Deep Parametric Time-to-Event Regression with Time-Varying Covariates

Chirag Nagpal* [1] CHIRAGN@CS.CMU.EDU
Vincent Jeanselme* [1,2] VINCENT.JEANSELME@MRC-BSU.CAM.AC.UK
Artur Dubrawski [1] AWD@CS.CMU.EDU
[1] *Auton Lab, School of Computer Science, Carnegie Mellon University*
[2] *MRC Biostatistics Unit, School of Clinical Medicine, University of Cambridge*

Abstract

Time-to-event regression in healthcare and other domains, such as predictive maintenance, require working with time-series (or time-varying) data such as continuously monitored vital signs, electronic health records, or sensor readings. In such scenarios, the event-time distribution may have temporal dependencies at different time scales that are not easily captured by classical survival models that assume training data points to be independent. In this paper, we describe a fully parametric approach to model censored time-to-event outcomes with time varying covariates. It involves learning representations of the input temporal data using Recurrent Neural Networks such as LSTMs and GRUs, followed by describing the conditional event distribution as a fixed mixture of parametric distributions. The use of the recurrent neural networks allows the learned representations to model long-term dependencies in the input data while jointly estimating the Time-to-Event. We benchmark our approach on MIMIC III: a large, publicly available dataset collected from Intensive Care Unit (ICU) patients, focusing on predicting duration of their ICU stays and their short term life expectancy, and we demonstrate competitive performance of the proposed approach compared to established time-to-event regression models.[1]

Keywords: Time-to-Event Regression, Survival Analysis, Recurrent Neural Networks

1. Introduction

Survival analysis is an important problem in statistical estimation with applications in healthcare, predictive maintenance, econometrics and actuarial sciences. In its most classical form, survival analysis (or time-to-event regression) involves estimation of the conditional survival function of an individual with a given set of covariates $\mathbf{x}, S(t|\mathbf{x}) = \mathbb{P}(T > t|X = \mathbf{x})$. Several important applications of survival analysis require working with interdependent temporal data such as multiple hemodynamic vital signs. Standard extensions to survival models for longitudinal data involve representing input covariates with aggregate statistics accrued over time in order to make them compatible with standard survival regression approaches.

However, some data modalities, such as time series, cannot always be sufficiently captured using statistical featurization of static snapshots of the input vectors, $\mathbf{x}$. Furthermore, in the case of discrete temporal data, such as electronic health records, certain historical events may be more informative and more consequential, with long term effects that require

[1] Software Package: autonlab.github.io/DeepSurvivalMachines/#recurrent-deep-survival-machines

modeling at multiple scales with factors more capacious than simple aggregate statistics such as moving averages and variances over time.

In time-to-event regression literature, such input data are known as time-varying covariates. While there is a large amount of existing work on extensions of classical statistical methods involving such data, modern machine learning approaches to model time-varying covariates are relatively understudied.

In this paper, we propose Recurrent Deep Survival Machines (RDSM), a fully parametric survival analysis method for modeling time-to-event data in the presence of time-varying covariates. RDSM builds on the original DSM model (Nagpal et al., 2021a) by replacing the learned representation with a Recurrent Neural Network (RNN) architecture, such as a standard RNN or its variants, e.g., GRU (Cho et al., 2014) or LSTM (Gers et al., 1999). As in the case of the original DSM model, we assume that once the representations are obtained, the event arrival times are distributed as a mixture of underlying parametric distributions. The parameters of these underlying distributions are also assumed to be functions of the obtained representations, and are learned jointly with the recurrent neural architectures. Our key contributions in this paper are:

- We propose a novel censored Time-to-Event regression model, Recurrent Deep Survival Machines, that allows modeling of time-varying coefficients by using RNN layers with flexible parametric choices on the event-time distribution.
- We demonstrate the utility of RDSM on Mortality and Length-of-Stay prediction in the MIMIC-III dataset of ICU patients, and compare performance of the proposed approach against established censored survival regression baselines.
- Finally, we release the RDSM model as part of the open-source `dsm` python package for wider dissemination with the survival analysis research community.

