# Clinical Time-to-Event Prediction Enhanced by Incorporating Compatible Related Outcomes

Yan Gao[1,2,*] and Yan Cui[1,2,3,*]

[1]Department of Genetics, Genomics and Informatics, University of Tennessee Health Science Center, Memphis, TN, 38163, USA

[2]Center for Integrative and Translational Genomics, University of Tennessee Health Science Center, Memphis, TN, 38163, USA

[3]Center for Cancer Research, University of Tennessee Health Science Center, Memphis, TN, 38163, USA

*corresponding authors: Yan Gao (ygao45@uthsc.edu), Yan Cui (ycui2@uthsc.edu)

## Abstract

Accurate time-to-event (TTE) prediction of clinical outcomes from personal biomedical data is essential for precision medicine. It has become increasingly common that clinical datasets contain information for multiple related patient outcomes from comorbid diseases or multifaceted endpoints of a single disease. Various TTE models have been developed to handle competing risks that are related to mutually exclusive events. However, clinical outcomes are often non-competing and can occur at the same time or sequentially. Here we develop TTE prediction models with the capacity of incorporating data of compatible related clinical outcomes. We test our method on real and synthetic data and find that the incorporation of related auxiliary clinical outcomes can: 1) significantly improve the TTE prediction performance of convention Cox model while maintaining its interpretability; 2) further improve the performance of the state-of-the-art deep learning based models. While the auxiliary outcomes are utilized for model training, the model deployment is not limited by the availability of the auxiliary outcome data because the auxiliary outcome information is not required for the prediction of the primary outcome once the model is trained.

1

## Introduction

With the rapid advances in health informatic technologies, comprehensive biomedical datasets with multiple related clinical outcomes have become increasingly common. The related clinical outcome variables may represent comorbidity or multifaceted endpoints of a single disease. The multifaceted endpoints provide the measurement of the disease outcomes from different perspectives. The Cancer Genome Atlas[1] (TCGA) clinical dataset[2], as a well-known example, includes four variables to characterize the cancer outcome endpoints: overall survival, disease-specific survival, disease-free interval, and progression-free interval. Studies of comorbidity, the coexistence of multiple diseases or disorders in relation to a primary disease in a patient[3], may also generate multiple related clinical outcome data such as the cardiovascular disease related outcomes in the Sleep Heart Health Study (SHHS) data[4,5]. These related clinical outcomes contain rich information from multiple aspects of disease progression and may stem from common molecular mechanisms or environmental factors.

The Cox proportional hazards (CPH)[6-8] model has been widely used for time-to-event (TTE) data analysis for decades. In the recent years, deep learning (DL) based methods have been developed for disease classification, diagnosis, and prognosis from personal biomedical data[9-16]. The DL-based TTE models generally outperformed the conventional CPH and other survival analysis models in recent studies[17-23]. However, the black-box nature of deep neural networks makes the DL-based models lack the interpretability that is critical for clinical applications. In these TTE prediction models, a single outcome variable was used as the prediction target, even if multiple related clinical outcome data were available. Developing models to utilize the information from the related outcomes may provide an effective approach to enhance TTE prediction. Various multi-task machine learning models have been developed to handle competing risks that are related to mutually exclusive events[24-28]. However, the widely available compatible related outcomes that may occur at the same time or sequentially have not been formally utilized in the TTE models to improve prediction accuracy.

In this study, we develop a new method, the related outcome incorporator (ROI), to improve the TTE prediction performance for the primary outcome by incorporating related clinical outcome (the auxiliary outcome) data during model training. However, the auxiliary outcome information is not

2

required for the prediction of the primary outcome after model training. Thus, the model application would not be limited by the availability of the auxiliary outcome data. We integrated the ROI with the conventional CPH model and the DL-based model. Using prognosis tasks assembled from real and synthetic datasets, we show that ROI can significantly improve the prediction performance of CPH model while maintaining its interpretability and the ROI can further improve the prediction performance of the state-of-the-art DL-based model.

