HHS Public Access

Author manuscript

*Stat Med*. Author manuscript; available in PMC 2019 August 30.

Published in final edited form as:

*Stat Med*. 2019 May 30; 38(12): 2139–2156. doi:10.1002/sim.8090.

# Extreme learning machine Cox model for high-dimensional survival analysis

**Hong Wang**[1] and **Gang Li**[2]

[1]School of Mathematics and Statistics, Central South University, Changsha, China

[2]Department of Biostatistics, UCLA Fielding School of Public Health, University of California, Los Angeles, California

## Abstract

Some interesting recent studies have shown that neural network models are useful alternatives in modeling survival data when the assumptions of a classical parametric or semiparametric survival model such as the Cox (1972) model are seriously violated. However, to the best of our knowledge, the plausibility of adapting the emerging extreme learning machine (ELM) algorithm for single-hidden-layer feedforward neural networks to survival analysis has not been explored. In this paper, we present a kernel ELM Cox model regularized by an $L_0$-based broken adaptive ridge (BAR) penalization method. Then, we demonstrate that the resulting method, referred to as ELMCoxBAR, can outperform some other state-of-art survival prediction methods such as $L_1$-or $L_2$-regularized Cox regression, random survival forest with various splitting rules, and boosted Cox model, in terms of its predictive performance using both simulated and real world datasets. In addition to its good predictive performance, we illustrate that the proposed method has a key computational advantage over the above competing methods in terms of computation time efficiency using an a real-world ultra-high-dimensional survival data.

### Keywords

censored data; extreme learning machine; machine learning; regularized Cox model; survival analysis

## 1| INTRODUCTION

Survival analysis for a time-to-event outcome has wide applications in biomedical research and other fields. One of the major goals of survival analysis is to predict survival from a set of a patient's covariates or predictors. Analysis of time-to-event data is usually complicated by censoring when the survival time of interest is not fully observed for every subject in the study. For example, in clinical studies, the exact time of the event is only known if a patient experiences the event during the study, and if not, then it is usually not possible to know when such event will occur after the study. Another unprecedented challenge posed by continuing advancements in data acquisition technology is that recent survival data such as

**Correspondence** Gang Li, Department of Biostatistics, UCLA Fielding School of Public Health, University of California, Los Angeles, CA 90095-1772., vli@ucla.edu.

Author Manuscript
Author Manuscript
Author Manuscript
Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

high throughput genomic data has become increasingly high dimensional or ultra-high dimensional, ie, the number of covariates often far exceeds the number of observations ($n \gg$ p).

To deal with the challenges of survival analysis with high-dimensional data, a large variety of parametric and semiparametric models such as regularized Cox models have been developed and analyzed.[1–6] However, the model assumptions underlying these models, such as the proportional hazard, independence between censoring and survival times, linearity in coefficients, may not be tenable in real applications. For this reason, nonparametric approaches such as neural networks[7–9] can be useful alternatives. In particular, neural networks methodology has been adapted to survival analysis for more than two decades. Based on how survival information is exploited in neural networks, one can classify the neural networks methods into three main categories: classification methods, partial logistic artificial neural network (PLANN), and the Faraggi-Simon method. Classification methods treat the survival outcome as an uncensored discrete variable with all censored samples omitted or included in the highest category[10–12] or transform the problem into a complex discrete multiple classification framework.[13,14] PLANN methods assume a proportional-odds form using a logit transfer function on the linear model predictor and treats the extension of Cox regression in discrete times by modeling the conditional event rate per time interval.[15–20] The Faraggi-Simon method is a feedforward neural network that provides the basis for a nonlinear proportional hazards model.[21] In this approach, the linear combination of covariates is replaced by the nonlinear output function of a neural network and coefficients can be estimated by optimizing a modified Cox partial likelihood. Hence, the Faraggi-Simon method is usually regarded a nonlinear extension of the Cox model and most of the advantages of a classical proportional hazards model are preserved.

Due to the structural simplicity and its popularity, several machine learning methods including the popular deep neural networks have been extended to Cox's regression analysis in conjunction with the Faraggi-Simon method.[7–9,22–24] But perhaps due to the computation burden incurred by the backpropagation algorithms that are widely employed in both multilayer perceptron (MLP) neural networks and deep neural networks, most of these proposals focus on low-dimensional survival data. These sophisticated deep learning methods have demonstrated their usefulness in high-dimensional setting.[8,9] However, other than the requirement of a high performance computing platform, their success also relies on a large number of training examples,[25–27] which are often not available in a real study. Hence, it is of high importance to develop new survival prediction methods for high-dimensional data that are computationally scalable to high-dimensional data and require a small or moderate sample size.

