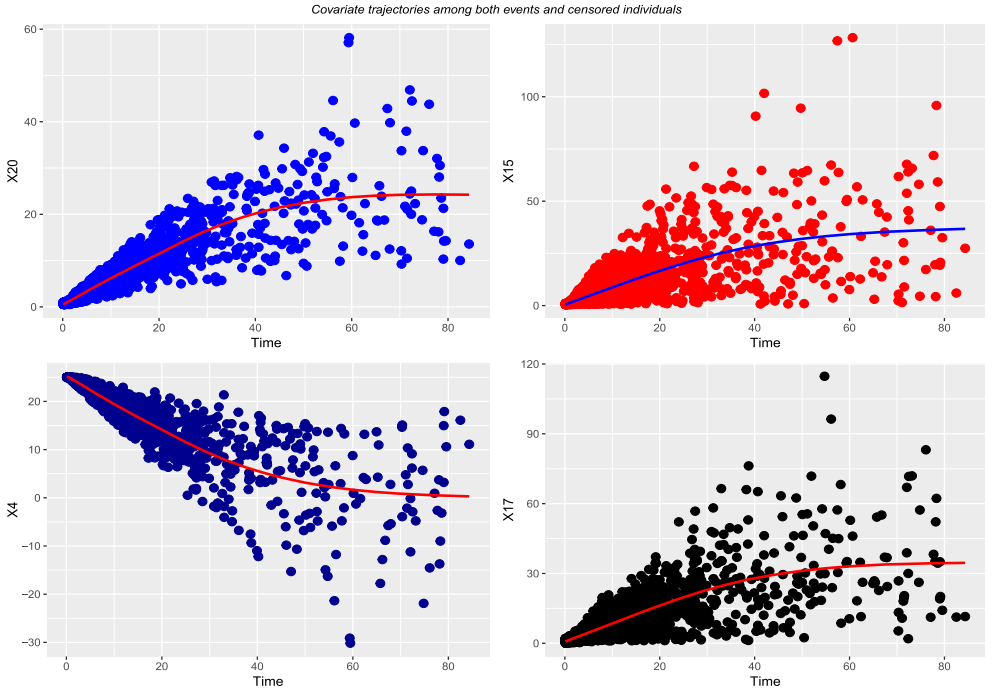
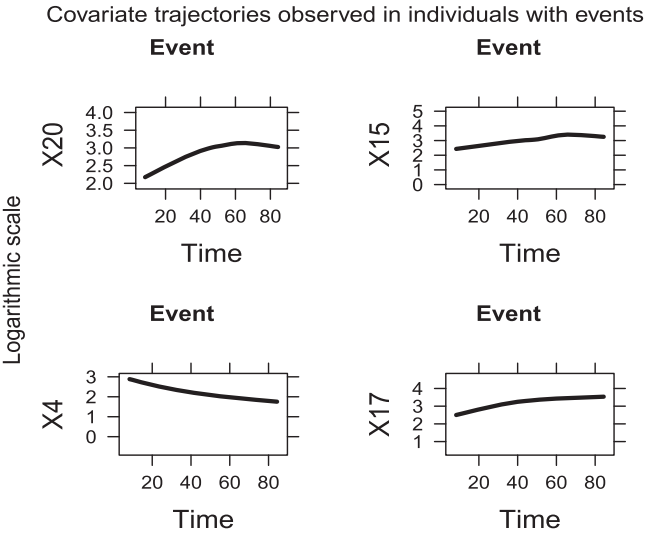


Figure 4. The trajectory and changes in the values of $\mathbf{X}_4$, $\mathbf{X}_{15}$, $\mathbf{X}_{17}$ and $\mathbf{X}_{20}$ with time.

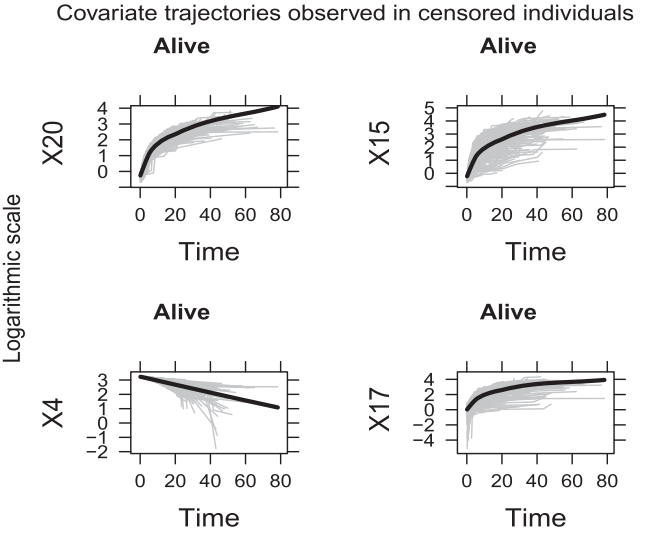
patients, we obtain an estimate such that the one who has the utmost anticipated survival is the one who would die last [11]. This method does not permit comparison between a censored observation and an observed even time. This is mainly because the precise moment of the occurrence of an event of interest is undiscovered. Other pairs of observations are very similar. Suppose we denote the observed and predicted survival times by Y_i and $\hat{Y}_i$, respectively. Then the C-index is defined as:

