Survival Analysis of Heart Failure Patients

Souradeep Sen
Department of Computer Science,
University of Exeter
(Dated: July, 2023)

The study aims to compare the performance of Deep Learning (DL) architectures for predicting mortality in heart failure (HF) patients, against traditional survival analysis techniques. The aim is to see if combining unsupervised and supervised learning can help estimate survival probabilities based on contextual and historic features from clinical data, and how these predictions perform against traditional techniques. By leveraging longitudinal patient data made available in the form of Electronic Health Records (EHR), an examination of the performance of machine learning risk prediction models is conducted against conventional survival analysis in HF patients.

1. INTRODUCTION

Heart failure is a clinical syndrome interfering with its ability to pump blood, leading to a reduction in systemic circulation performance. This condition is widespread globally - approximately 26 million people worldwide are estimated to be affected by heart failure [1]. Hence, accurately predicting risk is crucial for improving patient outcomes. Traditional survival analysis can be limited in its ability to handle complex non-linear dependencies as well as accounting for time-variant patient characteristics ¡this is not really true¿. Deep learning is an exciting tool in this aspect due to its innate ability to handle complex non-linear relationships ¡include citation - NNs are universal function approximators¿. While there has been work done on adapting deep learning to the survival analysis domain (see:), it is still a growing field. As DL is generally labeled as data-hungry ¡include citation¿, the advent of EHR data with its vast longitudinal bandwidth(??) looks promising to be used in conjunction with DL. This study will attempt to validate the hypothesis that HDL outperforms traditional survival analysis in terms of prediction accuracy using real-world EHR data. The findings may have important implications for clinical practice, healthcare resource allocation, and future research in risk prediction modeling for HF patients with frailty.

The paper is organized as follows. Section 2 contains a brief review of related work in this capacity. Section 3 presents a more rigorous understanding of survival analysis, before building up to how discrete hazard rates [2] ¡CHECK THIS?¿ (as used in this paper) can be parameterized by a neural network [3]. A suitable loss function is chosen from the available literature and is derived as per [3]. Section 4 delves into Uncertainty Quantification, by way of Monte Carlo (MC) dropout and looks at explainability of the model(s) built. Section 5 discusses the data used for the paper - MIMIC-IV. Section 6 looks at the results by means of experiments over real and synthetic data and compares the architechture to existing solutions from deep learning and traditional survival analysis. Section 7 is reserved for proposed extensions to the model and further work that could be done. Some of the supervised and unsupervised methods used in this paper (MLP, CNN, PCA) are detailed in the Appendices.

2. RELATED WORK

Traditional survival analysis has been used extensively to predict mortality in this patient population. Deep learning methods have also been employed - see [4], [5], [6], [7] and [8]. However, limited work has been done in predicting mortality and hospitalization in HF patients with frailty, especially using electronic health records (EHR) data. Several past papers have addressed predictive modeling for heart failure patients. A deep neural network model with learned medical feature embedding is proposed in [9] to address high dimensionality and temporality in electronic health record (EHR) data. Here, a convolutional neural network is used to capture non-linear longitudinal evolution of EHRs and local temporal dependency for risk prediction, and embed medical features to account for high dimensionality. Experiments show promising results in predicting risks for congestive heart failure.

Personalized predictive modeling is investigated in [10], which aims to build specific models for individual patients using similar patient cohorts to capture their specific characteristics. According to this study, although CNNs have shown promise on measuring patient similarity, one disadvantage is that they could not utilize temporal and contextual information of EHRs. To measure patient similarity using EHRs, the authors proposed a time-fusion CNN framework. A vector representation was generated for each patient, which was then utilized for measuring patient similarity and personalized disease prediction. Dynamic updates to a CNN model are explored in [11] as more data is gathered over time - this architecture lends itself well to real-time mortality risk prediction.

