Samantha Morrison, Constantine Gatsonis, Ani Eloyan and Jon Arni Steingrimsson*

Survival analysis using deep learning with medical imaging

https://doi.org/10.1515/ijb-2022-0113
Received September 14, 2022; accepted February 24, 2023; published online June 14, 2023

Abstract: There is widespread interest in using deep learning to build prediction models for medical imaging data. These deep learning methods capture the local structure of the image and require no manual feature extraction. Despite the importance of modeling survival in the context of medical data analysis, research on deep learning methods for modeling the relationship of imaging and time-to-event data is still under-developed. We provide an overview of deep learning methods for time-to-event outcomes and compare several deep learning methods to Cox model based methods through the analysis of a histology dataset of gliomas.

Keywords: convolutional neural networks; Doubly Robust estimation; survival analysis; time-to-event outcomes

1 Introduction

Time-to-event outcomes are common in medical research but analysis of such outcomes is often complicated by right-censoring, i.e., when a participant drops out of a study or has not experienced the event of interest at the end of follow-up. There are various classical methods for analysis of time-to-event outcomes under right-censoring such as the semi-parametric Cox proportional hazards (PH) model, accelerated failure time model, and additive model [1]. More recently, several machine learning methods have been developed for time-to-event data [2–6]. However, these methods are not directly applicable for analysis of time-to-event outcomes with imaging data as they require extracted imaging features rather than whole images as inputs. Extracting image features, such as shapes and pixel intensities in the image, requires input by experts, can be time consuming, and is susceptible to variability between experts.

In the absence of censoring, deep learning algorithms have been widely used for analyses of imaging data, such as classification, detection, segmentation, and registration of medical images [7]. In particular, convolutional neural networks are a type of deep learning algorithm that is tailored to image analysis. Unlike most other machine learning algorithms, convolutional neural networks use the whole image as an input and capture local structure within the image. Despite the rapid growth of approaches incorporating convolutional neural networks into medical image analyses, the combination of convolutional neural networks with survival analysis has received substantially less attention. Most prior work replaced the unobservable full data loss used in uncensored convolutional neural networks by the negative log-likelihood from a Cox proportional hazard model [8–10]. To return interpretable measures, such as survival probabilities or hazard rates, the output from the network needs to be combined with estimators of the baseline hazard from the proportional hazards model, therefore relying on the proportional hazards assumption.

***Corresponding author: Jon Arni Steingrimsson**, Department of Biostatistics, School of Public Health, Brown University, Providence, RI, USA, E-mail: jon_steingrimsson@brown.edu
Samantha Morrison, **Constantine Gatsonis and Ani Eloyan**, Department of Biostatistics, School of Public Health, Brown University, Providence, RI, USA

In the context of fully connected neural networks [6], developed deep learning algorithms that account for censoring by replacing the unobservable loss function with censoring unbiased loss functions. Censoring unbiased loss functions are unbiased estimators for the full data risk (expected loss) that can be calculated when some observations are censored and have the advantage of not relying on the proportional hazard assumption.

In this manuscript, we present an overview of previously developed deep learning methods for image analysis with time-to-event outcomes, and compare the performance of the methods using data from cancer patients with brain tumors to predict their survival.

2 Glioma biopsy imaging dataset and preprocessing

We present our methods using a database consisting of magnified slides of tissue biopsies from 769 patients with brain tumors and the patients' associated observed overall survival time (in days). The data was retrieved from the TCGA database (TCGA Research Network: https://www.cancer.gov/tcga) by Ref. [8] from the Brain LGG and GBM cohorts [11–13]. The magnified tissue biopsy slides of the 769 gliomas were H&E-stained whole slides that were processed to select regions of interest (ROIs) identified by experts using the Digital Slide Archive (ROI, 1024 × 1024 pixels) [8]. The ROI had primary tumor nuclei and were processed at "20× objective magnification using OpenSlide and color-normalized to a gold standard H&E calibration images" [8]. All of the whole slide images went through quality control review that included removal of images with image artifacts or with tissue processing issues (e.g., overstaining, understaining) [8]. Additionally, regions with geographic necrosis were excluded. One patient had a survival time of 0 days and was therefore not included in the analysis.

Each patient can have multiple whole slides and ROIs. In our analysis, we randomly sampled one ROI per patient ($n = 768$ ROIs). There was a 49.5 % censoring rate and for the analysis we split the data into 80 % training (615 observations) and 20 % testing (153 observations). Figure 1 shows two examples of ROIs, one from a patient with a survival time of 627 days and one with a survival time of 1077 days.

3 Brief overview of deep learning and convolutional neural networks

To establish the statistical framework and notation presented in the paper, we start by providing a description of the deep learning algorithm when there is no censoring [14]. Let $T_i \in \mathbb{R}^+$ denote the survival time for participant i and $W_i \in \mathbb{R}^p$ be the vector of image intensities for each pixel in the image associated with participant i where p is the total number of pixels. The data observed in the absence of censoring, hereafter referred to as the fully observed data, is $\mathcal{F} = \left\{ \left(T_i, W_i^T\right)^T : i = 1, \ldots, n \right\}$. We will focus on predictions of the form $E[h(T)|W]$, where h is a monotone transformation of the survival time $h: \mathbb{R}^+ \to \mathbb{R}$ (e.g., $h(T) = I(T > t)$ and $E[h(T)|W] = P(T > t|W)$). The final deep learning prediction model, denoted $f_\beta(W)$, consists of function compositions of several layers, where β refers to the unknown parameters (weights) that need to be estimated to fully specify the deep learning predictions. The parameters β are estimated by minimizing a loss based objective function (loss function:

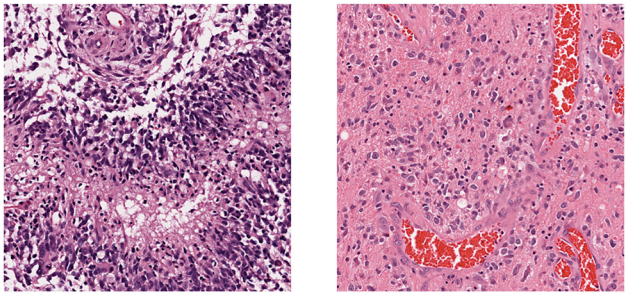

Figure 1: Example histology ROIs with survival time of 627 days (left) and 1077 days (right).

$L(h(T), f_\beta(W)))$ that compares the discrepancy between the observed outcome $h(T)$ and the predicted outcome $f_\beta(W)$. The deep learning algorithm described below can include both fully connected neural networks and convolutional neural networks as special cases. The algorithm is given by the following steps [14]:

(1) Define the structure (architecture) of the neural network prediction model. Let K be the number of layers in the model. For a given $k \in \{1, \ldots, K\}$, the output of the kth layer is defined as $f^{(k)}_{\beta_k}$ which consists of a nonlinear activation function that depends on the unknown parameters (β_k). The final prediction model from the algorithm is given by $f_\beta(w) = f^{(K)}_{\beta_K} \circ f^{(K-1)}_{\beta_{K-1}} \circ \ldots \circ f^{(1)}_{\beta_1}(w)$. Here, $\circ$ denotes function composition (i.e., $(c \circ d)(w) = c(d(w))$ for two functions $c(\cdot)$ and $d(\cdot)$). Denote the parameters (weights) as $\beta = \left(\beta_1^T, \ldots, \beta_{K-1}^T, \beta_K^T\right)^T$ where each β_k is a vectorized form of the matrix of weights associated with layer k.

(2) Estimate the parameters (weights). Estimation of the parameters relies on selection of an optimization procedure (e.g., stochastic gradient descent) and a loss function ($L(h(T), f_\beta(w))$). For a given penalization parameter η, the vector of unknown parameters β is estimated by minimizing

$$\frac{1}{n}\sum_{i=1}^{n} L(h(T_i), f_\beta(W_i)) + \eta\|\beta\|^2, \tag{1}$$

where $\|\beta\|^2 = \sum_{j=1}^{q}|\beta_j|^2$ and q is the dimensionality of β.

(3) Select the optimal penalization parameter η^* using cross-validation or using an independent test set. The final output from the prediction model is $f_{\beta^*}(W_i)$, where β^* is the value of β that minimizes expression (1) with $\eta = \eta^*$.

In this manuscript, we are interested in convolutional neural networks since they are deep learning algorithms that use architectures tailored to image recognition [15, 16]. Convolutional neural networks use local connectivity, parameter sharing, and pooling to capture local spatial information from the image and reduce the dimension of the data. Below we give heuristic explanations of local connectivity, parameter sharing, and pooling; however, one can refer to the Supplementary Web Appendix for more information and refer to Ref. [17] for precise mathematical definitions of these concepts.

Local connectivity restricts the output units of a layer so that it is only affected by input units spatially close to the output unit, while in fully connected neural networks the output of each layer is connected to every single input unit. This local connectivity restriction results in a reduction of the dimensionality of the weights (β). Mathematically, this property is represented by replacing the matrix multiplication used in fully connected layers by a convolution with a kernel (multi-dimensional array of weights β) that has a smaller dimension than that of the input features. Intuitively, local connectivity utilizes the strong 2D local structure in images since it assumes interactions between pixels that are spatially close ought to be more important than pixels that are further apart. A layer using a convolution is called a convolutional layer.

If an important feature is detected in the image and then the feature position in the input is shifted, it would be ideal to detect the same important feature only shifted by the same amount. For instance, for an algorithm identifying specific tumors, changes in the location of the tumors on a medical image should result in the same detection and representation of the tumor in the output, only shifted by the same amount. To achieve this, parameter sharing is implemented by constraining each layer in the network to use the same set of parameters (e.g., same kernel function) at different parts of the image. Parameter sharing also reduces the memory storage required. Finally, pooling replaces the output of the layer by summary statistics, most often the max, calculated over a local region (e.g., the maximum value of neighboring pixels). In addition to reducing dimensionality of the parameter vector, pooling helps to make the output invariant to small translations of the input [14].

If at least one of the layers in a neural network is a convolutional layer, then the corresponding network is referred to as a convolutional neural network. Convolutional neural networks usually include multiple convolutional layers– which consist of the convolutions, activation functions, and pooling – followed by fully connected layers. The only difference between convolutional and fully connected neural networks in the full data deep

learning algorithm is which type of layer is utilized (e.g., step one of the full data deep learning algorithm described earlier in this section).