In this paper, we extend a state-of-the-art extreme learning machine (ELM) algorithm[28] to survival analysis and propose a novel regularized Cox model within the simple framework of the Faraggi-Simon method. Besides its superb prediction capability, there are several reasons why we consider the ELM for single-hidden-layer feedforward neural network (SLFN) instead of other popular neural networks models. First, it has been shown that any continuous target function can be approximated by SLFNs with adjustable hidden nodes.[29,30] This means that complicated network structure such as MLP neural networks or deep

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

neural networks may not always be necessary. Second, conventional neural network training approaches, like the backpropagation or similar algorithms used in most deep learning neural networks, adjust both the input and output weights and the hidden layer bias values by applying gradient descend-based optimization. However, gradient descend-based learning techniques are generally slow and may decrease the network generalization ability since the solution may be trapped in local minima. In contrast, the ELM theory states that the hidden node parameters of SLFNs need not be tuned, ie, all these hidden node parameters can be randomly generated without knowing the training data.[31] Third, different from backpropagation approaches that consider MLP networks or deep networks as a black box, ELM handles SLFNs as a white box and can train hidden neurons layer by layer if multiple layers are necessary. Together with ELM, we adopt a recently proposed $L_0$-based penalization method, named CoxBAR,[32] for parameter estimation. In contrast to most regularization methods that require data-driven selection of the regularization tuning parameter(s), the CoxBAR method uses pre-specified tuning parameters and thus is computationally efficient.

In Section 2, we first review the ELM algorithm and the CoxBAR method and then propose a novel ELMCoxBAR survival model within the Faraggi-Simon framework. Evaluation metrics and some competing methods are described in Section 3. In Sections 4 and 5, the performance of ELMCoxBAR is illustrated, analyzed, and compared with the competing methods through extensive simulations and well-known benchmark real datasets. Section 6 provides concluding remarks and further discussion.

## 2 | METHODS

In this section, we first give an overview of the ELM technique and CoxBAR model to provide the necessary background. Then, we develop the ELMCoxBAR method using ELM and CoxBAR.

### 2.1 | Extreme learning machine

The ELM was originally proposed as a fast SLFN learning alternative for time-consuming backpropagation training approaches for classification and regression problems,[28,31] where the hidden layer output function (hidden layer mapping ELM feature space) can be any form of piecewise continuous functions such as sigmoid, Fourier series, and radial basis functions (RBFs).

Given training samples $\left\{\left(\mathbf{x}_i, y_i\right) | \mathbf{x}_i \in R^p, y_i \in R^m\right\}_{i=1}^n$, where $n$ denotes the number of observations, $p$ denotes the dimension of covariates, and $y_i \in R^m$ denote the target of each observation ($m = 1$ for regression and $m \geq 2$ for classification). SLFNs with $L$ hidden nodes can be represented by the following equation:

$$f_L(x) = \sum_{i=1}^{L} g\left(\mathbf{x}, w_i, b_i\right)\beta_i = \mathbf{h}(\mathbf{x})\beta, \quad (1)$$

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

where $g(\cdot)$ is the activation function, $w_i \in R^p$ denotes the input weight vector between the input layer and the output layer, $b_i$ is the bias of the ith hidden layer, and $\beta = [\beta_1, \dots, \beta_i, \dots, \beta_L]^T$ is the output weight vector.

The main theoretical result behind the feasibility of ELM is the following Universal Approximation Theorem.[33]

**Theorem 1.** Given any nonconstant piecewise continuous function g, if continuous target function f(x) can be approximated by SLFNs with adjustable hidden nodes g, then the hidden node parameters of such SLFNs need not be tuned. That is, for any continuous target function f(x) and any randomly generated sequence $\{w_i, b_i\}_{i=1}^{L}$,

$$\lim_{L \to \infty} \left\| f(x) - f_L(x) \right\| = \lim_{L \to \infty} \left\| f(x) - \sum_{i=1}^{L} g(\mathbf{x}, w_i, b_i)\beta_i \right\| = 0 \quad (2)$$

holds with probability one if β is chosen to minimize || f(x) – $f_L$(x) ||.

In other words, the hidden layer weights $w_i$ and bias values $b_i$ in ELM can be randomly generated and are independent of the training data.