Maintaining interpretability across deep learning models is explored in [12]. Many previous studies using machine learning for modeling the risk of HF in patients have focused on discretized outputs. This study aims to consider incidences as time-to-event to enable continuous probabilistic risk prediction for hospitalization and mortality, addressing a critical need in patient care. The use of EHR such as those available in CPRD, allows access to comprehensive longitudinal data, which captures the entire cycle of a patient's diagnosis and treatment. ¡Add citations for PyCox, Deep Survival Machines, Deep Survival Analysis, Faraggi-Simon, DeepSurv, RNN-SURV, ¿

3. SURVIVAL ANALYSIS

A. Basics

Survival analysis deals with the estimation of a survival distribution representing the probability of an event of interest, typically a failure, to occur beyond a certain time in the future [13]. One way to specify the survival distribution is through the survival function. The survival function defines the probability of surviving till point t [14].

$$S(t) = P(T > t),\ 0 < t < \infty$$

It can be thought of as the complement of the cumulative distribution function $F(t)$.

$$S(t) = P(T > t) = 1 - P(T \leq t) = 1 - F(t),\ 0 < t < \infty$$

Another way to specify the survival distribution is through the hazard function, which denotes the instantaneous rate of failure.

$$h(t) = \lim_{\delta \to 0} \frac{P(t < T < t + \delta | T > t)}{\delta}$$

For the continuous-time scenario, the hazard function and survival function are related as follows.

$$f(t) = \frac{d}{dt}F(t) = -\frac{d}{dt}S(t)$$
$$h(t) = \frac{f(t)}{S(t)}$$

where $f(t)$ is the probability mass function (PMF). This says that the hazard is the probability of the subject experiencing an event at time t, provided that the subject is alive till time t. It can be further simplified as

$$h(t) = -\frac{d}{dt}S(t)\frac{1}{S(t)}$$
$$\implies S(t) = exp\left(-\int_0^t h(u)du\right)$$

This relationship produces the survival function from a hazard function [14].

B. A Brief Review of Traditional Fitters

In the context of continuous time survival models, the Cox Proportional Hazards model has long been the 'gold-standard' for survival analysis ¡cite?¿. Extensions have been made for Cox models to incorporate time-varying covariates as well as introduce the capability to handle non-linear hazards ¡cite Katzman¿ (although the proportional hazards assumption remains). The assumption of a proportional hazards model is that the covariates have a multiplicative effect on the hazard.

$$h(t|x) = h_0(t)e^{w^T x}$$

where h_0 is called as the baseline hazard.

Proportional hazards models are learned by optimizing Cox's partial likelihood in classical survival analysis [15]. A general formulation is derived as follows [14]. Consider failure time t_i. The set of all subjects in the trial "at risk" for failure at this time is denoted by $j : Y_j > Y_i$. The probability of Patient i failing at this time is

$$L_i = \frac{h(t_i|x_i)}{\sum_{j:Y_j>Y_i} h(t_i|x_j)}$$

where $h(t|x)$ denotes the hazard for patient with covariates x at time t. The likelihood of the entire cohort is thus,

$$L = \prod_{i:C_i=1} L_i = \prod_{i:C_i=1} \frac{h(t_i|x_i)}{\sum_{j:Y_j>Y_i} h(t_i|x_j)}$$

where $C_i = 1$ denotes an observed (uncensored) event. The assumption of a proportional hazards model reduces this to

$$L = \prod_{C_i=1} \frac{e^{w^T x_i}}{\sum_{j:Y_j>Y_i} e^{w^T x_j}}$$

It is noted that the baseline hazard h_0 is canceled from both the numerator and the denominator [15]. The log likelihood therefore becomes

$$\ell = \sum_{i:C_i=1} \left(w^T x_i - log \sum_{j:Y_j>Y_i} e^{w^T x_j} \right)$$

This negative of this log-likelihood is used as the loss function for Deep Surv ¡cite katzman¿ and the Faraggi-Simon ¡cite faraggi-simon¿ implementation of deep Cox proportional hazards models.

Accelerated failure time models are also popular ¡cite¿, as they allow the incorporation of an acceleration (and conversely decelration) of the hazard rates.

