A deep convolutional neural network approach for predicting cumulative incidence based on pseudo-observations

Pablo Gonzalez Ginestet[1 *], Philippe Weitz[1] , Mattias Rantalainen[1] , Erin E Gabriel[1,2] ,

1 Department of Medical Epidemiology and Biostatistics, Karolinskta Institutet, Stockholm, Sweden
2 Section of Biostatistics, Department of Public Health, University of Copenhagen, Denmark.

* pablo.gonzalez.ginestet@ki.se

Abstract

Predictions of cumulative incidence are useful in many areas of medical research, and recent works have suggested that including information from medical images can improve these predictions. Medical images have been used to predict time-to-event outcomes by applying deep convolutional neural network (CNN) methods using a Cox partial likelihood loss function under the assumption of proportional hazards. This, however, gives a single prediction for all time points and enforces the proportional hazards assumption, reducing the flexibility of the algorithm. We propose a method to predict the cumulative incidence from images and structured clinical data using pseudo-observations as the response variable in deep CNNs for use with right-censored outcome data. There is a theoretical backing for the use of pseudo-observations to replace the right-censored response outcome, and this allows for algorithms and loss functions designed for continuous and uncensored data to be used. The performance of our proposed method is assessed in simulation studies and a real data example in breast cancer from The Cancer Genome Atlas (TCGA). The results are compared to the existing state-of-the art deep CNN with Cox loss. The results demonstrated that our proposed method outperformed the Cox loss in large sample settings, and performed similarly in small sample settings. Our proposed method provides a flexible alternative for the application of deep CNN methods to right-censored data and allows for the use of simple and easy to modify loss functions.

Introduction

It has recently been demonstrated that contemporary medical image analysis has the potential to improve the diagnostic and prognostic stratification of cancer patients [1–3]. In particular, the analysis of microscopic morphological patterns in histopathological tissue sections is a key component of routine care. For instance, in lung cancer, tumors with predominantly micro-papillary and solid patterns have been associated with a poorer prognosis [4]. With the advent of digital pathology, whole slide images (WSIs) of stained tissue sections are becoming increasingly available. This may provide the opportunity to more accurately predict individual prognoses using image data paired with other clinical information and provide clinicians and patients with decision support [1, 2].

Deep convolutional neural networks (CNNs) are currently at the forefront of image analysis and have become the state of the art in image-based precision medicine [5,6]. Deep CNN models are neural networks with several layers, including convolutional layers that are suitable for modelling of image data. Deep CNNs learn hierarchical representations directly from raw image data given a large dataset of labeled examples.

Few machine learning methods have been developed for survival outcomes originally, and thus, most existing machine learning methods for survival outcomes are adaptations. This is also true for image analysis methods. CNN methods have been used and adapted to address the task of predicting time-to-event outcomes from WSIs. Recent works [7–13] have used WSIs with CNN for survival predictions. They applied convolutional layers to extract features of the images using convolutional kernels and pooling operations, followed by a sequence of fully connected layers where the terminal layer outputs a predicted risk associated with the image. These risks are plugged into the Cox partial likelihood and the network is trained using a back-propagation procedure and optimization algorithm. These prior works combined modern CNN models with Cox regression for prediction of time-to-event outcomes, keeping the assumption of proportional hazards. This stipulates no effect modification by time, which can be restrictive or even unrealistic [14]. Furthermore, the negative partial log-likelihood is a relatively complicated loss function that can be challenging to implement in existing CNN frameworks, and is difficult to modify.

Recent works in other areas of machine learning have suggested data pre-processing steps that can be used to adapt common classes of machine learning methods to time-to-event outcomes [15–17]. Vock et al. [15] introduced the pre-processing step of reweighting the data using inverse probability of censoring weighting (IPCW). IPCW is a well-established technique for dealing with censored data [18] that reweights those individuals whose event is not missing by the inverse probability of remaining uncensored to represent the individuals whose event is not known. This pre-processing step creates a weighted dataset that can be analysed by any classification machine learning technique that can incorporate weights. Gonzalez Ginestet et al. [16] generalized this approach combining IPCW and bagging. The weights are incorporated in the resampling step and not in the algorithm as [15] and thus eliminating the need to directly adapt any of the ML algorithms. Zhao and Feng [17] used a conditional pseudo-observation approach as a pre-processing step. Just like the classical pseudo-observation developed by [19], this pre-processing step transforms the censored survival dataset into one with a new quantitative response variable enabling the application of multivariate regression. Zhao and Feng [17] replaced the observed survival times by jackknife pseudo survival probabilities in a discrete-time survival framework and used them as the new response variable in a simple neural network model. By using pseudo-observations, this work avoided the sophisticated loss functions for censored data or the proportional hazards assumption from previous work that modeled survival data using deep neural networks [20–24].

Similar to [17], we perform the pre-processing step of calculating pseudo-observations and then implementing existing CNN methods. However, we propose to combine classical pseudo-observations [19,25], instead of the conditional ones used in [17], with CNN models in order to make risk predictions based on medical images and clinical covariates in a context of right-censored outcome data. Moreover, we adapt the pseudo-observation approach in a continuous time and we also consider a network with multiple outcomes, one per each time point, in addition to including the the time points of interest as predictor in the single output network in a similar manner to [17].

The use of pseudo-observations as a pre-processing step enables us to avoid the proportional hazards assumption and any special tailored loss function to handle right-censoring that was made by recent previous works [7–12] that used medical images with CNN for survival predictions. We demonstrate our method in simulations based on the CIFAR-10 images [26] and a real data example in breast cancer from The Cancer Genome Atlas Breast Invasive Carcinoma data [27]. Though the results are based on one specific CNN model,the Residual Network model with 18 layers [28], our proposed method can be used with any CNN model, and thus, applies to all those publicly available in Pytorch [29] or TensorFlow [30].

Materials and methods

Setup and notation

Let $T^{(m)}$ denote the true event time for an individual m and $C^{(m)}$ the censoring time, $\tilde{T}^{(m)} = \min\{T^{(m)}, C^{(m)}\}$ the observed survival time and event indicator $\Delta^{(m)} = 1(T^{(m)} \leq C^{(m)})$. In addition, for each individual we observe a p-dimensional vector of clinical covariates at baseline $\mathbf{X}^{(m)}$ and a three-dimensional image data denoted as $\mathbf{I}^{(m)}$. Each image data is a three dimensional array of size $w \times h \times d$, where w and h are spatial dimensions and d is the channel dimension, where color images have three channels (red, green and blue (RGB)).

Without censoring, the sample data would be $\mathcal{D}^{ideal} = \{(\mathbf{I}^{(m)}, \mathbf{X}^{(m)}, T^{(m)}, y_\tau^{(m)})\}$ for $m = 1, \ldots, N$ where $y_\tau^{(m)}$ is the response variable for individual m indicating if the individual has experienced the event at a specific time τ, $y_\tau^{(m)} = 1(T^{(m)} \leq \tau)$. The goal is to predict the individual risk of experiencing the main event before time τ given his or her information based on sample data $\mathcal{D}^{ideal}$. However, in the presence of censoring, the response variable $y_\tau^{(m)}$ is not observed for all m. Instead, we observe the sample data $\mathcal{D} = \{(\mathbf{I}^{(m)}, \mathbf{X}^{(m)}, \tilde{T}^{(m)}, \tilde{y}_\tau^{(m)})\}$ for $m = 1, \ldots, N$ where $\tilde{y}_\tau^{(m)} = \Delta^{(m)} 1(\tilde{T}^{(m)} \leq \tau)$.

