Transformer-Based Deep Survival Analysis

Shi Hu S.HU@UVA.NL
University of Amsterdam

Egill A. Fridgeirsson E.AXFJORD@AMSTERDAMUMC.NL
Amsterdam UMC

Guido van Wingen G.A.VANWINGEN@AMSTERDAMUMC.NL
Amsterdam UMC

Max Welling M.WELLING@UVA.NL
University of Amsterdam

Abstract

In this work, we propose a new Transformer-based survival model which estimates the patient-specific survival distribution. Our contributions are twofold. First, to the best of our knowledge, existing deep survival models use either fully connected or recurrent networks, and we are the first to apply the Transformer in survival analysis. In addition, we use ordinal regression to optimize the survival probabilities over time, and penalize randomized discordant pairs. Second, many survival models are evaluated using only the ranking metrics such as the concordance index. We propose to also use the absolute error metric that evaluates the precise duration predictions on observed subjects. We demonstrate our model on two publicly available real-world datasets, and show that our mean absolute error results are significantly better than the current models, meanwhile, it is challenging to determine the best model under the concordance index.

Keywords: survival analysis, Transformers, deep learning.

1. Introduction

Survival analysis is an important branch in statistics, which estimates the expected duration of time until an event happens. It is used in a wide range of domains, such as medicine, engineering and economics. For example, a hospital can use survival analysis techniques to estimate how long a COVID-19 patient will stay in the intensive care unit (ICU), a factory can use these techniques to estimate the time until a machine breaks down, and a government can estimate the duration of an economic recession.

Earlier survival models, such as the Cox model (Cox, 1972), are linear models. Recently, a number of deep survival models, such as (Giunchiglia et al., 2018; Lee et al., 2018; Ren et al., 2019), have been proposed. These models can estimate the patient-specific survival probabilities over time, but use either fully connected or recurrent neural networks. Meanwhile, the Transformer (Vaswani et al., 2017) is an accurate and efficient model which handles sequential data, and has applications in several domains; however, to the best of our knowledge, it has not been used in survival analysis. Hence, in this work, we propose a new Transformer-based deep survival model as well as a new training objective. Further, in survival analysis, there are often censored subjects whose durations are not observed, e.g., in a clinical study, some subjects can be still alive when the study ends or were not followed after a certain time. For these subjects, we cannot evaluate the precise duration predictions. We can only compare them with the observed (and smaller) durations and evaluate the pairwise orderings. As a result, many survival models are evaluated using only the ranking metrics, such as the concordance index (C-index) (Harrell et al., 1982; Antolini et al., 2005). On the other hand, it is highly valuable if a model can accurately estimate the *precise* survival duration per subject. For example, in the current COVID-19 pandemic, if a hospital can predict how long each patient will stay in the ICU, as opposed to merely who leaves the ICU first, it can prioritize the medical resources and treat the patients more effectively. Thus, we propose to use both the C-index and the mean absolute error (MAE) metrics to evaluate the results. The former evaluates the pairwise orderings of the duration predictions on observed and censored subjects, while the latter evaluates the precise duration predictions on observed subjects.

We demonstrate our model on two real-world datasets with these two metrics. We show that our MAE results are significantly better than the current models, meanwhile, it is challenging to determine the best model under the C-index.

2. Notation

We denote the probability by P, time by t, T or τ, features by X, event density by $f(t)$, hazard function by $\lambda(t)$, cumulative hazard by $\Lambda(t)$, and survival probability by $S(t)$. The estimates are marked with the caret symbol, e.g., $\hat{S}(t)$ is an estimate of $S(t)$.

For continuous survival models, the hazard function represents the instantaneous failure rate, and the survival probability is $S(t) = \exp(-\Lambda(t))$. For discrete models, the hazard function represents the conditional probability that the patient dies at time t, given he/she was alive before t, and the survival probability is $S(t) = \prod_{\tau=0}^{t} 1 - \lambda(\tau)$.

3. Related Work

Proportional hazards models are a popular class of survival models, where the hazard function $\lambda(t \mid X)$ is the product of two parts: the base hazard function $\lambda_0(t)$ and the effect of the features $g(X)$. $\lambda_0(t)$ is predefined and depends only on time t, whereas $g(X)$ is learned during training. Since $\lambda_0(t)$ is not trained, these models are semi-parametric. The Cox model is a widely used example, where $g(X) = \exp(\theta^\top X)$. Subsequent works, such as (Faraggi and Simon, 1995; Luck et al., 2017; Katzman et al., 2018), extend this idea using more advanced models to compute $g(X)$, and more sophisticated training losses. Furthermore, the effect can be time-dependent, e.g. (Fernández et al., 2016) uses Gaussian processes to model the joint effect of the features and time.

Fully parametric models have also been used in survival analysis. For example, (Ranganath et al., 2016) and (Martinsson, 2017) assume the survival distribution of each patient is Weibull, and predict the event times using a deep hierarchical generative model and an RNN respectively. (Yang et al., 2017) and (Avati et al., 2019) assume each survival distribution is log-normal, and use RNNs to predict the event times with the MLE and Survival-CRPS training objectives. In sum, both semi-parametric and fully parametric models need to make assumptions about the survival distribution, and the predictions are accurate when these assumptions hold.