Random Survival Forest models have managed to achieve state-of-the-art performance. At the cost of higher fitting times, they deliver high discriminative power with impressive calibration ¡cite¿.

C. Data Setup

Before moving to the discrete setting, some formal notation for the data is introduced. The data is assumed to be right-censored. Hence, the data, $\mathcal{D}$ can be represented as a set of tuples $\{(x_i, t_i, d_i)\}_{i=1}^{N}$ [13]. Here, $x_i \in \mathbb{R}^d$ are covariates for patient i. t_i is the time of an event or censoring such that $t_i = min(T_i, C_i)$, where T_i and C_i respectively denote the times of event and censoring. A subject is assumed to have either experienced an event or have been censored, but not both. d_i is an indicator that signifies whether t_i is event time or censoring time. $d = 1$ for a subject that experiences the event (uncensored) while $d = 0$ for a subject that is censored before experiencing the event. More formally, $d = \mathbb{1}\{T_i \leq C_i\}$. Later in the paper, experiments with time-variant covariates will necessitate the use of a null masking matrix, $\mathcal{M}$ ¡cite DeepHit¿.

D. Discrete-Time Survival Analysis

For hazard and survival calculation in a discrete-time setting, the following formulation from [3] and earlier [2]¡CHECK THIS?¿ is presented. Let $\mathbb{T} = \{\tau_1, \tau_2, \dots\}$ denote the timestamps, i.e. the indices of the discrete times corresponding to different subjects in the data. The event timestamp is $T* \in \mathbb{T}$. The definitions of PMF and survival function follow as

$$f(\tau_j) = P(T* = \tau_j),$$
$$S(\tau_j) = P(T* > \tau_j) = \sum_{k>j} f(\tau_k)$$

It can be seen that the hazard at time τ_j $h(\tau_j)$, is just the probability of an event happening at time τ_j, given the subject has survived till the previous time step τ_{j-1}.

$$h(\tau_j) = P(T* = \tau_j | T* > \tau_{j-1}) = \frac{f(\tau_j)}{S(\tau_{j-1})} \quad (1)$$

$$\implies h(\tau_j) = \frac{S(\tau_{j-1}) - S(\tau_j)}{S(\tau_{j-1})} \quad (2)$$

$$\implies S(\tau_j) = (1 - h(\tau_j))S(\tau_{j-1}) \quad (3)$$

Recursively, the survival function can be parameterized wholly in terms of the hazard function as

$$S(\tau_j) = \prod_{k=1}^{j} (1 - h(\tau_k)) \quad (4)$$

If there were no censoring involved, the likelihood of observing n failures (events) is of the form

$$L(t_1, t_2, \dots, t_n) = f(t_1)f(t_2)\dots f(t_n) = \prod_{i=1}^{n} f(t_i)$$

As per [14], for an observed event, the pdf is retained. But for a right-censored observation, it is replaced by the survival function, as that observation is known only to exceed a particular value. The likelihood then becomes

$$L(t_1, t_2, \dots, t_n) = \prod_{i=1}^{n} f(t_i)^{\delta_i} S(t_i)^{1-\delta_i} = \prod_{i=1}^{n} h(t_i)^{\delta_i} S(t_i)$$

E. Loss Function(s)

The architecture incorporates two loss functions - one being the negative log likelihood of the data [3], another being a lower bound on the concordance index [15].