## Results

### TTE prediction experiments on TCGA data

We compared the TTE prediction models with and without the ROI using the TCGA data. The TCGA dataset contains 4 clinical outcome endpoints for 40 types and categories of cancers (Supplementary Table 1): overall survival (OS), disease-specific survival (DSS), disease-free interval (DFI), and progression-free interval (PFI). We assembled 160 time-to-event prediction tasks using the TCGA clinical data and protein expression data. We filtered out the tasks with less than 50 patients or less than 20 observed events. We further filtered out the tasks with an event-to-censored ratio (E/C) less than 0.2. Finally, we removed the tasks with concordance-index (C-index) less than 0.6 from the baseline CPH model, resulting in 19 tasks from 9 cancer types/categories and 4 different outcomes (**Table 1**). We tested four different TTE prediction models on each task: CPH, CPH with ROI (CPH_ROI), CPH integrated with a deep neural network (CPH_DL), and CPH_DL with ROI (CPH_DL_ROI). The models are described in the Method section. OS and DSS are survival outcomes and are used as a pair of related outcomes for the ROI. Also, PFI and DFI are disease progression outcomes and are used as another pair of related outcomes.

In our experiments, the CPH_ROI outperformed CPH on 17 out of the 19 tasks and CPH_DL_ROI outperformed CPH_DL on 15 out of the 19 tasks and tied with CPH_DL on 2 tasks. The use of ROI improved the mean C-index for the convention Cox model (CPH) by 6.1%, and the mean C-index for the deep Cox model (CPH_DL) by 2.9%. The p-values indicate that the performance improvements by ROI are statistically significant (**Fig.1**).

3

**Table 1** TTE prediction model performance comparison on TCGA data

| Cancer Type* | Primary Outcome | Auxiliary Outcome | Censored | Events | E/C | C-Index | | | |
| | | | | | | CPH | CPH_ROI | CPH_DL | CPH_DL_ROI |
|---|---|---|---|---|---|---|---|---|---|
| GBMLGG | OS | DSS | 391 | 274 | 0.70 | 0.74 | **0.79** | 0.80 | **0.81** |
| GBMLGG | DSS | OS | 397 | 246 | 0.62 | 0.74 | **0.80** | **0.81** | 0.81 |
| GBMLGG | PFI | DFI | 325 | 340 | 1.05 | 0.72 | **0.76** | **0.77** | 0.76 |
| KIPAN | OS | DSS | 548 | 206 | 0.38 | 0.71 | **0.73** | 0.73 | **0.74** |
| KIRC | DSS | OS | 364 | 104 | 0.29 | 0.70 | **0.72** | 0.71 | **0.74** |
| KIPAN | DSS | OS | 609 | 133 | 0.22 | 0.70 | **0.76** | 0.78 | **0.80** |
| PanGyn | PFI | DFI | 622 | 459 | 0.74 | 0.69 | **0.70** | **0.70** | 0.69 |
| PanGyn | DFI | PFI | 479 | 216 | 0.45 | 0.67 | **0.70** | 0.71 | **0.73** |
| KIPAN | PFI | DFI | 539 | 213 | 0.40 | 0.67 | **0.70** | **0.72** | 0.72 |
| LGG | OS | DSS | 330 | 98 | 0.30 | 0.66 | **0.73** | 0.75 | **0.76** |
| CESC | PFI | DFI | 141 | 32 | 0.23 | 0.64 | **0.70** | 0.61 | **0.64** |
| LGG | DSS | OS | 333 | 89 | 0.27 | 0.63 | **0.72** | 0.76 | **0.77** |
| PanGyn | OS | DSS | 693 | 388 | 0.56 | 0.63 | **0.64** | 0.66 | **0.67** |
| KIRC | OS | DSS | 312 | 166 | 0.53 | 0.62 | **0.69** | 0.68 | **0.70** |
| PanGyn | DSS | OS | 734 | 313 | 0.43 | 0.62 | **0.67** | 0.68 | **0.70** |
| COADREAD | OS | DSS | 383 | 104 | 0.27 | 0.62 | **0.64** | 0.63 | **0.64** |
| LUAD | DSS | OS | 244 | 80 | 0.33 | **0.61** | 0.52 | 0.53 | **0.58** |
| PanGI | DFI | PFI | 376 | 80 | 0.21 | **0.61** | 0.58 | 0.54 | **0.64** |
| KIRC | PFI | DFI | 322 | 155 | 0.48 | 0.60 | **0.65** | 0.68 | **0.69** |
| **Mean** | | | 428 | 195 | 0.44 | 0.66 | **0.69** | 0.70 | **0.72** |
| **Median** | | | 383 | 166 | 0.40 | 0.66 | **0.70** | 0.71 | **0.72** |