The Kaplan-Meier (Kaplan and Meier, 1958) and Nelson-Aalen estimators (Nelson, 1969, 1972; Aalen, 1978) are widely used non-parametric models. They estimate the survival distribution of the entire population using the times of observed and censored patients. However, they cannot estimate the patient-specific survival distribution since they do not use the features, while other non-parametric models can, e.g., random survival forests (Ishwaran et al., 2008).

A number of deep survival models have recently been proposed to estimate the patient-specific survival distribution. For example, DeepHit (Lee et al., 2018) combines fully connected networks to learn the joint probability distribution of the first hitting time and the competing risk, and RNN-SURV (Giunchiglia et al., 2018) and DRSA (Ren et al., 2019) use recurrent models to estimate the survival probabilities over time.

Temporal point processes (TPPs) are related to survival analysis, which estimate the time of the next occurrence of an event given its previous occurrences. TPPs have been modeled using recurrent neural networks (Du et al., 2016; Mei and Eisner, 2017) and the Transformer (Zhang et al., 2020). In survival analysis, however, we usually do not have the history of the event (e.g. death), as it occurs only once; in addition, we need to handle censored subjects.

Lastly, the absolute error evaluation metrics have been used in previous works, such as (Yu et al., 2011) and (Yang et al., 2017), though neither uses the ranking metrics. In this work, we use both metrics as they complement each other.

4. Method

In survival analysis, the training data consists of the features and time pairs (X_i, T_i), where T_i can be observed or censored (we consider only right censoring). If we fit continuous models, we can maximize the following log-likelihood function:

$$\begin{aligned}\mathcal{L}_{\text{continuous}} &= \sum_{i \in \text{observed}} \log f(T_i \mid X_i) + \sum_{i \in \text{censored}} \log S(T_i \mid X_i) && (1)\\ &= \sum_{i \in \text{observed}} \log \lambda(T_i \mid X_i) + \log S(T_i \mid X_i) + \sum_{i \in \text{censored}} \log S(T_i \mid X_i) && (2)\\ &= \sum_{i \in \text{observed}} \log \lambda(T_i \mid X_i) - \Lambda(T_i \mid X_i) + \sum_{i \in \text{censored}} -\Lambda(T_i \mid X_i). && (3)\end{aligned}$$

However, it is easier to fit discrete models on computers. In this case, the hazard function represents the conditional probability that the patient dies at time t, given he/she was alive before t as follows:

$$\lambda(t \mid X) = \mathsf{P}_X(T = t \mid T > t - 1). \tag{4}$$

Let $q(t \mid X) = 1 - \lambda(t \mid X)$, then the survival probability can be written as:

$$\begin{aligned} S(t \mid X) =& \mathsf{P}_X(T > t \mid T > t-1) \cdot \mathsf{P}_X(T > t-1) && (5)\\ =&(1 - \mathsf{P}_X(T = t \mid T > t-1)) \cdot \mathsf{P}_X(T > t-1) && (6)\\ =&q(t \mid X) \cdot S(t-1 \mid X). && (7) \end{aligned}$$

By recursively expanding Eq. 7, we obtain the final expression of $S(t \mid X)$ as follows:

$$S(t \mid X) = \prod_{\tau=0}^{t} q(\tau \mid X), \tag{8}$$

where $q(0 \mid X) = 1 - \mathsf{P}_X(T = 0)$. Then, for each patient, we use ordinal regression to optimize the survival probabilities $S(t \mid X)$ for $t = 0, 1, 2, \ldots,$ and use a second loss to penalize the discordant pairs. The details of the losses will be discussed later.

The Transformer model was originally designed to solve NLP tasks, where the inputs are sentences. In our case, for each patient, we use the encoder of the Transformer to predict the complement of the hazard function $q(t \mid X)$ for all times up to $T_{\max}$, where $T_{\max}$ is a hyperparameter. We treat each patient as a 'sentence', and each 'word' is the sum of the feature embedding and the positional encoding of a time t, where $t = 0, 1, 2, \ldots, T_{\max}$ (the inputs are not masked). Thus, each 'word' represents the interaction between the patient and time t. A diagram of our model is shown in Figure 1. Compared to the recurrent models that compute the outputs sequentially, such as (Giunchiglia et al., 2018; Ren et al., 2019), the Transformer computes them in parallel, which is much more efficient.

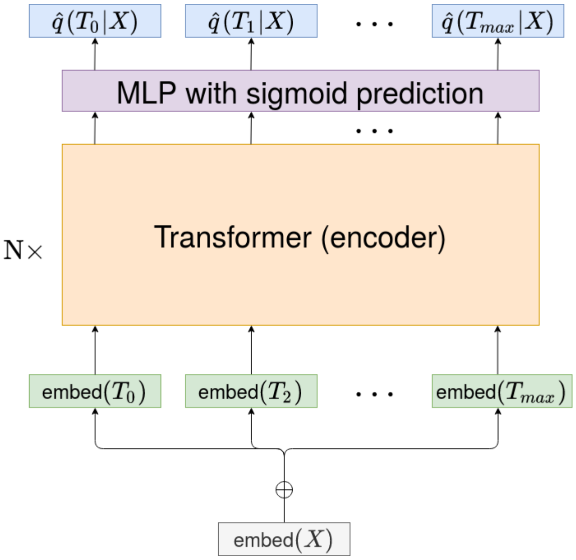

Figure 1: A diagram of the Transformer-based deep survival model.