Negative Log-Likelihood - hazard function

The following derivation is taken from [3]. For notational convenience, let $\kappa(t) \in \{0, \dots, m\}$ define the index of the discrete time t, meaning $t = \tau_{\kappa(t)}$. Thus, the likelihood contribution for individual i is seen to be

$$\begin{aligned} L_i &= f(t_i)^{\delta_i} S(t_i)^{1-\delta_i} \\ &= [h(t_i)S(\tau_{\kappa(t_i)-1})]^{d_i}][(1 - h(t_i))S(\tau_{\kappa(t_i)-1})]^{1-d_i} \\ &= h(t_i)^{d_i}[1 - h(t_i)]^{1-d_i} S(\tau_{\kappa(t_i)-1})] \\ &= h(t_i)^{d_i}[1 - h(t_i)]^{1-d_i} \prod_{\kappa_{t_i}-1}^{j=1} [1 - h(\tau_j)] \end{aligned}$$

For all the individuals, the combined log-likelihood is

$$L = \prod_{i=1}^{n} L_i = \prod_{i=1}^{n} \left(h(t_i)^{d_i}[1 - h(t_i)]^{1-d_i} \prod_{\kappa_{t_i}-1}^{j=1} [1 - h(\tau_j)] \right)$$

From here, the loss function for a batch can be constructed as the negative log likelihood.

$$log(L) = \sum_{n}^{i=1} \Bigg(d_i log[h(t_i|x_i)] + (1 - d_i) log[1 - h(t_i|x_i)] +$$
$$\sum_{j=1}^{\kappa(t_i)-1} log[1 - h(\tau_j|x_i)] \Bigg)$$

To adjust for varying batch sizes, the mean negative log likelihood is taken as the loss.

$$\mathcal{L}_{nll} = -\frac{1}{n} \sum_{n}^{i=1} \Bigg(d_i log[h(t_i|x_i)] + (1 - d_i) log[1 - h(t_i|x_i)] +$$
$$\sum_{\kappa(t_i)-1}^{j=1} log[1 - h(\tau_j|x_i)] \Bigg)$$
$$= \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{\kappa_{t_i}} (y_{ij} log[h(\tau_j|x_i)] + (1 - y_{ij}) log[1 - h(\tau_j|x_i)])$$

This can now be minimized by gradient-based methods, thus making it a useful loss function for a neural network to work with. Here, y_{ij} is an indicator variable corresponding to 1 if and only if an event is experienced by the individual i at time t_j. Hence, y is a sparse matrix consisting of mostly 0 with 1 only present when the time t_i represents an observed event $d_i = 1$. The loss can be thought of as the negative log likelihood of Bernoulli data $\theta^p(1-\theta)^{1-p}$, where $\theta = h(\tau_j|x_i)$ and $p = y_{ij}$(¡noted by Brown 1975¿) and can be computed using existing functions in the PyTorch library.

Lower-Bound on C-Index

The following loss function is taken from [15]. The sigmoid function defined as $\sigma = (1+e^{-z})^{-1}$ is an approximation to the indicator function $\mathbb{1}_{z>0}$. However, it is not a lower bound. For this, the scaled version of its log is taken.

$$\mathbb{1}_{z>0} \geq log\ [2\sigma(z)]/log\ 2$$
$$\implies \mathbb{1}_{z>0} \geq 1 + (log\ \sigma(z)/log\ 2)$$

For a more rigourous treatment of this inequality, see the original paper [15]. It follows that the lower bound on the concordance index then becomes

$$c = \frac{1}{|\mathcal{E}|}\sum_{i:C_i=1}\sum_{T_j>T_i}\mathbb{1}_{\eta_j-\eta_i>0}$$
$$c \geq \frac{1}{|\mathcal{E}|}\sum_{i:C_i=1}\sum_{T_j>T_i} 1 + (log\ \sigma(\eta_j-\eta_i)/log\ 2)$$

The negative of this lower-bound is chosen to be the loss function, which can therefore be minimized using gradient-based methods (again, with the help of a library like PyTorch).

$$\mathcal{L}_{lbo} = -\frac{1}{|\mathcal{E}|}\sum_{i:C_i=1}\sum_{T_j>T_i} 1 + (log\ \sigma(\eta_j-\eta_i)/log\ 2)$$