Denote the hidden layer output matrix by

$$\mathbf{H} = \begin{bmatrix} h(\mathbf{x}_1) \\ h(\mathbf{x}_2) \\ \vdots \\ h(\mathbf{x}_n) \end{bmatrix} = \begin{bmatrix} g(w_1, b_1, \mathbf{x}_1) & \cdots & g(w_L, b_L, \mathbf{x}_1) \\ g(w_1, b_1, \mathbf{x}_2) & \cdots & g(w_L, b_L, \mathbf{x}_2) \\ \vdots & \cdots & \vdots \\ g(w_1, b_1, \mathbf{x}_n) & \cdots & g(w_L, b_L, \mathbf{x}_n) \end{bmatrix}_{(n \times L)}, \quad (3)$$

and denote the target matrix by

$$Y = \begin{bmatrix} y_1^T \\ y_2^T \\ \vdots \\ y_n^T \end{bmatrix} = \begin{bmatrix} y_{11} & \cdots & y_{1m} \\ y_{21} & \cdots & y_{2m} \\ \vdots & \cdots & \vdots \\ y_{n1} & \cdots & y_{nm} \end{bmatrix}. \quad (4)$$

The output weights $\beta$ is solved analytically using the following formula:

$$\hat{\beta} = \mathbf{H}^{+} Y, \quad (5)$$

where $\mathbf{H}^{+} = (\mathbf{H}'\mathbf{H})^{-1}\mathbf{H}'$ is the generalized pseudo-inverse of **H**, and an efficient closed-form solution with ridge regression[34] is given by

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

$$\hat{\beta} = \mathbf{H}^{Y}\left(\frac{1}{C} + \mathbf{H}\mathbf{H}^{Y}\right)^{-1}Y, \quad (6)$$

where $I$ is an $n \times n$ matrix.

If the hidden layer mapping $h(\cdot)$ is unknown, ie, an implicit function, one can define a kernel matrix for ELM that takes the following form

$$K\left(\mathbf{x}_i, \mathbf{x}_j\right) = h\left(\mathbf{x}_i\right) \cdot h\left(\mathbf{x}_j\right), \quad (7)$$

where $K(.,.)$ can be any type of strict positive definite kernel.

A kernel ELM or an SFLN with a kernel function using $L$ supporting vectors[35] can be modeled as

$$f_L\left(x_i\right) = \sum_{j=1}^{L} K\left(\mathbf{x}_i, \mathbf{x}_j\right)\beta_j, \quad i = 1, 2, \ldots, n \quad (8)$$

or in a compact matrix form

$$f_L(\mathbf{x}) = K_{n \times L}\beta\,. \quad (9)$$

It is further proved that an SLFN with $L = n$ random support vectors (involving all training samples) can learn $n$ distinct samples at zero error.[35]

## 2.2 | The CoxBAR model

Assume **x** to be a variable set of $p$ covariates and $D$ be the dataset containing the training samples in a form of $n \times (p + 2)$ matrix, namely, $D = (\boldsymbol{\pi}_i, \delta_i, \boldsymbol{X})$, $i = 1, 2, \ldots, n$. In the case of right-censored data, $\tau_i = \min\left(T_i, C_i\right)$, Where $T_i$ is the true survival time, $C_i$ the censoring time, and $\delta_i = I(T_i \le C_i)$ the censoring indicator.

In a Cox model,[36] the hazard of a patient can then be rewritten as the product of a baseline hazard $\boldsymbol{\lambda_0}(t)$ and a positive function of the covariates

$$\lambda_i(t) = \lambda_0(t)\exp\left(f\left(x_i\right)\right), \quad (10)$$

where $f(x_i)$ can be any function of covariates $x_i$, and for a traditional Cox model, $f(x_i) = \mathbf{x}_i\boldsymbol{\beta}$. Hence, in Cox regression, the hazards $\lambda_i(t)$ and $\lambda_k(t)$ of any pair of instances $(i, k)$ are proportional.

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

Denote $\mathscr{R}(t_i) = \{k | t_k \geq t_i\}$ the set of patients still at risk right before time $t_i$. The partial log likelihood of the Cox model can be written as

$$l(\beta) = \sum_{i=1}^{n} \delta_i \mathbf{x}_i \beta - \sum_{i=1}^{n} \delta_i \log \sum_{j \in \mathscr{R}(\tau_i)} \exp(\mathbf{x}_j \beta). \quad (11)$$

Because of its semiparametric nature and nice interpretation of covariate coefficients, the Cox model is the most popular approaches in survival analysis on low-dimensional data.[4] To model high-dimensional survival data ($p > n$), penalized likelihood-based extensions of the Cox model are often considered and the corresponding penalized version of log likelihood is given by

$$l(\beta) - \sum_{j=1}^{p} p_\lambda(|\beta_j|), \quad (12)$$

where $p_\lambda(|\beta_j|)$ is the penalty term with tuning parameter $\lambda$. Recently, a detailed comparison across various penalty terms such as Lasso, Ridge, Elastic Net, and SCAD has been performed.[4] Regardless of which penalty term is incorporated, a time-consuming cross-validation procedure is generally needed to tune parameters.[4]

