Reproducibility of SurvTRACE: Transformers for Survival Analysis with Competing Events

Martin Becerra and Brian Reinbold
{carlosb3, brianjr3}@illinois.edu

Group ID: 56
Paper ID: 313, Difficulty: Hard
Presentation link: https://youtu.be/ql0gty92EuA
Code link: https://gitlab.engr.illinois.edu/carlosb3/survtrace

1 Introduction

In medicine, survival analysis studies the time duration to events of interest such as mortality. One major challenge is how to deal with multiple competing events (e.g., multiple disease diagnoses). Survival analysis using TRAansformers with Competing Events (SurvTRACE) proposes the use of a transformer based model capable of handling competing events (Wang and Sun, 2021).

The approach taken by the authors is to do survival analysis using transformers with competing events using the following techniques:

- Uses an inverse propensity score (IPS) to deal with selection biases, which is present in competing events.
- Automatic feature engineering using transformers with attentive encoders to learn high-order interactions between covariates.
- Incorporates multi-task (MTL) learning for downstream prediction tasks, which makes for a more robust and accurate model.

The paper evaluates different datasets such as the Molecular Taxonomy of Breast Cancer International Consortium (METABRIC), Study to Understand Prognoses Preferences Outcomes and Risks of Treatment (SUPPORT) and SEER made with data of 470k patients to validate that SurvTRACE is superior.

2 Scope of Reproducibility

This paper introduces the use of transformers with competing events for doing survival analysis that outperforms previous experiments.

Inverse propensity score and multi-task shared representation learning on SurvTRACE will have higher performance per the time-dependent concordance index than CPH, DeepSurv, DeepHit, Random Survival Forest, Piecewise Constant Hazard Deep Survival Machines on datasets METABRIC, SUPPORT, and SEER.

2.1 Addressed claims from the original paper:

- SurvTRACE outperforms in single events.
- SurvTRACE outperforms in competing events.
- MTL in SurvTRACE improves performance.
- IPS in SurvTRACE improves performance.

3 Methodology

We reproduced the same experiments as the authors explained in their paper. We used the authors' code for the SurvTRACE model, and then we implemented the other models mentioned in the paper using the libraries pycox[1], auton_survival,[2] and scikit-survival[3]. The official SEER dataset was used for competing events, and we got METABRIC and SUPPORT from the pycox library to evaluate on single events.

3.1 Model descriptions

Cox proportional hazards (CPH) is a semi-parametric method that models an individual's survival function given their features x (Cox, 1972). The hazard function is assumed to have the form $\lambda(t|x) = \lambda_0(t) \cdot e^{x^{\beta)}}$, where $\lambda_0(t)$ is the baseline hazard function. CPH is estimated by finding the β that maximizes the Cox partial likelihood function $L_c(\beta) = \prod_{i:E_i=1} \frac{e^{x_i\beta}}{\sum_{j\in\Re(T_{ij})} e^{x_i\beta}}$, where T_i is the event time, E_i is the event indicator, and $\Re(t)$ is the set of patients still at risk of failure at time t. The product is defined over the set of patients with an observable event.

[1] https://github.com/havakv/pycox

[2] https://autonlab.github.io/auton-survival/

[3] https://scikit-survival.readthedocs.io/en/stable/index.html

DeepSurv is a non-linear extension of CPH (J. L. Katzman and Kluger, 2018). It takes the patient's features as input and propagates it through a deep, feed-forward neural network consisting of a series of fully-connected layers and a nonlinear SELU activation function followed by a dropout layer. The output is an estimate of the patient's log-risk function from the CPH model. The objective function is the average negative log of the Cox partial likelihood with regularization.

DeepHit uses a deep neural network to learn the distribution of survival times directly (Lee et al., 2018). Importantly, DeepHit can handle competing events. A shared representation of the input passes through K independent, cause-specific layers, and finally the output of each cause-specific network is concatenated and passed through a single softmax layer to learn the joint distribution $\mathbf{y} = \hat{P}(t, k|x)$, which is the probability of event k happening at time t. The loss function is the sum of the log-likelihood of the joint distribution of the first occurrence of an event modified to take account of the right-censoring of the data, and the second loss incorporates estimated cumulative incidence functions at different times. The first loss function helps the model to generalize the joint distribution $\mathbf{y}$ while the second loss function finetunes the network to each cause-specific estimate.