These two loss functions are linearly combined based on a hyperparameter $\alpha \in [0,1]$ to make the total loss

$$\mathcal{L}_{total} = \alpha\mathcal{L}_{nll} + (1-\alpha)\mathcal{L}_{lbo}$$

F. Neural Network Parameterization

As hazards are conditional probabilities (see Eq. 1), they must lie within $[0,1]$. A handy function for this is the sigmoid non-linearity

$$g(x) = \frac{1}{1+e^{-x}}$$

For a neural network taking x_i (the covariates of patient i) as input and producing m outputs denoting m discrete timesteps in the patient's journey, applying the sigmoid function effectively transforms the outputs into valid hazard rates.

$$h(\tau_j|x_i) = g(\phi_j(x_i)) = \frac{1}{1+e^{-\phi_j(x_i)}}$$

where $\phi_j(x_i)$ is the output of the neural network at node j corresponding to the hazard function over the time period $[m_{j-1}, m_j)$.

Returning to the problem at hand, for such a parameterization, the continuous time scale $\mathbb{T}$ needs to be discretized into m pieces. A simple way to do this is to equally divide up the set $\mathbb{T}$ into m equal parts. The hazards across these m timesteps can be cumulatively multiplied to find the survival function as seen in Eq. 4.

4. UNCERTAINTY QUANTIFICATION AND EXPLAINABILITY

An addition this work attempts to make to the above is by introducing a layer of uncertainty quantification via Monte-Carlo dropout [16]. Dropout is generally reserved for training as a means to enforce regularization on the network. When applied during the testing or prediction phase as well, a probability distribution of the output is created, allowing for probabilistic inference instead of point inference as present in regular neural networks. Being uncertainty-aware ties in with the idea of interpretability of modern deep learning methods, which is often a barrier towards the adoption of such methods into highly regulated industries such as finance or medicine. SHAP (SHapley Additive exPlanations) values [17], extended from the field of cooperative game theory provide a unified, model-agnostic method to explain the contributions of covariates to a model's output.

5. MODEL ARCHITECTURE

This is a placeholder for the model architecture. This paper attempts to model survival functions from both a time-invariant perspective (using mean statistics of covariates) and from a time-variant perspective (using temporal dependencies of covariates ¡??rephrase¿). The latter takes some more preprocessing and a more complex architecture.

A. Time-Invariant

The simpler time-invariant version of the architecture is a multi-layer perceptron (MLP), otherwise referred to as a fully-connected neural network. Consider a batch of input covariates $x \in \mathbb{R}^{n\times d}$, where n is the batch size and d is the number of input covariates. The basic architecture is as outlines in Table I.

Layer Type	Shape/Rate
Dense (nn.Linear)	(d, h)
ReLU (nn.ReLU)	-
Batch Normalization (nn.BatchNorm1d)	h
Dropout (nn.Dropout)	0.5
Dense (nn.Linear)	(h, h)
ReLU (nn.ReLU)	-
Batch Normalization (nn.BatchNorm1d)	h
Dropout (nn.Dropout)	0.5
Dense (nn.Linear)	(h, m)

TABLE I. General Architecture -

Here, m is the output size, denoting the number of discretized time indices across time $max(\mathbb{T})$, while h denotes the number of hidden nodes in the linear layers.
Optimizer - Nesterov Momentum, Learning Rate scheduler - decay!

6. DATASET

The study uses the large publicly available database MIMIC-IV [18], which consists of critical care data from hospital and ICU admissions for over 40,000 patients admitted to intensive care units at the Beth Israel Deaconess Medical Center (BIDMC).

A. Description

B. Preprocess

To prepare the data, all admissions associated with a Heart Failure ICD-10 code were selected - this forms the base pool of patients. Admission and discharge times are collected along with static covariates such as patients' gender, age and date of death. Patients' ethnicity is collected and grouped into six broad categories - Native, Asian, Black, Hispanic, White and Other. (This goes in footnote - To avoid multi-collinearity issues arising later, the 'OTHER' variable is dropped after one-hot encoding)

Time-variant records such as BMI, Weight and Height are collected from Online Medical Records (OMR). Such records have an associated chart-time or chart-date, denoting when they were collected. Lab tests corresponding to cholesterol, sodium intake, lymphocyte count and hemoglobin levels ¡CITE!? Why just these?¿. Medication administered from the classes of angiotensin-converting enzyme blockers, angiotensin receptor blockers, calcium channel blockers and beta blockers are also taken as features/ covariates. Vital signs such as temperature, heartrate, respiratory rate, O_2 saturation, systolic and diastolic blood pressure are also taken. Patients who have at least one record for all of the above 4 datasets (OMR, lab test, medication and vital signs) are retained.