More recently, a novel sparse Cox regression via broken adaptive ridge (CoxBAR) is developed using $L_0$-based iteratively reweighted $L_2$-penalized Cox regression.[32] The CoxBAR estimation of $\beta$ starts with an initial Cox ridge regression estimator

$$\hat{\beta}^{(0)} = \underset{\beta}{\operatorname{argmin}} \left\{ -2l(\beta) + \xi_n \sum_{j=1}^{p_n} \beta_j^2 \right\}, \quad (13)$$

which is updated iteratively by a reweighed $L_2$-penalized Cox regression estimator

$$\hat{\beta}^{(k)} = \underset{\beta}{\operatorname{argmin}} \left\{ -2l(\beta) + \lambda_n \sum_{j=1}^{p_n} \frac{\beta_j^2}{\left(\hat{\beta}_j^{(k-1)}\right)^2} \right\}, \quad k \geq 1, \quad (14)$$

where $\xi_n$ and $\lambda_n$ are nonnegative penalization tuning parameters. The CoxBAR estimator is defined as

$$\hat{\beta} = \lim_{k \to \infty} \hat{\beta}^{(k)}. \quad (15)$$

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

The above CoxBAR estimator inherits some appealing properties of both $L_0$- and $L_2$-penalized regressions, such as an oracle property for selection and estimation and a grouping property for highly correlated covariates. Moreover, the CoxBAR estimator with $\lambda = \ln(n)$ is insensitive to the choice of $\xi_n$.[32] These properties suggest that the CoxBAR estimator can be a very competitive candidate for Cox models with common penalty terms in computation-intensive setting such as high dimensional or large-scale survival data.

### 2.3 | ELM Cox model with BAR penalization

A potential limitation of the above penalized Cox models is that these models are built upon the underlying proportional hazard and linear in coefficients assumptions. When these assumptions do not hold, Cox models might not work properly. In this case, a nonparametric and nonlinear neural network model such as ELM may be useful. Here, we adopt the Faraggi-Simon framework in which a Cox-like model can be obtained via replacing the linear combination of covariates in (11) by the output layer function in the neural network. In our case, the kernel ELM output function (8) is applied and the number of support vectors $L$ is set to $n$ for a maximum accuracy.

Based on the above considerations, we propose an ELMCoxBAR survival model based on the kernel ELM neural network and the CoxBAR model. A high-level description of the ELMCoxBAR algorithm is stated as follows.

**a.** Choose an appropriate kernel function and calculate the kernel matrix $K_{n \times n}$ as in (9).

**b.** Replace the linear function in (11) by $K_{n \times n}\beta$ and similar to the kernel Cox regression,[37] we obtain the following partial log likelihood for ELMCoxBAR model:

$$g(\beta) = \sum_{i=1}^{n} \delta_i K_{n \times n}\beta - \sum_{i=1}^{n} \delta_i \log \sum_{j \in \mathscr{R}(\tau_1)} \exp\left(K_{n \times n}\beta\right) - 0.5\ln(n) \sum_{j=1}^{p} \frac{\beta_j^2}{\tilde{\beta}_j^2}. \qquad (16)$$

**c.** Let $\tilde{\beta}^{(0)}$ be a Cox ridge estimator (the ridge penalty parameter need not to be optimal and can be simply set to $\ln(n)$).[32] Estimate $\beta$ by maximizing $g(\beta)$ via a simple Newton-Raphson procedure[38] for a small or moderate $n$ or a sophisticated optimization algorithm[6] for a large $n$.

**d.** Iterate the previous two steps to update $\beta$ until convergence or a maximum iteration times *maxiter* is reached.

**e.** In prediction, the hazard of a new patient with covariates $\chi$ can be derived via

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

$$\begin{aligned}\lambda_i(t|\chi) &= \lambda_0(t)\exp(f(\chi)) \qquad (17)\\ &= \lambda_0(t)\exp\left(\sum_{j=1}^{n} K\left(\chi, \mathbf{x}_j\right)\beta_j\right).\end{aligned}$$

In this study, two popular kernel functions are implemented:

- the linear kernel: $K\left(\mathbf{x}_i, \mathbf{x}_j\right) = \mathbf{x}_i^T\mathbf{x}_j + c$;
- the RBF kernel: $K\left(\mathbf{x}_i, \mathbf{x}_j\right) = \exp\left(-\left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2/c\right)$.