Convolutional Neural Network

A convolutional neural network is a specialized neural network for processing image data. A CNN is typically composed of three types of layers: convolution, pooling, and fully connected layers. We give an overview of these concepts and we refer the readers to [5, 31] for a detailed exposition.

The convolutional layer performs feature extraction by applying a combination of linear and nonlinear operations. The linear operation is the convolution or cross-correlation operation where a weight matrix, also called convolution kernel or filter, is used to compute an element-wise product between each element of the kernel and the input. This operation is repeated spatially sliding the kernel over the input. The kernel contains learnable parameters that are shared across all positions of the input. The output of the cross-correlation operations is passed through a non-linear transformation, called activation function, such as Rectified Linear Unit (ReLU) defined as $ReLU(x) = max(0, x)$.

The pooling layer aggregates information over a neighborhood defined by a fixed shape window that is slid over the input. It is typically applied after a convolution layer and it does not contain any new parameters to learn. It serves the purpose of mitigating the sensitivity of convolutional layers to location and improving computational efficiency due to the fact that the next layer has a lower dimension input to process. Two common pooling operations are maximum pooling and average pooling that output the maximum and average value in each neighborhood, respectively.

The last layers of the CNN architectures are usually fully connected (FC) layers which correspond to classical multilayer perceptron. The output of the final convolution or pooling layer is flattened and connected to one or more FC layers where each input is connected to the output by learnable weights. Each fully connected layer is followed by a nonlinear function. The number of outputs of the final FC layer is equal to the number of classes. In this work, there are two classes: $y_\tau^{(m)} = 1$ or $y_\tau^{(m)} = 0$.

Combining CNN with Cox regression (CoxCNN)

The work of [20] was one of the first to combine Cox regression with a simple neural network to predict time-to-event outcomes from clinical data. This idea was revisited by [21] adapting a deep neural network and showing that the neural network generalization of the proportional hazards model was able to outperform classical Cox regression. The Cox regression approach was then extended to images. The following works [7–9] combined CNN models with a Cox proportional hazards model to predict time-to-event data from medical images. Mobadersany et al. [7] used elements of a type of CNN model developed in [32]. Li et al. [9] proposed their own CNN model, instead of relying on a publicly available architecture. Wulczyn et al. [8] used a similar CNN model to the MobileNet [33]. The predicted risk associated with the medical image that is output at the terminal FC layer is plugged into the negative partial log-likelihood that is used as a loss function to handle censored data. This loss function has been written in many different ways in the literature, but we reproduce here the equation given in [34]

$$L(risk) = -\sum_{i=1}^{N} \Delta^{(i)} \big(risk_i - \log \sum_{j \in R_i} \exp(risk_j) \big) \tag{1}$$

where R_i is the risk set at time $\tilde{T}^{(i)}$ that includes those who have survived at least to time $\tilde{T}^{(i)}$; $R_i = \{1 \leq j \leq N : \tilde{T}^{(j)} \geq \tilde{T}^{(i)}\}$. The CoxCNN is trained using the observed sample data $\mathcal{D} = \{(\mathbf{I}^{(m)}, \mathbf{X}^{(m)}, \tilde{T}^{(m)}, \tilde{y}_\tau^{(m)})\}$ paired with an optimization algorithm such as Adam or Adagrad that minimizes the loss function.

Pseudo-Observations

Andersen et al. [19] introduced a strategy to transform a censored problem into an uncensored one in order to be able to apply standard methods for complete data such as regression models. If y_τ were not subject to censoring, we could use it directly to model the cumulative incidence. In the presence of censoring, the pseudo-observation approach replaces the censored response variable $y_\tau^{(m)}$ of each individual m by a jackknife pseudo-observation, which can be used as a new response variable to fit models. Pseudo-observations can be based on a number of estimators. We will use the Kaplan-Meier (KM) estimator of the cumulative incidence of failure before time τ. However, other estimators of the unconditional cumulative incidence could be used. In the absence of competing risks, the cumulative incidence of the event of interest is given by $\theta(\tau) = 1 - S(\tau)$, where $S(\tau)$ is the survival function. The pseudo-observation (PO) cumulative incidence for individual m at time τ is computed as

$$\hat{\theta}^{(m)}(\tau) = N \times \hat{\theta}(\tau) - (N-1) \times \hat{\theta}^{(-m)}(\tau) \tag{2}$$

where $\hat{\theta}(\tau) = 1 - \hat{S}_{KM}(\tau)$ and $\hat{S}_{KM}(\tau)$ is the the KM estimator of the survival function based on all the examples and $\hat{\theta}^{(-m)}(t) = 1 - \hat{S}_{KM}^{(-m)}(\tau)$ is obtained by eliminating individual m from the data. The PO are used as a replacement for the incompletely observed random variable $y_\tau^{(m)}$ for each individual. The asymptotic justification of the pseudo-observation approach requires that $\hat{\theta}(\tau)$ be a consistent estimator of $\theta(\tau)$, and that the right-censoring be independent of the survival time and any covariates one intends to include in the model [35].

In cases where the censoring is potentially dependent on covariates, one can model the censoring and use inverse probability of censoring weighted (IPCW) methods to consistently estimate the survival function [18,36]. In order to perform IPCW, one estimates the conditional censoring survival function at time τ, denoted by

$$G^{(m)}(\tau) = P(C^{(m)} > \tau \mid \mathbf{X}^{(m)}), \tag{3}$$

and weights the estimator by the inverse of the estimated probability at the minimum of τ and their censoring time.

The weighted pseudo-observation, IPCW-PO, cumulative incidence for individual m at time τ replaces $\hat{S}_{KM}(\tau)$ by $\hat{S}^W(\tau)$ in Equation 2 where $\hat{S}^W(\tau)$ is defined as in [37]. The censoring survivor function $G(\cdot)$ is typically unknown and needs to be estimated. Appropriate procedures to estimate $G(\cdot)$ are the Cox proportional hazards model [38] and more flexible models such as Aalen's linear hazard model [39], boosted Cox regression [40] or random forest [41]. We applied the R package *eventglm* [42] to estimate the IPCW-PO using a Cox's proportional hazards model for the censoring weights.

Once pseudo-observations are obtained, the sample data for the analysis $\mathcal{D}_{\mathcal{PO}} = \{(\mathbf{I}^{(1)}, \mathbf{X}^{(1)}, \hat{\theta}^{(1)}(\tau)), \ldots, (\mathbf{I}^{(N)}, \mathbf{X}^{(N)}, \hat{\theta}^{(N)}(\tau))\}$ can be used to train any CNN model to predict the individual risk of experiencing the main event before time τ, which would have been similar to basing our predictions on $\mathcal{D}^{ideal}$ if this sample data were available. Note that $\mathcal{D}_{\mathcal{PO}}$ that uses IPCW-PO includes all individuals (those with unknown and known time of the event) as observations in the analysis. This contrast the IPCW approach in [15] and [16] where only individuals who are not censored are included as observations in the weighted analysis.