*See Supplementary Table 1 for the annotations of cancer type abbreviations.
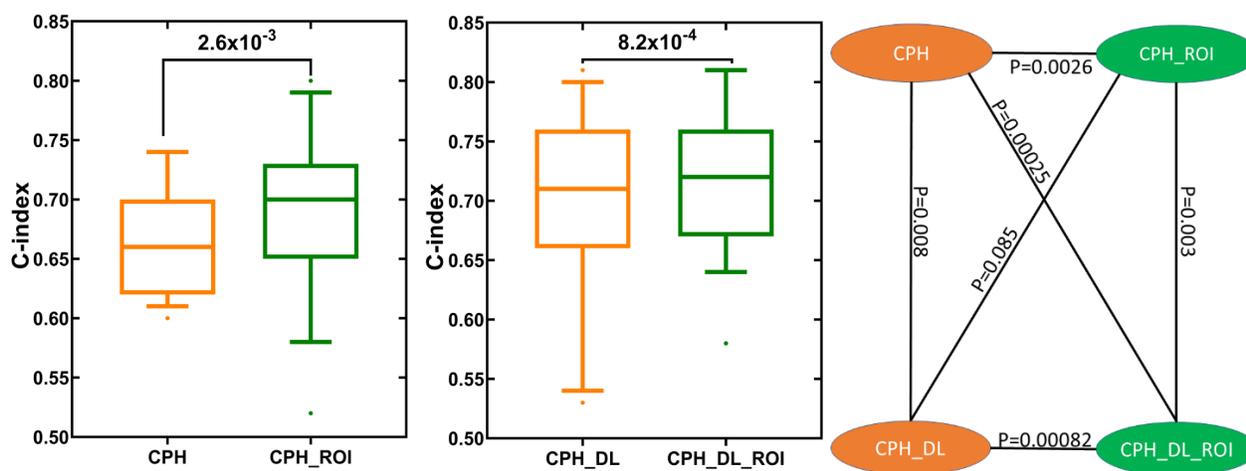


**Fig 1**. TTE prediction performance comparison between models with and without the related outcome incorporator (ROI). Each box plot shows the distribution of C-index values for the 19 TTE prediction tasks (Table 1). The p values were calculated using one-sided Wilcoxon signed-rank test.

4

## TTE prediction experiments on SHHS data

To further test the ROI method, we assembled TTE prediction tasks using the data from Sleep Heart Health Study (SHHS). The SHHS dataset contains 5804 samples from 10 different types of cardiovascular disease (CVD), and each sample comprises 1279 clinical features, as well as the follow-up event time. We used Angina as the primary outcome, and Congestive Heart Failure (CHF) and Stroke as the auxiliary tasks are (Table 2). In these experiments, the CPH_ROI model outperformed the CPH model, and the CPH_DL_ROI model outperformed the CPH_DL model. The results show that using ROI can improve the TTE predictions.

**Table 2** TTE prediction model performance comparison on SHHS data

| Primary outcome | Auxiliary outcome | C-Index | | | |
|---|---|---|---|---|---|
| | | CPH | CPH_ROI | CPH_DL | CPH_DL_ROI |
| Angina | CHF | 0.49 | **0.60** | 0.61 | **0.64** |
| Angina | Stroke | 0.49 | **0.59** | 0.59 | **0.62** |

## TTE prediction experiments on synthetic data

To further study our method and test the performance improvement of ROI, we performed an experiment on a synthetic dataset. The synthetic dataset has a total of 300 patients, 200 patients with observed events, and 100 patients with censorship. Each patient has 150 covariates to simulate the high-dimensional biomedical features. The time to the primary outcome (Outcome1) event and auxiliary outcome (Outcome2) event for each patient were generated using the exponential Cox model[29], in which the hazard function is a linear combination of features $h(x) = \beta \times x$. To simulate the relevance between Outcome1 and Outcome2, we set the weights for their hazard functions have a correlation value greater than 0.8. The results for the primary outcome and the auxiliary outcome are shown in Table 3. We observed a significant improvement in the performance of the models with ROI (CPH_ROI and CPH_DL_ROI) compared to the corresponding models without ROI (CPH and

5

CPH_DL). The results on the synthetic data show that incorporating a related outcome can improve time-to-event predictions, which is consistent with the observations from the real datasets.