However, in the case of high or ultra-high-dimensional settings, one may just use a linear kernel since the linear kernel is as good as the RBF kernel but is significantly faster.[39,40]

Due to the construct of kernel matrix ($K_{n \times L}$), ELMCoxBAR's efficiency can be degraded by a large sample size n. In this case, as suggested in the work of Deng et al,[35] we can set $L$ (the number of support vectors) to a small value and make a trade-off between accuracy and efficiency.

## 3 | MODEL EVALUATION

### 3.1 | Evaluation metrics

We adopt two popular criteria for censored data analysis, ie, integrated Brier score (IBS)[41] and Harrell's concordance index (C-index or CI)[42] to evaluate the accuracy of survival models in later experioments.

- *IBS:* The integrated Brier score over a certain period of time for the survival model is defined by

$$IBS = \frac{1}{\max\left(\tau_i\right)}\int_0^{\max\left(\tau_1\right)} BS(t)dt,$$

Where $BS(t)$ is the Brier score at time $t$. The Brier score can be regarded as the mean square error between the estimated survival function $\hat{S}(t|\mathrm{X})$ and the test data weighted by inverse probability of censoring

$$BS(t) = \frac{1}{n}\sum_{i=1}^{n}\left\{\left[0 - \hat{S}\left(t\middle|\mathbf{X}_i\right)\right]^2 I\left(\tau_i \leq t, \sigma_i = 1\right)\left(1/\hat{G}\left(\tau_i\right)\right) + \left[1 - \hat{S}\left(t\middle|\mathbf{X}_i\right)\right]^2 I\left(\tau_i \geq t\right)\left(1/\hat{G}(t)\right)\right\},$$

where $n$ is the sample size of the test data, $I$ is an indicator function, and $\hat{G}(t)$ is the nonparametric Kaplan-Meier estimate of the censoring distribution. We prefer survival models with small IBS values. When the IBS value equals 0, it means a perfect prediction.

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

- *C-index:* The concordance index (C-index) denotes the fraction of all pairs of observations whose survival predictions have correct ranks over the valid pairs. It can be calculated as follows:
  - Create all time pairs of observed survival times and initialize sum $s = 0$
  - For all comparable time pairs (pairs where time $T_{j1}$ is greater than time $T_{j2}$), evaluate whether the corresponding survival probabilities or hazard rates are concordant, ie, $\eta_{j1} > \eta_{j2}$. If so, $s = s + 1$; If $\eta_{j1} = \eta_{j2}$, $s = s + 0.5$; If $\eta_{j1} < \eta_{j2}$, $s$ remains the same.
  - Let $q$ denote the number of comparable time pairs. Divide the total sum $s$ by $q$ and C-index equals $s/q$.

If C-index equals i, it means a perfect prediction, and if C-index ≤ 0.5, it implies that the model behaves like or worse than a random guess. Different from IBS, a larger C-index value implies a better performance.

## 3.2 | Comparing models

To demonstrate the effectiveness of the proposed ELMCoxBAR approach, we will compare our method with several popular semiparametric and nonparametric models widely used in real applications.

- *Penalized Cox Models.* When maximizing the partial log likelihood of Cox model $l(\beta) = \sum_{i=1}^{n} \delta_i x_i^T \beta - \sum_{i=1}^{n} \delta_i \log \sum_{j \in \mathscr{R}(\tau_1)} \exp\left(x_j^T \beta\right)$ where $\beta = (\beta_1, \ldots, \beta_p)^T$ is an unknown vector and $\mathscr{R}(t)$ is the risk set right before the time $t$, one can either regularize it using the $L_1$ penalty (CoxLasso)[1]: $\hat{\beta} = \mathrm{argmax} l(\beta), \|\beta\|_1 \le s$ or using the $L_2$ penalty (CoxRidge)[5]: $\hat{\beta} = \mathrm{argmax} l(\beta), \|\beta\|_2 \le s$, where $s$ is a positive constant.
- *Random Survival Forest.* Random survival forest is a censored extension of random forest,[43] where the ensemble survival function is aggregated by tree-based Nelson-Aalen estimators. We consider several random survival forest models with different splitting rules: the traditional log-rank rule (RSFL),[44] two state-of-art splitting rules based on C-index (RSFCI),[45] and maximally selected rank statistics (RSFM).[46]
- *Boosted Cox Model.* CoxBoost is one of the few approaches that allow the implementation of popular boosting teniques in conjunction with the Cox models.[47] In CoxBoost, an offset-based likelihood boosting approach is applied to estimate coefficients of Cox proportional hazards models.