For feature embeddings, we use a simple MLP with one fully connected layer of shape $M \times d_{\text{model}}$ followed by layer normalization (Ba et al., 2016), where M is the number of input features and d_{model} is the embedding dimension of the Transformer model. For time embeddings, we use the same sine and cosine functions of different frequencies as in (Vaswani et al., 2017):

$$\mathrm{TE}(t, 2i) = \sin\big(\frac{t}{10000^{2i/d_{\text{model}}}}\big), \tag{9}$$

$$\mathrm{TE}(t, 2i+1) = \cos\big(\frac{t}{10000^{2i/d_{\text{model}}}}\big). \tag{10}$$

The encoder output for each time t is the transformed embedding of shape $1 \times d_{\text{model}}$, and we use an MLP to predict the complement of the hazard function. First, we multiply each output embedding with a fully connected layer of shape $d_{\text{model}} \times \frac{d_{\text{model}}}{2}$ followed by the rectified linear unit (ReLU) (Nair and Hinton, 2010) and layer normalization, then we predict $q(t \mid X)$ using a second fully connected layer of shape $\frac{d_{\text{model}}}{2} \times 1$ followed by sigmoid. Thus, for each patient, the outputs of the encoder are a vector of shape $1 \times (T_{\max} + 1)$ as follows:

$$\hat{y}_X = \big[\hat{q}(0 \mid X), \hat{q}(1 \mid X), \ldots, \hat{q}(T_{\max} \mid X)\big]. \tag{11}$$

In the continuous-time survival analysis, the mean lifetime of a patient is the area under the survival curve (using integration by parts). In the discrete-time, we can approximate it by the sum of the survival probabilities up to $T_{\max}$ as follows:

$$\mu_X = \int_0^\infty t f(t \mid X) dt = \int_0^\infty S(t \mid X) dt \approx \sum_{t=0}^{T_{\max}} S(t \mid X). \tag{12}$$

Further, by expanding $S(t \mid X)$ using Eq. 8, the mean lifetime can be estimated as:

$$\hat{\mu}_X = \sum_{t=0}^{T_{\max}} \hat{S}(t \mid X) = \sum_{t=0}^{T_{\max}} \prod_{\tau=0}^{t} \hat{q}(\tau \mid X). \tag{13}$$

Alternatively, we could directly predict the survival probability $S(t \mid X)$ at each time t. However, $S(t \mid X)$ is a monotonically decreasing function, so all model weights need to have the same sign to preserve monotonicity (Omi et al., 2019). In contrast, our model does not have this constraint, since the survival probability at time t is the product of that at time $t-1$ and the complement of the hazard at t, which is between 0 and 1.

Next, for the observed case, we let T denote the event time, and minimize the following ordinal regression loss:

$$\mathcal{L}_X^{obs} = -\sum_{t=0}^{T-1} \log \hat{S}(t \mid X) - \sum_{t=T}^{T_{\max}} \log\big[1 - \hat{S}(t \mid X)\big]. \tag{14}$$

In other words, we maximize the survival probabilities $\hat{S}(t \mid X)$ for $t < T$, and minimize them for $t \geq T$. For the right-censored case, T denotes the censoring time, and we minimize the following truncated ordinal regression loss:

$$\mathcal{L}_X^{cen} = -\sum_{t=0}^{T} \log \hat{S}(t \mid X), \tag{15}$$

which means we maximize the survival probabilities $\hat{S}(t \mid X)$ for $t \leq T$. In sum, for the observed case, the ordinal regression loss optimizes all survival probabilities up to $T_{\max}$, and for the censored case, it optimizes up to T. This is different from the previous approaches, such as DeepHit, which use the softmax classifier to predict the survival distribution. Our approach can achieve better results since ordinal regression is known to perform better than softmax on ordinal data (Cheng et al., 2008).

We found the following loss, which penalizes the randomized discordant pairs, further improves the performance. Denote T_i and T_j as the times for patients i and j, where T_i is observed and $T_i < T_j$. The predicted survival durations $\hat{T}_i$ and $\hat{T}_j$ (by Eq. 13) are discordant if $\hat{T}_i > \hat{T}_j$, and we want to reduce the number of discordant pairs. In theory, we could penalize all of them during training. However, for a training set of size N, enumerating all discordant pairs takes $\mathcal{O}(N^2)$ time, which is highly inefficient even for a moderate size, say $N = 1000$. Instead, we propose the following randomized algorithm with an $\mathcal{O}(N)$ runtime: for each observed patient i in the training set, we randomly sample another patient j where $T_j > T_i$ (in the experiments, j is sampled with replacement). Since T_j can be censored, the true survival duration for j cannot be smaller than T_j. Thus, the difference between $\hat{T}_j$ and $\hat{T}_i$ should be at least $T_j - T_i$. Furthermore, since the true duration T_i is observed, the predicted duration $\hat{T}_i$ should be close to T_i, and we use the MAE loss $|T_i - \hat{T}_i|$ to penalize their difference. In sum, the loss that penalizes the discordant pairs is:

$$\mathcal{L}_{X_i}^{disc} = \max\left[0, (T_j - T_i) - (\hat{T}_j - \hat{T}_i)\right] + |T_i - \hat{T}_i|. \tag{16}$$

If $\hat{T}_i$ and $\hat{T}_j$ are concordant and separated by at least $T_j - T_i$, then $\mathcal{L}_{X_i}^{disc}$ becomes the MAE loss.