**Table 3** TTE prediction model performance comparison on synthetic data

| Primary Outcome | Auxiliary Outcome | C-Index | | | | |
|---|---|---|---|---|---|---|
| | | CPH | CPH_ROI | | CPH_DL | CPH_DL_ROI |
| Outcome1 | Outcome2 | 0.67 | **0.81** | | 0.79 | **0.84** |
| Outcome2 | Outcome1 | 0.63 | **0.78** | | 0.78 | **0.80** |

## Discussion

We developed a method to improve TTE prediction by incorporating related clinical outcomes (ROI) in model training. We tested the utility of the ROI method in two types of models: the conventional CPH model and the deep learning based model that integrates deep neural network and Cox regression (CPH_DL). Consisting with previous observations[17-23], the DL structures can generate better feature representations and therefore are able to improve the TTE predictions. However, the low interpretability of DL-based models is a major obstacle to their application in healthcare where user (patients and clinicians) trust is critical[30-32]. The use of ROI can improve the TTE prediction performance of the CPH model while maintain its full interpretability. In our experiments on the real and synthetic datasets, the use of ROI improved the prediction performance of the CPH model to the level that is comparable to that of the DL-based model (CPH_ROI vs CPH_DL, Table 1-3). This interesting observation suggests that the use of ROI with CPH model and the integration of DL with CPH model may lead to a comparable level of prediction performance improvement. The use of ROI also further improved the performance of the DL-based model. Time-to-event analysis is widely used in many areas beyond medicine, including engineering, economics and finance. We expect that the ROI framework may also be applicable to TTE predictions in more broader areas.

# Methods

## Datasets

We used the TCGA clinical and protein expression datasets from the Genome Data Commons (GDC, https://gdc.cancer.gov). We filtered the proteins with missing values, resulting in 189 protein features. Patients with missing values for follow-up time or event indicators were removed. For each survival analysis task, the protein expression values were standardized by removing the mean and scaled to unit variance. The clinical outcome is the time in days and the event indicator for endpoints: OS, PFI, DFI, and DSS.

For the Sleep Heart Health Study (SHHS)[4,5] dataset, we acquired the data from the National Sleep Research Resource (https://sleepdata.org/datasets/shhs). We removed the records with 20% missing features and dropped the features with missing values. We converted the $rcrdtime$ (total recording time) into the number of minutes and filtered out samples without follow-up time information. After the preprocessing, we got 1514 uncensored events from 10 outcomes, and each record has 374 features. For each event, we applied an unsupervised feature selection method to select 150 features with top mean absolute deviation values. We then filtered out the clinical outcomes with less than 50 samples. The final dataset contains three clinical outcomes: Angina (121 patients), Congestive Heart Failure (197 patients), and Stroke (79 patients).

## The baseline model

In our study, for each survival analysis task, the baseline model we used is the Cox proportional-hazards (CPH) model[33] (Fig 2A) implemented in the python lifeline package[34]. To improve the robustness of the model, we set the penalty weight of 0.0001 to the coefficients during fitting. All other parameters remain unchanged.