4 Deep learning for survival data

In the presence of censoring, the true survival time T is not always observed. Instead we observe $\tilde{T} = \min(T, C)$, where C is the censoring time, and a failure time indicator $\Delta = I(T \leq C)$. The observed data is denoted as $\mathcal{O} = \left\{ \left(\tilde{T}_i, \Delta_i, W_i^T \right)^T : i = 1, \ldots, n \right\}$. When an observation is censored ($\Delta = 0$) the true survival time is unknown so the contribution of that observation to the full data loss function (1) is unknown. Recall that we are interested in predicting $E[h(T)|W]$, a naive approach is to fit the deep learning algorithm using the observed time $\tilde{T}$ instead of the true failure time T. However, using this in connection with the deep learning algorithm leads to an estimator of $E[h(\tilde{T})|W]$ which is biased for the target parameter $E[h(T)|W]$. We will now describe several alternative methods to this naive approach that deal with censoring in the deep learning algorithm. These methods fall into two major categories, those that incorporate the proportional hazards likelihood into the deep learning algorithm and those that act as direct extensions of the previously presented full data deep learning algorithm to censored data structures. The latter focuses on replacing the unobservable full data loss (1) in the full data deep learning algorithm by an unbiased estimator for the full data loss that can be calculated in the presence of censoring.

4.1 Methods based on proportional hazards likelihood

The Cox proportional hazard model assumes that the hazard rate for the failure time $\lambda(t|W)$ satisfies $\lambda(t|W) = \lambda_0(t)e^{\beta^T W}$, where $\lambda_0(t)$ is the baseline hazard. Denote $\Lambda_0(t) = \int_0^t \lambda_0(u)du$ as the cumulative baseline hazard. Under the non-informative censoring assumption, "profiling" out the baseline hazard from the full likelihood [18] leads to the Cox proportional hazards partial likelihood

$$L(\beta) = \prod_{i:\Delta_i=1} \frac{e^{\beta^T W_i}}{\sum_{j=1}^{n} e^{\beta^T W_j} I(\tilde{T}_j \geq \tilde{T}_i)}.$$

The Cox proportional hazards model has been used to build survival prediction models in the context of image data analysis. Using clinician or radiologist hand-derived features from the images as covariates in a Cox proportional hazards model has been a popular approach to create risk prediction models for censored data [19]. Since there are numerous features that can be extracted from the images, selection of the covariates that are utilized in the Cox proportional hazards model can be done via penalized regression (ridge, LASSO) or fitting the Cox PH model with a few principal components [20]. More recently, automatic extraction of imaging features has received substantial attention [21]. One type of automated extraction is running images through a pre-trained deep learning model (such as VGG16 or ResNet50) built on large imaging datasets. The Cancer Imaging Phenomics Toolkit (CaPTk) is another technology recently developed for analysis of radiographic cancer images which provides a graphical user interface for extraction and analysis of radiomic features from images [22, 23].

Authors have also incorporated censored outcomes into the deep learning algorithm by using Cox proportional hazards based loss functions. This method originated with Ref. [24] who used a single layer neural network to estimate the functional form of the covariates in a Cox proportional hazard model by minimizing the negative of the Cox proportional likelihood via Newton-Raphson. This was extended by Refs. [25, 26] who considered networks with additional layers and different optimization procedures; this adapted algorithm was referred to as DeepSurv [10] provided an extension of DeepSurv to convolutional neural network architectures. This was later adapted by Ref. [8] who structured their algorithm similarly to the Imagenet VGG19 natural image analysis network and used the negative log partial Cox PH likelihood as the loss function. All of these neural network methods produce estimators for the functional form of the covariates in a Cox proportional hazard

model $g(W, \widehat{\beta})$. Estimation of survival probabilities of the form $\widehat{S}_0(t|W) = \widehat{P}(T > t|W) = e^{-e^{g(W,\widehat{\beta})}\widehat{\Lambda}_0(t)}$ requires estimating the cumulative baseline hazard $\widehat{\Lambda}_0(t)$ (e.g., using the Breslow estimator) and relies on the proportional hazard assumption.

4.2 Methods based on censoring unbiased transformations

As previously described, the main complication with adjusting the full data deep learning algorithm to censored outcomes is that the full data loss function cannot be calculated for censored outcomes. A censoring unbiased loss function is a function of the observed data $\mathcal{O}$ that is an unbiased estimator of the full data risk [27]. A simple and intuitive censoring unbiased loss function is the Buckley–James loss

$$L_{BJ}(\mathcal{O}, \beta, \widehat{S}) = \frac{1}{n}\sum_{i=1}^{n}\left(\Delta_i L(h(T_i), \beta(W_i)) + (1-\Delta_i)E_{\widehat{S}}[L(h(T), \beta(W))|T \geq C_i, W_i]\right),$$

where

$$E_{\widehat{S}}[L(h(T), \beta(W))|T \geq u, W] = -\int_u^{\infty} \frac{L(h(t), \beta(W))d\widehat{S}(t|W)}{\widehat{S}(u|W)}$$

for some estimator $\widehat{S}(u|W)$ for the survival curve $S_0(u|W) = P(T > u|W)$. If the true survival time is known, then the contribution of that observation to the Buckley–James loss is the same as it would contribute to the full data loss $L(h(T), \beta(W))$. However, if the true survival time is unknown (censored), the contribution is an estimator for the expected value of the full data loss given the observed information $E_{\widehat{S}}[L(h(T), \beta(W))|T \geq C, W]$. The Buckley–James loss is a consistent estimator of the full data risk $E[L(h(T), \beta(W))]$ if the estimator for $\widehat{S}(u|W)$ is correctly specified.