$$C = \frac{\sum_{i:\delta_i=1} \sum_{j:Y_i<Y_j} I(S(\hat{Y}_i|\mathbf{X}_i) < S(\hat{Y}_j|\mathbf{X}_j))}{|\text{Set of all concordant pairs}|}, \tag{12}$$

where $S(\cdot)$ is the estimated survival probability and $I(\cdot)$ is an indicator function. When the C-index is above 0.5, it suggests that there is better agreement among the pairs that are comparable [25], whereas a value of exactly 0.5 represents an average index for a random model. A value of 1 represents an ideal grading for the event times [16], which is indicative of a strong model. This study uses the C-index obtained from Leave-one-out cross-validation (LOOCV) to perform the model evaluation. The LOOCV is a special case of k-fold cross-validation in which the number of folds is the same as the number of instances [35]. It is one of the cross-validation methods in which each observation in the data set is considered the validation set and the rest of the observations (N-1) are considered the training set. It is used to estimate the performance of machine learning models when they are used to make predictions on data not used to train the model. The LOOCV is appropriate when you have a small dataset and it results in a reliable and accurate estimate of model performance [35].



Covariate trajectories among individuals with events.



Covariate trajectories among the censored individuals.

Figure 5. The plot shows the trajectory and changes in the values of $\mathbf{X}_4$, $\mathbf{X}_{15}$, $\mathbf{X}_{17}$ and $\mathbf{X}_{20}$ with time among the events and the censored individuals.

3.2.2. Brier score

The Brier score, as a proper scoring rule, concurrently addresses calibration, which refers to the statistical consistency between the predicted probability and the data, as well as sharpness, which refers to the predictive distribution's concentration [28]. The brier score is

defined as:

$$BS = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{P}(y_i = 1 \mid \mathbf{x}_i))^2, \tag{13}$$

where $\hat{P}(y_i = 1 \mid \mathbf{x}_i)$ is the estimated probability of event.

3.3. Results

In this study, we simulated two datasets with 200 unique individuals; such that the first dataset contained all the covariates that were simulated to be highly significant. The second dataset, contained all the covariates that were simulated to be non-significant. In addition, the two simulated datasets were analyzed using the extended Cox model, the Dynamic-DeepHit, multivariate joint model and the DSNN. For the DSNN, the following hyperparameters were used; a learning rate of 0.001, the Adam optimiser, a batch size of 256, and the model was trained for a total of 1000 epochs. The simulations were repeated 10 times at five different seeds giving a total of 50 independent simulations. All the mentioned models were trained and tested on the simulated datasets. The results including the estimated coefficients and the corresponding standard errors (se) from fitting the extended Cox model to the simulated data with highly significant time-varying covariates are presented in Table 1.

The results in Table 1 show that with one single run of the extended Cox model on the dataset containing highly significant time-varying covariates, all the covariates that were simulated to be significant are indeed demonstrating a significant effect on the outcome.

A single analysis on the extended Cox model on the dataset simulated with non-significant time-varying covariates is also presented in Table 2. The results presented in Table 2 show that almost all the covariates are non-significant at 0.01 level of significance in explaining the survival outcome.

Table 1. The extended Cox regression results for the simulated data with highly significant time-varying covariates.

	coef	exp(coef)	se(coef)	z	p
X1	0.73	2.07	0.25	2.86	0.00
X2	1.12	3.07	0.41	2.76	0.01
X3	−0.96	0.38	0.23	−4.18	0.00
X4	−2.00	0.13	0.42	−4.77	0.00
X5	2.64	13.98	0.20	13.25	0.00
X6	−2.44	0.09	0.22	−11.26	0.00
X7	−2.27	0.10	0.43	−5.29	0.00
X8	1.91	6.75	0.39	4.92	0.00
X9	1.57	4.81	0.14	11.59	0.00
X10	−1.59	0.20	0.42	−3.83	0.00
X11	−2.18	0.11	0.27	−7.97	0.00
X12	0.98	2.66	0.15	6.39	0.00
X13	−0.03	0.97	0.99	−0.03	0.97
X14	−1.61	0.20	0.14	−11.70	0.00
X15	0.85	2.34	0.68	1.26	0.21
X16	1.57	4.78	0.24	6.44	0.00
X17	0.34	1.41	1.66	0.21	0.84
X18	−2.32	0.10	0.86	−2.68	0.01
X19	0.98	2.66	0.23	4.20	0.00
X20	−0.72	0.48	0.33	−2.21	0.03

Table 2. The extended Cox regression results for the simulated dataset with non-significant time-varying covariates.

	coef	exp(coef)	se(coef)	z	p
X1	0.57	1.77	0.29	1.95	0.05
X2	0.12	1.13	0.55	0.22	0.83
X3	0.23	1.26	0.29	0.82	0.41
X4	−0.03	0.97	0.04	−0.89	0.37
X5	0.16	1.17	0.10	1.55	0.12
X6	−0.12	0.89	0.32	−0.36	0.72
X7	0.72	2.06	0.52	1.40	0.16
X8	−0.25	0.78	0.48	−0.52	0.60
X9	−0.05	0.95	0.10	−0.48	0.63
X10	0.40	1.48	0.48	0.82	0.41
X11	0.13	1.14	0.29	0.45	0.65
X12	−0.43	0.65	0.17	−2.49	0.01
X13	−0.65	0.52	0.60	−1.08	0.28
X14	−0.17	0.84	0.10	−1.70	0.09
X15	0.01	1.01	0.01	0.98	0.33
X16	−0.18	0.83	0.25	−0.74	0.46
X17	−0.01	0.99	0.01	−0.61	0.54
X18	−0.09	0.91	0.39	−0.23	0.82
X19	0.07	1.07	0.29	0.24	0.81
X20	−0.05	0.96	0.04	−1.25	0.21

The predictive performance of the four used models on the simulated datasets from two different scenarios are presented in Figures 6 and 7 using the C-index and the Brier score.