The total loss is the combination of the above three losses as follows:

$$\mathcal{L} = \sum_{i \in \text{observed}} \left[\mathcal{L}_{X_i}^{obs} + \alpha \mathcal{L}_{X_i}^{disc}\right] + \sum_{i \in \text{censored}} \mathcal{L}_{X_i}^{cen}, \tag{17}$$

where α is a hyperparameter.

5. Experiments

5.1. Datasets

We use two publicly available real-world datasets[1]: the Molecular Taxonomy of Breast Cancer International Consortium (METABRIC) (Curtis et al., 2012) and Study to Under-

[1] They can be found on Github at: https://github.com/jaredleekatzman/DeepSurv/tree/master/experiments/data.

stand Prognoses Preferences Outcomes and Risks of Treatment (SUPPORT) (Knaus et al., 1995). We use the same data pre-processing steps as (Katzman et al., 2018) but merge their training and test sets. The data descriptions are shown below.

- **METABRIC**. The pre-processed dataset contains the records of 1904 breast cancer patients. It has 9 features, which include 4 gene indicators: MKI67, EGFR, PGR and ERBB2, and 5 clinical features: hormone treatment indicator, radiotherapy indicator, chemotherapy indicator, ER-positive indicator and age at diagnosis.
- **SUPPORT**. The pre-processed dataset contains the records of 8873 seriously ill hospitalized adults. It has 14 features, which are age, sex, race, number of comorbidities, presence of diabetes, presence of dementia, presence of cancer, mean arterial blood pressure, heart rate, respiration rate, temperature, white blood cell count, serum's sodium and serum's creatinine.

We predict the survival duration of each patient (month is used as the unit of time). The results are evaluated using cross-validation, where we randomly split each dataset 4 times into a training, validation and test set with ratio 7/1/2. For each split, we train the model on the training set, select the best model on the validation set, and evaluate the results on the test set. For ease of comparison, we do not perform input normalization or exclude any test subject. Table 1 shows the datasets details.

Table 1: Details of the METABRIC and SUPPORT datasets.

Dataset	Total	Observed (%)	Censored (%)	Features	Max Month	Min Month
META	1904	1103 (58%)	801 (42%)	9	355	0
SUPP	8873	6036 (68%)	2837 (32%)	14	68	0

5.2. Evaluation Metrics

We use the widely adopted C-index as the first evaluation metric. It is a ranking metric defined as the ratio of the number of concordant pairs to the total comparable pairs, and we compute it in a similar way to (Uno et al., 2011) as the following:

$$\text{C-index} = \frac{\sum_{i,j} \mathbb{1}[T_i < T_j] \cdot \mathbb{1}[\hat{T}_i < \hat{T}_j] \cdot \delta_i}{\sum_{i,j} \mathbb{1}[T_i < T_j] \cdot \delta_i}, \tag{18}$$

where $\delta_i = 1$ if T_i is observed, and 0 otherwise. Since the C-index does not evaluate the exact survival durations, an inaccurate model can still achieve a high score. For example, if a model has a significant systematic error that adds 10 years to every prediction, the C-index will be the same, but the absolute error can be arbitrarily high. Therefore, as a complement, we propose to use MAE as the second metric, which evaluates the precise duration predictions on observed subjects as follows:

$$\text{MAE} = \frac{\sum_i |T_i - \hat{T}_i| \cdot \delta_i}{\sum_i \delta_i}. \tag{19}$$

This metric is especially relevant when the majority are observed, which is the case for both datasets in Table 1. In comparison, there are two types of pairs that the C-index cannot evaluate: 1. both patients are censored, and 2. one patient is observed and the other censored, and the observed time is greater than the censoring time. Hence, the C-index can evaluate at most $1 - (1 - 0.58)^2 \approx 82\%$ and $1 - (1 - 0.68)^2 \approx 90\%$ of pairs. In sum, neither metric covers all subjects. The C-index covers more than MAE, but it is a ranking metric, and therefore less precise.

5.3. Results

We compare with four baseline models, which are the Cox model (Cox, 1972), random survival forests (RSF) (Ishwaran et al., 2008), DeepSurv (Katzman et al., 2018) and DeepHit (Lee et al., 2018). We cannot compare with DRSA (Ren et al., 2019), since it requires customized input encoding. For implementations, we use lifelines[2] for the Cox model, scikit-survival[3] for RSF, and pycox[4] for both DeepSurv and DeepHit. Their hyperparameter spaces and optimal values are shown in Appendix A.