Another appealing censoring unbiased loss function is the Doubly Robust loss function [28]

$$L_{DR}(O, \beta, \widehat{G}, \widehat{S}) = \frac{1}{n}\sum_{i=1}^{n}\left\{\frac{\Delta_i L(h(T_i), \beta(W_i))}{\widehat{G}(\tilde{T}_i|W_i)} + \frac{(1-\Delta_i)E_{\widehat{S}}[L(h(T), W)|T \geq C_i, W_i]}{\widehat{G}(\tilde{T}_i|W_i)}\right.$$
$$\left. - \int_0^{\widetilde{T}_i} \frac{E_{\widehat{S}}[L(h(T), W)|T \geq u, W_i]}{\widehat{G}(u|W_i)} \mathrm{d}\Lambda_{\widehat{G}}(u|W_i)\right\},$$

where $\widehat{G}(u|W)$ is an estimator for the censoring survival curve $G_0(u|W) = P(C > u|W)$ and $\Lambda_{\widehat{G}}(u|W)$ is an estimator for the cumulative hazard function for the censoring distribution. Implementation requires estimation of the survival curve $S_0(u|W)$ and the censoring curve $G_0(u|W)$. The Doubly Robust loss is a consistent estimator for the full data risk if at least one of the estimators is correctly specified, but not necessarily both (referred to as the doubly robustness property).

Replacing the full data loss in the full data deep learning algorithm (1) by a censoring unbiased loss function results in a deep learning algorithm applicable to censored outcomes. When the full data loss is selected as the Brier Loss, $L(h(T), \beta(W)) = (I(T > t) - \beta(W))^2$, the estimated target parameter is the survival probability $P(T > t|W)$. If interest lies in estimating restricted mean survival ($E[\min(T, \tau)|W]$ for some pre-determined constant τ), the full data loss can be selected as $(\min(T, \tau) - \beta(W))^2$. Both the Brier loss and the loss for estimating restricted mean survival can be incorporated into the framework described earlier by using the L_2 loss with $h(T) = I(T > t)$ and $h(T) = \min(T, \tau)$, respectively.

5 Comparisons of methods on the glioma biopsy imaging dataset

We compared the prediction accuracy of the classical Cox PH regression model, DeepSurv, and the deep learning algorithms that use the Buckley–James and the Doubly Robust loss functions using the glioma histology imaging dataset. We focused on estimating $P(T > 914|W)$, where 914 days was chosen as it is the median of the marginal survival curve estimated using a Kaplan–Meier estimator.

5.1 Implementation

The deep learning algorithms can be implemented using an end-to-end convolutional neural network; however, the size of the training set is fairly small compared to the number of parameters needed to be estimated when training such networks. Additionally, training multi-layer neural networks from scratch is computationally expensive [29]. A popular approach to overcome these limitations is to use convolutional neural networks that have been pre-trained on large imaging datasets [30]. These pre-trained networks extract image features that are then fed into fully connected layers. Intuitively, the image features extracted are related to shapes that are relevant to imaging based prediction problems (such as corners). Using a pre-trained network allows us to retrieve those relevant features without requiring the massive amounts of data needed to tune the parameters in the convolutional layers.

A popular pre-trained network is the ImageNet VGG16, a convolutional neural network developed by the Oxford Visual Geometry Group as a submission to the Imagenet Challenge 2014. The VGG16 was trained on a variety of augmented natural images and improved accuracy by increasing the depth of the neural network with several small (3×3) convolutional layers. It was furthermore shown to generalize well to other datasets by Ref. [30].

We utilized the VGG16 (configuration D with pretrained weights) to extract image features from the histology ROI images by removing the last 3 fully connected layers of the VGG16 and feeding our images through the convolutional layers with fixed pre-trained weights. Our input of a 1024×1024 RGB ROI histology image returned a tensor of image features ($32 \times 32 \times 512$) after feeding it through the VGG16. The tensor of image features ($32 \times 32 \times 512$) was then flattened and fed into five different methods: LASSO Cox PH regression, Cox PH regression with principal component analysis (PCA), DeepSurv, and fully connected Buckley–James and Doubly Robust neural networks (see the Supplementary Web Appendix for a flowchart of the analysis). Two of these methods fit a classical Cox regression (LASSO, PCA) to the image features and three of these methods (Deep Surv, BJ, DR) used the deep learning algorithm by fitting fully connected neural networks on the image features. Below we will describe how we trained each of these five models.

5.1.1 CoxPH (LASSO and PCA)

After flattening the image feature tensor obtained from VGG16 there were 524,288 features for each participant. To account for the high-dimensionality of the input vector in the Cox PH regression, we considered two analyses (1) using LASSO penalized regression and (2) using principal component analysis to reduce the dimension of the parameters. We fit a constrained Cox PH model using LASSO with the penalization parameter obtained via 5-fold cross validation. Ridge regression was also considered but resulted in an uninformative model that had predicted probabilities all close to 0.5. The LASSO model selected 23 features which were then used in an unconstrained Cox PH model fit on the entire training set (615 patients). For the principal component analysis we used the first 3 principal components that explain 91.4 % percent of variability in the features. Further details on the principal component analysis can be found in the Supplementary Web Appendix. We checked the proportional hazards assumption on the 23 image features using the `cox.zph()` function in the survival package in R. That test did not provide strong evidence that the proportional hazard assumption was violated (p-value of a global test equal to 0.364); thus it is reasonable to expect a method that relies on the proportional hazard assumption to perform well. For the Cox PH PCA model the proportional hazards assumption on the 3 principal components

also did not provide evidence that the proportional hazard assumption was violated (global test p-value equal to 0.999).

