SURVIVAL NEURAL NETWORK WITH SELF-ATTENTION FOR INTERPRETATION OF BIOMARKERS INTERACTIONS

A PREPRINT

Margarita Zapevalina
Life Sciences

Vlad Fedotov
Life Sciences

Pavel Muravski
Petroleum Engineering

Artem Erkhov
Petroleum Engineering

Batyr Khabibullin
Petroleum Engineering

November 2, 2024

ABSTRACT

Survival analysis is widely used in clinical research to compare treatment therapies, detect biomarkers, which influence survival, disease progression and response to the medicine. It is particularly important to account for genomics features as well as to identify and interpret their interactions. In our research we created deep learning survival model with multi-head attention layer, which shows genomics biomarkers interactions and has performance comparable with the most used deep learning survival model DeepSurv.

Keywords Survival analysis · Deep Learning · Self-attention · Features interaction interpretability

1 Introduction

Survival analysis is a set of methods for data analysis where the outcome variable of interest is time until an event occurs (disease progression or death). This analysis is widely used in clinical research to estimate the probabilities of patients' survival under different treatments accounting for various human health records features and identifying the most influential biomarkers.

In survival analysis we aim to predict not the overall survival, but survival probabilities for each patient for each time point. However, there is a very specific challenge - data censoring, which means that not all the patients experienced the event of interest (many of them were withdrawn from experiment at some time point or have not faced the event till the end of the experiment). Therefore, in survival models the target is 2-dimensional: firstly, it contains information if the patient experienced death or was censored and, secondly, time until this event. Thus, to account for it, survival deep learning models use a special loss function, which is derived from likelihood in the traditional statistical survival model named CoxPH. Wang et al. [2019]

It is particularly important to incorporate genomics data into survival analysis as it may shed light on hub genes and key pathways in human survival and disease emergence. However, genomics data is very sparse with lots of nonlinear relationships between biomarkers. Moreover, the huge challenge in survival analysis with genomics is data dimensionality: due to the high cost of the analysis there are usually not many patients, but lots of genomics biomarkers.

Nowadays, the most wide-spread model in deep learning survival analysis is DeepSurv from pycox package. Recently, to increase the accuracy of analysis transformer-based models were proposed Wang and Sun [2021], Arango et al. [2021], Hu et al. [2021], Wang et al. [2024]. In our study we created deep learning survival model with multi-head attention layer based on DeepSurv model from pycox package. The usage of self-attention might increase accuracy and, most importantly, show interactions of genomics features, making model interpretable, which is highly valuable for further biological research.

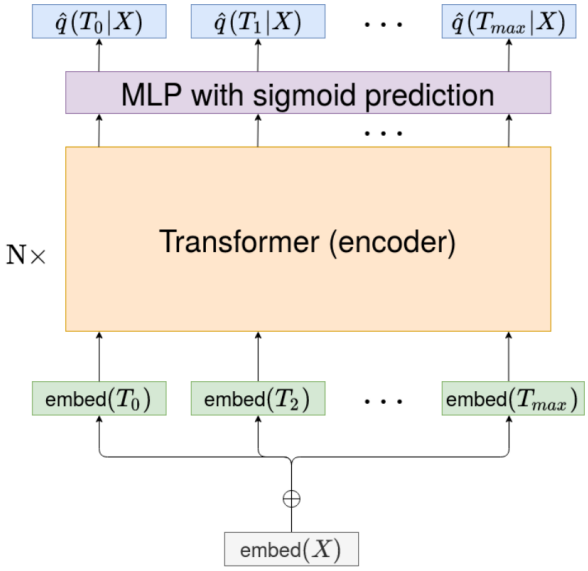

Figure 1: Architecture of the model.