The results shown in Figure 6 indicate that the DSNN model exhibits a C-index ranging from 0.57–0.59 across all arbitrary time points. This performance is above 0.5, however, it is not large enough to conclude that the model has a high predictive performance on the dataset used. Given that we simulated data with covariates that are not significantly linked to the survival outcome, the predictive performance should have been 0.5. The Cox model, the Dynamic-DeepHit have the same performance in discrimination ability as the DSNN. The multivariate joint model has a C-index ranging from 0.47–0.61, showing its discrimination superiority in the presence of nonsignificant covariates. The results for Brier score in Figure 6 show that all the models have Brier score values that are approximately above 0.25 at most of the time-points. This shows that all the models including the DSNN model were good at calibrating and discriminating in the presence of nonsignificant covariates.

The results in Figure 7 show that the both DSNN model and the extended Cox model at all the arbitrary time points have a C-index above 0.8 however, the DSNN was able to learn better the predictive pattern in the dataset. The C-index for the joint model has a range of 0.57–0.72, indicating an acceptable performance. On the other hand, the discrimination ability of the Dynamic-DeepHit is lowest among the four models. Additionally, Figure 7 shows that the Brier score values for all models are below 0.2, except for the DSNN model, which exhibited poor performance in the time interval between 20 and 25 points.

4. Real data analysis

All the four models were trained on the Primary Biliary Cirrhosis (PBC) dataset. The used hyperparameters for the DSNN model are indicated in Table 3.

We have used the PBC, a publicly available dataset as a case study to assess how well the DSNN model performs compared to the extended Cox model, Dynamic-DeepHit and the