Comparisons with these models are conducted with corresponding “glmnet,” “ranger,” and “CoxBoost” packages in R. All default settings of these methods are applied: for CoxLasso and CoxRidge, a cross-validation version (cv.glmnent) is applied and nfolds = 10; for all RSF models, $\mathrm{mtry} = \sqrt{p}$, num.trees = 500, nodesize = 3; for CoxBoost, stepno = 100 (number of boosting steps).

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

For the ease of notation, models with longer names, ie, ELMCoxBAR, CoxLasso, CoxRidge, and CoxBoost are abbreviated as ECoxBAR, CoxL, CoxR, and CoxBst, respectively, when necessary.

## 4 | SIMULATED SURVIVAL DATA

We use simulation studies to evaluate the effectiveness of the proposed method for survival prediction in both high and ultra-high-dimensional settings. We also investigate the parameter sensitivity of the proposed ELMCoxBAR using simulated ultra-high-dimensional survival data with different censoring rates.

### 4.1 | Simulated survival datasets

In our simulation studies, the sample size of each simulated dataset is 300. For each observation, values for the $p$ covariates are generated from a multivariate normal distribution with mean 5 and autoregressive correlation structure $\text{corr}(x_i, x_j) = p^{|i-j|}$ ($p = 0.7$) for two covariates $i$ and $j$. Then, all these covariates all transformed by a certain type of kernel. A Weibull distribution with the scale parameter of 2 is used for the baseline hazard function. Censoring times are drawn from an uniform distribution [0,8] and are independent of the survival times. Approximately 25% censoring rate is expected.

### 4.2 | Performance comparison on simulated datasets

In the experiments, 4 different dimensions (numbers of covariates) ranging from 500 to 5000 and a Weibull distribution with three shape parameters (0.5 for decrease hazard, 1 for constant hazard, and 3 for increase hazard) are considered.

In real applications, the embedding kernels cannot be known beforehand; thus, choice of kernels on the prediction results needs to be investigated. We consider two cases here.

- Case 1: A kernel is correctly specified, ie, the data is generated and predicted using the same type of kernel.
- Case 2: When a zkernel is misspecified, ie, the data is generated using one type of kernel (say, an RBF kernel) but is modeled using another one (a linear kernel).

Hence, all together, 24 scenarios of survival data are simulated. For all simulations, twofold cross-validation is chosen,ie, the dataset is randomly divided into two halves, the first half is used for training and the other half for testing and viceversa. This process is repeated 10 times for each simulation for ELMCoxBAR and all the compared models.

**4.2.1 | Case 1: correctly specified kernels—**First, we consider the case when the kernel is correctly specified: allsurvival datasets here are simulated and transformed using linear kernels and then predicted by ELMCoxBAR with the same type of kernels.

Figure 1 presents the box plots of the prediction performance in terms of C-index for Case 1 when $p$ =500,1000,2000, 5000 (from top to bottom), respectively.

Figure 2 presents the box plots of the prediction performance in terms of IBS for Case 1 when $p$ =500,1000, 2000,5000 (from top to bottom), respectively.

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

In general, the proposed model does fairly well in predictive performance on all 12 simulation settings. This is because, the data is generated with a linear kernel and then predicted by ELMCoxBAR with the same type of kernel. However, there are some unique differences between different hazard types that cannot be ignored. From the first columns of both Figures, one can observe that, when survival times have a decreasing hazard rate (*shape* = 0.5), the proposed method is marginally better to other models in terms of C-index. In terms of IBS, its performance is similar to or comparable with other models, and looking at the second columns (*shape* = 1) of these two Figures, when the hazard function is constant, the proposed method is superior to the other six survival models in terms of C-index and at par in terms of IBS. While according to the last columns (*shape* = 3) of two Figures, when the hazard rate is increasing over time, ELMCoxBAR outperforms other models by a large margin in terms of C-index and has a better performance interms of IBS in most cases.

The results on these simulated data therefore indicates that, when the underlying kernel is correctly specified, ELMCoxBAR has a better predictive performance with an increasing hazard rate. Besides, the superiority of the proposed method becomes more evident with the increase of dimensionality.

#### 4.2.2 | Case 2: misspecified kernels—

Next, we consider the case when the kernel is not correctly specified: all survival datasets in this case are simulated and transformed using RBF kernels and then predicted by ELMCoxBAR with linear kernels.

Figure 3 presents the box plots of the prediction performance in terms of C-index for Case 2 when $p$ =500,1000,2000, 5000 (from top to bottom), respectively.