7
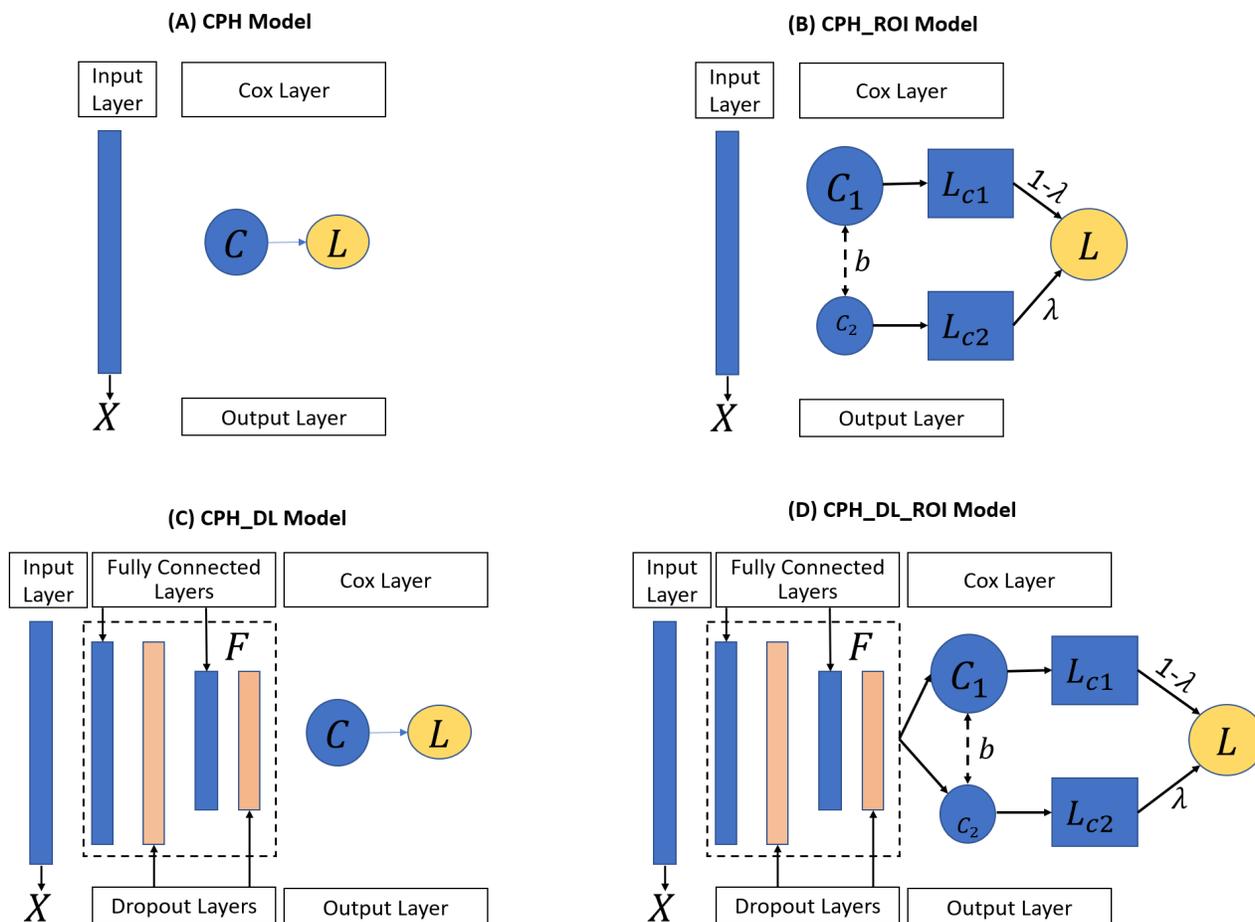
**Fig 2.** The four TTE prediction models: (A) CPH Model, (B) CPH_ROI Model, (C) CPH_DL Model and (D) CPH_DL_ROI Model. $X$ represents the input features, $C$ is the Cox regression layer, $L$ is the partial hazard loss function, $\lambda$ is the weight to balance outcomes, and $F$ is the feature extractor to map the input feature into an embedding space. **CPH**: Cox proportional hazards. **ROI**: related outcome incorporator. **DL**: deep learning.

## The CPH_ROI model

The proposed CPH_ROI model has two parts, and each part is a regularized CPH model (Fig. 2B). The loss function of CPH_ROI (Equation 1) is a linear combination of two related events, the loss of the main task and the loss for the related task. In the equation, $\lambda$ is a hyperparameter used to balance the loss between the primary outcome and the auxiliary outcome, $\beta_1$ and $\beta_2$ are the log partial hazard ratio for the main and the auxiliary outcomes, $b$ is the bias, $U_1$ and $U_2$ are the set of

8

uncensored patients for the main and the auxiliary outcomes. In equation 2, $\sum_{i\in U}^{m} L(\beta, b)$ is the log partial likelihood which is defined in Equation 3, $U$ is the set of uncensored patients, and $\lambda_1$ is the weight of the $L_2$ regularization term. In Equation 3, $m$ is the number of patients, $R(T_i)$ is the risk group in which each patient's survival time is greater than $T_i$, $\delta_i$ is the event status of patient $i$, $\delta_i = 1$ if an event (like death) occurred, and $\delta_i = 0$ when there is a censorship. The $h(X_i, \beta, b)$ in equation 4 is the linear hazard function, where $X_{ij}$ is the $j^{th}$ feature of patient $i$, $\beta_j$ is $j^{th}$ weight of $\beta$, and $b$ is the bias. During the training, we set $\lambda$ to 0.2, the parameters of $\beta_1, \beta_2, b$ were randomly initialized and optimized by minimizing the objective function in equation 1. Adopting the idea of transfer learning for high-dimensional linear regression[35], we set the two linear functions to share the same bias.