Pseudo-Observation (PO) CNN

Our proposed POCNN procedure enables a CNN to be fitted using a PO-based response to predict the cumulative incidence from images and structured clinical data handling right-censoring without resorting to the Cox partial likelihood. The pipeline of the proposed framework is shown in Figure 1 and Figure 2 and can be summarized as follows.

i) For a finite number of time points, compute the PO (or IPCW-PO) cumulative incidence for each individual using $\mathcal{D}$ to construct $\mathcal{D}_{\mathcal{PO}}$.

ii) Choose a CNN model and add additional fully connected layers at the terminal layer of the CNN and the clinical data as intermediate input for multiple outputs (Figure 2.d) OR include the time points as input to for a single output implementation (Figure 2.c).

iii) Train the POCNN (or IPCW-POCNN) using a mean squared error (MSE) based loss function of your choice.

The implementation that only includes the clinical data as intermediate input is a multi-output regression, which is related to multi-task supervised learning approach [43]. In Figure 2.d, each output at a different time point is regarded as a specific output and each of them have several task-specific layers while sharing all previous layers. Thus, there is no need to add the time points as intermediate predictors. We denote this implementation as multi-output. Unlike the single output implementation, the multi-output minimizes the combined MSE of each output values together. Although we simply sum the different losses, this can be tailored as desired. The single output loss function can also be tailored as desired to more highly weight a particular time point, or can only use a single time point. Although we do not investigate this further, this simple modification of the loss function may be of great advantage over the existing Cox-loss methods, which is much more difficult to modify in a targeted manner. In what follows, we use the default average MSE over all included time points in both POCNN approaches.

It is of note that although it is technically possible to use the image information in the fitting of the censoring model, we do not believe this is practical or necessary. Instead, fitting a model for the censoring distribution based solely on the set of available clinical covariates is likely sufficient and much more feasible in practice. Thus, we assume that it is sufficient to condition on the clinical covariates when modelling the censoring in the IPCW-PO. This is slightly more restrictive than a CoxCNN treatment of censoring as all inputs in the CNN are included in some way in the final layer and thus are accounting for censoring. We investigate this in the simulations in case 3 and 6 where censoring depends on the image information.

Fig 1. Pseudo-observations cumulative incidence are computed at a finite number of time points for each individual to be used as the new response variable

Fig 2. (b) Medical images are passed-through the CNN model chosen; (c) single output, and (d) multi-output.

Results

For all methods that follow, POCNN and CoxCNN, we used a Residual Network model [28] with 18 layers (ResNet18), although any model could be used. The typical block of layer in ResNet is i) convolution; ii) non-linearity activation function; iii) batch normalization; iv) pooling and v) dropout. We used a pre-trained ResNet18 model on ImageNet [44, 45]. We replaced the last 1000-category linear layer used in ImageNet with a linear layer with a single node. The output of this new linear layer is paired with the structured data (clinical covariates and/or time points) and used as input in a new linear layer added with a single node. The LeakyReLU activation function is used in the final layer in POCNN, single and multi output, to allow small negative values. On the other hand, a linear activation function with no intercept is applied in the CoxCNN. We froze all previous existing ResNet layers and, in the training step, we only updated the parameters in these two new linear layers. Table 1 summarizes the implementation of each method. We used Pytorch for the CNN implementation with Adam [46] as optimizer. To obtain the PO and IPCW-PO, we used the R packages prodlim [47] and eventglm [42], respectively, where the IPC weights were estimated using a Cox regression model based on the full set of clinical covariates. Code containing details of all procedures is available at `https://github.com/pablogonzalezginestet/POCNN`.

Simulations based on CIFAR-10 dataset

Previous works [9, 12] have used greyscale images from MNIST handwritten digit database to generate a simulated dataset to evaluate their methods. In this experiment, we used images from the CIFAR-10 dataset as the basis of our simulated data. The CIFAR-10 dataset consists of colour images ($32 \times 32 \times 3$) in 10 classes. We denote the classes with y where $y_i \in \{0, \dots, 9\}$ is the class for the image I_i. We generated the true survival time based on the classes that each image represents as well as independent covariates. We generated nine independent covariates $X_1, \dots, X_9$ from the standard normal distribution and one binary covariate X_{10} from a binomial distribution as the full set of clinical covariates. We presented six cases corresponding to different survival and censoring time models. The six cases are as follow:

Case 1. The true survival time was generated from a proportional hazards model. T was generated with hazard function:

$$\lambda_T(t \mid y, X) = \lambda_{T,0}(t) \exp\big\{1.7y + (0.3 + 0.6\cos(y))X_{10} + 0.2X_1\big\}$$

where $\lambda_{T,0}(t) = 2t$. We randomly selected around 30% observations to be right-censored at time C generated from a uniform distribution on $(0, T)$.

Case 2. The true survival time was generated under a proportional hazards model as Case 1 but the censoring time is generated from

$$\lambda_C(t \mid X) = \lambda_{C,0}(t) \exp\big\{1.4X_{10} + 2.6X_1 - 0.2X_2\big\}$$

where $\lambda_{C,0}(t) = 12t$. The censoring percentage is around 20%.

Case 3. The true survival time was generated from a proportional hazards model where

$$\lambda_T(t \mid y, X) = \lambda_{T,0}(t) \exp\big\{y - 1.6\cos(y)X_{10} + 0.3X_1X_{10}\big\}$$

and $\lambda_{T,0}(t) = 0.7t$. The censoring time was generated using a gamma distribution with shape parameter equal to $\exp\big\{-1.8X_{10} + 1.4X_1 + 1.5X_{10}X_1\big\}$ and scale parameter equal to y. The censoring percentage is around 43%.

Case 4. The true model for survival time was generated using a gamma distribution with shape parameter equal to $\exp\big\{0.5y + 0.2X_{10}\cos(y) + 1.5X_1 + 1.2X_{10}\big\}$. We randomly selected 30% observations to be right-censored at time C generated from a uniform distribution on $(0, T)$.

Case 5. The true survival times are non-proportional hazards as *Case 4* and the censoring time was generated

$$\lambda_C(t \mid X) = \lambda_{C,0}(t) \exp\big\{-3.4X_{10} + 0.6X_1 - 2.2X_2\big\}$$

where $\lambda_{C,0}(t) = 0.01t$. The censoring percentage is around 60%.

Case 6. The true survival times and the censoring times are both generated using a gamma distribution. The shape parameter is $\exp\big\{0.7y + 0.4X_{10}y - 0.1X_1X_{10} + 0.1yX_1\big\}$ and $\exp\big\{3.8X_{10} + 5.2X_1 - 3.3X_{10}X_1\big\}$ for the survival and censoring time, respectively. The scale parameter was set equal to y in both censoring and survival models. The censoring percentage is around 65%.

To make the involvement of the images in the simulation concrete one could think of them as tumors. In the first three cases higher digits could be thought as a more deadly tumors and in the last three cases lower digits are related to more deadly tumors (see S7 Appendix for Kaplan-Meier curves stratified by image's classes). For each case, we consider a sample size of $N = 1000$ and 5000. From each sample size, 80% of the observations were randomly sampled for training, while the remaining 20% were set aside as a test set. The simulations were repeated 100 times each. The accuracy of the prediction of the cumulative incidence at the four percentiles observed times was assessed using the area under the ROC curve (AUC). The prediction at each time point were compared to the true binary outcome of having an event prior to a given time point of interest. The AUC in the simulations is based on the uncensored true binary indicator, since we know the exact survival time. The PO and IPCW-PO were computed for a grid of time points corresponding to 20th, 30th, 40th and 50th percentiles of the overall time distribution. We did not tune any hyper-parameters. All simulations were run using a learning rate of 0.01. Comparisons of performance values used Wilcoxon signed-rank test with CoxCNN as reference. Two sided p-values were reported.