2. Recurrent Deep Survival Machines

2.1. Notation and Setting

We assume that we want to model a *right-censored* dataset, implying that our data, $\mathcal{D}$ is a set of tuples $\{(\mathcal{X}_i, t_i, \delta_i)\}_{i=1}^{N}$, where t_i is the time at which an event of interest took place, or the censoring time, and δ_i is an indicator that signifies whether t_i is the event time or censoring time. $\mathcal{X}_i$ is the set of features sampled over time along with the corresponding timestamp at which the data was sampled $\mathcal{X}_i := \{(\mathbf{x}_i^1, \tau_i^1), (\mathbf{x}_i^2, \tau_i^2), ...(\mathbf{x}_i^j, \tau_i^j)\}$. We notate $\mathcal{X}_i^j$ as the set of all covariates observed before time-step j and introduce remaining time-to-event at a time-step j as $r^j = (t - \tau^j)$. Thus the learning problem reduces to estimating the distributions of the conditional remaining time-to-event at each time step j, $\widehat{\mathbb{P}}(T(j) \mid \mathcal{X}_i^j)$.

We assume that we either only observe the actual failure or censoring time, but not both, for each individual. Furthermore, for the purposes of identification, it is assumed that the true data generating process is such that the censoring process is independent of the actual time to failure.

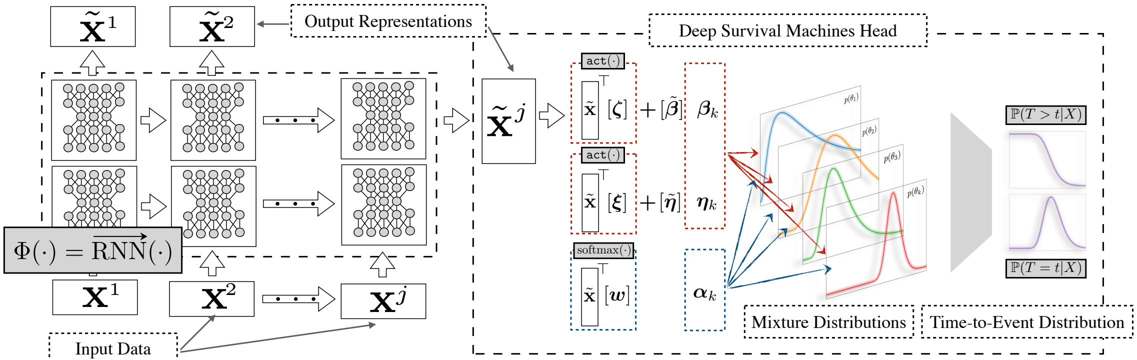

Figure 1: **Recurrent Deep Survival Machines:** The model involves first passing the set of input covariates sampled over time $\{\mathbf{x}^1, \mathbf{x}^2, ..., \mathbf{x}^j\}$ through the RNN, $\Phi(\cdot)$ to obtain the corresponding set of representations $\{\tilde{\mathbf{x}}^1, \tilde{\mathbf{x}}_i^2, ..., \tilde{\mathbf{x}}^j\}$ at each time step. The remaining Time-to-Event distribution is then modeled as a mixture of parametric distributions where the parameters of the distributions are functions of the input representations. The use of RNNs allows us to learn representations that retain knowledge from previous times steps.

2.2. Deep Survival Machines

The key idea behind the original Deep Survival Machines model is to assume that the conditional survival distribution of an individual with covariates $\mathbf{x}$ is a mixture of a fixed-size parametric distributions like the Weibull or Log-Normal. The shape and scale parameters of the underlying distributions, as well as the mixing weights, are implemented as a function of the input covariates using neural networks. Thus:

$$\mathbb{P}(T = t|X = \mathbf{x}) = \sum_k \mathbb{P}(Z = k|X = \mathbf{x})\mathbb{P}(T = t|X = \mathbf{x}, Z = k). \tag{1}$$

Here, $\mathbb{P}(Z = k|X = \mathbf{x}) = \mathrm{softmax}\big(f(\Phi(\mathbf{x}))\big)$ where f is a linear function and $\mathbb{P}(T = t|X = \mathbf{x}, Z = k)$ is Weibull or Log-Normal with shape and scale parameterized as functions of the representation $\Phi(\mathbf{x})$.

$$\begin{aligned} \ln \mathbb{P}(T = t|X = \mathbf{x}) &= \ln \sum_k \mathbb{P}(Z = k|X = \mathbf{x})\mathbb{P}(T = t|X = \mathbf{x}, Z = k) \\ &\geq \sum_k \mathbb{P}(Z = k|X = \mathbf{x}) \ln \mathbb{P}(T = t|X = \mathbf{x}, Z = k). \end{aligned} \tag{2}$$

In the original DSM model, the authors proposed to perform inference using a lower bound to the maximum likelihood estimate under the above model as in Equation 2.