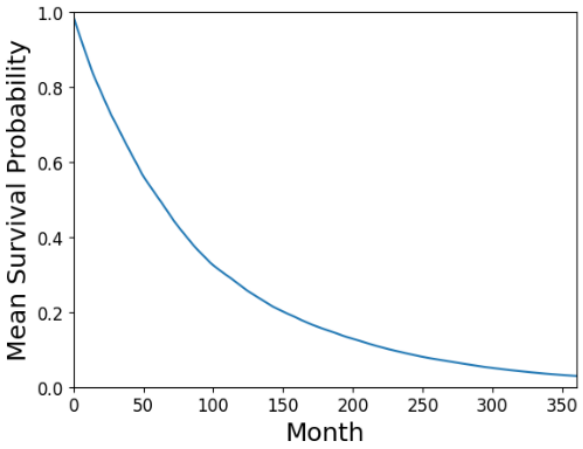

Figure 2: Predicted survival probability plot.

2 Data and Methods

2.1 Dataset

We used publicly available Breast cancer dataset Metabric with RNA levels and mutation from Cbioportal [1].

Dataset included 1900 patients, dozens of clinical features and nearly 500 genomics features including m-RNA levels scores for 331 genes, and mutation for 175 genes. In our research we studied the influence of clinical biomarkers and m-RNA level scores on survival. We substantially filtered features for subsequent analysis with deep learning model containing attention layer. Firstly, we removed highly correlated biomarkers and then filtered features, leaving only those which were shown to be important by survival xgboost and survival random forest analysis.

2.2 Model from the article "Transformer-Based Deep Survival Analysis"

We implemented model developed in the article [Hu et al., 2021]. This is fully parametric transformer based model that was first attempt to apply the transformer in survival analysis. For each patient encoder of the Transformer predicts the complement of the hazard function for all times up to T_{max}, where T_{max} is a hyperparameter. Each patient is treated as a 'sentence', and each 'word' is the sum of the feature embedding and the positional encoding of a time t, where $t = 0, 1, 2, \ldots, T_{max}$. Thus, each 'word' represents the interaction between the patient and time t. Figure 1 demonstrates the architecture of the model.

Also this model has special loss function and prediction. Figure 2 shows how mean prediction of this model looks like. The loss function consists of several parts: for the observed cases, for the right-censored case and the loss that penalizes the discordant pairs. The total loss is the combination of the above three losses.

For the observed cases we maximize the survival probabilities $\hat{S}(t|X)$ for $t \leq T$, and minimize them for $t \geq T$. For the right-censored case we maximize the survival probabilities $\hat{S}(t|X)$ 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. Also one part of the loss penalizes the randomized discordant pairs. Denote T_i and T_j as the times for patients i and j, where T_i is observed and $T_i \leq T_j$. The predicted survival durations $\hat{T}_i$ and $\hat{T}_j$ are discordant if $\hat{T}_i \geq \hat{T}_j$, and we want to reduce the number of discordant pairs. The following randomized algorithm with an $O(N)$ runtime used for this purpose: for each observed patient i in the training set, we randomly sample another patient j where $T_j \geq T_i$. 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 T_i should be close to T_i, and we use the MAE loss $|T_i - \hat{T}_i|$ to penalize their difference.

[1] https://www.cbioportal.org/study/summary?id=brca_metabric

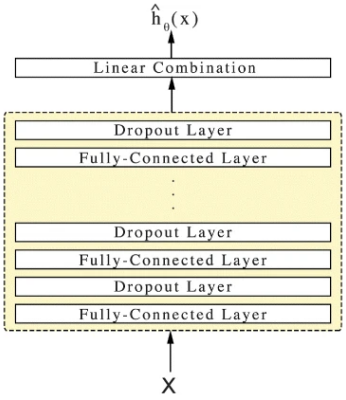

Figure 3: Architecture of the DeepSurv model.

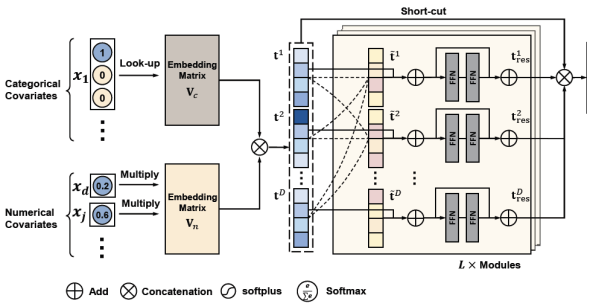

Figure 4: Encoder block of the SurvTRACE model.