Figure 3 and Figure 4 show the simulation results. The learning curves of each model across cases and sample sizes are provided in the S6 Appendix. Our proposed method POCNN, single or multi-output and weighted or unweighted, showed better performance than CoxCNN when censoring was independent (case 1 and case 4). Under presence of dependent censoring (case 2 and case 5), where correct modeling of the censoring mechanism resulted in better performance than CoxCNN. The difference in accuracy between POCNN's models and CoxCNN were larger when the true survival time was generated under a proportional hazard model and for earlier time points. Though in small sample size CoxCNN performed comparably to some POCNN's models (there were no significant differences in case 5), POCNN's models showed better performance when sample size was increased, with significant gains in performance for all cases other than 6.

When censoring time followed a non-proportional hazard model (case 3 and case 6), unweighted POCNN outperformed CoxCNN and IPCW-POCNN in the large sample setting. The weighted POCNN exhibited the worst performance in case 6, and had high variability in case 5 and case 3. The poor behaviour was expected in cases 3 and 6 because the censoring model is misspecified, but the high variability in case 5 where censoring could have been correctly modelled using the assumed Cox censoring model. A different method of including the weights in survival estimator might improve variability, and this is a topic for future investigation.

Over all sample sizes and time points, simulations highlighted that single and multi output performed similarly, though single output demonstrated slightly better accuracy in small sample setting. Although IPC-weighting seems to improve results slightly when the censoring is dependent and the censoring model is correct in larger sample sizes and overall in smaller sample sizes, the potential losses due to incorrect modelling, as demonstrated in case 6 and sample size 5000, likely makes chasing these minor improvements inadvisable. Therefore, based on the simulations, one would expect that there is potential gains in predictive accuracy using either the unweighted single output or multi-output POCNN over the CoxCNN, especially when one has access to a large dataset.

Real data application in breast cancer

We illustrate our proposed method using whole-slide histopathology images of breast tumors and clinical structured data obtained from The Cancer Genome Atlas Breast Invasive Carcinoma (TCGA-BRCA) [27]. The event of interest was time to death from first diagnosis of breast cancer at four time points: 2 years, 3.5 years, 5 years and 8 years. We implemented our proposed method by computing pseudo-observations for these four time points. We selected the following clinical predictors from the clinical data: race, ethnicity, age, pathologic stage and molecular subtype [48, 49]. Table 2 summarizes the predictor variables and event time-to-death, measured by days, and the vital status of the patient, if they are alive/censored (status=0) or dead (status=1).

We used the TCGA breast histopathology image datasets as it was pre-processed in these papers [50, 51]. For a brief explanation of the pre-processing steps see S4 Appendix. The image datasets are composed of 710 WSIs and all of them contain invasive breast cancer. Each WSI was tiled into image patches that span 598 x 598 pixels at 20X magnification. Tiling WSIs into smaller image patches and assigning the patient-level label to each image patch is a common strategy in digital pathology due to current memory constraints. Each WSI, which is associated to a unique patient, is linked to the clinical data. Patients were divided randomly into training (64%), validation (16%) and test (20%) data sets, respectively. The number of tiles in the train/validation/test was 4,494,472/1,061,601/1,417,900, respectively. Due to differences in tumor size and variations in the sectioning, patients have differing numbers of tiles. To sample equivalent numbers of tiles per patient, we decided to augment the original number of tiles of all patients to the extent of balancing the number of tiles per patient.

We performed data augmentation as a form of regularization, including random horizontal flip and random rotation from -90º to 90º. For all models, we only tuned the learning rate using the package Ray Tune [52, 53] for a maximum of 30 epochs in each trial. The mean absolute error was used as evaluation metric in the validation set for POCNN and IPCW-POCNN, whereas the average of the AUC for each time point was used for the CoxCNN. We trained the CNN model on per-tile basis. The final per-slide prediction, which is our interest, was obtained by applying a tile aggregation method. We considered the average and the 75th percentile of the per-tile scores across all tiles as a patient-level prediction. For evaluation in the test dataset, we used the time-dependent area under the ROC curve (AUC) for right-censored time-to-event data [54, 55]. The AUC is estimated at the four different time points of interest. See S5 Appendix for the learning curves of each model.

Results using the averaging procedure are presented in Figure 5. The results obtained using the averaging procedure are similar to those using the 75th percentile aggregation procedure, which are reported in the S1 Appendix. Our proposed method POCNN single output, weighted and unweighted, had the best performance for prediction at all time-points except for the earliest time where POCNN multi output resulted in better accuracy. CoxCNN performed better than POCNN multi output for the last two time points. One explanation for the fall in the accuracy of the weighted version of both POCNN at time 3.5 years could be explained by the misspecification of the censoring model at that time point. In S3 Appendix, we provided heatmap visualizations of risk predictions of POCNN single output model at the four time-points generated by the patch-level model from three patients. As it was expected, the risk increases over time.

Figure 6 shows Kaplan-Meier curves for survival within risk groups defined by quartiles of the distribution of the predicted risk of death at the four time-points for the CoxCNN and POCNN single output. The log-rank test for comparing the survival curves as well as the survival curves for the rest of the models are reported in S2 Appendix. The models showed some discrimination. For instance, POCNN were able to distinguish between group one (Q1) and four (Q4), while CoxCNN group three (Q3) and four (Q4). Using either method, we are less able to distinguish between the two intermediate groups, Q2 and Q3 (see S2 Appendix).

Discussion

Improved prognostic models, including those based on routine histopathology image data, are of high clinical relevance as they can provide information that is important for decision making. The proposed method, based on pseudo-observations, provides an efficient approach to fit deep CNN models to right-censored time-to-event outcomes using standard loss functions, making implementation straightforward while providing similar or improved model performance in comparison to Cox-based approaches. We have shown this by evaluating a large set of simulated scenarios and a real data example.

The results in the real data example are consistent with the results found in the simulation for a small sample size, but are highly variable over the time points of interest. This may be because individual tiles used to train the model in the real application are not discriminative at earlier time points and are biasing the predictions [56]. It may be possible to improve this by modifying a loss function to more highly weight time points with better discrimination, something that is much more easily implemented using the POCNN approaches than the CoxCNN.

Despite the fact that training a CNN model requires a large amount of images, in the area of medical research it often happens that the real application dataset is based on a small number of whole slide images as training samples. This fact limits the performance of CNN methods in many medical contexts, but right-censored time-to-event outcomes may also appear in settings where it is easy to generate images or there are a large number of existing images, such as ecology or robotics.

Lastly, we have not investigated tuning hyper-parameters in great detail other than the learning rate. The CNN model as well as the two implementations (single output and multi-output) can be tuned and we think that with further hyper-parameter tuning, better performance can likely be achieved. In the multi-output implementation, the criteria used to combine the loss of the multiple outputs is an important hyper-parameter to tune. One can also tune the aggregation procedure to go from a per-slide prediction to a per individual. Also, one might consider weighting the loss function at each time point to account for the heterogeneity across time points. more precise tuning of these and potential other parameters is a future line of work for the authors.