2.3. Structure of Recurrent DSM

Figure 1 gives a schematic description of our proposed approach. In order to allow RDSM to incorporate streaming data inputs, we will extend our discussion and consider the remaining time-to-event distribution $T(j)$ at a timestep, j. Instead of treating this distribution as static as in standard survival settings, modeling it as a function of time allows capturing the time varying effects of the input covariates.

Assumption 1 (Independent Censoring)**.** *For an individual with remaining time-to-event distribution, $T(j)$, censoring time C and set of observed covariates, $\mathcal{X}^j$ till time j;*

$$T(j) \perp C \;\mid\; \mathcal{X}^j$$

Assumption 1 is analogous to the standard assumptions of random (non-informative) censoring in static survival analysis. This assumption is required for the purpose of identifiability. Assuming independent censoring, we can factorize the log-likelihood of the data from an individual at a time step j with remaining time to event $r^j = t - \tau^j$ over the censored and uncensored likelihood. We thus arrive at the following:

$$\mathbb{P}(\{\mathcal{X}^j, r^j, \delta\}) = \mathbb{P}(T = r^j|\mathcal{X}^j)^{\delta}\mathbb{P}(T > r^j|\mathcal{X}^j)^{(1-\delta)}. \tag{3}$$

Assumption 2 (Statefulness)**.** *For an individual with event-time distribution at time j, $T(j)$ and the set of covariates observed till time-step j, $\mathcal{X}^j$;*

$$T(j) \perp (\mathcal{X} \setminus \mathcal{X}^j) \;\mid\; \mathcal{X}^j$$

Assumption 2 essentially states that the distribution of the remaining time-to-event at time j, $T(j)$ is completely characterized by the data at time steps preceding j. Although, in practice, later events in the data stream might effect the final event-time t, notice here that we consider the instantaneous event time distribution at time j, $T(j)$ instead, which we allow to evolve as more data is accrued. We require to make this assumption in order to enable inferring event risks dynamically, in a streaming fashion.

2.4. Learning

In the case of time-varying covariates, we have access to the full data stream $\mathcal{X}$ and not just a single feature vector $\mathbf{x}$. We thus replace the learning objective with the following objective modified to allow streaming (time-varying) data. For Maximum Likelihood Estimation we can represent the log-likelihood of a single time-step of the data stream as follows:

$$\mathcal{L}(\{\mathcal{X}^j, r^j, \delta\}; \theta) = (1-\delta)\ln \mathbb{P}(T > r^j|\theta, \mathcal{X}^j) + \delta \ln \mathbb{P}(T = r^j|\theta, \mathcal{X}^j). \tag{4}$$

Where δ is an indicator of whether the individual experienced the event or was lost to follow-up (censored) and θ is the set of parameters to be inferred. This is a direct consequence of Equation 3. Now, leveraging the assumed statefulness, we can rewrite the loss factorized over each time-step as:

$$\mathbb{P}(\{r^1, r^2, ...r^j\}|\theta, \mathcal{X}^j) = \prod_j \mathbb{P}(T = r^j|\theta, \mathcal{X}^j), \text{ or } \prod_j \mathbb{P}(T > r^j|\theta, \mathcal{X}^j). \tag{5}$$

From Equation 5 we can rewrite the loss in Equation 4 as a sum over each time-step as

$$\mathcal{L}(\{\mathcal{X}, t, \delta\}; \theta) = \sum_j (1-\delta)\ln \mathbb{P}(T > r^j|\theta, \mathcal{X}^j) + \delta \ln \mathbb{P}(T = r^j|\theta, \mathcal{X}^j).$$