2.3 DeepSurv baseline model

During implementation of previous model, we faced many issues with data preprocessing, training pipeline and metrics computation. All these issues are resolved in PyCox framework, which is a python package for survival analysis and time-to-event prediction with PyTorch. It contains implementations of various survival models, some useful evaluation metrics, and a collection of useful preprocessing tools. We used Cox proportional hazards model (CoxPH) also referred to as [Katzman, 2018], as a baseline (Figure 3). The model assumes that the hazard function is composed of two non-negative functions: a baseline hazard function, $\lambda_0(t)$, and a risk score, $r(x) = e^{h(x)}$, defined as the effect of an individual's observed covariates on the baseline hazard. We denote $h(x)$ as the log-risk function. The hazard function is assumed to have the form:

$$\lambda(t|x) = \lambda_0(t) \cdot e^{h(x)} \tag{1}$$

The loss function in Pycox CoxPH model(DeepSurv) is derived from classical Cox model and is computed with the formula below:

$$l(\theta) := -\frac{1}{N_{E=1}} \sum_{i:E_i=1} \left(\hat{h}_\theta(x_i) - \log \sum_{j \in \Re(T_i)} \exp[\hat{h}_\theta(x_j)] \right) + \lambda \cdot ||\theta||_2^2, \tag{2}$$

Where x represents patients biomarkers, θ – weights of the network and is the output of the network, $\hat{h}_\theta$ – a single node with a linear activation which estimates the log-risk function for patients.

2.4 Integration of Attention mechanism in DeepSurv

To increase accuracy of the baseline model and show interactions of genomic features, we added transformer encoder block to the beginning of the model. In this block we implemented some ideas from [Wang and Sun, 2021], such as separate numerical and categorical features encoding. Figure 4 demonstrates the architecture of the encoder block of there model.

2.4.1 Searching for hyperparameters and experimenting with layers

All experiments and parameter tuning were carried out for 50 runs, after which the average value of Brier and Concorde scores was taken.

After initial formation, the neural network contained an MLP block containing two hidden layers. After that this MLP block was replaced by ResidualMLPBlock also containing two layers, but as it turned out this replacement did not really affect the results as in both cases Concorde and Brier showed similar results so for MLP block with two layers scores were 0.663 and 0.183 and for ResidualMLPBlock 0.671 and 0.181 Concorde and Brier respectively. Therefore, we simply took the ResidualMLPBlock.

After that we experimented with the number of hidden layers or rather 1, 2, 4, 6, 8 layers were tried. After that, they were compared on the same metrics that were stated above. The results of selecting the number of layers did not give

anything, as you can see on the figures (7, 8). Therefore, it was decided to continue the rest of the experiments on two layers.

The next point of selection was the number of neurons in the hidden layer. The values varied: 20, 40, 60, 80. But this, as well as the previous attempts, was not successful. We took the number of neurons equal to 50. Graphs of metrics at variation on neurons are presented on the figures (9, 10).

The number of heads of the Multihead model was also enumerated. The heads were oversampled between 1, 2, 4, 8, 16. In this case, the best performance can be achieved at the number of heads equal to 16. As you can see in the figures (11, 12). On 16 heads, Brien achieved 0.180 and Concorde 0.680. It sometimes achieves a worse metric, but on average, it is on 16 heads that the best performance is achieved in the runs. Therefore, it was decided to use this number of heads.

All these changes have very little impact as the dataset is specific and contains only 1904 rows, some of which will still go to validation and test As a result, it retrains very quickly, so to achieve and improve speeds is quite a problematic task.

3 Results and Discussion

3.1 Comparison of models

3.1.1 Concordance index

All of the three tested models were evaluated for concordance index comparison. The following results can be found on Figure 5. X-axis shows model type and Y axis shows concordance index.

As we can see, the performance of models are nearly the same, but we can make some inference even from these data. It is clearly seen that base CoxPH model gives predictions with more uncertainty than other two models.However, we can also see that the predictions are nearly the same for all three models. So, if we would like to find out which model in the winner, we can't rely only on concordance index, as our model shows only slight improvement here.

3.1.2 Brier score

Figure 6 shows Brier score calculated for all three models.