$$l(\beta_1, \beta_2, b) = (1-\lambda)l(\beta_1, b)_{i\in U_1} + \lambda l(\beta_2, b)_{i\in U_2} \tag{1}$$

$$l(\beta, b)_{i\in U} = -\sum_{i\in U}^{m} L(\beta, b) + \lambda_1 \|\beta, b\|^2 \tag{2}$$

$$L(\beta, b) = log\left[\prod_{i=1}^{m} \frac{e^{h(X_i, \beta, b)}}{\sum_{j\in R(T_i)} e^{h(X_j, \beta, b)}}\right]^{\delta_i} = \sum_{i=1}^{m} \delta_i \left[h(X_i, \beta, b) - log\left\{\sum_{j\in R(T_i)} e^{h(X_j, \beta, b)}\right\}\right] \tag{3}$$

$$h(X_i, \beta, b) = \sum_{j=1}^{n} \beta_j * X_{ij} + b \tag{4}$$

## The CPH_DL model

In the CPH_DL model, a Cox regression layer was added on top of a deep neural network structure (Fig. 2C). The loss function is shown in Equation 5, which has two parts: a deep neural network-based feature extractor $F$ and a Cox regression layer $C$, where $F$ maps the high-dimensional input feature into an embedded space $Z$, and then $C$ makes a prediction from $Z$. In Equation5, $\lambda_1$ is the $L_2$ regularization weight, $W$ is the weight of $F$, $\beta$ and $b$ are the log partial hazard ratio and the bias of $C$. All other notions are the same as the CPH_ROI model. The feature extractor $F$ is a 4-layer deep neural network structure: a fully connected (FC) layer of 100 nodes followed by a dropout layer (with p=0.5), an FC layer of 50 nodes followed by a dropout layer (with p=0.5). These two dropout layers and the regularization term are used to avoid overfitting during training.

$$l(F, C) = -\sum_{i\in U}^{m} L(F, \beta, b) + \lambda_1 \|W, \beta, b\|^2 \tag{5}$$

9

$$L(F,\beta,b) = log\left[\prod_{i=1}^{m}\frac{e^{h(F(X_i),\beta,b)}}{\sum_{j\epsilon R(T_i)}e^{h(F(X_j),\beta,b)}}\right]^{\delta_i} = \sum_{i=1}^{m}\delta_i\left[h(F(X_i),\beta,b) - log\left\{\sum_{j\epsilon R(T_i)}e^{h(F(X_j),\beta,b)}\right\}\right]$$

(6)

## The CPH_DL_ROI model

The CPH_DL_ROI model (Fig. 2D) incorporates the related outcome in its loss function (Equation 7), where $\lambda$ is the adjustment weight, $F$ is the feature extractor, $C_1$ and $C_2$ are the Cox regression layer for each outcome. Similar to the CPH_ROI model, we set $C_1$ and $C_2$ share the bias and set $\lambda$ to 0.2. The notion of $l(F, C)$, $U_1$ and $U_2$ are same as the notions defined in the CPH_ROI and CPH_DL models.

$$l(F, C_1, C_2) = (1 - \lambda)l(F, C_1)_{i\in U_1} + \lambda l(F, C_2)_{i\in U_2} \tag{7}$$

## Experiment setup

For a given task with two related outcomes $D = \{x_i, t_i, e_i, t_i', e_i'\}_{i=1}^{m}$, $m$ is the number of patients, and for patient $i$, $x_i$ is the input feature, $t_i$ and $e_i$ are the event time and status of the primary outcome, $t_i'$ and $e_i'$ are the time and event for the auxiliary outcome. We formulate it as $D = \{D_1 \cup D_2\}$, where $D_1 = \{x_i, t_i, e_i\}_{i=1}^{m}$ is the main task and $D_2 = \{x_i, t_i', e_i'\}_{i=1}^{m}$ is the auxiliary task. For each task $D_i$, we applied a stratified ten-fold cross-validation for training and testing splitting. The dataset was stratified by the event status to ensure a uniform distribution of events across each fold. In each fold, the training set of $D_1$ was used to train a CPH and CPH_DL models, while the training sets of $D_1$ and $D_2$ were used to train the CPH_ROI and CPH_DL_ROI models. For the CPH_ROI and CPH_DL_ROI models, in the testing stage, only the input features from the testing set of $D_1$ were used to calculate the risk of each sample, and these risk scores were used for evaluation.

## Evaluation metric

We evaluated the prediction performance of each model using the concordance index (C-index)[36], which measures the proportion of concordant pairs among the total number of possible pairs. Those

10

pairs were discarded if the earlier time is censored. For a testing set, its risk scores with event time and event status were used to calculate the C-index. A higher C-index value indicates a better TTE prediction model. A C-index of 1.0 indicates a perfect prediction model, while a C-index of 0.5 indicates a totally random prediction model.