For our Transformer model, we use the implementation by OpenNMT (Klein et al., 2017). Since the two datasets in Table 1 are much smaller than the standard NLP datasets (e.g., the WMT 2014 English-German dataset used in (Vaswani et al., 2017) contains about 4.5 million sentence pairs), we use the following hyperparameter spaces, which are also smaller than the original:

- number of attention layers: $\{1, 2, 3, 4\}$
- number of heads: $\{1, 2, 4, 8\}$
- d_{model}: $\{256, 512\}$
- dropout (Srivastava et al., 2014) rate (DR): $\{0.0, 0.1, 0.3, 0.5\}$
- Adam optimizer (Kingma and Ba, 2015) learning rate (LR): {1e-4, 5e-4, 1e-3} (for simplicity, we use a fixed learning rate)
- batch size: $\{4, 8, 16, 32\}$
- α: $\{0.05, 0.1, 0.5, 1\}$
- $T_{\max}$ for METABRIC: $\{400, 450\}$
- $T_{\max}$ for SUPPORT: $\{80, 100\}$.

[2] https://lifelines.readthedocs.io/en/latest/
[3] https://scikit-survival.readthedocs.io/en/latest/
[4] https://pypi.org/project/pycox/

We use the *mean* lifetime to estimate the survival duration of each patient (we cannot use the *median*, since models such as Cox and RSF predict 'infinity' on some test subjects). Further, since the C-index covers more subjects than MAE, we use it as the early stopping criterion, and select the best model that achieves the highest mean C-index over the 4 validation sets. In addition, we use MAE as the tiebreaker. Table 2 shows the optimal hyperparameters of our model.

Table 2: The optimal hyperparameters of our model.

Dataset	Layers	Heads	d_{model}	d_{ff}	DR	LR	Batch	α	Epochs	T_{max}
META	4	4	512	2048	0.1	1e-4	16	0.1	200	400
SUPP	4	4	512	2048	0.1	1e-4	16	0.1	400	80

Next, we compute the mean result for each test set, then report their mean ± standard deviation (SD) over the 4 test sets. We use the paired t-test to determine the statistical significance, and the p-values can be found in Table 4 of Appendix B.

Figure 2 compares the C-index test results using SD error bars (the numeric results are shown in Table 5 of Appendix C). For either dataset, the difference in the mean results between the best and worst is at most 0.015 (except the Cox model in SUPPORT, which is much worse than the rest). Further, only RSF in SUPPORT is better than ours with statistical significance, but its mean result is worse than ours in METABRIC. Therefore, it is difficult to determine the best model under the C-index.

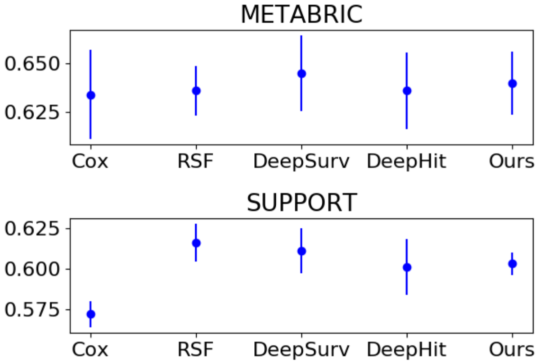

Figure 2: Patient-specific C-index test results in SD error bars.

Table 3 compares the MAE test results, and ours are better than all baseline models with statistical significance. In particular, the difference in the mean results between ours and the second-best in each dataset is 14.6 and 8.5 months. In addition, Figure 3 compares different mean survival curves with the Kaplan-Meier curve on a randomly chosen test set, where only observed patients are used (in this case, the Kaplan-Meier curve is equivalent to the empirical survival distribution of the observed patients). The figure shows that in both cases, our mean survival curve is the closest to the Kaplan-Meier curve.

We note that our MAE results can further improve if we reduce the $\mathcal{L}_{X_i}^{disc}$ loss of Eq. 16 to the MAE loss $|T_i - \hat{T}_i|$, namely, we penalize only the inaccurate duration predictions and not discordant pairs. However, this reduces the C-index accuracy.

Table 3: Patient-specific MAE test results (mean ± SD). Best results are in bold, and $\star$ means statistically significant.

Model	METABRIC	SUPPORT
Cox	78.61 ± 4.11	19.50 ± 0.24
RSF	84.17 ± 4.91	16.97 ± 0.31
DeepSurv	76.60 ± 4.82	18.24 ± 0.33
DeepHit	68.96 ± 6.74	17.11 ± 2.46
Ours	**54.33 ± 5.91***	**8.52 ± 0.73***