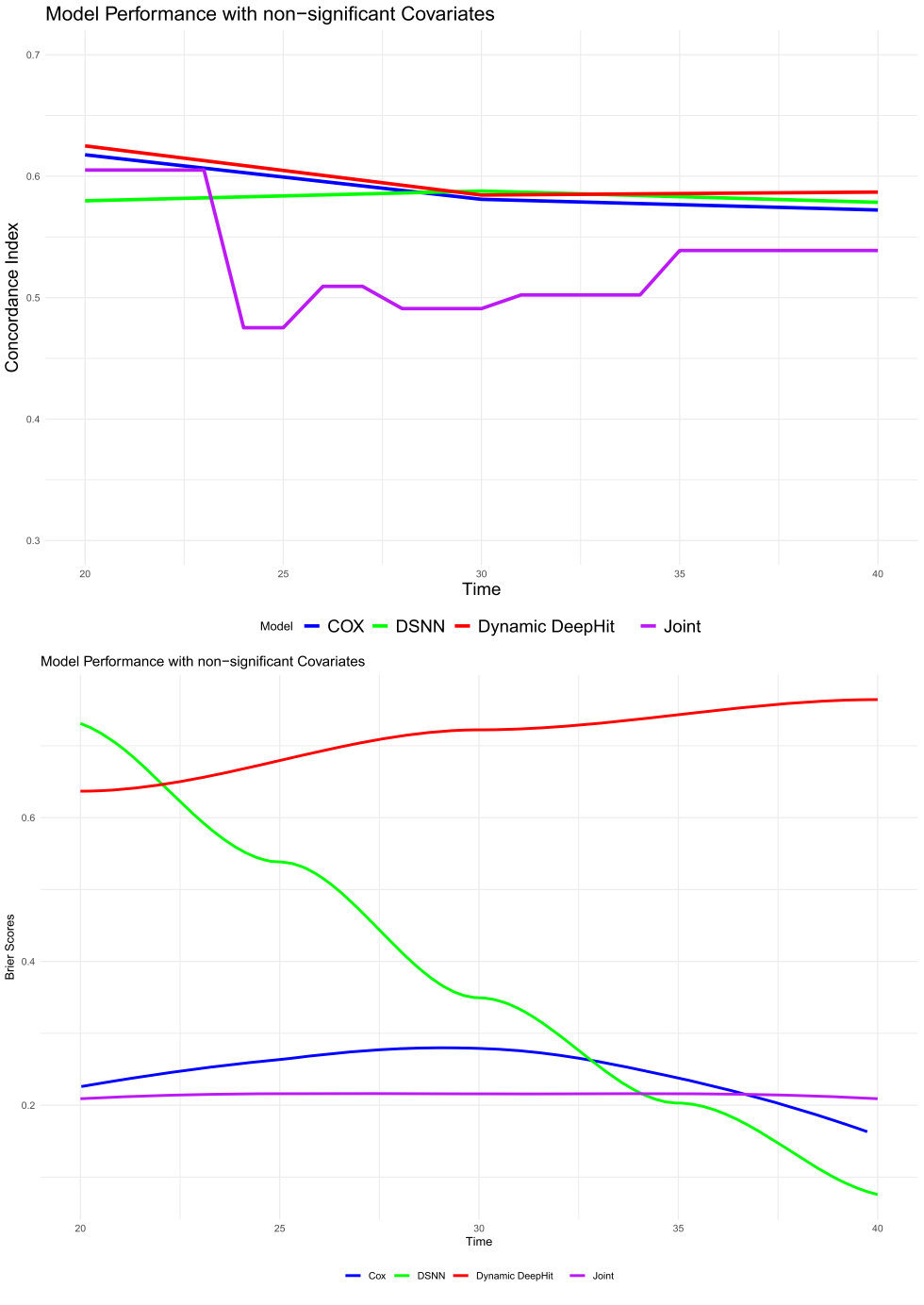


Figure 6. Models evaluation from the simulated dataset with nonsignificant covariates using C-index and Brier score.

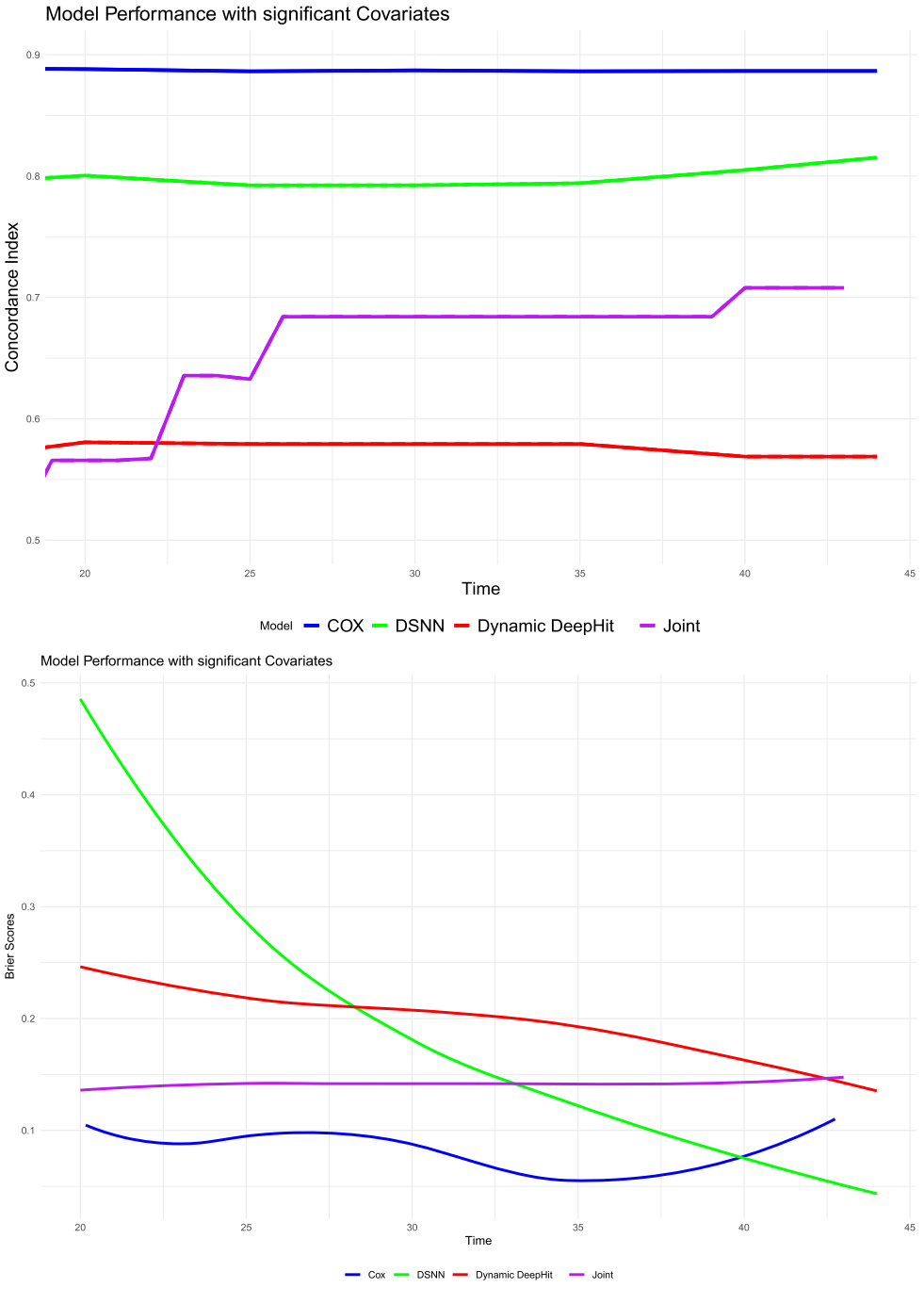


Figure 7. Models evaluation from the simulated dataset with significant covariates using C-index and Brier score.

Table 3. Hyperparameters for DSNN model used for the PBC real data analysis.

Hyperparameter	DSNN
Activation	'Relu'
Bias Initializer	Constant (0.0)
Dropout Rate	0.2
Optimizer Learning Rate	0.001
Loss Function	Mean Squared Error
Validation Split	0.2
Epochs	1000
Batch Size	256

multivariate joint model when it is used to estimate the survival function in the presence of time-varying covariates. A brief description of the dataset is given.