Random survival forests (RSF) is an ensemble tree method for analysis of right-censored data (H. Ishwaran, 2008). The algorithm first draws B bootstrap samples of the data. Then it creates a tree for each bootstrap sample, and then the node is split based on which candidate variables maximize survival difference. The tree is grown until the terminal node has no more than $d_0 > 0$ deaths. Finally, a cumulative hazard function is calculated for each three and then averaged across all trees.

Piecewise Constant Hazard (PC-Hazard) is a continuous-time method that assumes the hazard function is constant in predefined intervals (Kvamme and Borgan, 2021). It minimizes the negative log-likelihood.

Cause Specific Cox Proportional Hazards (CS-CPH) and Cause Specific Piecewise Constant Hazard (CS-PC-Hazard) train each competing event separately and treats the other event as censored, i.e. that it did not occur. This assumes that the competing events are independent.

Deep Survival Machines (DSM) estimate the conditional survival function as a mixture of individual parametric survival distributions, like Weibul and log-normal distributions (C. Nagpal and Dubrawski, 2021). A softmax layer creates a weighted average of the distributions. Competing risks are handled by passing the share representation from the fully-connected layer to independent primitive distribution layers to estimate conditional probability of each event separately. The loss function is the sum of a maximum likelihood estimator on the uncensored loss, a maximum likelihood estimator on the censored loss, and a term to govern the strength of priors on the primitive distributions.

Survival analysis using TRAansformers with Competing Events (SurvTRACE) proposes a transformer based model that does not make assumptions for the underlying survival distribution and is capable of handling competing events. The implementation includes a baseline covariates embedding module, a deep-stacked attentive encoder module where covariate embeddings interact to form high-order combinatorial embeddings, and the alignment and subnetwork prediction module to generate the predicted hazard function for each discrete, duration index. The loss function is the inverse propensity score-based piecewise constant hazard: $\ell = \frac{1}{nK_E} \sum_{i,k} \frac{\mathbb{1}_{ik} l_{ik}}{\pi_{ik}}$. When multi-task learning is incorporated, subnetworks generate predictions for whether any event happened for a patient and the event/censoring time. The loss functions then incorporates binary cross entropy and mean squared error respectively.

3.2 Data descriptions

The authors utilized three datasets to evaluate SurvTRACE. The METABRIC and the SUPPORT dataset are used for single events, and the SEER dataset is used for competing events. For all datasets, we re-used the authors' function to load and process the datasets. Part of the processing includes dividing the data at different quantiles of event times and standardizing continuous covariates. 60% of the data is used for training, and 10% for validation, and 30% for testing.

METABRIC The Molecular Taxonomy of Breast Cancer International Consortium (METABRIC) uses gene and protein expressions to determine new breast cancer subgroups that can help physicians provide better treatment recommendations (C. Curtis, 2012). The dataset has nine covariates: hormone treatment indicator, radiotherapy indicator, chemotherapy indicator, ER-positive indicator, age at diagnosis, and four gene indicators (MKI67, EGFR, PGR, and ERBB2). There are 1,904 patients, and 57.9% have an event. The median event duration is 85.9 months, and the median censoring time is 158 months. The data comes from the pycox library.

SUPPORT The Study to Understand Prognoses Preferences Outcomes and Risks of Treatment (SUPPORT) tracks the survival time of gravely ill hospitalized adults (Knaus et al., 1995). The dataset has fourteen covariates: 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. There are 9,105 patients, and 68% have an event. The median event duration is 57 days, and the median censoring time is 918 days. The data also comes from the pycox library.