By looking on these boxplots we can see that the worst performance in the term of Brier score is achieved by SA Transformer model (the one explored by us from the article). Such results indicate that despite that this model has a good concordance index, it can't predict probabilities of the events with high accuracy, which is crucial for a good SA model. At the same time we can see that the baseline DeepSurv and our model show very similar results, that are much lower than SA Transformer.

In the end we can say that our model totally wins SA Transformer by metrics, but comes somewhere near with base DeepSurv MLP. This is already a good result as we get the same good performance, but additionally have an extra feature with attention.

3.2 Attention maps

The main idea of this project was to find a way to make survival analysis model interpretable. This, potentially, can help us start full personalized healthcare, as we could get interpretable results for every patient that goes through analysis. To achieve this goal we implemented multihead attention mechanism along with a residual MLP architecture, so, we can train our model and get attention weights for each feature afterward. We've taken patients from our dataset and divided them into two groups: those who died in the lower quantile of durations time (0.2) and in the higher (0.8). The resulting maps are shown on figure 7 and 8.

Despite that both groups consist of patients who died, the fact that it happened in different time produces different patterns in attention. Both groups have cancer markers highlighted with high attention weight scores, however, for patients that died in the lower quantile the most prominent gene is ef23, which is shown to have a big impact on tumorogenesis, as it influences stem cancer cells growth. At the same time, we can see that the highest score for upper quantile is achieved with shbg gene, which has not so prominent role in cancerogenesis and mostly associated with sexual hoemone interactions. So, we can see that at some level the extracted attention maps can even have some meaning and logic in them, which can be a signal of choosing a right direction in studying the interpretability of survival analysis predictions.

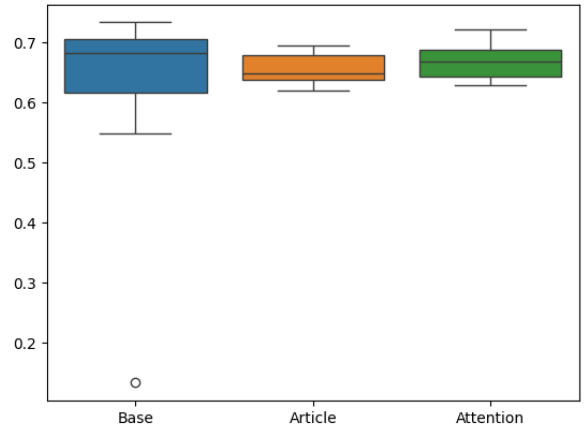

Figure 5: Concordance index for explored models.

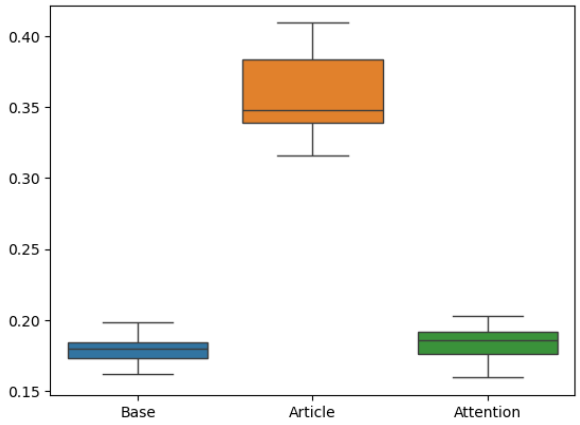

Figure 6: Brier score for explored models.

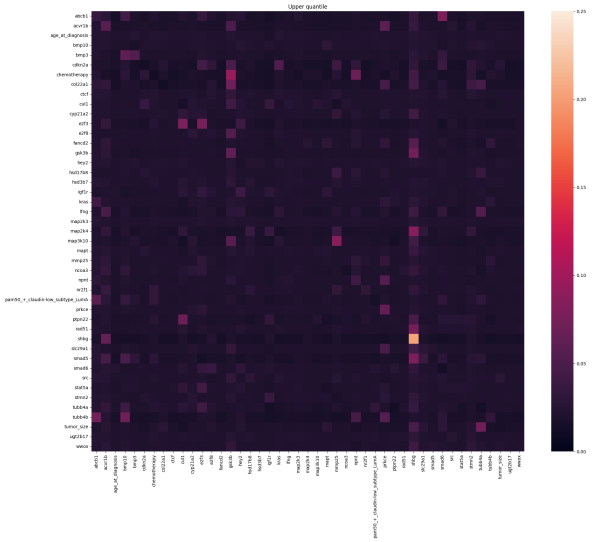

Figure 7: Attention maps for patients died in the upper quantile.