Figure 4 presents the box plots of the prediction performance in terms of IBS for Case 2 when $p$ =500,1000, 2000,5000 (from top to bottom), respectively.

The results from Figures 3 and 4 show that, if the data is generated using a kernel different from the one adopted by ELMCoxBAR, then ELMCoxBAR’s performance is degraded and this is not something unexpected. However, even using misspecified kernels, ELMCoxBAR’s performance is also comparable to other models in terms of both C-index and IBS. In addition, judging from the second and last columns of these two Figures, when the hazard rate is a constant or an increasing function, ELMCoxBAR even takes the lead twice in terms of C-index and three times in terms of IBS across all eight simulations.

Hence, with regard to the testing of kernels, ELMCoxBAR appears to be somewhat robust toward a misspecified kernel in these simulations. Since the computation overhead incurred by an RBF kernel during training and prediction is many times more than that by a linear kernel, we prefer the use of linear kernels over RBF kernels in case of large sample size and/or high dimensionality.

### 4.3 | Parameter sensitivity analysis

Here, we want to examine the sensitivity of ELMCoxBAR to the choice of parameters under different censoring rates using simulated ultra-high-dimensional data ($n$ = 300, $p$ = 5000, *shape* = 3). Compared with traditional neural networks based Cox models, much less parameters need to be chosen in the training process of ELMCoxBAR. From the previous section, we can see that two tuning parameters should be properly specified, ie, the parameter *c* for kernel functions and the number of maximum iterations *maxiter*.

In the following experiments, we assume that the type of kernel is correctly specified and 3 different censoring rates ranging from 25 to 75% are investigated.

**4.3.1 | Kernel parameter *c*—**First, we want to test the performance of ELMCoxBAR with different kernel parameters *c*. It is obvious that the parameter of a linear kernel has no effect on prediction if we standardize the kernel matrix before computing its likelihood with a BAR penalty.

Thus, in the following, we only test the sensitivity of ELMCoxBAR with RBF kernels. To see how the choice of *c* mayinfluence ELMCoxBAR's performance, we test *c* using a variety of values ranging from 500 to 100 000.

Figure 5 illustrate the performance of ELMCoxBAR with different values of *c* under three degrees of censoring using 20 simulated datasets (different line colors denote different datasets).

From Figure 5, one may observe that, in terms of C-index, sensitivity of *c* on ELMCoxBAR (with an RBF kernel) is somewhat related to the censoring rates: in case of a high censoring rate (75%), ELMCoxBAR is not too much sensitive to changes of *c* values; under a medium censoring rate (50%) and when $c > 2000$, a larger *c* implies a better performance in most cases; when a low censoring rate (25%) is present, ELMCoxBAR's performance is generally on the decrease with the increase of *c*.

In terms of IBS, things are drastically different. Under all degrees of censoring, ELMCoxBAR's performance is rather stable and seems insensitive to changes of *c* if $c \leq 10$ 000; when $c > 10$ 000, however, ELMCoxBAR's prediction capability begins to deteriorate rapidly with the increase of *c*.

From the above, one can easily observe that, to obtain a satisfactory result, parameter *c* needs to be selected carefully for ELMCoxBARwith an RBF kernel. This result agrees with other RBF kernel based approaches such as ELM[34,40,48] and Support Vector Machine.

In case an RBF kernel is necessary, we prefer to set a large value for *c* (eg, 5000–10 000) for ELMCoxBAR, and this setting is consistent with previous studies obtained for ELM in the classification and regression contexts.[34]

All considered, as far as both the computation efficiency and predication accuracy are concerned, we recommend to use a linear kernel as a first choice and if it fails, we can try an RBF kernel with a well-tuned parameter.

**4.3.2 | Choice of *maxiter*—**Next, we want to determine the maximum iteration needed to obtain a optimal $\beta$ before ELMCoxBAR can produce a satisfactory result. If *maxiter* is small, the obtained coefficients $\beta$ might not be optimal while if *maxiter* is large, the

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

ELMCoxBAR procedure will take a relatively long time to converge. Here, we only test ELMCoxBAR with *maxiter* values ranging from 2 and 100 in the experiments.

Figure 6 illustrate the performance of ELMCoxBAR with different values of *maxiter* under three degrees of censoring using 20 simulated datasets (different line colors denote different datasets).

From Figure 6, it is clear that ELMCoxBAR's performance is not sensitive to choice of *maxiter* in terms of both C-index and IBS metrics. We know that a more accurate estimation of $\theta$ is usually guaranteed with a larger number of iterations. If one examines Figure 6 carefully, one may observe that there is a slight improvement in performance in terms of C-index with more iterations if $maxiter < 10$. However, it does not always imply an improved predictive performance when $maxiter > 10$. To make a trade-off between accuracy and efficiency, we set $maxiter = 5$ as the default value.