Here, $\mathbb{P}(T > r^j|\theta, \mathcal{X}^j)$ and $\mathbb{P}(T = r^j|\theta, \mathcal{X}^j)$ are functions of the input representation:

$$\mathbb{P}(T = r^j|\theta, \mathcal{X}^j) = \sum_k \text{softmax}(f\big(\overrightarrow{\text{RNN}}(\mathcal{X}^j)\big))\mathbb{P}\big((T = r^j|Z = k, \overrightarrow{\text{RNN}}(\mathcal{X}^j)\big).$$

Note that here the term $\overrightarrow{\text{RNN}}(\cdot)$ refers to the output of the Recurrent Neural Network (LSTM or GRU). $\mathbb{P}(T = r^j|Z = k, \overrightarrow{\text{RNN}}(\mathcal{X}^j))$ is obtained by making a parametric choice (like the 'Weibull' distribution) and the shape and scale parameters for each Z modelled as functions of $\overrightarrow{\text{RNN}}(\mathcal{X}^j)$. $\mathbb{P}(T = r^j|\theta, \mathcal{X}^j)$ is defined similarly for the uncensored data.

And finally, the full log-likelihood over the entire dataset, $\mathcal{D} := \{(\mathcal{X}_i, t_i, \delta_i)\}_{i=1}^N$ is:

$$\mathcal{L}(\mathcal{D};\theta) = \sum_i^{|\mathcal{D}|}\sum_j (1-\delta_i)\ln \mathbb{P}(T > r_i^j|\theta, \mathcal{X}_i^j) + \delta_i \ln \mathbb{P}(T = r_i^j|\theta, \mathcal{X}_i^j).$$

Here we introduce subscript i to refer to a single individual in the dataset. This likelihood can be optimized using a gradient based, first order optimizer. In practice, we optimize a lower bound of the following likelihood similar to Equation 2 as was proposed for DSM.

3. Experiments

3.1. Task and Dataset

We work with the large publicly available dataset of over 50,000 ICU admissions from the Beth Israel Deaconess Medical Center, MIMIC III (Johnson et al., 2016). In critical care scenarios, it is of interest to be able to accurately quantify the Length-of-Stay (LOS) of an admitted patient, and their in-hospital mortality. Such estimation allows decision makers to help allocate adequate healthcare resources including staff and equipment, as well as get a sense of the seriousness of the patient's condition and guide their triage upon admission, and treatment thereafter. For our experiments, we use MIMIC-extract (Wang et al., 2020) to obtain a subset of all patients who were in the ICU for at least 30 hours. Our subset includes all measurements including vital signs and medications administered, sampled every hour. A stay at ICU can be terminated by either of the two competing events: Discharge, or Death. For mortality prediction, we define the target variable to be time in the ICU from admission till death, with discharge being a censoring event. For Length-of-Stay prediction, the target variable is time from admission till discharge, with death being the censoring event.

3.2. Evaluation

We use the first 24 hours of data from admission to estimate if the remaining ICU Length of Stay would exceed 1, 3 or 7 days. Simultaneously, we aim to estimate if a patient would experience death within the next 1, 3 or 7 days. Among 24,430 patients spending at least 30 hours in the ICU, 4,886 were used as a test set to evaluate the performance of models, while the rest were left for training and hyper-parameter tuning. For both RDSM and the baselines, the best set of hyperparameters were chosen to maximize the log-likelihood on a development set consisting of 2,443 patients.

We evaluate performance in terms of Area under the Receiver Operating Characteristic curve (AuROC) and Brier score on a left-out test set for the two tasks: 'Length of Stay' and 'Death'. Metrics were measured on the agglomerated risks computed every hour during the first 24 hours after admission.

During evaluation, the AuROC and Brier Scores are computed by considering censoring events as positive (negative) for Length-of-Stay (Mortality) prediction.[2] The metrics could also be adjusted to account for the censoring using Inverse Propensity of Censoring Weighting (IPCW) by employing a Kaplan-Meier estimator of the censoring distribution (Uno et al., 2007; Hung and Chiang, 2010) as is popular in survival analysis. We however avoid this approach as in ICU or critical-care settings time-to-event (death) and censoring time (discharge) are not independent, leading to biased estimates when adjusted using the Kaplan-Meier estimator.