4.1. Data description

The Mayo Clinic Primary Biliary Cirrhosis Data, available in the R package survival, was utilized to study the effectiveness of D-penicillamine as a treatment. This dataset comprises 312 patients with primary biliary cirrhosis (PBC) who were enrolled in a randomized medical trial at the Mayo Clinic from January 1974 to May 1984 [5,24]. In this study, the primary outcome of interest is the time to death for patients with PBC. Medical measurements and other patient information were recorded as covariate values both at entry and at yearly intervals The Cox model was used to estimate the survival function for patients with PBC based on 12 noninvasive, easily collected covariates that require only a blood sample and clinical evaluation. These 12 covariates include age at entry, alkaline phosphatase (U/L), logarithm of serum albumin (g/dL), presence of ascites, aspartate aminotransferase (U/mL), logarithm of serum bilirubin (mg/dL), serum cholesterol (mg/dL), condition of edema, presence of hepatomegaly or enlarged liver, platelet count, logarithm of prothrombin time and presence or absence of spiders. As the study extended for another four years, a total of 1945 visits were generated. All 12 covariates, except age, became time-varying covariates in the follow-up data. By the end of the follow-up study, 169 out of the 312 patients were still alive, 140 had died, and three were lost to follow-up. The extended study enabled researchers to examine changes in prognostic variables as time-varying covariates. Figure 8 displays the time-varying trajectories of the covariates in the PBC dataset, confirming their time-dependent nature. The data is available in 'survival' package in R.

4.2. Results

The performance of all the models was evaluated on the PBC dataset in Figure 9 using the Brier score and the C-index. The results show that the C-index values are above 0.75 for all the models and the Brier scores are below 0.25, which underscore the promising performance of the DSNN model as well as its competitors on the real dataset.

5. Discussion

At all time points, the DSNN model achieved higher C-index values and lower Brier scores on the test data. The C-index values were above 0.7 and the Brier scores were below 0.25

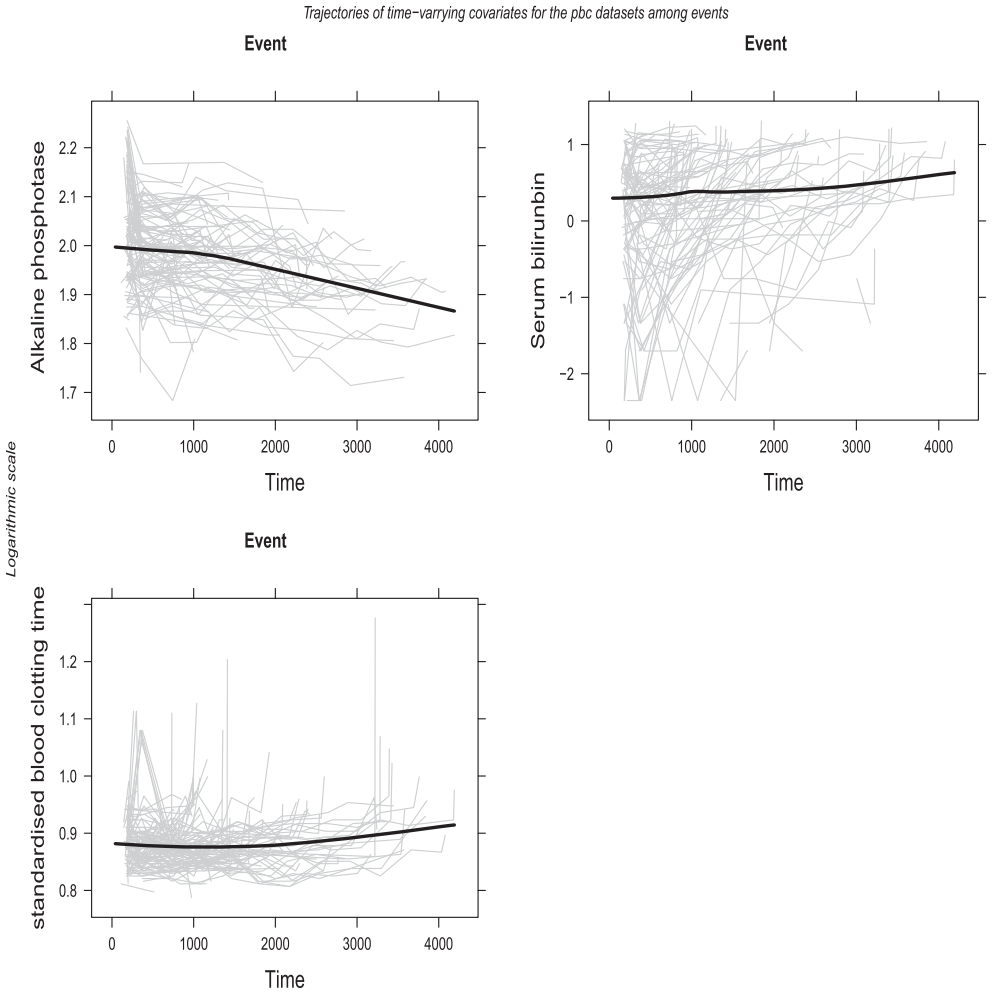


Figure 8. The trajectories in the values of the continuous time-varying covariates for the PBC over time.