5.1.2 DeepSurv

The DeepSurv algorithm [25] is implemented using the TFDeepSurv package in Python available on Github. It uses the negative log partial likelihood from a proportional hazard model as the loss function. We first tried to train DeepSurv on the flattened image features that were obtained by running the histology ROIs through the VGG16 convolutional layers. Despite varying several tuning parameters in DeepSurv, such as the choice of activation function and the number of layers, we could not obtain an informative model based on all the image features. All of the models resulted in the exact same or nearly the same risk score $g(w, \widehat{\beta})$ for all training observations, and hence predicted similar probabilities for all observations in the test set (regardless of the observations' image feature values). Because of that, we fit the DeepSurv algorithm on the 23 features selected by the LASSO Cox PH model. For this model we used the default parameters provided by the TFDeepSurv package https://github.com/liupei101/TFDeepSurv except we changed the *hidden_layer_nodes* to [2, 1]. Thus, our final DeepSurv model was a 2 layer Neural Network (with a hidden layer of width 2) fit on the 23 features selected by the LASSO Cox PH model.

5.1.3 Buckley–James and Doubly Robust fully connected neural networks

For both the Buckley–James and Doubly Robust models, we defined the structure of the neural networks to have $K = 2$ layers of the form $f_\beta(w) = f^{(2)}_{\beta_2} \circ f^{(1)}_{\beta_1}(w)$ where the input (w) was the 524,288 flattened image features and the output returned an estimator for $P(T > 914|W)$. These neural networks used the loss functions $L_{BJ}(\mathcal{O}, \beta, \widehat{S}_0)$ for the Buckley–James method and $L_{DR}(\mathcal{O}, \beta, \widehat{G}_0, \widehat{S}_0)$ for the Doubly Robust method with the full data loss, $L(h(T), \beta(W))$, selected as the Brier loss ($(I(T > 914) - \beta(W))^2$) reflecting that the target parameter is $P(T > t|W)$ (see Supplementary Web Appendix). Both networks used the rectified linear units (ReLU) activation function for the hidden layer, the sigmoid function for the output layer, and Adam as the optimizer (learning rate 1×10^{-6}) [31].

We used L2 regularization for penalization as described by Equation (1) and selected both the penalization parameter λ and the size of the hidden layer (l) using 5 fold cross validation that searched over the grid $\lambda \in \{0.001, 0.01, 0.1, 1, 10\}$ and $l \in \{5, 10, 100, 256, 1000\}$. The values of λ and l were chosen as those with the lowest average cross validated Brier score [32]. As expected, increasing the penalty parameter λ required a larger hidden layer width l to get good model performance. This is reasonable since λ constrains the model complexity and increasing the width l increases the complexity. Our chosen values for λ and l lie in the middle of the ranges of values we considered (see Supplementary Web Appendix). For both the Buckley–James and Doubly Robust algorithms the optimal parameters were $\lambda = 0.01$ and $l = 100$. This architecture resulted in 52,429,001 parameters (weights and biases) being estimated when fitting each model.

For the Buckley–James model, we estimated $E_{\widehat{S}}[L(h(T), \beta(W))|T \geq C, W]$ using the Cox PH regression model fit on the 23 image features. For the Doubly Robust model, we needed to estimate both $E_{\widehat{S}}[L(h(T), \beta(W))|T \geq C, W]$ and $G_0(.|W_i)$. We estimated G_0 with a survival tree [33] and $E_{\widehat{S}}[L(h(T), \beta(W))|T \geq C, W]$ using the Cox PH regression model fit on the 23 image features.

Both the Buckley–James and the Doubly Robust deep learning algorithms are implemented using a form of response imputation as described in Ref. [6] using the Python package Tensorflow.

6 Results

To compare the different methods we used the weighted Brier score [32, 34] and calibration error evaluated on the test set. The weighted Brier score weights each observation by the inverse probability of censoring using the formula:

Table 1: Weighted Brier score (lower is better) and calibration error (lower is better) estimated on a test set for a LASSO Cox proportional hazards model, Cox proportional hazards PCA model, DeepSurv deep learning algorithm, and Buckley–James and Doubly Robust deep learning algorithms.

	Weighted Brier	Calibration error
BJ model	0.203	0.002
Cox PH regression (PCA)	0.207	0.020
DeepSurv	0.232	0.009
DR model	0.213	0.086
LASSO Cox PH regression	0.198	0.045
Random guess (predict 0.5)[a]	0.268	0

[a]Random guess refers to predicting all probabilities as 0.5.

$$\frac{1}{n}\sum_{i=1}^{n}\frac{I(\min(T_i, 914) \leq C_i)}{\widehat{G}_0(\min(T_i, C_i, 914))}[I(\tilde{T}_i > 914) - \widehat{P}(T > 914|W_i)]^2. \tag{2}$$

For implementation of the weighted Brier score, $\widehat{G}_0$ was estimated using a Kaplan-Meier estimator. The motivation behind using the weighted Brier score is that it is an unbiased estimator for the Brier Risk $E[(I(T > t) - P(T > t|W))^2]$ (if the model for $\widehat{G}_0$ is correctly specified). We also used boxplots to visually assess the ability of the models to discriminate between the two classes $T \leq 914$ and $T > 914$.