3.3. Baselines

We compare the performance of RDSM to the standard Deep Survival Machines (DSM) model which assumes data at each time-step to be independent. We also benchmark performance against the *DeepSurv* model that assumes proportional hazards and *DeepHit* model that discretizes event times.

3.4. Hyperparameter Choices

We performed a simple grid search to coarsely optimize performance of the RDSM model. The choices of hyperparameters include the type of the recurrent neural network cell (selected from vanilla RNNs, LSTMs or GRUs), the batch size (selected from $\{125, 250\}$), the learning rate ($\{1 \times 10^{-3}, 1 \times 10^{-4}\}$), the number of hidden layers ($\{1, 2, 3\}$), the dimensionality of the hidden layer ($\{50, 100, 200\}$), and the number of underlying parametric distributions to use for the mixture ($k \in \{3, 4, 6\}$). For fair comparison the baseline models were also optimized over the same grid design as for RDSM. Additionally, all models were trained using the Adam optimizer (Kingma and Ba, 2014). For fair comparison, we employed early stopping by evaluating the likelihood on a 10% subset of the training set.

3.5. Results

As summarized in Tables 1 and 2, performance of the DSM model is improved by the use of Recurrent Neural Networks across all the tasks, a clear indicator that considering time-varying covariates allows to better model the multivariate longitudinal time series data such as MIMIC-III. Furthermore RDSM demonstrates competitive performance when compared to other published deep learning based survival approaches. As compared to approaches such as DeepHit, RDSM does not involve discretization of the event times leading to risk estimates that are better calibrated as well as making inference scalable.

[2] This is an intuitive choice: a discharged patient is not likely to experience mortality immediately post ICU admission. Similarly, a dead patient most likely had poor physiology and would end up with a longer ICU length of stay. **Note**: This adjustment is only made during evaluation. At training time censored observations are treated as censored.

Model	Death<1	Death<3	Death<7	LOS>1	LOS>3	LOS>7
DeepSurv	0.897 (0.004)	0.859 (0.003)	0.841 (0.002)	0.740 (0.002)	0.715 (0.002)	0.756 (0.003)
DeepHit	0.897 (0.004)	0.859 (0.003)	0.798 (0.003)	0.851 (0.001)	0.724 (0.001)	0.786 (0.002)
DSM	0.898 (0.004)	0.863 (0.002)	0.846 (0.002)	0.819 (0.002)	0.737 (0.001)	**0.801 (0.001)**
RDSM	**0.923 (0.003)**	**0.890 (0.002)**	**0.872 (0.002)**	**0.864 (0.002)**	**0.740 (0.002)**	0.796 (0.003)

Table 1: Area under ROC Curves on the MIMIC-III dataset for Length of Stay (LOS) and Mortality Prediction (Death). (95% CIs were generated by bootstrapping the test set, DSM: Deep Survival Machines, RDSM: Recurrent Deep Survival Machines)

Model	Death<1	Death<3	Death<7	LOS>1	LOS>3	LOS>7
DeepSurv	0.005 (<0.001)	0.027 (<0.001)	0.058 (<0.001)	0.115 (<0.001)	0.215 (0.001)	0.096 (0.001)
DeepHit	0.005 (<0.001)	0.028 (<0.001)	0.168 (<0.001)	0.084 (0.001)	0.224 (<0.001)	0.100 (0.001)
DSM	0.005 (<0.001)	0.027 (<0.001)	0.059 (<0.001)	0.080 (0.001)	0.198 (<0.001)	**0.088 (0.001)**
RDSM	0.005 (<0.001)	**0.026 (0.001)**	0.059 (0.002)	**0.073 (<0.001)**	**0.193 (0.001)**	0.091 (0.001)

Table 2: Brier score on the MIMIC-III dataset for Length of Stay (LOS) and Mortality Prediction (Death). (95% CIs were generated by bootstrapping the test set, DSM: Deep Survival Machines, RDSM: Recurrent Deep Survival Machines)

4. Discussion

We observe that the proposed approach shows leading performance when making predictions at shorter horizons of time, while still being comparable at longer horizons. We believe that these results can be further improved by making more appropriate parametric choices to model the event time distributions. Further research will involve more flexible parametric assumptions to encourage robust performance across a wide range of time horizons.