indicating that the model made accurate predictions at the randomly selected time point. A model with a C-index > 0.7 and a Brier score < 0.25 indicates good performance [3,14,16]. This evidence strengthens the conclusion that the DSNN model performed effectively on the dataset where the covariates were simulated to be highly associated with the survival outcome. The DSNN model also performed well on the PBC dataset. The study yielded C-index values exceeding 0.78 and Brier scores below 0.25. These results underscore the promising performance of the DSNN model on the real dataset, providing additional statistical evidence of its predictive capability. Thus, the model's high predictability at the arbitrarily selected time-point in the PBC dataset supports its potential in real-world applications. Zeng et al. [39] also found that the covariates used in the PBC dataset are highly predictive of the survival outcome. The DSNN model performed comparebly well to the other models including the extended Cox model, Dynamic-DeepHit and the multivariate joint model.

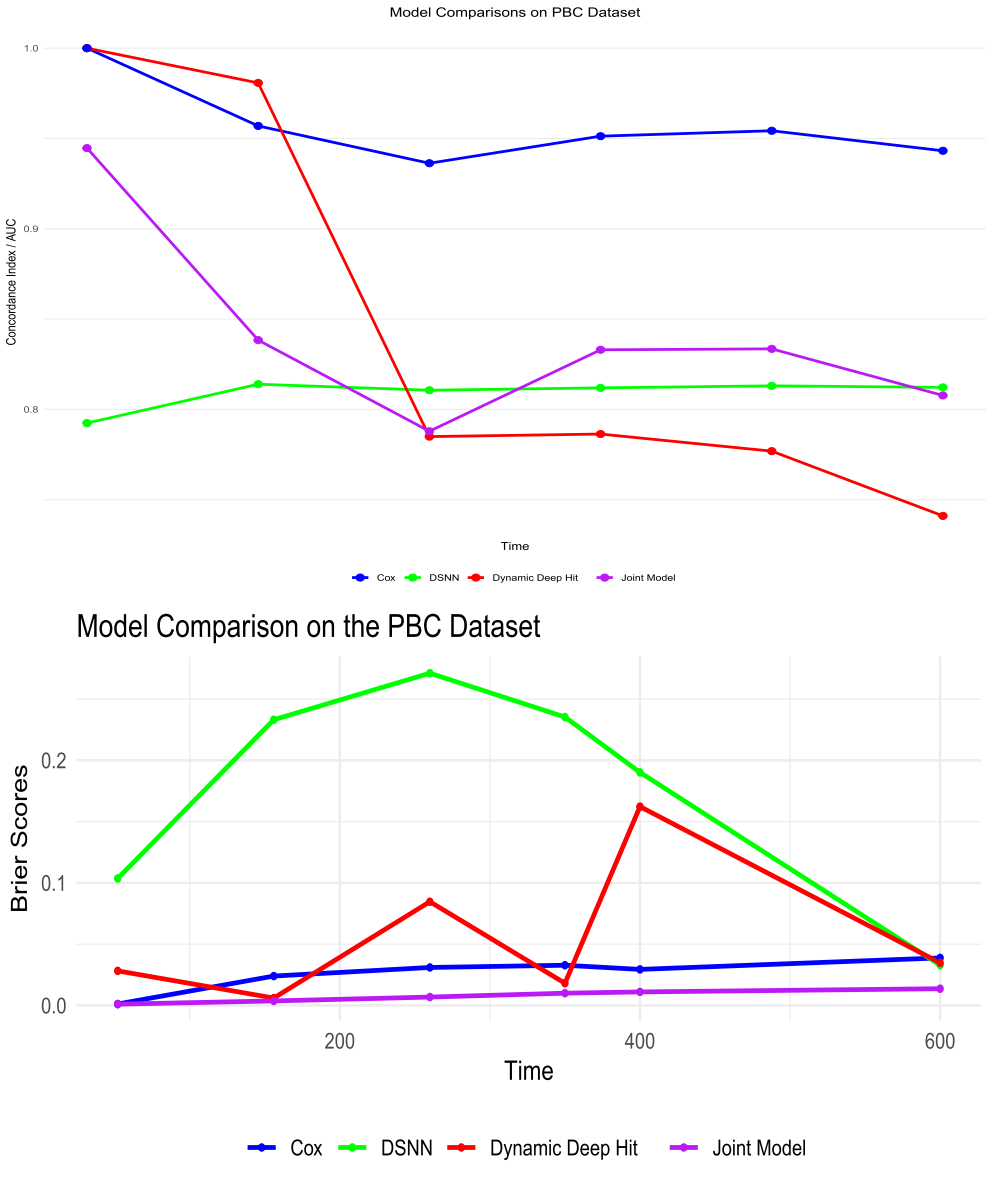


Figure 9. The figure displays the performance of both the DSNN model and the extended Cox model in the PBC dataset, evaluated based on the Brier score and the C-index.

6. Conclusion

In conclusion, the findings from this study indicate that the DSNN model performed well on the simulated dataset that was intentionally designed to have highly predictive time-varying covariates. This is evidenced by the high C-index values and low Brier scores obtained on the test datasets, indicating its ability to accurately predict outcomes at the selected random time point. The DSNN also showed a good performance in real-world application by providing a good predictive performance on the PBC dateset.