Conclusions

This work contributes to modern image-based precision medicine by providing an alternative to Cox loss in CNN image analysis for prediction of cumulative incidence at a given time point. Our proposed method uses classical pseudo-observations as the outcome in deep CNN methods to predict the cumulative incidence, which as demonstrated in both simulation and a real data example outperformed the state-of-the-art Cox loss based methods in many plausible settings. Moreover, our proposed method is more flexible as it does not assume proportional hazards and allows for simpler and easily modified loss functions.

Supporting information

Appendix

Acknowledgments

None

References

1. Colling R, Pitman H, Oien K, Rajpoot N, Macklin P, in Histopathology Working Group CPA, et al. Artificial intelligence in digital pathology: a roadmap to routine use in clinical practice. The Journal of Pathology. 2019;249(2):143–150.

2. Ehteshami Bejnordi B, Veta M, Johannes van Diest P, van Ginneken B, Karssemeijer N, Litjens G, et al. Diagnostic assessment of deep learning

algorithms for detection of lymph node metastases in women with breast cancer. JAMA. 2017;318(22):2199–2210.

3. Yoo H, Kim KH, Singh R, Digumarthy SR, Kalra MK. Validation of a deep learning algorithm for the detection of malignant pulmonary nodules in chest radiographs. JAMA Network Open. 2020;3(9):e2017135–e2017135.
4. Ma M, She Y, Ren Y, Dai C, Zhang L, kang Xie H, et al. Micropapillary or solid pattern predicts recurrence free survival benefit from adjuvant chemotherapy in patients with stage IB lung adenocarcinoma. Journal of thoracic disease. 2018;10 9:5384–5393.
5. LeCun Y, Bengio Y, Hinton G. Deep Learning. Nature. 2015;521:436–44. doi:10.1038/nature14539.
6. Lu L, Zheng Y, Carneiro G, Yang L. Deep learning and convolutional neural networks for medical image computing: precision medicine, high performance and large-scale datasets. Advances in Computer Vision and Pattern Recognition. Springer; 2017.
7. Mobadersany P, Yousefi S, Amgad M, Gutman DA, Barnholtz-Sloan JS, Velázquez Vega JE, et al. Predicting cancer outcomes from histology and genomics using convolutional networks. Proceedings of the National Academy of Sciences. 2018;115(13):E2970–E2979. doi:10.1073/pnas.1717139115.
8. Wulczyn E, Steiner DF, Xu Z, Sadhwani A, Wang H, Flament-Auvigne I, et al. Deep learning-based survival prediction for multiple cancer types using histopathology images. PLOS ONE. 2020;15(6):1–18. doi:10.1371/journal.pone.0233678.
9. Li H, Boimel P, Janopaul-Naylor J, Zhong H, Xiao Y, Ben-Josef E, et al. Deep convolutional neural networks for imaging data based survival analysis of rectal cancer. In: 2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019); 2019. p. 846–849.
10. Hao J, Kosaraju SC, Tsaku N, Song D, Kang M. PAGE-Net: Interpretable and integrative deep learning for survival analysis using histopathological images and genomic data. Pacific Symposium on Biocomputing. 2020;25:355–366.
11. 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); 2016. p. 544–547.
12. Gensheimer MF, Narasimhan B. A scalable discrete-time survival model for neural networks. PeerJ. 2019;7:e6257. doi:10.7717/peerj.6257.
13. Sun D, Li A, Tang B, Wang M. Integrating genomic data and pathological images to effectively predict breast cancer clinical outcome. Computer methods and programs in biomedicine. 2018;161:45–53.
14. HernÃ¡n M. The Hazards of Hazard Ratios. Epidemiology. 2010;21:13–5.
15. Vock DM, Wolfson J, Bandyopadhyay S, Adomavicius G, Johnson PE, Vazquez-Benitez G, et al. Adapting machine learning techniques to censored time-to-event health record data: a general-purpose approach using inverse probability of censoring weighting. Journal of Biomedical Informatics. 2016;61:119–131. doi:10.1016/j.jbi.2016.03.009.

16. Gonzalez Ginestet P, Kotalik A, Vock DM, Wolfson J, Gabriel EE. Stacked inverse probability of censoring weighted bagging: A case study in the InfCareHIV Register. Journal of the Royal Statistical Society: Series C (Applied Statistics). 2020;70(1):51–65.

17. Zhao L, Feng D. Deep neural networks for survival analysis using pseudo values. IEEE Journal of Biomedical and Health Informatics. 2020;.

18. Robins J, Finkelstein D. Correcting for noncompliance and dependent censoring in an AIDS clinical trial with inverse Probability of Censoring Weighted (IPCW) Log-Rank Tests. Biometrics. 2000;56(3):779–788.

19. Andersen PK, Klein JP, RosthÃ¸j S. Generalised linear models for correlated pseudo-observations, with applications to multi-state models. Biometrika. 2003;90(1):15–27.

20. Faraggi D, Simon R. A neural network model for survival data. Statistics in Medicine. 1995;14(1):73–82. doi:10.1002/sim.4780140108.

21. 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 Medical Research Methodology. 2018;18(24).

22. Ching T, Zhu X, Garmire LX. Cox-nnet: An artificial neural network method for prognosis prediction of high-throughput omics data. PLOS Computational Biology. 2018;14(4):1–18. doi:10.1371/journal.pcbi.1006076.

23. Luck M, Sylvain T, Cardinal H, Lodi A, Bengio Y. Deep learning for patient-specific kidney graft survival analysis. ArXiv. 2017;abs/1705.10245.

24. Hao J, Kim Y, Mallavarapu T, Oh JH, Kang M. Interpretable deep neural network for cancer survival analysis by integrating genomic and clinical data. BMC Medical Genomics. 2019;12:189. doi:10.1186/s12920-019-0624-2.

25. Andersen P, Perme M. Pseudo-observations in survival analysis. Statistical methods in medical research. 2009;19:71–99. doi:10.1177/0962280209105020.

26. Krizhevsky A. Learning multiple layers of features from tiny images; 2009.

27. Gutman D, Cobb J, Somanna D, Yuna P, Wang F, Kurc T, et al. Cancer Digital Slide Archive: an informatics resource to support integrated in silico analysis of TCGA pathology data. Journal of the American Medical Informatics Association : JAMIA. 2013;20.

28. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2015;.

29. Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, et al. PyTorch: An imperative style, high-performance deep learning library. In: NeurIPS; 2019.

30. Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, et al.. TensorFlow: Large-scale machine learning on heterogeneous systems; 2015.

31. Goodfellow I, Bengio Y, Courville A. Deep Learning. MIT Press; 2016.

32. Simonyan K, Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition. In: International Conference on Learning Representations; 2015.

33. Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, et al. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. ArXiv. 2017;abs/1704.04861.

34. Collett D. Modelling survival data in medical research. (3rd ed.). Chapman & Hall/CRC; 2014.

35. Graw F, Gerds T, Schumacher M. On pseudo-values for regression analysis in competing risks models. Lifetime Data Anal. 2009;15:241–255.

36. Satten GA, Datta S, Robins J. Estimating the marginal survival function in the presence of time dependent covariates. Statistics and Probability Letters. 2001;54(4):397 – 403.

37. Overgaard M, Parner ET, Pedersen J. Pseudo-Observations Under Covariate-Dependent Censoring. Journal of Statistical Planning and Inference. 2019;202:112–122.