SEER The Surveillance, Epidemiology, and End Results (SEER) Program provides information on cancer statistics in an effort to reduce the cancer burden among the U.S. population. Data is retrieved from the SEER database, and we follow the authors' methodology to select patients for the dataset. They choose breast cancer patients between 2004 and 2014, with a follow-up period no more than ten years. Furthermore, they subset the data to include patients who also have heart disease. Therefore breast cancer and heart disease are competing events. In total, there are 476,746 patients in the dataset. 77.1% of events are censored. Of the events that occurred, 80% are breast cancer, so heart disease is a rare event. There are eighteen covariates including age, race, sex, diagnostic confirmation, morphology information sucha as primary site, laterality, and histologic type; tumor information like size, type, and number; and surgery information. Missing values were filled with the mean of continuous covariates and with the mode for categorical covariates.

3.3 Hyperparameters

We used the same batch size as the SurvTRACE paper for all experiments. The batch size is 64 for METABRIC, 128 for SUPPORT, and 1,024 for SEER. All models except CPH, RSF, and DSM have a dropout layer with a dropout rate of 0.1. Unless otherwise noted, all experiments used 100 epochs with an early stopping criteria that stopped training if the best validation loss so far had not improved in 20 epochs except for SurvTRACE on SEER, which was stoped after 8. We tuned the hyperparameters for the baselines using a grid search based on criteria specified in their respective papers, and we selected the best parametrization by maximizing the time dependent concordance index averaged over the three time-horizon quantiles for all events. Table 1 has some common hyperparameters used in many of the models for each dataset as well as the number of trainable parameters. Hidden layer node size is set to the same value for all hidden layers, and the number of hidden layers in SurvTRACE refers to the number of transformers.

We do not set any specific hyperparameters for CPH, and we run the default model from scikit-survival. We apply the same hyperparameters for all datasets in RSF. We choose to create 200 trees in the forests and set the max depth of each tree to four per (C. Nagpal and Dubrawski, 2021). The rest of the parameters are the default settings in scitkit-survival.

We use the same hyperparameters found in the authors' repo for SurvTRACE as we assumed they represented the optimal parameterization that replicates their results.For MTL, we add the parameters γ_1 and γ_2 to scale the contribution of the mortality and the length-of-stay to the loss function. γ_1 was set to 1 for all datasets, and γ_2 was 0.3 for both METABRIC and SUPPORT and 0.8 for SEER. We do not vary the hyperparameters based on SurvTRACE variants, but only based on the dataset. We set the early stop patience to 8 epochs for SEER.

3.4 Implementation

The authors' provide a repo to implement SurvTRACE[4]. They also provide three notebooks to replicate their results for each dataset. However, they do not have MTL implemented. Also, they do not provide implementations for the baselines. We use pytorch and pycox to implement DeepHit, DeepSurv, and PC-Hazard. Pycox provides classes to incorporate the specific loss and survival functions used in each of these models, however, the user has to provide the neural network structure. We program each neural network in pytorch according to the structure outlined by the authors in their respective paper. We used the library scikit-survival to implement cox proportional hazards and random survival forests, and we used the li-

[4]https://github.com/RyanWangZf/SurvTRACE

	METABRIC				SUPPORT				SEER			
	# Hidden Layers	Hidden Layer Node Size	Learning Rate	# Learnable Parameters	# Hidden Layers	Hidden Layer Node Size	Learning Rate	# Learnable Parameters	# Hidden Layers	Hidden Layer Node Size	Learning Rate	# Learnable Parameters
DeepSurv	2	32	0.01	1,537	2	32	0.01	1,697	2	32	0.01	1,825
PC-Hazard	2	64	0.01	5,316	2	32	0.01	1,796	4	32	0.001	4,164
DeepHit	1	16	0.01	980	2	16	0.01	1,748	2	32	0.001	8,744
DSM	2	100	0.001	12,116	2	100	0.001	12,616	2	100	0.0001	14,232
SurvTRACE	3	64	0.001	38,694	3	64	0.001	54,374	2	64	0.0001	66,138

Table 1: Models' hyperparameters and total number of trainable parameters.

brary auton_survival to implement Deep Survival Machines.

To implement multi-task learning into SurvTRACE, we used the fact that the encoder learns a shared representation of the features that is used for all downstream tasks, and each downstream task had a similar structure except the output layer was subject to a different loss function. We modified the neural network class "BertCLS" to vary the number of outputs and then derived two classes from it - "BertCLSEvent" and "BertCLSTime."