Acknowledgments

We would like to thank the three reviewers for their valuable comments, which have significantly improved our work. We would like to thank the Emmy Noether Junior Research Group: Beyond Prediction – Statistical Inference with Machine Learning at the Leibniz Institute for Prevention Research and Epidemiology – BIPS, Bremen, Germany for providing the computational resources used in this study.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This research is supported in part by the RDP grant at University of Pretoria, National Research Foundation (NRF) of South Africa, Ref.: RA210106581084, grant No. 150170, the South African DST-NRF-MRC SARChI Research Chair in Biostatistics (Grant No. 114613) and DSI-NRF Centre of Excellence in Mathematical and Statistical Sciences (CoE-MaSS), South Africa. The opinions expressed and conclusions arrived at are those of the authors and are not necessarily to be attributed to the CoE-MaSS or the NRF.

References

[1] N. Ata and M.T. Sözer, *Cox regression models with nonproportional hazards applied to lung cancer survival data*, Hacettepe J. Math. Stat. 36 (2007), pp. 157–167.
[2] S.F. Brown, A.J. Branford, and W. Moran, *On the use of artificial neural networks for the analysis of survival data*, IEEE Trans. Neural Netw. 8 (1997), pp. 1071–1077.
[3] T. Ching, X. Zhu, and L.X. Garmire, *Cox-nnet: an artificial neural network method for prognosis prediction of high-throughput omics data*, PLoS. Comput. Biol. 14 (2018), p. e1006076.
[4] D.R. Cox, *Regression models and life-tables*, J. R. Stat. Soc.: Ser. B (Methodological) 34 (1972), pp. 187–202.
[5] E.R. Dickson, P.M. Grambsch, T.R. Fleming, L.D. Fisher, and A. Langworthy, *Prognosis in primary biliary cirrhosis: model for decision making*, Hepatology 10 (1989), pp. 1–7.
[6] D. Faraggi and R. Simon, *A neural network model for survival data*, Stat. Med. 14 (1995), pp. 73–82.
[7] D. Feng and L. Zhao, *Bdnnsurv: bayesian deep neural networks for survival analysis using pseudo values*, preprint (2021). Available at arXiv:2101.03170.
[8] S.P. Fotso, *Deep neural networks for survival analysis based on a multi-task framework*, preprint (2018). Available at arXiv:1801.05512.
[9] M.F. Gensheimer and B. Narasimhan, *A scalable discrete-time survival model for neural networks*, PeerJ 7 (2019), p. e6257.
[10] E. Giunchiglia, A. Nemchenko, and M. van der Schaar, *Rnn-surv: a deep recurrent model for survival analysis*, in *International Conference on Artificial Neural Networks*, Springer, 2018, pp. 23–32.
[11] F.E. Harrell Jr, K.L. Lee, R.M. Califf, D.B. Pryor, and R.A. Rosati, *Regression modelling strategies for improved prognostic prediction*, Stat. Med. 3 (1984), pp. 143–152.
[12] G.L. Hickey, P. Philipson, A. Jorgensen, and R. Kolamunnage-Dona, *Joint modelling of time-to-event and multivariate longitudinal outcomes: recent developments and issues*, BMC. Med. Res. Methodol. 16 (2016), pp. 1–15.
[13] H. Ishwaran, U.B. Kogalur, E.H. Blackstone, and M.S. Lauer, *Random survival forests*, Ann. Appl. Stat. 2 (2008), pp. 841–860.
[14] Y. Kalakoti, S. Yadav, and D. Sundar, *Survcnn: a discrete time-to-event cancer survival estimation framework using image representations of omics data*, Cancers 13 (2021), p. 3106.