38. Cox DR. Regression models and life-tables. Journal of the Royal Statistical Society Series B (Methodological). 1972;34(2):187–220.

39. Aalen OO. A linear regression model for the analysis of life times. Statistics in Medicine. 1989;8(8):907–925.

40. Binder H. CoxBoost: Cox models by likelihood based boosting for a single survival endpoint or competing risks; 2013. Available from: `https://cran.r-project.org/package=CoxBoost`.

41. Ishwaran H, Kogalur UB. randomForestSRC: Fast Unified Random Forests for Survival, Regression, and Classification (RF-SRC).; 2020. Available from: `https://cran.r-project.org/package=randomForestSRC`.

42. Sachs MC, Gabriel EE, Overgaard M, Gerds TA, Therneau T. eventglm: Regression models for event history outcomes.; 2020. Available from: `https://sachsmc.github.io/eventglm`.

43. Zhang Y, Yang Q. An overview of multi-task learning. National Science Review. 2017;5(1):30–43. doi:10.1093/nsr/nwx105.

44. Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, et al. ImageNet large scale visual recognition challenge. International Journal of Computer Vision. 2014;115. doi:10.1007/s11263-015-0816-y.

45. Raghu M, Zhang C, Kleinberg J, Bengio S. In: Transfusion: Understanding Transfer Learning for Medical Imaging. Red Hook, NY, USA: Curran Associates Inc.; 2019.

46. Kingma D, Ba J. Adam: A method for stochastic optimization. International Conference on Learning Representations. 2014;.

47. Gerds TA. prodlim: Product-limit estimation for censored event history analysis.; 2020. Available from: `https://cran.r-project.org/package=prodlim`.

48. Russo J, Frederick J, Ownby HE, Fine G, Hussain M, Krickstein HI, et al. Predictors of recurrence and survival of patients with breast cancer. American Journal of Clinical Pathology. 1987;88(2):123–131.

49. Lee J, Kim S, Kang B. Prognostic factors of disease recurrence in breast cancer using quantitative and qualitative magnetic resonance imaging (MRI) parameters. Scientific Reports. 2020;10.

50. Wang Y, Acs B, Robertson S, Liu B, Solorzano L, WÃ¤hlby C, et al. Improved breast cancer histological grading using deep learning. Annals of Oncology. 2022;33(1):89–98. doi:https://doi.org/10.1016/j.annonc.2021.09.007.

51. Wang Y, Kartasalo K, Weitz P, Ã cs B, Valkonen M, Larsson C, et al. Predicting Molecular Phenotypes from Histopathology Images: A Transcriptome-Wide Expressionâ€“Morphology Analysis in Breast Cancer. Cancer Research. 2021;81(19):5115–5126. doi:10.1158/0008-5472.CAN-21-0482.

52. Liaw R, Liang E, Nishihara R, Moritz P, Gonzalez JE, Stoica I. Tune: A research platform for distributed model selection and training. ArXiv. 2018;abs/1807.05118.

53. Moritz P, Nishihara R, Wang S, Tumanov A, Liaw R, Liang E, et al. Ray: A distributed framework for emerging AI applications. In: Proceedings of the 13th USENIX Conference on Operating Systems Design and Implementation. OSDI'18. USA: USENIX Association; 2018. p. 561â€“577.

54. Uno H, Cai T, Tian L, Wei L. Evaluating prediction rules for t-year survivors with censored regression models. Journal of the American Statistical Association. 2007;102:527 – 537.

55. Kamarudin AN, Cox T, Kolamunnage-DonÃ R. Time-dependent ROC curve analysis in medical research: Current methods and applications. BMC Medical Research Methodology. 2017;17.

56. Hou L, Samaras D, KurÃ§ T, Gao Y, Davis JE, Saltz J. Patch-based convolutional neural network for whole slide tissue image classification. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2016; p. 2424–2433.

Table 1. Summary of the models with their loss function implemented in the simulations and real data application. The last fully connected layer of the pre-trained ResNet18 model was altered by a single neuron. All previous existing layers of the ResNet18 were frozen. In the training, the only parameters that are updated are of the altered fully connected and the new one added. LeakyReLU activation function is defined as $f(x) = 0.01x$ if $x < 0$ and $f(x) = x$ if $x \geq 0$. LF denotes loss function. $R_i(t_i)$ denotes the risk set at failure time t_i. N and $R_i(t_i)$ in the LF of the CoxCNN were restricted to samples within batch, as opposed to the standard Cox partial likelihood.

CoxCNN	(IPCW) POCNN single-output	(IPCW) POCNN multi-output

$y =$ ResNet18(Images)
$X =$ concatenate(y,clinical covariates)
output $= X \times w$
	$y =$ ResNet18(Images)
	$X =$
	concatenate(y,clinical covariates,time points)
	output = LeakyReLU($X \times w + b$)
		$y =$ ResNet18(Images)
		$X =$ concatenate(y,clinical covariates)
		output$_{t_1}$ $= LeakyReLU($ X$\times w_{t_1} + b_{t_1})$
		$output_{t_2} =$ LeakyReLU$(X \times w_{t_2} + b_{t_2})$
		$output_{t_3} =$ LeakyReLU$(X \times w_{t_3} + b_{t_3})$
		$output_{t_4} =$ LeakyReLU$(X \times w_{t_4} + b_{t_4})$

$LF = -\sum_{i \in N} \Delta^{(i)} \big(output_i - \log \sum_{j \in R_i(t_i)} \exp(output_j)\big)$	$LF = MSE(output)$	$LF = \sum_k MSE(output_{t_k})$