From Table 1 we see that all five of our models perform reasonably. Among the three methods that utilize fully connected neural networks (BJ, DeepSurv, DR) the Buckley–James method has the best performance with the lowest Brier score and the lowest calibration error. For the Cox regression models, the Cox PH PCA model has higher Brier score and higher calibration error compared to the Buckley–James model. The LASSO Cox model has lower Brier score than the Buckley–James model, but higher calibration error.

The boxplots in Figure 2 show the predicted probabilities for each participant in the test set separated by if a participant survived beyond 914 days, died before 914 days, or if the participant was censored before 914 days. For those censored before 914 days it is unknown whether they survived past 914 days or not. We can see from the boxplots that all methods are fairly good at discriminating between the two classes $T \leq 914$ and $T > 914$.

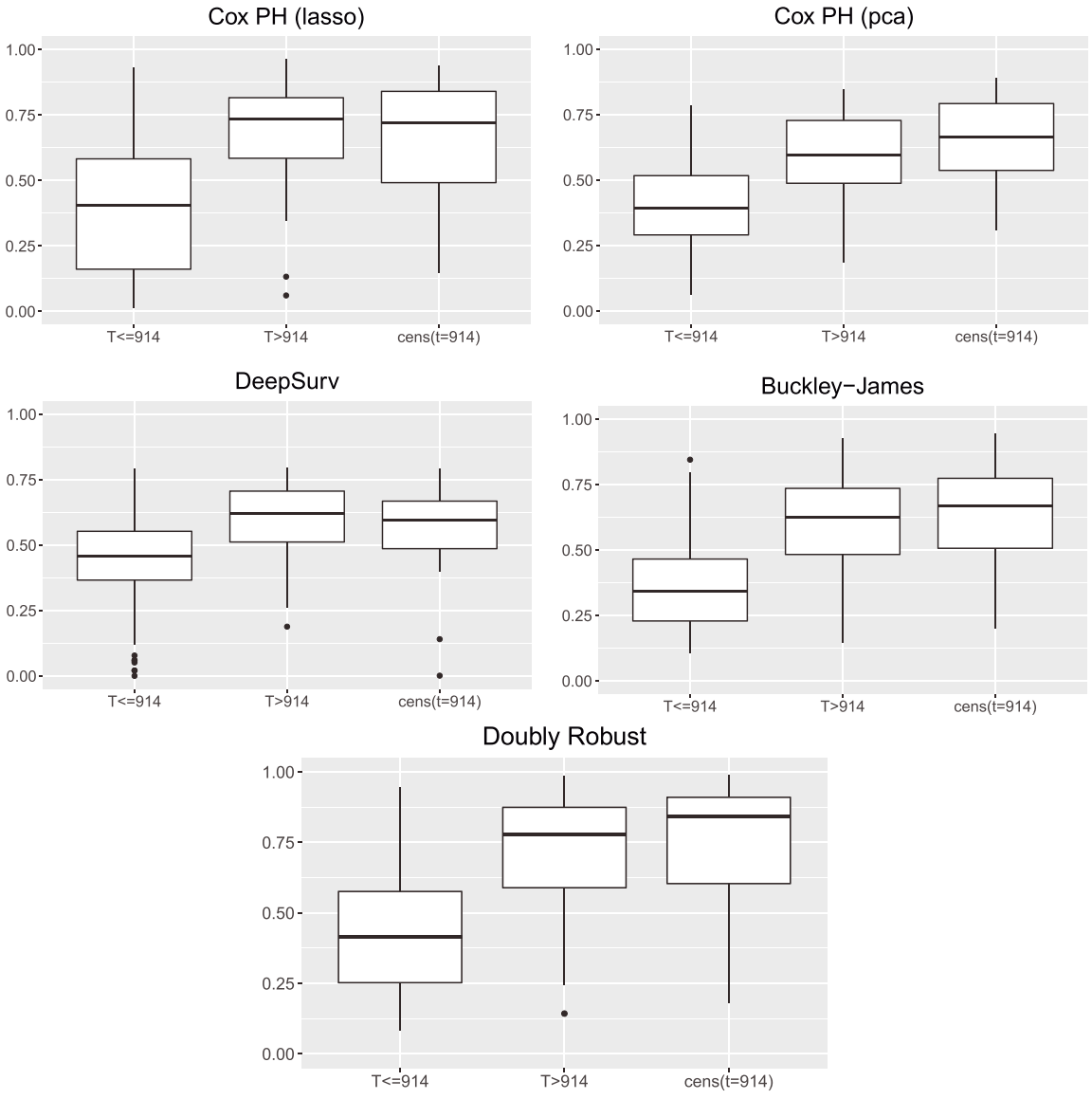

Figure 2: Predicted probabilities of survival past 914 days on the test set grouped by the outcome values. The outcome groups consist of patients who died before 914 days, those who survived past 914 days, and those who were censored before 914 days (denoted cens [$T = 914$]). For those censored before 914 days it is unknown whether they survived past 914 days or not.