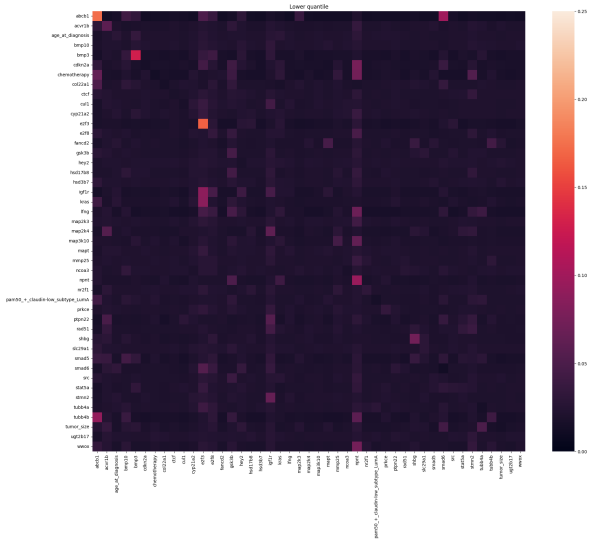

Figure 8: Attention maps for patients died in the lower quantile.

4 Code availability

All code can be accessed in Github repository [2]

5 Contribution

Margarita Zapevalina – data preprocessing, baseline deep learning survival model DeepSurv
Vlad Fedotov – attention map extraction and interpretation
Artem Erkhov and Pavel Muravskii – incorporation of multi-head self-attention in DeepSurv pycox framework
Batyr Khabibullin – implementation of transformer-based survival model from the article

References

Ping Wang, Yan Li, and Chandan K. Reddy. Machine learning for survival analysis: A survey. *ACM Comput. Surv.*, 51 (6), feb 2019. ISSN 0360-0300. doi:10.1145/3214306. URL `https://doi.org/10.1145/3214306`.

[2] https://github.com/ErkhovArtem/DL_Project

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

Gustavo Arango, Elly Kipkogei, Etai Jacob, Ioannis Kagiampakis, and Arijit Patra. Explainable transformer-based neural network forthe prediction of survival outcomes in non-smallcell lung cancer (nsclc), 10 2021.

S. Hu, E.A. Fridgeirsson, G. van Wingen, and M. Welling. Transformer-based deep survival analysis. *Proceedings of Machine Learning Research*, 146:132–148, 2021.

Yuchen Wang, Xianchun Kong, Xiao Bi, Lizhen Cui, and Hao Wu. Resdeepsurv: A survival model for deep neural networks based on residual blocks and self-attention mechanism. *Interdisciplinary sciences, computational life sciences*, 03 2024. doi:10.1007/s12539-024-00617-y.

Cloninger A. et al. Katzman, J.L. Shaham U. Deepsurv: personalized treatment recommender system using a cox proportional hazards deep neural network. *BMC Med Res Methodol*, 2018.

A Appendix

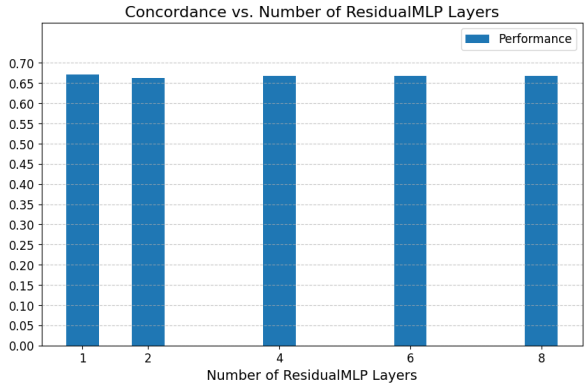

Figure 9: Concordance when iterating over the number of layers.