"BertCLSEvent" learns to predict mortality. Note that in competing events, it does not need to distinguish amongst events, but simply learn whether *any* event occurred. The forward function uses the sigmoid function to output the probability of mortality, and the loss function uses binary cross entropy.

"BertCLSTime" learns to predict the time at which an event or censoring occurred. The model is not trained using the event time but instead the index of the quantile of event times (i.e. 0 is 25% quantile, 1 is 50% quantile, ect.). Therefore, this network also predicts this index. The forward function incorporates a sigmoid function and the probability is scaled by the number of quantiles and rounded to the nearest integer. The loss function is the mean squared error.

3.5 Computational requirements

We ran all code locally using a NVIDIA GeForce RTX 2060 GPU, which has 1,920 CUDA cores, 14,254 MB of total available memory, and a memory data rate of 11 GB/s. Table 2 shows the average total training time in seconds, total number of epochs trained, and seconds per epoch with the standard deviation of ten runs in parenthesis. CPH and DSM did not have a way to retrieve the number of iterations performed during optimization. Furthermore, the training process for DSM is split into two steps: a pre-training of up to 10,000 iterations followed by training for 100 epochs. Reporting number of epochs in this context may not completely make sense in this context so it is omitted. Epochs in RSF refers to the number of trees in the forest.

4 Results

Each model is evaluated by calculating the time-dependent concordance index, which is a measure of rank correlation between predicted risk scores. We successfully replicated the majority of their results since 84% of our results are within three percent of the authors' results, and 54% of our results are within one percent. In fact, 37% of our results are above the authors' results. Also, our results confirm many of the authors' key conclusions. Table 3 presents our results for all experiments.

4.1 Result 1: SurvTRACE outperforms baselines in single events.

Most of our results on METABRIC and SUPPORT are within one percent of the authors' results, and only two results are lower than 3% of the authors'. Our results for RSF on METABRIC outperformed the authors' for all three time horizons. Our results for SurvTRACE and SurvTRACE w/o MTL on SUPPORT were in line with the authors' and outperformed all baselines, however, SurvTRACE performance on METABRIC was a bit low. The best performing model was at the 25% quantile on METABRIC was DSM, and a variant of SurvTRACE was the best performing model at the 50% and 75% quantile.

All in all we find some evidence that SurvTRACE improves performance in single events since a SurvTRACE variant outperforms a baseline model for all but one time horizon in these two datasets. However, the difference in means between the best performing SurvTRACE variant and the best baseline model is only statistically significant at the 25% and 50% time horizon on the SUPPORT dataset unlike the authors' whose results for SurvTRACE are statistically significant for both datasets at all time horizons. Therefore, it would be too strong of a claim to say that SurvTRACE outperforms the baselines in single events, but it is very competitive.