We have demonstrated that the use of recurrent neural architectures helps improve representation learning capability for data with time-varying covariates such as time series vital signs. Currently, the MIMIC data we work with were preprocessed to obtain hourly resolution data by aggregating observations at regular intervals and imputing missing entries. There are significant challenges with such real-world healthcare data including missing values and irregular sampling frequencies. Extensions to RDSM could involve integrating handling of missing data into the overall process. The use of RNNs for data imputation as has been demonstrated to have utility in clinical contexts like for example, the GRU-D model (Che et al., 2018). In addition, irregularly or asynchronously sampled time series could be directly handled by time aware RNNs without imputation as shown in Baytas et al. (2017) and Rubanova et al. (2019).

Extensions could also involve the use of attention in order to capture and characterize specific events in the patients history relevant to the outcome of interest enabling studies towards establishing causal relationships between observable factors and outcomes in such data. Future work could also involve the use of generative models or adversarial training to better learn robust representations of the temporal data.

5. Related Work

The surge of deep learning methods for machine learning have prompted related research in the use of deep learning for augmenting classic survival models. Katzman et al. (2016)

propose *DeepSurv*, a proportional hazard model where relative risks are estimated using a neural network. DeepSurv allows to model non-linear proportional hazards however, is still restricted to the strong cox assumptions of proportional hazards. Recently Nagpal et al. (2021b) proposed a mixture of Cox models to overcome the PH assumption.

Lee et al. (2019) propose *DeepHit* that involves discretizing the event space with fixed intervals and treating the survival analysis problem as binary classification over these horizons. *DeepHit* has competitive performance by alleviating the proportional hazards assumption but scales poorly to large datasets especially at longer horizons. Ren et al. (2019) propose an RNN based model but only in the context of static features and do not discuss time-varying covariates.

Apart from Deep Learning approaches popular approaches for survival analysis also include non-parametric techniques including Random Survival Forests (Ishwaran et al., 2008) and Gaussian Processes (Fernández et al., 2016; Alaa and van der Schaar, 2017).

Statistical methods for longitudinal data have proposed to model censored survival data by analysing different follow up times, commonly known as landmark times. These approaches involve building separate models for each time (Van Houwelingen, 2007). Two stage landmarking have been explored to model the disease evolution within those intervals and extract meaningful representations which are then used to build standard survival regression models van Houwelingen and Putter (2011).

Another approach aims to model longitudinal and survival outcomes simultaneously by sharing a latent representation (Henderson et al., 2000). This joint modeling benefits from shared knowledge between models but suffers from intractability and computational complexity.

Modern ML methods for modeling time-to-event outcomes in the presence of time-varying covariates are relative understudied. Lee et al. (2019) propose *Dynamic-DeepHit* involving recurrent neural networks but suffers from similar limitations as the original *DeepHit* model. In a similar vein, Jarrett et al. (2019) further propose *MatchNet* involving temporal convolution networks.

6. Conclusion

We proposed an extension of the Deep Survival Machines model to handle temporal data with time-varying coefficients. Our approach uses recurrent neural networks to handle long term temporal dependencies in the input data stream. Our approach obtained favorable results from empirical evaluation of ourimodel on predicting in-hosptial ICU mortality and length of stay. We also made the software implementation of our tool available to the survival analysis research community online as an open source package.

In the future, we envision extensions that employ generative techniques to impute missing data in clinical time series (Lipton et al., 2016) as well as subgroup discovery to identify most at-risk sub-populations (Nagpal et al., 2020).

Acknowledgments

The authors would like to thank the reviewers and program commitee of the 2021 AAAI Spring Symposium on Survival Analysis. This work was partially supported by the Defense

Advanced Research Projects Agency under award FA8750-17-2-0130. Vincent Jeanselme received funding from the UKRI Medical Research Council MC-UU-00002/5.

References

Ahmed M Alaa and Mihaela van der Schaar. Deep multi-task gaussian processes for survival analysis with competing risks. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 2326–2334, 2017.