7 Discussion

In this manuscript we reviewed and compared several convolutional neural networks methods adapted to right censored outcomes and more traditional Cox models using a glioma biopsy imaging dataset. One difficulty with this analysis was the relatively small number of images (training set consists of 615 images). We used a pre-trained convolutional neural network, VGG16, that was trained on a variety of natural images to create image features that were further utilized by the different methods to create a risk prediction model. VGG16 was trained on a variety of natural images, not medical images, which could impact the image features extracted and model fitting process. Previous published work has studied the impact of using pre-trained natural image convolutional neural networks (CNN) for medical image analysis by comparing built from scratch CNNs to pre-trained CNNs that were fine tuned to the medical dataset [29]. They found deep fine tuning of pre-trained neural networks

often performs as well as or outperforms the networks that are built from scratch. They would consider our analyses with the VGG16 as a type of shallow tuning because we only trained the last two fully connected layers; they suggest the best method is to start with shallow tuning and increase the level of fine tuning to find the best depth for the application [29]. Thus, it is unknown if the VGG16 used was fine tuned to a sufficient depth. To address this question, in a non exhaustive analyses, we looked at fine tuning the last convolutional block and fully connected layers of the VGG16 to our specific histology dataset, a deep level of fine tuning, and found no improvement in performance. Future work could include more extensive studies of the impact of deeper fine tuning on our model.

The glioma dataset contained multiple ROIs per patient. In the absence of censoring [35], suggested to account for multiple ROIs using an EM based model to predict which patches were discriminative and then combine the patch level predictions by a decision fusion model (e.g., multi-class logistic regression) [36]. proposed an algorithm that adaptively sampled patches from the whole slide image, grouped the patches into clusters based on the pixel data, and built a prediction model on the informative patches. The output of the patches was weighted and aggregated into a final patient level prediction.

The analysis randomly selected one ROI per patient; however, in the Supplementary Web Appendix we explored the impact of incorporating the multiple ROIs into our analysis. All ROIs associated with patients in the training set were used for training (the test and training set split was done at the patient level to ensure independence between the test and training sets). The ROI level predictions on the test set were aggregated by using the minimum predicted probability over all ROIs for a particular patient. The prediction accuracy of the methods (LASSO Cox PH regression, Cox PH PCA, DeepSurv, BJ, DR) did not change substantially after accounting for the multiple images (see Supplementary Web Appendix). Future work could include utilizing more sophisticated algorithms to account for the multiple ROIs obtained from the images of each participant.

Deep learning algorithms suffer from lack of interpretability on how the inputs contribute to the final prediction model. To address this lack of interpretability, methods have been developed for convolutional neural networks which produce visual representations (e.g., heat maps/highlighted regions) of important pixels and segments in the input image [37, 38]. However, understanding the complex interactions in the deep learning algorithm still remains challenging. The classical Cox PH regression model has the advantage over the black box deep learning algorithms in that the coefficients are interpretable and quantification of uncertainty in parameter estimation is easy. When pre-trained networks such as VGG16, Xception, ResNet, or ResNet50 are used to create image features that then are used as inputs into the Cox model, interpretability of the Cox model parameters is less beneficial. This is because the features are outputs from complex convolutional neural networks and are therefore not easily interpretable from the original image. DeepSurv differs from the pre-trained networks + Cox PH since all parts of the model are neural network layers. There have already been developments to incorporate convolutional neural networks and the DeepSurv framework together into one neural network called DeepConvSurv, however software is not currently available [10].

The censoring unbiased loss function based methods (Buckley–James and Doubly Robust) also consist solely of neural network layers but differ in that they do not rely on the proportional hazard assumption. Although a Cox PH model can be used to get an initial estimator of the survival function, various other estimators that do not rely on the PH assumption (e.g., from AFT model, random survival forests, survival trees) can be used to implement the censoring unbiased loss function methods. The main difference between the Buckley–James and the Doubly Robust loss functions is in their statistical properties and thus their required estimators. To implement the Buckley–James method, an estimator for the survival function is needed, while the Doubly Robust method requires estimating both the survival curve and the censoring survival curve. Theoretically, the Doubly Robust method is more robust to misspecification of those estimators.

Author contributions: All the authors have accepted responsibility for the entire content of this submitted manuscript and approved submission.
Research funding: This work was supported by the National Institute of General Medical Sciences (U54GM115677) and National Cancer Institute (U10CA180794, U10CA180820).
Conflict of interest statement: The authors declare no conflicts of interest regarding this article.