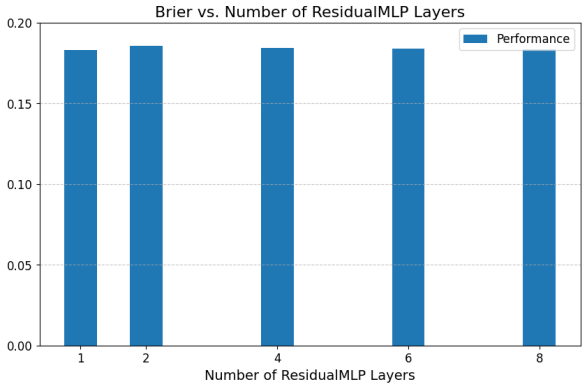

Figure 10: Brier when iterating over the number of layers.

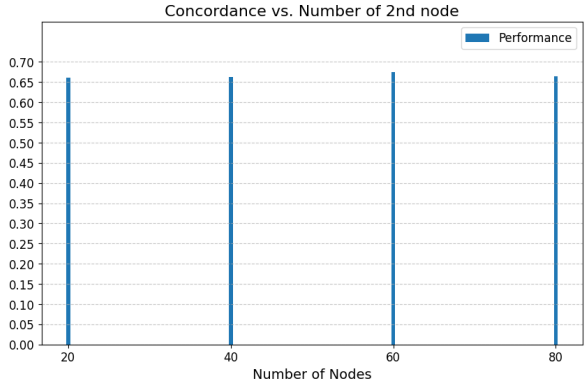

Figure 11: Concordance when iterating over the number of nodes.

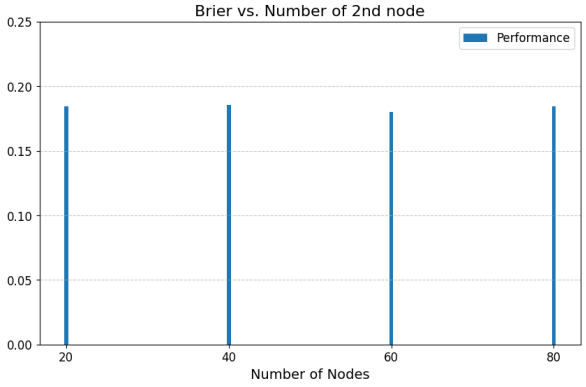

Figure 12: Brier when iterating over the number of nodes.

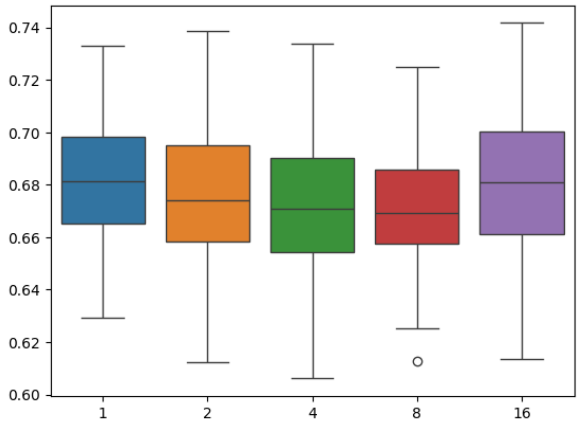

Figure 13: Concordance when iterating over the number of Heads.

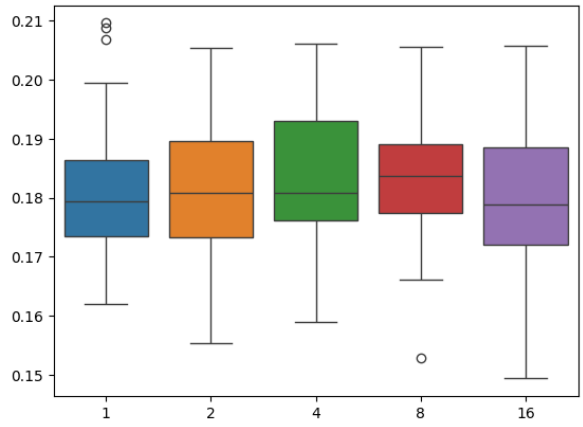

Figure 14: Brier when iterating over the number of Heads.