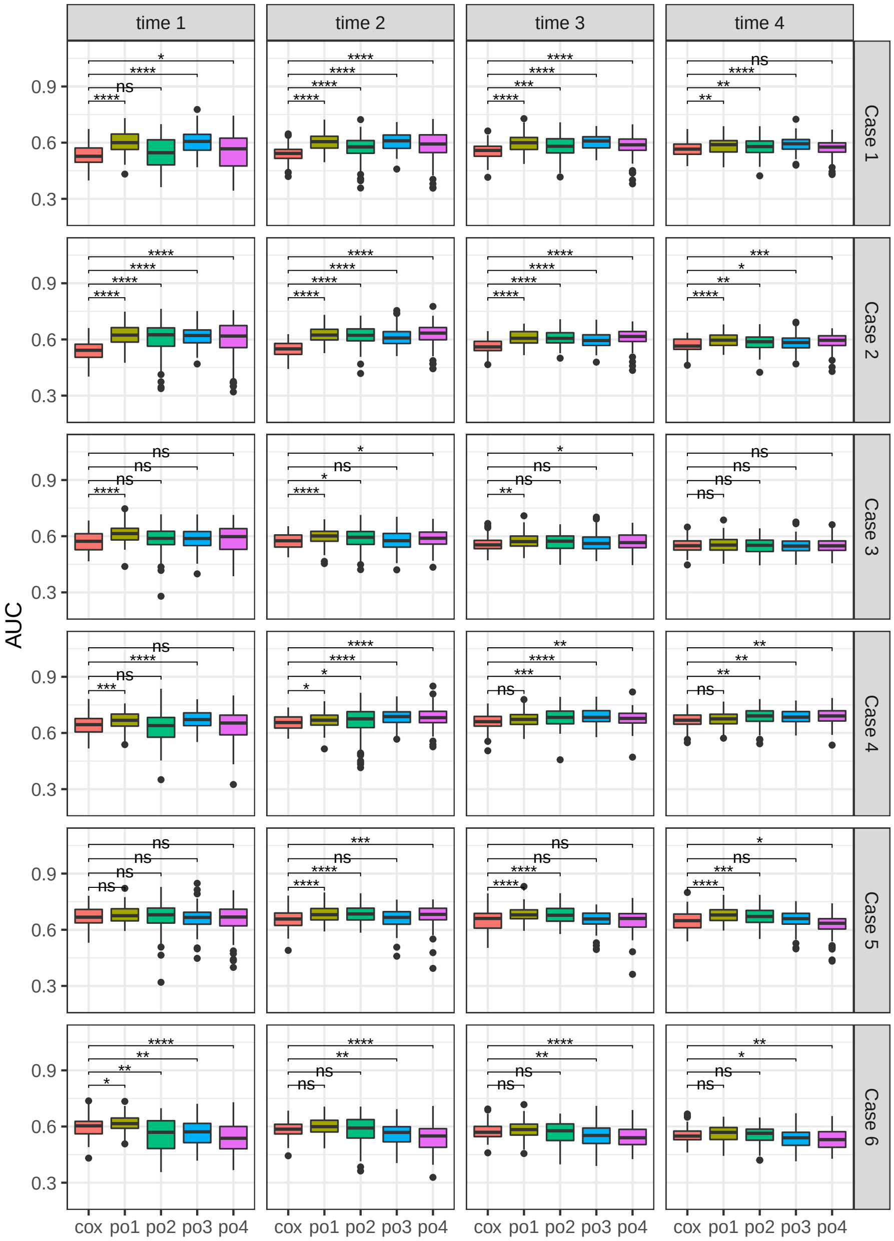


Fig 3. Boxplots of AUC values for the prediction of the cumulative incidence at 20th (time 1), 30th (time 2), 40th (time 3) and 50th (time 4) percentile of the overall time across 100 simulated datasets of sample size 1000 using different methods: Cox, CNN with Cox PH layer; po1, POCNN single output; po2, POCNN multi-output; po3, IPCW-POCNN single output, and po4, IPCW-POCNN multi-output. Pairwise comparison against the reference CoxCNN were compared using Wilcoxon signed-rank test. $****, p \leq 0.0001; ***, p \leq 0.001; **, p \leq 0.01; *, p \leq 0.05; ns$**, not significant.**

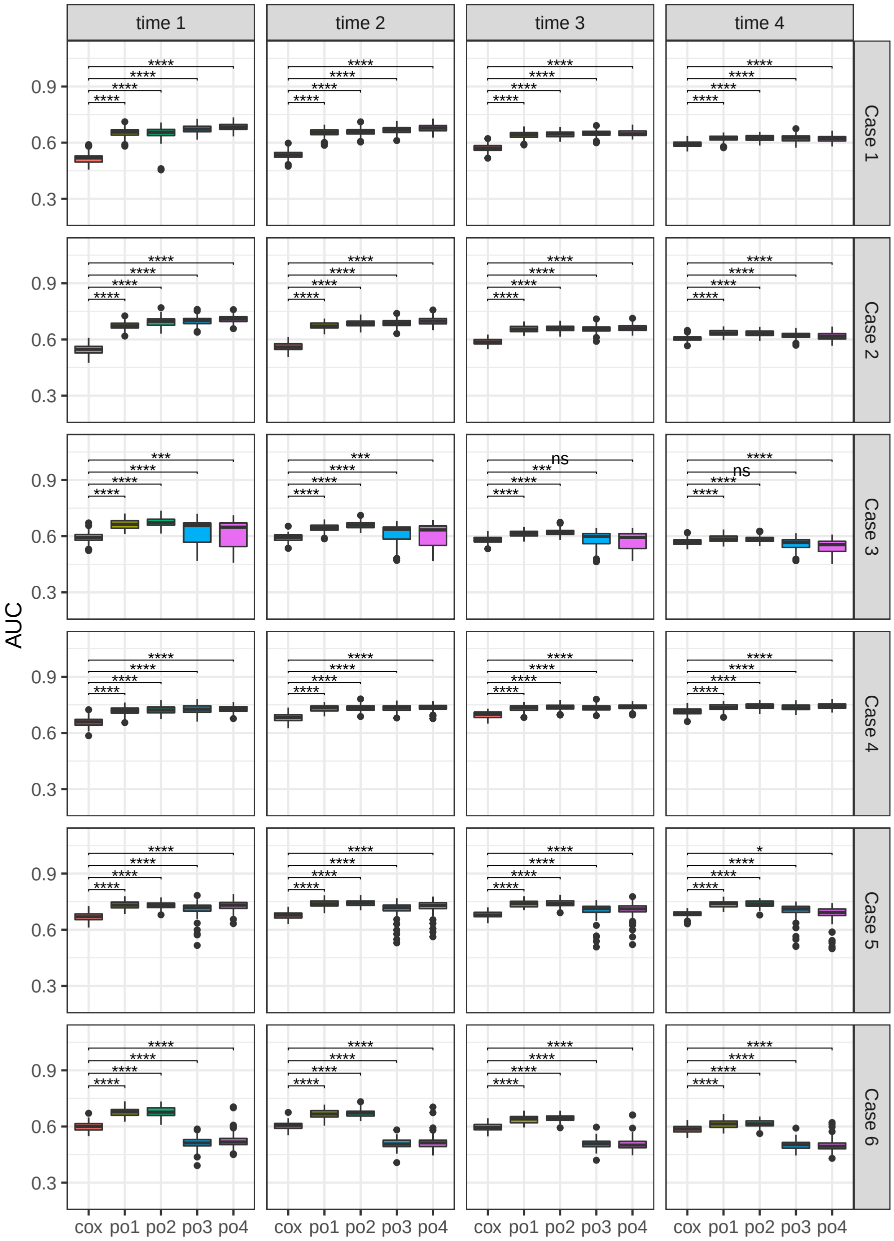


Fig 4. Boxplots of AUC values for the prediction of the cumulative incidence at 20th (time 1), 30th (time 2), 40th (time 3) and 50th (time 4) percentile of the overall time across 100 simulated datasets of sample size 5000 using different methods: Cox, CNN with Cox PH layer; po1, POCNN single output; po2, POCNN multi-output; po3, IPCW-POCNN single output, and po4, IPCW-POCNN multi-output. Pairwise comparison against the reference CoxCNN were compared using Wilcoxon signed-rank test. $****, p \leq 0.0001; ***, p \leq 0.001; **, p \leq 0.01; *, p \leq 0.05; ns$**, not significant.**

Table 2. Summary of the clinical information of breast cancer patients included in the analysis

	Overall
Sample size	710
Days to death (mean (SD))	1351.32 (1267.36)
Status (mean (SD))	0.16 (0.37)
Age (mean (SD))	58.33 (12.97)
Race (%)	
black	87 (12.3)
other	56 (7.9)
white	567 (79.9)
Ethnicity = other (%)	87 (12.3)
Pathologic stage (%)	
Stage I	62 (8.7)
Stage II	397 (55.9)
Stage III	151 (21.3)
StageX	100 (14.1)
Molecular subtype (%)	
Basal	129 (18.2)
Her2	57 (8.0)
LumA	369 (52.0)
LumB	132 (18.6)
Normal	23 (3.2)