References

1. Klein JP, Moeschberger ML. Survival analysis: techniques for censored and truncated data. New York: Springer Science & Business Media; 2006.
2. Zhao L. Deep neural networks for predicting restricted mean survival times. Bioinformatics 2020;36:5672—7.
3. Sun Y, Chiou SH, Wang MC. Roc-guided survival trees and ensembles. Biometrics 2020;76:1177—89.
4. Bou-Hamad I, Larocque D, Ben-Ameur H. A review of survival trees. Stat Surv 2011;5:44—71.
5. Cui Y, Zhu R, Kosorok M. Tree based weighted learning for estimating individualized treatment rules with censored data. Electron J Stat 2017;11:3927.
6. Steingrimsson JA, Morrison S. Deep learning for survival outcomes. Stat Med 2020;39:2339—49.
7. Litjens G, Kooi T, Bejnordi BE, Setio AAA, Ciompi F, Ghafoorian M, et al. A survey on deep learning in medical image analysis. Med Image Anal 2017;42:60—88.
8. Mobadersany P, Yousefi S, Amgad M, Gutman DA, Barnholtz-Sloan JS, Vega JEV, et al. Predicting cancer outcomes from histology and genomics using convolutional networks. Proc Natl Acad Sci USA 2018;115:E2970—79.
9. Li H, Boimel P, Janopaul-Naylor J, Zhong H, Xiao Y, Ben-Josef E, Fan Y. Deep convolutional neural networks for imaging data based survival analysis of rectal cancer. In: 2019 IEEE 16th international symposium on biomedical imaging (ISBI 2019). IEEE; 2019:846—9 pp.
10. Zhu X, Yao J, Huang J. Deep convolutional neural network for survival analysis with pathological images. In: 2016 IEEE international conference on bioinformatics and biomedicine (BIBM). IEEE; 2016:544—7 pp.
11. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, et al. Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2015:1—9 pp.
12. Pedano N, Flanders A, Scarpace L, Mikkelsen T, Eschbacher J, Hermes B, Ostrom Q. Radiology data from the cancer genome atlas low grade glioma [tcga-lgg] collection. The Cancer Imaging Archive; 2016. Available from: https://www.cancerimagingarchive.net/.
13. Clark K, Vendt B, Smith K, Freymann J, Kirby J, Koppel P, et al. The cancer imaging archive (tcia): maintaining and operating a public information repository. J Digit Imag 2013;26:1045—57.
14. Goodfellow I, Bengio Y, Courville A. Deep Learning. MIT Press; 2016. Available from: http://www.deeplearningbook.org.
15. LeCun Y, Bengio Y. Convolutional networks for images, speech, and time series. In: The handbook of brain theory and neural networks. Cambridge: MIT Press; 1995, 3361:1995 p.
16. LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. Proc IEEE 1998;86:2278—324.
17. Caterini AL, Chang DE. Deep neural networks in a mathematical framework. New York: Springer; 2018.
18. Murphy SA, Van der Vaart AW. On profile likelihood. J Am Stat Assoc 2000;95:449—65.
19. Lao J, Chen Y, Li ZC, Li Q, Zhang J, Liu J, et al. A deep learning-based radiomics model for prediction of survival in glioblastoma multiforme. Sci Rep 2017;7:10353.
20. Friedman J, Hastie T, Tibshirani R. Regularization paths for generalized linear models via coordinate descent. J Stat Software 2010;33:1.
21. Aerts HJ, Velazquez ER, Leijenaar RT, Parmar C, Grossmann P, Carvalho S, et al. Decoding tumour phenotype by noninvasive imaging using a quantitative radiomics approach. Nat Commun 2014;5:1—9.
22. Davatzikos C, Rathore S, Bakas S, Pati S, Bergman M, Kalarot R, et al. Cancer imaging phenomics toolkit: quantitative imaging analytics for precision diagnostics and predictive modeling of clinical outcome. J Med Imag 2018;5:011018.
23. Pati S, Singh A, Rathore S, Gastounioti A, Bergman M, Ngo P, et al. The cancer imaging phenomics toolkit (captk): technical overview. In: International MICCAI brainlesion workshop. Springer; 2019:380—94 pp.
24. Faraggi D, Simon R. A neural network model for survival data. Stat Med 1995;14:73—82.
25. Katzman JL, Shaham U, Cloninger A, Bates J, Jiang T, Kluger Y. DeepSurv: personalized treatment recommender system using a cox proportional hazards deep neural network. BMC Med Res Methodol 2018;18:1—12.
26. Yousefi S, Amrollahi F, Amgad M, Dong C, Lewis JE, Song C, et al. Predicting clinical outcomes from large scale cancer genomic profiles with deep survival models. Sci Rep 2017;7:11707.
27. Steingrimsson JA, Diao L, Strawderman RL. Censoring unbiased regression trees and ensembles. J Am Stat Assoc 2019;114:370—83.
28. Steingrimsson JA, Diao L, Molinaro AM, Strawderman RL. Doubly robust survival trees. Stat Med 2016;35:3595—612.
29. Tajbakhsh N, Shin JY, Gurudu SR, Hurst RT, Kendall CB, Gotway MB, et al. Convolutional neural networks for medical image analysis: full training or fine tuning? IEEE Trans Med Imag 2016;35:1299—312.
30. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
31. Ruder S. An overview of gradient descent optimization algorithms. 2016. *arXiv preprint arXiv:1609.04747*.
32. Graf E, Schmoor C, Sauerbrei W, Schumacher M. Assessment and comparison of prognostic classification schemes for survival data. Stat Med 1999;18:2529—45.
33. LeBlanc M, Crowley J. Survival trees by goodness of split. J Am Stat Assoc 1993;88:457—67.
34. Lostritto K, Strawderman RL, Molinaro AM. A partitioning deletion/substitution/addition algorithm for creating survival risk groups. Biometrics 2012;68:1146—56.

35. Hou L, Samaras D, Kurc TM, Gao Y, Davis JE, Saltz JH. Patch-based convolutional neural network for whole slide tissue image classification. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2016:2424—33 pp.
36. Zhu X, Yao J, Zhu F, Huang J. Wsisa: making survival prediction from whole slide histopathological images. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2017:7234—42 pp.
37. Ribeiro MT, Singh S, Guestrin C. "Why should I trust you?" Explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining; 2016:1135—44 pp.
38. Wagner J, Kohler JM, Gindele T, Hetzel L, Wiedemer JT, Behnke S. Interpretable and fine-grained visual explanations for convolutional neural networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition; 2019:9097—107 pp.

Supplementary Material: This article contains supplementary material (https://doi.org/10.1515/ijb-2022-0113).