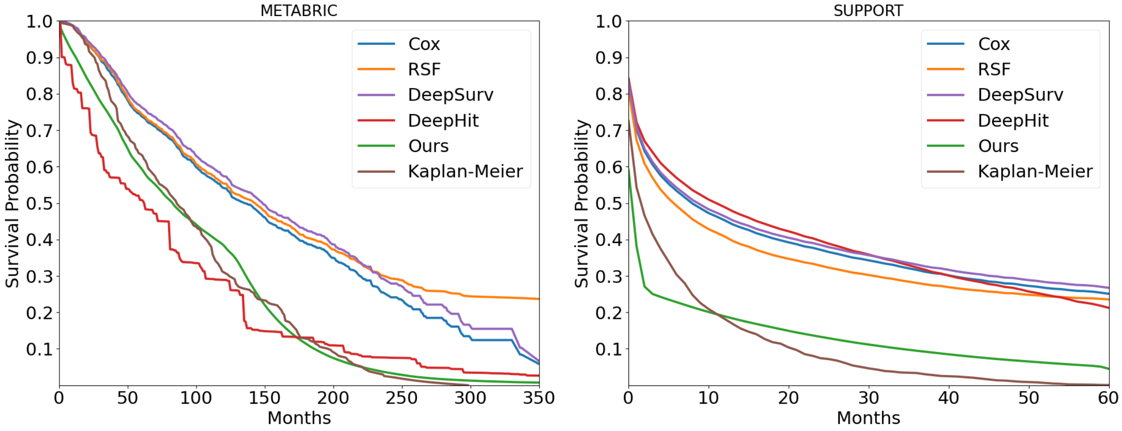

Figure 3: Comparisons of different mean survival curves with the Kaplan-Meier curve (for observed patients in test).

Lastly, Appendix E shows an ablation study which aims to determine the source of performance gain. In sum, the Transformer and fully connected models perform similarly on the two datasets when they both use the same training objective that we propose. However, the two datasets have only 1.9K and 8.9K patients (‘sentences’), which are much smaller than the standard NLP datasets. Hence, it is possible that the Transformer can achieve better results on much bigger datasets.

6. Conclusions and Future Work

In this paper, we propose a Transformer-based deep survival model that estimates the patient-specific survival distribution. Our contributions are twofold. First, to the best of our knowledge, we are the first to apply the Transformer model to survival analysis. In addition, we use ordinal regression to optimize the survival probabilities over time, and penalize randomized discordant pairs. Second, we show the C-index alone cannot adequately evaluate the survival models, since it is a ranking metric which does not evaluate the precise duration predictions. As a complement, we propose to evaluate MAE on observed subjects. We demonstrate our model on two real-world datasets, and show our MAE results are significantly better than the current models, meanwhile, it is challenging to determine the best model under the C-index.

There are several directions for future work. First, we would like to compare the performance of our Transformer model with fully connected models on bigger datasets using the same training objective. Second, to improve prediction accuracy, we can pretrain the model on a large NLP dataset, then fine-tune on the smaller survival dataset, as transfer learning has been shown to be effective for the Transformer model (Devlin et al., 2019). Third, we can use our model to handle sequential input data, such as a patient's periodic clinical measurements over time (in contrast, it is challenging for the fully connected models to process this type of data). Finally, we can improve the survival distribution predictions on censored subjects, e.g., we can penalize the duration predictions that are shorter than the censoring times.

Acknowledgments

We thank Shu Liu, Rajat Thomas and Changhee Lee for helpful discussions regarding the datasets. This research was supported by NWO Perspective Grant DLMedIA as well as the in-cash and in-kind contributions by Philips.

References

Odd Aalen. Nonparametric inference for a family of counting processes. In *Annals of Statistics*, 1978.

Laura Antolini, Patrizia Boracchi, and Elia Biganzoli. A time-dependent discrimination index for survival data. In *Statistics in Medicine*, 2005.

Anand Avati, Tony Duan, Sharon Zhou, Kenneth Jung, Nigam H. Shah, and Andrew Ng. Countdown regression: Sharp and calibrated survival predictions. In *UAI*, 2019.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. In *arXiv preprint arXiv:1607.06450*, 2016.

Jianlin Cheng, Zheng Wang, and Gianluca Pollastri. A neural network approach to ordinal regression. In *IEEE International Joint Conference on Neural Networks*, 2008.

David R. Cox. Regression models and life-tables. In *Journal of the Royal Statistical Society*, 1972.

Christina Curtis, Sohrab P. Shah, Suet-Feung Chin, Gulisa Turashvili, Oscar M. Rueda, Mark J. Dunning, Doug Speed, Andy G. Lynch, Shamith Samarajiwa, Yinyin Yuan, Stefan Graf, Gavin Ha, Gholamreza Haffari, Ali Bashashati, Roslin Russell, Steven McKinney, Anita Langerød, Andrew Green, Elena Provenzano, Gordon Wishart, Sarah Pinder, Peter Watson, Florian Markowetz, Leigh Murphy, Ian Ellis, Arnie Purushotham, Anne-Lise Børresen-Dale, James D. Brenton, Simon Tavare, Carlos Caldas, and Samuel Aparicio. The genomic and transcriptomic architecture of 2,000 breast tumours reveals novel subgroups. In *Nature*, 2012.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 2019.

Nan Du, Hanjun Dai, Rakshit Trivedi, Utkarsh Upadhyay, Manuel Gomez Rodriguez, and Le Song. Recurrent marked temporal point processes: Embedding event history to vector. In *KDD*, 2016.

David Faraggi and Richard Simon. A neural network model for survival data. In *Statistics in Medicine*, 1995.

Tamara Fernández, Nicolás Rivera, and Yee Whye Teh. Gaussian processes for survival analysis. In *NIPS*, 2016.