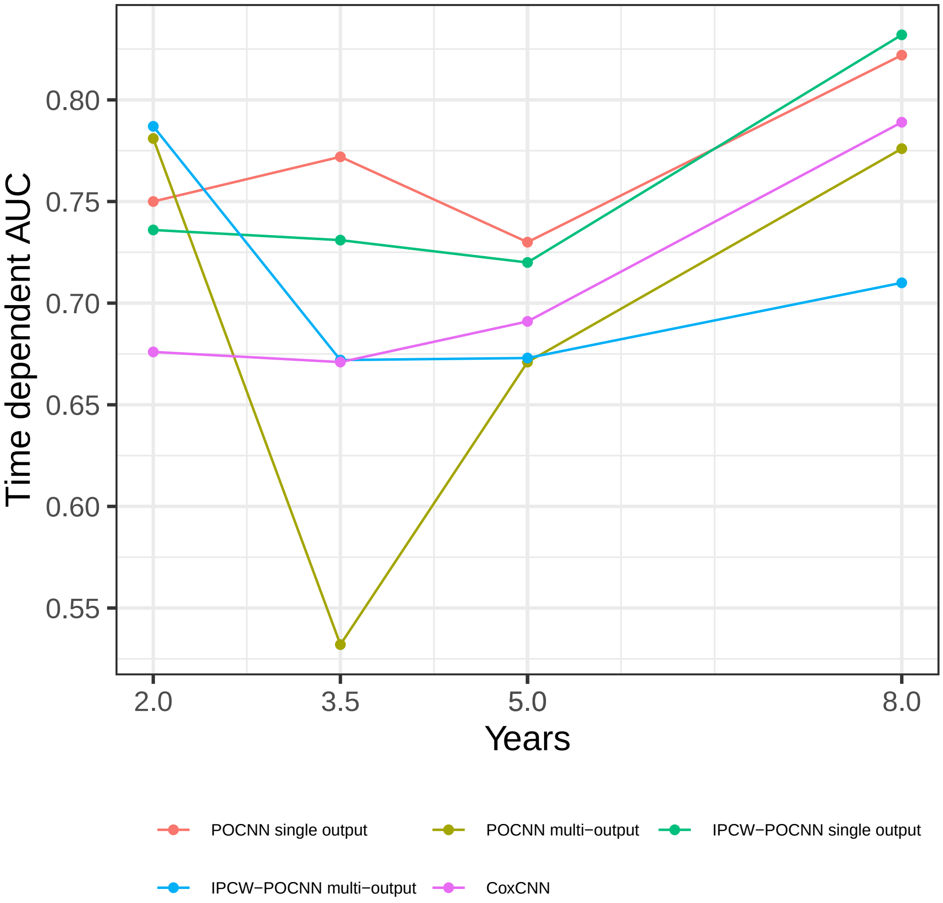
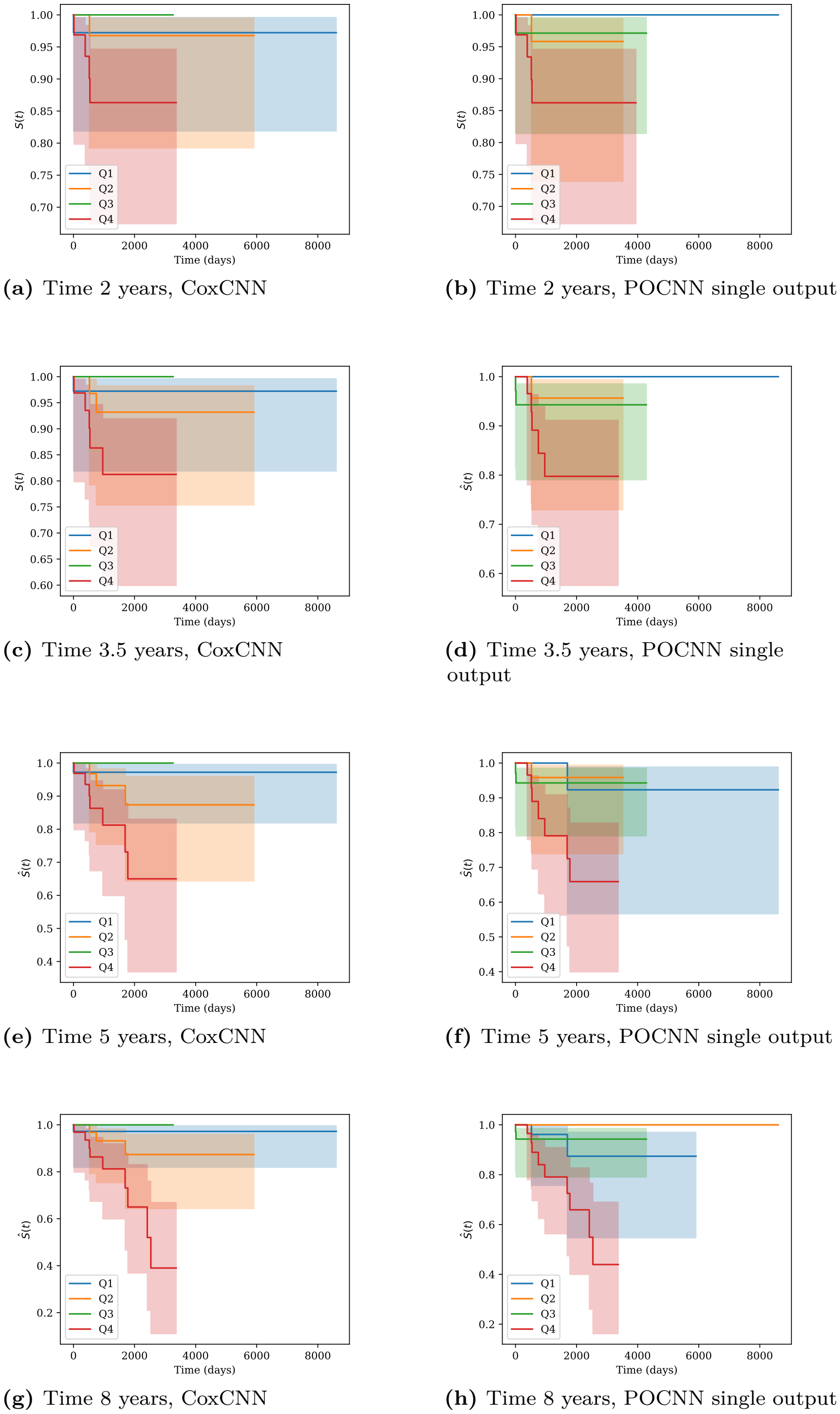


Fig 5. Estimated AUCs for predicting death at 2-year, 3.5-year, 5-year and 8-year, using the average criteria to obtain a whole's slide prediction.



(a) Time 2 years, CoxCNN

(b) Time 2 years, POCNN single output

(c) Time 3.5 years, CoxCNN

(d) Time 3.5 years, POCNN single output

(e) Time 5 years, CoxCNN

(f) Time 5 years, POCNN single output

(g) Time 8 years, CoxCNN

(h) Time 8 years, POCNN single output

Fig 6. Kaplan-Meier survival curves by quartiles (Q1-Q4) of the distribution of the predicted risk of death at 2; 3.5; 5 and 8 years in the TCGA test set based on the CoxCNN and POCNN single output.