	Total Training Time (Seconds)	Total Epochs Trained	Time per Epoch (Seconds)	Total Training Time (Seconds)	Total Epochs Trained	Time per Epoch (Seconds)
		METABRIC			**SUPPORT**	
CPH	0.05(0.011)	N/A	N/A	0.252(0.05)	N/A	N/A
DSM	7.494(5.899)	N/A	N/A	6.583(2.471)	N/A	N/A
DeepHit	6.612(2.559)	45.0(16.014)	0.147(0.027)	21.23(2.965)	46.9(8.399)	0.457(0.038)
DeepSurv	3.148(0.946)	35.8(10.799)	0.089(0.011)	9.161(2.388)	44.6(12.465)	0.207(0.012)
PC-Hazard	7.037(2.009)	51.5(12.651)	0.137(0.026)	12.822(2.521)	45.0(7.916)	0.285(0.018)
RSF	53.685(0.809)	200.0(0.0)	0.268(0.004)	50.213(2.099)	200.0(0.0)	0.251(0.01)
SurvTRACE	44.304(15.772)	36.0(12.41)	1.225(0.074)	100.088(30.617)	41.0(12.41)	2.438(0.129)
SurvTRACE w/o IPS*	42.146(14.527)	37.4(13.327)	1.132(0.053)	84.227(22.713)	35.8(9.566)	2.351(0.044)
SurvTRACE w/o IPS & MTL*	25.134(9.364)	32.6(12.677)	0.777(0.09)	51.752(10.682)	34.0(5.676)	1.519(0.171)
SurvTRACE w/o MTL	42.85(14.242)	39.3(12.824)	1.092(0.07)	63.335(16.281)	32.4(7.975)	1.951(0.069)
		SEER - Breast Cancer			**SEER - Heart Disease**	
CS-CPH	19.397(5.323)	N/A	N/A	16.439(3.489)	N/A	N/A
CS-DeepSurv*	64.198(21.099)	46.1(15.3)	1.394(0.021)	75.272(33.326)	52.8(23.442)	1.426(0.039)
CS-PC-Hazard	154.69(5.438)	50.9(0.316)	3.039(0.102)	147.343(5.806)	50.4(0.966)	2.923(0.102)
CS-RSF*	408.076(15.592)	200.0(0.0)	2.04(0.078)	450.24(19.285)	200.0(0.0)	2.251(0.096)
		SEER				
DSM	433.353(86.666)	N/A	N/A			
DeepHit	377.643(121.064)	64.2(19.64)	5.865(0.483)			
SurvTRACE	1101.338(251.979)	48.6(10.946)	22.644(0.403)			
SurvTRACE w/o IPS	1067.762(283.505)	46.9(11.958)	22.719(0.679)			
SurvTRACE w/o IPS & MTL*	758.111(190.388)	41.1(9.96)	18.415(0.382)			
SurvTRACE w/o MTL	750.155(144.31)	38.7(7.439)	19.379(0.38)			

Table 2: Average total training time in seconds, total number of epochs trained, and seconds per epoch with the standard deviation of ten runs in parenthesis. * indicates an ablation. Epochs in RSF refers to the number of trees in the forest.

	25%	50%	75%	25%	50%	75%
		METABRIC			**SUPPORT**	
CPH	0.641(0.039)	0.629(0.014)	0.63(0.012)	0.544(0.013)	0.554(0.009)	0.565(0.005)
DSM	**0.723(0.026)**	0.664(0.013)	0.634(0.013)	0.644(0.007)	0.61(0.005)	0.589(0.007)
DeepHit	0.699(0.032)	0.655(0.012)	0.602(0.021)	0.648(0.015)	0.609(0.009)	0.581(0.011)
DeepSurv	0.655(0.033)	0.646(0.019)	0.639(0.015)	0.589(0.038)	0.595(0.015)	0.608(0.01)
PC-Hazard	0.709(0.025)	0.676(0.008)	0.646(0.014)	0.645(0.011)	0.617(0.008)	0.607(0.009)
RSF	0.712(0.023)	0.67(0.014)	0.639(0.015)	0.650(0.014)	0.619(0.009)	0.606(0.007)
SurvTRACE	0.705(0.033)	0.673(0.012)	0.642(0.012)	0.664(0.014)	0.632(0.007)	0.619(0.007)
SurvTRACE w/o IPS*	0.702(0.025)	0.675(0.012)	0.646(0.013)	**0.665(0.013)**	0.631(0.005)	**0.621(0.005)**
SurvTRACE w/o IPS & MTL*	0.707(0.023)	0.672(0.01)	0.643(0.011)	0.664(0.012)	**0.634(0.005)**	**0.621(0.005)**
SurvTRACE w/o MTL	0.709(0.029)	**0.678(0.012)**	**0.648(0.012)**	0.663(0.016)	0.629(0.007)	0.618(0.008)
		SEER - Breast Cancer			**SEER - Heart Disease**	
CS-CPH	0.5000(0.0000)	0.5000(0.0000)	0.5000(0.0000)	0.5000(0.0000)	0.5000(0.0000)	0.5000(0.0000)
CS-DeepSurv*	0.8744(0.0078)	0.8531(0.0091)	0.8374(0.0095)	0.7366(0.0154)	0.7313(0.0131)	0.7221(0.0127)
CS-PC-Hazard	0.8878(0.0012)	0.8657(0.0013)	0.8476(0.0014)	0.7703(0.0042)	0.7604(0.0049)	0.7485(0.0034)
CS-RSF*	0.8896(0.0012)	0.8666(0.0012)	0.8487(0.0012)	0.777(0.005)	0.7664(0.0045)	0.7557(0.0026)
DSM	0.8303(0.0047)	0.802(0.0068)	0.7874(0.0066)	0.7185(0.0069)	0.7052(0.0084)	0.6957(0.0073)
DeepHit	0.8888(0.0036)	0.8665(0.0049)	0.8473(0.0053)	0.7453(0.0057)	0.7325(0.0060)	0.7195(0.0067)
SurvTRACE	**0.9040(0.0013)**	**0.8827(0.0009)**	**0.8651(0.0007)**	**0.8002(0.0059)**	**0.7927(0.0041)**	**0.7806(0.0033)**
SurvTRACE w/o IPS	0.9036(0.0013)	0.8822(0.0009)	0.8646(0.0005)	0.7982(0.0062)	0.7907(0.0039)	0.7787(0.0030)
SurvTRACE w/o IPS & MTL*	0.9035(0.0012)	0.8821(0.001)	0.8645(0.0006)	0.7983(0.0061)	0.7908(0.0045)	0.7791(0.0035)
SurvTRACE w/o MTL	0.9039(0.0015)	0.8824(0.0011)	0.8648(0.0009)	0.7996(0.0063)	0.7925(0.0041)	0.7805(0.0031)