## Synthetic data generator

We developed a synthetic data generator to simulate the dataset for our simulation study. The simulated dataset can be formulated as $D = \{x_{ij}, t_i, e_i, t', e_i'\}_{i=1}^M$, where $M$ is the total number of patients, $x_{ij}$ is the $j^{th}$ input feature of patient $i$, $t_i$ and $e_i$ are the event time and status of the primary outcome, $t_i'$ and $e_i'$ are the event time and statues of the auxiliary outcome of patient $i$. We simulated the feature matrix $x_{ij}$ from a uniform distribution with the lower boundary $-1$ and higher boundary of 1. For each patient, we set the feature size to 150 to simulate the high-dimensional biomedical data. The main event time for patient $i$ was simulated using the exponential Cox model, as shown in Equation 8, where $\lambda$ is the baseline function, $EXP$ is an exponential distribution with a $mean = 3000$, and $\beta_j$ is the weight of feature $j$ which was generated from a uniform distribution. With the simulated event time, we set a cut-off time threshold to simulate the "end-of-study" to keep a portion of desire uncensored samples in the dataset.

To simulate the event time and status of the related outcome, we added a random noise to the weight of the primary outcome, $\beta_j' = \beta_j + \xi$, where $\xi$ is a uniform distribution with a range of 0 and 1. We control the correlation between $\beta$ and $\beta'$ to be greater than 0.8 to ensure the relevance of two outcomes. For both events, we generated a total of 300 patients, 200 patients with observed events, and 100 patients with censorship.

$$T_i \sim Exp\big(\lambda(t; x_i)\big) = \frac{EXP}{exp\big(\sum_{j=1}^m \beta_j * x_{ij}\big)} \tag{8}$$

## Data availability

The TCGA data was downloaded from the Genome Data Commons (https://gdc.cancer.gov/about-data/publications/pancanatlas). The SHHS data was downloaded from the National Sleep Research Resource website (https://sleepdata.org/datasets/shhs).

## Acknowledgements

## References

1. The Cancer Genome Atlas Program. (https://www.cancer.gov/about-nci/organization/ccg/research/structural-genomics/tcga).
2. Liu, J. *et al.* An integrated TCGA pan-cancer clinical data resource to drive high-quality survival outcome analytics. *Cell* **173**, 400-416. e11 (2018).
3. Moni, M.A., Xu, H. & Lio, P. Cytocom: a cytoscape app to visualize, query and analyse disease comorbidity networks. *Bioinformatics* **31**, 969-971 (2015).
4. Quan, S.F. *et al.* The sleep heart health study: design, rationale, and methods. *Sleep* **20**, 1077-1085 (1997).
5. Zhang, G.-Q. *et al.* The National Sleep Research Resource: towards a sleep data commons. *Journal of the American Medical Informatics Association* **25**, 1351-1358 (2018).
6. Cox, D.R. Regression models and life‐tables. *Journal of the Royal Statistical Society: Series B (Methodological)* **34**, 187-202 (1972).
7. Lin, D.Y. & Wei, L.-J. The robust inference for the Cox proportional hazards model. *Journal of the American statistical Association* **84**, 1074-1078 (1989).
8. Kumar, D. & Klefsjö, B. Proportional hazards model: a review. *Reliability Engineering & System Safety* **44**, 177-188 (1994).
9. Cheerla, A. & Gevaert, O. Deep learning with multimodal representation for pancancer prognosis prediction. *Bioinformatics* **35**, i446-i454 (2019).
10. She, Y. *et al.* Development and Validation of a Deep Learning Model for Non–Small Cell Lung Cancer Survival. *JAMA Network Open* **3**, e205842-e205842 (2020).

11. Singh, V., Baranwal, N., Sevakula, R.K., Verma, N.K. & Cui, Y. Layerwise feature selection in Stacked Sparse Auto-Encoder for tumor type prediction. in *2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)* 1542-1548 (2016).

12. Sevakula, R.K., Singh, V., Verma, N.K., Kumar, C. & Cui, Y. Transfer Learning for Molecular Cancer classification using Deep Neural Networks. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 1-1 (2018).

13. Poirion, O.B., Jing, Z., Chaudhary, K., Huang, S. & Garmire, L.X. DeepProg: an ensemble of deep-learning and machine-learning models for prognosis prediction using multi-omics data. *Genome Medicine* **13**, 112 (2021).