Eleonora Giunchiglia, Anton Nemchenko, and Mihaela van der Schaar. Rnn-surv: A deep recurrent model for survival analysis. In *ICANN*, 2018.

Frank E. Harrell, Robert M. Califf, David B. Pryor, Kerry L. Lee, and Robert A. Rosati. Evaluating the yield of medical tests. In *Journal of the American Medical Association*, 1982.

Hemant Ishwaran, Udaya B. Kogalur, Eugene H. Blackstone, and Michael S. Lauer. Random survival forests. In *The Annals of Applied Statistics*, 2008.

E. L. Kaplan and Paul Meier. Nonparametric estimation from incomplete observations. In *Journal of the American Statistical Association*, 1958.

Jared L. Katzman, Uri Shaham, Alexander Cloninger, Jonathan Bates, Tingting Jiang, and Yuval Kluger. Deepsurv: personalized treatment recommender system using a cox proportional hazards deep neural network. In *BMC Medical Research Methodology*, 2018.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. Opennmt: Open-source toolkit for neural machine translation. In *Proc. ACL*, 2017. doi: 10.18653/v1/P17-4012. URL https://doi.org/10.18653/v1/P17-4012.

W A Knaus, F E Harrell Jr, J Lynn, L Goldman, R S Phillips, A F Connors Jr, N V Dawson, W J Fulkerson Jr, R M Califf, N Desbiens, P Layde, R K Oye, P E Bellamy, R B Hakim, and D P Wagner. The support prognostic model. objective estimates of survival for seriously ill hospitalized adults. study to understand prognoses and preferences for outcomes and risks of treatments. In *Annals of Internal Medicine*, 1995.

Changhee Lee, William R. Zame, Jinsung Yoon, and Mihaela van der Schaar. Deephit: A deep learning approach to survival analysis with competing risks. In *AAAI*, 2018.

Margaux Luck, Tristan Sylvain, Héloïse Cardinal, Andrea Lodi, and Yoshua Bengio. Deep learning for patient-specific kidney graft survival analysis. In *arXiv preprint arXiv:1705.10245*, 2017.

Egil Martinsson. Wtte-rnn : Weibull time to event recurrent neural network. In *Master's thesis, Chalmers University of Technology*, 2017.

Hongyuan Mei and Jason Eisner. The neural hawkes process: A neurally self-modulating multivariate point process. In *NIPS*, 2017.

Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, 2010.

Wayne Nelson. Hazard plotting for incomplete failure data. In *Journal of Quality Technology*, 1969.

Wayne Nelson. Theory and applications of hazard plotting for censored failure data. In *Technometrics*, 1972.

Takahiro Omi, Naonori Ueda, and Kazuyuki Aihara. Fully neural network based model for general temporal point processes. In *NeurIPS*, 2019.

Rajesh Ranganath, Adler Perotte, Noémie Elhadad, and David Blei. Deep survival analysis. In *Machine Learning and Healthcare Conference*, 2016.

Kan Ren, Jiarui Qin, Lei Zheng, Zhengyu Yang, Weinan Zhang, Lin Qiu, and Yong Yu. Deep recurrent survival analysis. In *AAAI*, 2019.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. In *JMLR*, 2014.

Hajime Uno, Tianxi Cai, Michael J. Pencina, Ralph B. D'Agostino, and L. J. Wei. On the c-statistics for evaluating overall adequacy of risk prediction procedures with censored survival data. In *Statistics in Medicine*, 2011.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

Yinchong Yang, Peter A. Fasching, and Volker Tresp. Modeling progression free survival in breast cancer with tensorized recurrent neural networks and accelerated failure time models. In *Machine Learning for Healthcare*, 2017.

Chun-Nam Yu, Russell Greiner, Hsiu-Chin Lin, and Vickie Baracos. Learning patient-specific cancer survival distributions as a sequence of dependent regressors. In *NIPS*, 2011.

Qiang Zhang, Aldo Lipani, Omer Kirnap, and Emine Yilmaz. Self-attentive hawkes processes. In *ICML*, 2020.

Appendix A. Hyperparameters of the Baseline Models

A.1. The Cox Model

The hyperparameter spaces are:

- penalizer: $\{0, 0.001, 0.01, 0.1, 0.5\}$
- l1_ratio: $\{0, 0.001, 0.01, 0.1, 0.5\}$.

The optimal hyperparameters are as follows.
METABRIC:
penalizer=0, l1_ratio=0.01.
SUPPORT:
penalizer=0.01, l1_ratio=0.5.

A.2. Random Survival Forest

The hyperparameter spaces are:

- n_trees: $\{100, 500, 1000, 1200, 1500, 1700, 2000\}$
- min_samples_split: $\{3, 5, 10, 15, 20\}$
- min_samples_leaf: $\{3, 5, 15, 20, 25\}$
- max_features: sqrt.

The optimal hyperparameters are as follows.
METABRIC:
n_trees=100, min_samples_split=3, min_samples_leaf=3.
SUPPORT:
n_trees=1500, min_samples_split=20, min_samples_leaf=3.

A.3. DeepSurv

The hyperparameter spaces are:

- num_layers: $\{1, 2, 4\}$
- node_size: $\{64, 128, 256, 512\}$

- dropout: $\{0, 0.1, 0.3, 0.6\}$
- batch_size: $\{64, 128, 256\}$.