[15] J.D. Kalbfleisch and R.L. Prentice, *The Statistical Analysis of Failure Time Data*, John Wiley & Sons, Hoboken, NJ, 2011.
[16] J.L. Katzman, U. Shaham, A. Cloninger, J. Bates, T. Jiang, and Y. Kluger, *Deepsurv: personalized treatment recommender system using a cox proportional hazards deep neural network*, BMC. Med. Res. Methodol. 18 (2018), p. 24.
[17] D.G. Kleinbaum and M. Klein, *Survival Analysis: A Self-learning Text*, Springer, 1996.
[18] J.A. Koziol and Z. Jia, *The concordance index c and the mann–whitney parameter pr(x > y) with randomly censored data*, Biometrical J.: J. Math. Methods Biosci. 51 (2009), pp. 467–474.
[19] I. Kuitunen, V.T. Ponkilainen, M.M. Uimonen, A. Eskelinen, and A. Reito, *Testing the proportional hazards assumption in cox regression and dealing with possible non-proportionality in total joint arthroplasty research: methodological perspectives and review*, BMC. Musculoskelet. Disord. 22 (2021), pp. 1–7.
[20] Y. LeCun, Y. Bengio, and G. Hinton, *Deep learning*, Nature 521 (2015), pp. 436–444.
[21] C. Lee, J. Yoon, and M. Van Der Schaar, *Dynamic-deephit: A deep learning approach for dynamic survival analysis with competing risks based on longitudinal data*, IEEE Trans. Biomed. Eng. 67 (2019), pp. 122–133.
[22] M. Luck, T. Sylvain, H. Cardinal, A. Lodi, and Y. Bengio, *Deep learning for patient-specific kidney graft survival analysis*, preprint (2017). Available at arXiv:1705.10245.
[23] L. Mariani, D. Coradini, E. Biganzoli, P. Boracchi, E. Marubini, S. Pilotti, B. Salvadori, R. Silvestrini, U. Veronesi, R. Zucali, and F. Rilke, *Prognostic factors for metachronous contralateral breast cancer: a comparison of the linear cox regression model and its artificial neural network extension*, Breast. Cancer. Res. Treat. 44 (1997), pp. 167–178.
[24] P.A. Murtaugh, E.R. Dickson, G.M. Van Dam, M. Malinchoc, P.M. Grambsch, A.L. Langworthy, and C.H. Gips, *Primary biliary cirrhosis: prediction of short-term survival based on repeated patient visits*, Hepatology 20 (1994), pp. 126–134.
[25] J.B. Nasejje, R. Mbuvha, and H. Mwambi, *Use of a deep learning and random forest approach to track changes in the predictive nature of socioeconomic drivers of under-5 mortality rates in sub-saharan africa*, BMJ. Open. 12 (2022), p. e049786.
[26] J.B. Nasejje, H. Mwambi, K. Dheda, and M. Lesosky, *A comparison of the conditional inference survival forest model to random survival forests based on a simulation study as well as on two applications with time-to-event data*, BMC. Med. Res. Methodol. 17 (2017), p. 115.
[27] J.B. Nasejje, H.G. Mwambi, and T.N. Achia, *Understanding the determinants of under-five child mortality in uganda including the estimation of unobserved household and community effects using both frequentist and bayesian survival analysis approaches*, BMC. Public. Health. 15 (2015), p. 1003.
[28] K. Rufibach, *Use of brier score to assess binary predictions*, J. Clin. Epidemiol. 63 (2010), pp. 938–939.
[29] D.J. Sargent, *Comparison of artificial neural networks with other statistical approaches: results from medical data sets*, Cancer: Interdisciplinary Int. J. Am. Cancer Soc. 91 (2001), pp. 1636–1642.
[30] J.M. Satagopan, L. Ben-Porat, M. Berwick, M. Robson, D. Kutler, and A.D. Auerbach, *A note on competing risks in survival data analysis*, Br. J. Cancer. 91 (2004), pp. 1229–1235.
[31] H. Steck, B. Krishnapuram, C. Dehing-Oberije, P. Lambin, and V.C. Raykar, *On ranking in survival analysis: bounds on the concordance index*, in *Advances in Neural Information Processing Systems*, Vancouver, BC, Canada, December 3–6, Vol. 20, 2007.
[32] J.A. Steingrimsson and S. Morrison, *Deep learning for survival outcomes*, Stat. Med. 39 (2020), pp. 2339–2349.
[33] T. Therneau, C. Crowson, and E. Atkinson, *Using time dependent covariates and time dependent coefficients in the cox model*, Survival Vignettes 2 (2017), pp. 1–25.
[34] T.M. Therneau and P.M. Grambsch, *The Cox Model*, Springer, New York, 2000.
[35] T.-T. Wong, *Performance evaluation of classification algorithms by k-fold and leave-one-out cross validation*, Pattern. Recognit. 48 (2015), pp. 2839–2846.
[36] A. Xiang, P. Lapuerta, A. Ryutov, J. Buckley, and S. Azen, *Comparison of the performance of neural network methods and cox regression for censored survival data*, Comput. Stat. Data Anal. 34 (2000), pp. 243–257.

[37] W. Yao, H. Frydman, D. Larocque, and J.S. Simonoff, *Ensemble methods for survival function estimation with time-varying covariates*, Stat. Methods. Med. Res. 31 (2022), pp. 2217–2236.
[38] S. Yousefi, F. Amrollahi, M. Amgad, C. Dong, J.E. Lewis, C. Song, D.A. Gutman, S.H. Halani, J.E.V. Vega, D.J. Brat, and L.A.D. Cooper, *Predicting clinical outcomes from large scale cancer genomic profiles with deep survival models*, Sci. Rep. 7 (2017), p. 11707.
[39] L. Zeng, J. Zhang, W. Chen, and Y. Ding, *Dynamic prediction using time-dependent cox survival neural network*, preprint (2023). Available at arXiv:2307.05881.
[40] J. Zhang, L. Chen, Y. Ye, G. Guo, R. Chen, A. Vanasse, and S. Wang, *Survival neural networks for time-to-event prediction in longitudinal study*, Knowl. Inf. Syst. 62 (2020), pp. 3727–3751.
[41] Z. Zhang, J. Reinikainen, K.A. Adeleke, M.E. Pieterse, and C.G. Groothuis-Oudshoorn, *Time-varying covariates and coefficients in cox regression models*, Ann. Transl. Med. 6 (2018), p. 121.
[42] L. Zhao and D. Feng, *Deep neural networks for survival analysis using pseudo values*, IEEE. J. Biomed. Health. Inform. 24 (2020), pp. 3308–3314.
[43] X. Zhu, J. Yao, and J. Huang, *Deep convolutional neural network for survival analysis with pathological images*, in *2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, IEEE, 2016, pp. 544–547.