Table 3: Time-dependent concordance index at different quantiles of event times with the standard deviation of ten runs in parenthesis. * indicates an ablation.

4.2 Result 2: SurvTRACE outperforms baselines in competing events.

Our experiment for CS-CPH was a complete failure. We tried varying a regularization parameter, and tried another implementation from the library lifelines[5] but acquired the same results.The results for DSM were also low relative to the authors'. Despite our extensive hyperparameter search, it seems we could not find a parametrization to match the authors' results.

On the other hand, the results for CS-PC-Hazard, DeepHit, SurvTRACE, and SurvTRACE w/o MTL are all within 3% of the authors'. SurvTRACE results on breast cancer events are nearly identical to the authors', and they are slightly above the authors' results at all quantiles for heart disease events.

Every variant of SurvTRACE outperforms the baselines on competing events in SEER, and the difference in means between SurvTRACE and the best baseline model is statistically significant in all cases. Our results strongly support the central claim of the authors' paper that SurvTRACE performs well with competing events, especially when one of these events is rare.

4.3 Result 3: Multi-task learning in SurvTRACE improves performance.

On single events, SurvTRACE variants without MTL were higher than SurvTRACE, however, this

[5]https://lifelines.readthedocs.io/en/latest/

difference is not statistically significant. In competing events, the results for SurvTRACE are slightly above SurvTRACE w/o MTL, but these differences are not statistically significant. These results suggest that MTL may be helpful on competing events, but not single events, but our evidence is weak.

4.4 Result 4: IPS in SurvTRACE improves performance.

The results for SurvTRACE are slightly above SurvTRACE w/o IPS, but these differences are not statistically significant, however, they are stronger than the differences between SurvTRACE and SurvTRACE w/o MTL. These results suggest that IPS is helpful in competing events, but our evidence is a bit weak.

4.5 Ablations

We did seven ablations: SurvTRACE w/o IPS and SurvTRACE w/o IPS & MTL on both Metabric and SUPPORT; and cause-specific DeepSurv, cause-specific random survival forests and SurvTRACE w/o IPS & MTL on SEER. SurvTRACE w/o IPS and SurvTRACE w/o IPS & MTL on Metabric and SUPPORT does not seem to make a statistically significant difference between the results of other SurvTRACE variants. This makes sense since with single events, there is no need to de-bias events, so IPS should not make a difference. SurvTRACE w/o IPS & MTL results on SEER are less than both SurvTRACE and SurvTRACE w/o MTL, which further suggests the importance of IPS and the need to de-bias competing events. Finally, the performance of CS-DeepSurv is inline with the other baseline, and CS-RSF is the best performing baseline on SEER. Still, SurvTRACE and its variants outperform them, providing further evidence that SurvTRACE can handle competing events well.