The optimal hyperparameters are as follows.
METABRIC:
num_layers=4, node_size=256, dropout=0.1, batch_size=256.
SUPPORT:
num_layers=2, node_size=128, dropout=0.1, batch_size=256.

A.4. DeepHit

The hyperparameter spaces are:

- num_layers: $\{1, 2, 4\}$
- node_size: $\{64, 128, 256, 512\}$
- dropout: $\{0, 0.1, 0.3, 0.6\}$
- batch_size: $\{64, 128, 256\}$
- alpha: $\{0, 0.001, 0.1, 0.2, 0.5, 0.8, 0.9, 0.99, 0.999, 1\}$
- sigma: $\{0.01, 0.1, 0.25, 0.5, 1, 10, 100\}$
- time horizon: 1.2 × longest duration.

The optimal hyperparameters are as follows.
METABRIC:
num_layers=2, node_size=128, dropout=0.1, batch_size=128, alpha=0.001, sigma=1.
SUPPORT:
num_layers=2, node_size=256, dropout=0.3, batch_size=128, alpha=0.1, sigma=100.

Appendix B. Statistical Significance

The p-values are shown in Table 4.

Table 4: P-values of the paired t-tests for the C-index (left) and MAE (right) results.

Model	METABRIC	SUPPORT
Cox	0.636	0.003
RSF	0.809	0.004
DeepSurv	0.585	0.207
DeepHit	0.622	0.205

Model	METABRIC	SUPPORT
Cox	0.001	0.000
RSF	0.005	0.000
DeepSurv	0.005	0.000
DeepHit	0.006	0.000

Table 5: Patient-specific C-index test results (mean ± SD), and ⋆ means statistically significant.

Model	METABRIC	SUPPORT
Cox	0.634 ± 0.023	0.572 ± 0.008⋆
RSF	0.636 ± 0.013	0.616 ± 0.012⋆
DeepSurv	0.645 ± 0.020	0.611 ± 0.014
DeepHit	0.636 ± 0.020	0.601 ± 0.017
Ours	0.640 ± 0.016	0.603 ± 0.007

Appendix C. Numeric C-index Test Results

The results are shown in Table 5.

Appendix D. Mean Survival Curves Using all Patients

Figure 4 compares the mean survival curves of different models with the Kaplan-Meier curve on the same random test set, where all patients are used. Our mean survival curve does not match well with the Kaplan-Meier in this case, which indicates that our survival probability predictions on censored patients can be improved.

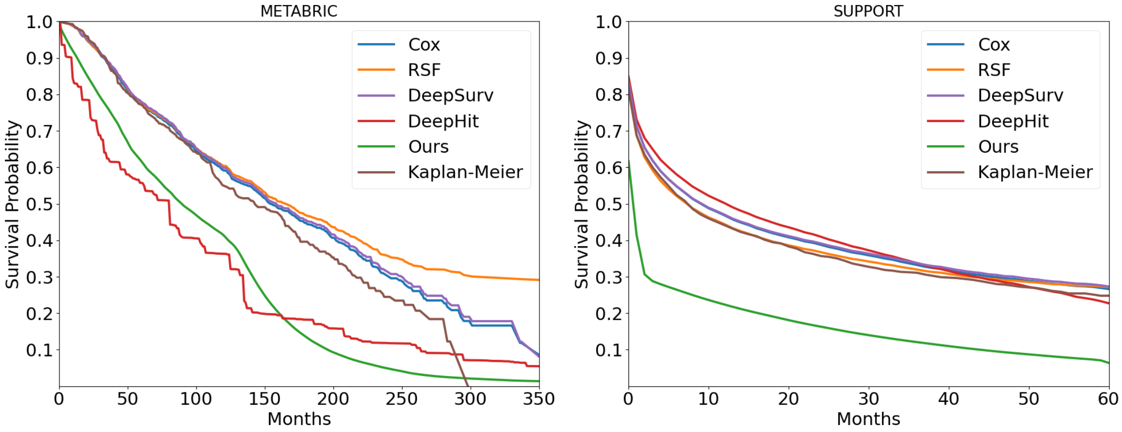

Figure 4: Comparisons of different mean survival curves with the Kaplan-Meier curve (for all patients in test).

Appendix E. An Ablation Study

As per a reviewer's suggestion, we perform this ablation study to determine the source of performance gain. We swap the Transformer model with the 'optimal' fully connected model shown in Appendix A.4, and use the same training hyperparameters in Table 2, and the same early stopping criterion to select the best models. The results are shown in Table 6. We note that if we perform an exhaustive parameter search, the results can be better or worse than these, as they depend on how well the validation accuracy correlates with the test accuracy.

Table 6: Comparisons of the Transformer and fully connected models using the same training objective that we propose.

	METABRIC		SUPPORT	
Model	C-Index ↑	MAE ↓	C-Index ↑	MAE ↓
Transformer	0.640 ± 0.016	54.33 ± 5.91	0.603 ± 0.007	8.52 ± 0.73
Fully Connected	0.631 ± 0.016	55.24 ± 2.52	0.583 ± 0.013	7.63 ± 0.67