Inci M Baytas, Cao Xiao, Xi Zhang, Fei Wang, Anil K Jain, and Jiayu Zhou. Patient subtyping via time-aware lstm networks. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 65–74, 2017.

Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. Recurrent neural networks for multivariate time series with missing values. *Scientific reports*, 8(1):1–12, 2018.

Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.

Tamara Fernández, Nicolás Rivera, and Yee Whye Teh. Gaussian processes for survival analysis. *Advances in Neural Information Processing Systems*, 29:5021–5029, 2016.

Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. Learning to forget: Continual prediction with lstm. 1999.

Robin Henderson, Peter Diggle, and Angela Dobson. Joint modelling of longitudinal measurements and event time data. *Biostatistics*, 1(4):465–480, 2000.

Hung Hung and Chin-Tsang Chiang. Estimation methods for time-dependent auc models with survival data. *Canadian Journal of Statistics*, 38(1):8–26, 2010.

Hemant Ishwaran, Udaya B Kogalur, Eugene H Blackstone, Michael S Lauer, et al. Random survival forests. *The annals of applied statistics*, 2(3):841–860, 2008.

Daniel Jarrett, Jinsung Yoon, and Mihaela van der Schaar. Dynamic prediction in clinical survival analysis using temporal convolutional networks. *IEEE Journal of Biomedical and Health Informatics*, 24(2):424–436, 2019.

Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-Wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3(1):1–9, 2016.

Jared L Katzman, Uri Shaham, Alexander Cloninger, Jonathan Bates, Tingting Jiang, and Yuval Kluger. Deep survival: A deep cox proportional hazards network. *stat*, 1050(2), 2016.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Changhee Lee, Jinsung Yoon, and Mihaela Van Der Schaar. Dynamic-deephit: A deep learning approach for dynamic survival analysis with competing risks based on longitudinal data. *IEEE Transactions on Biomedical Engineering*, 67(1):122–133, 2019.

Zachary C Lipton, David C Kale, Randall Wetzel, et al. Modeling missing data in clinical time series with rnns. *Machine Learning for Healthcare*, 56, 2016.

Chirag Nagpal, Dennis Wei, Bhanukiran Vinzamuri, Monica Shekhar, Sara E Berger, Subhro Das, and Kush R Varshney. Interpretable subgroup discovery in treatment effect estimation with application to opioid prescribing guidelines. In *Proceedings of the ACM Conference on Health, Inference, and Learning*, pages 19–29, 2020.

Chirag Nagpal, Xinyu Rachel Li, and Artur Dubrawski. Deep survival machines: Fully parametric survival regression and representation learning for censored data with competing risks. *IEEE Journal of Biomedical and Health Informatics*, 2021a.

Chirag Nagpal, Steve Yadlowsky, Negar Rostamzadeh, and Katherine Heller. Deep cox mixtures for survival regression. *arXiv preprint arXiv:2101.06536*, 2021b.

Kan Ren, Jiarui Qin, Lei Zheng, Zhengyu Yang, Weinan Zhang, Lin Qiu, and Yong Yu. Deep recurrent survival analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4798–4805, 2019.

Yulia Rubanova, Ricky TQ Chen, and David Duvenaud. Latent odes for irregularly-sampled time series. *arXiv preprint arXiv:1907.03907*, 2019.

Hajime Uno, Tianxi Cai, Lu Tian, and Lee-Jen Wei. Evaluating prediction rules for t-year survivors with censored regression models. *Journal of the American Statistical Association*, 102(478):527–537, 2007.

Hans van Houwelingen and Hein Putter. *Dynamic prediction in clinical survival analysis.* CRC Press, 2011.

Hans C Van Houwelingen. Dynamic prediction by landmarking in event history analysis. *Scandinavian Journal of Statistics*, 34(1):70–85, 2007.

Shirly Wang, Matthew BA McDermott, Geeticka Chauhan, Marzyeh Ghassemi, Michael C Hughes, and Tristan Naumann. Mimic-extract: A data extraction, preprocessing, and representation pipeline for mimic-iii. In *Proceedings of the ACM Conference on Health, Inference, and Learning*, pages 222–235, 2020.