14. Tran, K.A. *et al.* Deep learning in cancer diagnosis, prognosis and treatment selection. *Genome Medicine* **13**, 152 (2021).

15. Gao, Y. & Cui, Y. Deep transfer learning for reducing health care disparities arising from biomedical data inequality. *Nature Communications* **11**, 5131 (2020).

16. Gao, Y. & Cui, Y. Multi-ethnic Survival Analysis: Transfer Learning with Cox Neural Networks. in *Proceedings of AAAI Spring Symposium on Survival Prediction - Algorithms, Challenges, and Applications 2021* Vol. 146 (eds Russ, G., Neeraj, K., Thomas Alexander, G. & Mihaela van der, S.) 252--257 (PMLR, Proceedings of Machine Learning Research, 2021).

17. Qiu, Y.L., Zheng, H., Devos, A., Selby, H. & Gevaert, O. A meta-learning approach for genomic survival analysis. *Nature communications* **11**, 1-11 (2020).

18. Yousefi, S. *et al.* Predicting clinical outcomes from large scale cancer genomic profiles with deep survival models. *Scientific Reports* **7**, 11707 (2017).

19. Ching, T., Zhu, X. & Garmire, L.X. Cox-nnet: An artificial neural network method for prognosis prediction of high-throughput omics data. *PLOS Computational Biology* **14**, e1006076 (2018).

20. Katzman, J.L. *et al.* DeepSurv: personalized treatment recommender system using a Cox proportional hazards deep neural network. *BMC medical research methodology* **18**, 24 (2018).

21. Luck, M., Sylvain, T., Cardinal, H., Lodi, A. & Bengio, Y. Deep learning for patient-specific kidney graft survival analysis. *arXiv preprint arXiv:1705.10245* (2017).

22. Kvamme, H., Borgan, Ø. & Scheel, I. Time-to-Event Prediction with Neural Networks and Cox Regression. *Journal of Machine Learning Research* **20**, 1-30 (2019).

23. Wang, D., Jing, Z., He, K. & Garmire, L.X. Cox-nnet v2.0: improved neural-network based survival prediction extended to large-scale EMR data. *Bioinformatics* (2021).

24. Alaa, A.M. & van der Schaar, M. Deep multi-task gaussian processes for survival analysis with competing risks. in *Proceedings of the 31st International Conference on Neural Information Processing Systems* 2326-2334 (2017).

25. Bellot, A. & van der Schaar, M. Multitask boosting for survival analysis with competing risks. *Advances in Neural Information Processing Systems* **31**, 1390-1399 (2018).

26. Wang, Z. & Sun, J. SurvTRACE: Transformers for Survival Analysis with Competing Events. *arXiv preprint arXiv:2110.00855* (2021).

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

28. Lee, C., Zame, W.R., Yoon, J. & van der Schaar, M. Deephit: A deep learning approach to survival analysis with competing risks. in *Thirty-second AAAI conference on artificial intelligence* (2018).

29.    Austin, P.C. Generating survival times to simulate Cox proportional hazards models with time‐varying covariates. *Statistics in medicine* **31**, 3946-3958 (2012).
30.    Fan, F.-L., Xiong, J., Li, M. & Wang, G. On interpretability of artificial neural networks: A survey. *IEEE Transactions on Radiation and Plasma Medical Sciences* (2021).
31.    Barredo Arrieta, A. *et al.* Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion* **58**, 82-115 (2020).
32.    Yoon, C.H., Torrance, R. & Scheinerman, N. Machine learning in medicine: should the pursuit of enhanced interpretability be abandoned? *Journal of Medical Ethics*, medethics-2020-107102 (2021).
33.    Fox, J. & Weisberg, S. Cox proportional-hazards regression for survival data. *An R and S-PLUS companion to applied regression* **2002**(2002).
34.    Davidson-Pilon, C. lifelines 0.25.9 Survival analysis in Python, including Kaplan Meier, Nelson Aalen and regression. *https://pypi.org/project/lifelines/0.25.9/* (2021).
35.    Li, S., Cai, T.T. & Li, H. Transfer learning for high-dimensional linear regression: Prediction, estimation, and minimax optimality. *arXiv preprint arXiv:2006.10593* (2020).
36.    Harrell, F.E., Califf, R.M., Pryor, D.B., Lee, K.L. & Rosati, R.A. Evaluating the yield of medical tests. *Jama* **247**, 2543-2546 (1982).