Patients are randomly distributed into train, test and validation pools. Across each pool, their survival time is discretized into $q = 15$ buckets or time intervals, with $m_i = j$ denoting that the i^{th} patient experienced an event or was censored over the m^{th} time interval.

C. Censoring

Patients' earliest admission time is considered as their 'start' date. If their death date is not captured in the data ¡add footnote here¿, their last known discharge time is known as their 'end' date - these patients are considered to be censored. Otherwise, their date of death is taken to be the 'end' date - these are the uncensored patients. The distribution of event/censoring times is quite similar across both censored and uncensored cohorts ¡Add image of histogram¿. Standard pipelines for dummy-encoding, train-test-validation splitting, imputation and scaling are carried out. Columns with zero variance in the training data are discarded from all three datasets - train, test and validation.

7. EVALUATION METRICS

As survival analysis differs from ordinary linear regression in the aspect that not all the survival times are known (right-censoring), there is merit in reviewing the evaluation metrics for this task. The two metrics used in the experimental setup are the time-dependent concordance index (in favour of the regular concordance index) and the Brier score (along with its aggregated version, the integrated Brier score).

A. Time-Dependent Concordance Index (td-concordance index)

The C-index, often referred to as "Harrell's C-Index" or simply as the C-statistic is a measure of the discriminative capacity of a model. In essence, it is the generalization of the ROC (Receiver Operator Characteristic) curve - AUC in a survival analysis setting ¡Rephrase??¿ [19] [20]. The formula for the c-index is given as

$$c = \frac{1}{|\mathcal{E}|} \sum_{i:C_i=1} \sum_{T_j>T_i} \mathbb{1}_{\eta_j>\eta_i}$$

where η_i denotes the predicted survival time for subject i. As c-index is rank-based index, it can be substituted with the survival probability of subject i as well. $|\mathcal{E}|$ is the number of pairs that *can* be compared.

It can be interpreted as the fraction of all pairs of subjects whose predicted survival times are correctly ordered among all subjects that can actually be ordered [21]. A pair (i, j) is considered 'comparable' if the one with the lower observed time is uncensored, that is, when $T_i < T_j$, then $d_i = 1$. A pair is considered 'concordant' when the model predicts a higher risk for the patient with a lower survival time. Thus, it follows that

$$c - index = \begin{cases} 1.0, & \text{perfect concordance,} \\ 0.5, & \text{equivalent to random classification,} \\ 0.0, & \text{perfect anti-concordance.} \end{cases}$$

There are limitations with the traditional c-index as highlighted by [22], an important one being the assumption that the risk scores do not change over time. [23] propose a time-dependent concordance index which essentially takes a weighted average of time-specific C-index values across the entire time scale.

B. Brier Score

Contrasting the discriminatory power of the C-index, the Brier score provides a measure of how well the model is calibrated. It represents the distance between observed and predicted survival probability - hence 0 is the most desirable value. Given the data, $\mathcal{D}$, a set of tuples $\{(x_i, t_i, d_i)\}_{i=1}^{N}$ and the predicted survival function $\hat{S}(t, x_i)$, $\forall t \in \mathbb{R}^+$, the Brier score (without right-censored observations) assumes a form similar to the mean-squared-error.

$$BS(t) = \frac{1}{N} \sum_{i=1}^{N} (\mathbb{I}_{T_i > t} - \hat{S}(t, x_i))^2$$