5 Discussion

Overall, the paper was reproducible. Our results support the most important claim in the paper that SurvTRACE is very effective at handling competing events in survival analysis. It can also handle single events well too. We show some evidence that IPS and MTL in SurvTRACE are beneficial, but it is weaker than the authors'. Interestingly, our results of SurvTRACE w/o MTL, which is the authors' original model unedited, basically matched the authors' results of SurvTRACE (which includes MTL). Our results could suggest that the transformer architecture alone does an extraordinary job learning the data and MTL is unnecessary. Still, we did have to implement MTL on our own, and it is possible that we made a mistake.

5.1 What was easy

Processing the data was the easiest part of the replication. The authors' provided detailed instructions on how to obtain the SEER dataset, a file to query the data from the SEER database, and a script to process the data. Also, running SurvTRACE without MTL was simple. The authors' provided notebooks to evaluate the model on each dataset.

5.2 What was difficult

The most time consuming part of the replication is that we had to implement all of the baseline models. Fortunately, we found several libraries with high-level APIs to implement them. Generally, they were not difficult to use and usually had sufficient documentation, however, they can require different data structures and data types, which can take a bit of time to deduce. However, finding optimal hyperparameters for each baseline model was time consuming

5.3 Recommendations for Reproducibility

Researchers, in general, should publish the optimal hyperparameters they used. Some papers described their procedure to find optimal hyperparameters, but hyperparameter tuning can be a slow, expensive process (especially when doing a grid search). In addition, researchers should provide specific details on how they implemented their baseline models. It should be made clear what deviations they may have made to the original implementation of that model. Researchers should also publish their data processing scripts as the authors in this paper did. Data processing is often overlooked since it is not the main message of the research, but is one of the most important aspects of reproducible research.

6 Communication with Original Authors

We asked several questions to the authors about their implementation of SurvTRACE, but unfortunately we did not receive a response. We asked about the optimal parameters used in SurvTRACE, whether they calculated the quantiles of event time separately for cause-specific models, and if the length-of-stay task learned the discrete index of duration or the continuous raw durations.

References

S. F. Chin G. Turashvili O. M. Rueda M. J. Dunning D. Speed A. G. Lynch S. Samarajiwa Y. Yuan C. Curtis, S. P. Shah. 2012. The genomic and transcriptomic architecture of 2,000 breast tumors reveals novel subgroups. *Nature*, 486(7403):346–352.

X. R. Li C. Nagpal and A. Dubrawski. 2021. Deep survival machines: Fully parametric survival regression and representation learning for censored data with competing risks. *IEEE Journal of Biomedical and Health Informatics*, 25(8):3163–3175.

D. R. Cox. 1972. Regression models and life-tables. *Journal of the Royal Statistical Society: Series B (Methodological)*, 34(2):187–202.

E. H. Blackstone M. S. Lauer H. Ishwaran, U. B. Kogalur. 2008. Random survival forests. *The Annals of Applied Statistics*, 2(3):841–860.

A. Cloninger J. Bates T. Jiang J. L. Katzman, U. Shaham and Y. Kluger. 2018. Deepsurv: Personalized treatment recommender system using a cox proportional hazards deep neural network. *BMC Medical Research Methodology*, 18(1):1–12.

W. A. Knaus, F. E. Harrell, J. Lynn, L. Goldman, R. S. Phillips, A. F. Connors, N. V. Dawson, W. J. Fulkerson, R. M. Califf, N. Desbiens, P. Layde, R. K. Oye, P. E. Bellamy, R. B. Hakim, and D. P. Wagner. 1995. The SUPPORT prognostic model: Objective estimates of survival for seriously ill hospitalized adults. *Ann Int Med*, 122:191–203.

H.; Kvamme and Ø Borgan. 2021. Continuous and discrete-time survival prediction with neural networks. *Lifetime Data Anal*, 27:710–736.

Changhee Lee, William Zame, Jinsung Yoon, and Mihaela van der Schaar. 2018. Deephit: A deep learning approach to survival analysis with competing risks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).

Zifeng Wang and Jimeng Sun. 2021. SurvTRACE: Transformers for survival analysis with competing events.