With the occurence of right-censored data, the formula needs to be adjusted by the inverse probability of censoring weights method ¡cite?¿. Let $\hat{G}(t) = P[C > t]$ be the estimator of the conditional survival function of the censoring times calculated using the Kaplan-Meier method, where C is the censoring time.

$$BS(t) = \frac{1}{N} \sum_{i=1}^{N} \left(\frac{(0 - \hat{S}(t, x_i))^2 \cdot \mathbb{I}_{T_i \leq t, d_i = 1}}{\hat{G}(T_i)} + \frac{(1 - \hat{S}(t, x_i))^2 \cdot \mathbb{I}_{T_i > t}}{\hat{G}(t)} \right)$$

An aggregated version of this score provides a simpler (albeit more reductive) summary of the calibration of a model.

$$IBS(t_{max}) = \int_0^{t_{max}} BS(t) dt$$

8. EXPERIMENTS AND FITS

A series of experiments were conducted on this data from both a time-invariant and a time-varying perspective. The survival distributions were examined with non-parametric fitters - Kaplan-Meier fitter and Nelson-Aalen fitter. A short write-up on these methods can be found in the Appendix A.

Next, a range of parametric, semi-parametric, tree-based and neural network-based methods were applied to the data and their evaluation criteria noted. To establish confidence intervals on said criteria, fits were run multiple times. The details are listed in Table II.

9. CONCLUSIONS

Amongst the neural network models, the architecture is sacrificing calibration for discriminatory power.
[]

[1] G. Savarese and L. H. Lund, Global public health burden of heart failure, Cardiac Failure Review **03**, 7 (2017).

[2] M. F. Gensheimer and B. Narasimhan, A scalable discrete-time survival model for neural networks, PeerJ **7**, 10.7717/peerj.6257 (2019).

[3] H. Kvamme and Ø. Borgan, Continuous and discrete-time survival prediction with neural networks (2019), arXiv:1910.06724 [cs, stat].

[4] M. Gjoreski, A. Gradisek, B. Budna, M. Gams, and G. Poglajen, Machine learning and end-to-end deep learning for the detection of chronic heart failure from heart sounds, IEEE Access **8**, 20313 (2020).

[5] J. J. Nirschl, A. Janowczyk, E. G. Peyster, R. Frank, K. B. Margulies, M. D. Feldman, and A. Madabhushi, A deep-learning classifier identifies patients with clinical heart failure using whole-slide images of h&e tissue, PloS one **13**, e0192726 (2018).

[6] L. Wang, L. Sha, J. R. Lakin, J. Bynum, D. W. Bates, P. Hong, and L. Zhou, Development and Validation of a Deep Learning Algorithm for Mortality Prediction in Selecting Patients With Dementia for Earlier Palliative Care Interventions, JAMA Network Open **2**, e196972 (2019).

[7] J. R. Ayala Solares, F. E. Diletta Raimondi, Y. Zhu, F. Rahimian, D. Canoy, J. Tran, A. C. Pinho Gomes, A. H. Payberah, M. Zottoli, M. Nazarzadeh, N. Conrad, K. Rahimi, and G. Salimi-Khorshidi, Deep learning for electronic health records: A comparative review of

Model	**HyperParameters**	**C-Index**	**IBS**
Cox Proportional Hazards	penalizer = 0.1, step_size = 0.1	0.6808	0.1685
Weibull AFT	penalizer = 0.01	0.6804	0.1748
Random Survival Forest	n_estimators=1000, min_samples_split=10, min_samples_leaf=15, n_jobs=-1, random_state=1234, oob_score = True	0.6975	0.1956
PyCox (with Logistic Hazard loss function [3])	no. of discretization=15, layers=[256,256], batch_norm=True, dropout=0.5, batch_size=256, epochs=500	0.5974	0.1768
Deep Survival Machhines	k=6, distribution=LogNormal, learning_rate=1e-3, layers=[100, 100]	0.6441	0.2724
DeepSurv	batch_size=256, num_epochs=1000, learning_rate=0.01, patience=50, hidden_size=100, droutput_rate = 0.1	0.4259	-

TABLE II. Evaluation metrics

multiple deep neural architectures, Journal of Biomedical Informatics **101**, 103337 (2020).

[8] G. Lorenzoni, S. S. Sabato, C. Lanera, D. Bottigliengo, C. Minto, H. Ocagli, P. De Paolis, D. Gregori, S. Iliceto, F. Pisanò, and et al., Comparison of machine learning techniques for prediction of hospitalization in heart failure patients, Journal of Clinical Medicine **8**, 1298 (2019).

[9] Z. Che, Y. Cheng, Z. Sun, and Y. Liu, Exploiting convolutional neural network for risk prediction with medical feature embedding, arXiv preprint arXiv:1701.07474 (2017).

[10] Q. Suo, F. Ma, Y. Yuan, M. Huai, W. Zhong, A. Zhang, and J. Gao, Personalized disease prediction using a cnn-based similarity learning method, in *2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)* (IEEE, 2017) pp. 811–816.

[11] L. Brand, A. Patel, I. Singh, and C. Brand, Real time mortality risk prediction: A convolutional neural network approach., in *HEALTHINF* (2018) pp. 463–470.

[12] W. Caicedo-Torres and J. Gutierrez, Iseeu: Visually interpretable deep learning for mortality prediction inside the icu, Journal of biomedical informatics **98**, 103269 (2019).

[13] C. Nagpal, X. R. Li, and A. Dubrawski, Deep survival machines: Fully parametric survival regression and representation learning for censored data with competing risks (2021), arXiv:2003.01176 [cs, stat].

[14] D. F. Moore, Chapter 2 basic principles of survival analysis, in *Applied Survival Analysis Using R* (Springer, 2016).

[15] R. Vikas C, H. Steck, B. Krishnapuram, C. Dehing-oberije, and P. Lambin, On ranking in survival analysis: Bounds on the concordance index, Advances in Neural Information Processing Systems (2007).

[16] Y. Gal and Z. Ghahramani, Dropout as a bayesian approximation: Representing model uncertainty in deep learning, in *Proceedings of The 33rd International Conference on Machine Learning*, Proceedings of Machine Learning Research, Vol. 48, edited by M. F. Balcan and K. Q. Weinberger (PMLR, New York, New York, USA, 2016) pp. 1050–1059.

[17] S. M. Lundberg and S.-I. Lee, A unified approach to interpreting model predictions (2017).

[18] A. E. Johnson, L. Bulgarelli, L. Shen, A. Gayles, A. Shammout, S. Horng, T. J. Pollard, S. Hao, B. Moody, B. Gow, and et al., Mimic-iv, a freely accessible electronic health record dataset, Scientific Data **10**, 10.1038/s41597-022-01899-x (2023).

[19] H. Uno, T. Cai, M. J. Pencina, R. B. D'Agostino, and L. J. Wei, On the c-statistics for evaluating overall adequacy of risk prediction procedures with censored survival data, Statistics in Medicine **30**, 1105–1117 (2011).

[20] Pysurvival c-index, https://square.github.io/pysurvival/metrics/c_ind
accessed: 2023-08-01.

[21] J. D. Pinto, A. M. Carvalho, and S. Vinga, Outlier detection in survival analysis based on the concordance c-index, Proceedings of the International Conference on Bioinformatics Models, Methods and Algorithms 10.5220/0005225300750082 (2015).

[22] N. Hartman, S. Kim, K. He, and J. D. Kalbfleisch, Pitfalls of the concordance index for survival outcomes, Statistics in Medicine **42**, 2179–2190 (2023).

[23] P. J. Heagerty and Y. Zheng, Survival model predictive accuracy and roc curves, Biometrics **61**, 92–105 (2005).

Appendix A: Appenidx

Some formal mathematical formulations of the methods used in this paper are elaborated in the following sections

1. Non-Parametric Fits

The Kaplan Meier fitter
The Nelson Aalen