# 5 | REAL DATA APPLICATIONS

To further investigate the results obtained from the simulation study, we compare ELMCoxBAR with popular survival models on nine benchmark survival datasets extensively analyzed in the statistical literature. Both high-dimensional and ultra-high-dimensional survival datasets are included to show the effectiveness of the proposed algorithm. In order to test ELMCoxBAR's performance in terms of time efficiency, we also report the running time of all compared models on a real ultra-high-dimensional survival dataset.

## 5.1 | Real survival datasets

All these nine real survival datasets are public available through their R packages on Bioconductor (https://www.bioconductor.org/) or through the given web addresses. All the datasets are further preprocessed by eliminating all columns have the same values or data with missing values. The dimension to sample size ratio of each dataset ranges from 312 to 5 and the related censoring rate varies from 32% to 79%. Please note that, in the experiments with ultra-high-dimensional survival data that may have multiple survival information, overall survival time is the primary survival endpoint; otherwise, relapse-free time is chosen.

A short introduction of the benchmark datasets are given in the following.

- The *DrAsGiven Dataset* contains gene expression profiles of ovarian cancer samples treated at Duke University Medical Center and H. Lee Moffitt Cancer Center and Research Institute.[49] In all 117 samples used in this study, 22115 gene features and 7 clinical covariates are provided. The data can be obtained from the R package "dressCheck" of "Bioconductor."
- The *EMTAB386 Dataset* incorporates angiogenic mRNA and microRNA gene expression signature on 107 advanced stage, high grade serous ovarian cancers.[50] After preprocessing, each patient sample contains 10 357 gene features and 7 clinical covariates. The data can be obtained from the "curatedOvarianData" R package.

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

- The *GSE14764 Dataset* used here includes prognostic gene expression indexes in 42 ovarian cancer observations.[51] For each observation, 13104 gene features and 9 clinical covariates are provided. The dataset is provided by the "curatedOvarianData" R package.
- The *GSE32062 Dataset* is a dataset containing 204 patients with advanced-stage high-grade serous ovarian cancer.[52] The dataset can be obtained via the "curatedOvarianData" R package.
- The *GSE49997Dataset* contains the expression values of 193 epithelial ovarian cancer patients.[53] In each patient, 16 048 gene state information and 9 clinical covariates are provided. The dataset is available from the "curatedOvarianData" R package.
- The *LungBeer Dataset* contains 86 patients with lung adenocarcinoma.[54] A total of 7131 covariates are available in the dataset. This dataset can be downloaded from http://user.it.uu.se/~liuya610/.
- The *metabric Dataset* is a subset of Molecular Taxonomy of Breast Cancer International Consortium (METABRIC) data that contains clinical and gene expression information derived from breast tumors collected from participants of the METABRIC trial.[55] This dataset is provided by Sage-Bionetworks and can be downloaded from https://github.com/Sage-Bionetworks/predictiveModeling/.
- The *NSBCD Dataset* includes patterns of expression of 549 "intrinsic" genes of 115 malignant breast tumors.[56] This dataset can be downloaded from http://user.it.uu.se/~liuya610/.
- The *TCGAmirna Dataset* presents 187 patients with high-grade serous ovarian cancer.[57] For each sample, 799 gene features and 17 clinical covariates are provided. The dataset can be obtained via the "curatedOvarianData" R package.

The first five datasets are our ultra-high-dimensional group datasets and the last four are our high-dimensional group datasets. Summary information of all these datasets is given in Table 1:

For all real datasets, 50–2 fold cross-validation is chosen, ie, the dataset is randomly divided into two halves, the first half is used for training and the other half for testing and vice versa. This process is repeated fifty times for each dataset for ELMCoxBAR and all the compared models.

## 5.2 | Performance comparison on real datasets

In the following, we compare the performance of ELMCoxBAR with the same compared models in simulation study, namely, RSFL, RSFM, RSFCI, CoxLasso, CoxRidge, and CoxBoost in terms of both prediction accuracy and time efficiency.

**5.2.1 | Performance comparison results—**Here, we report the experimental results on prediction performance of all seven models across nine benchmark datasets.

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

Figure 7 reports the performance of ELMCoxBAR, RSFL, RSFM, RSFCI, CoxLasso, CoxRidge, and CoxBoost algorithms in term of C-index on 100 runs of all these benchmark datasets.

From Figure 7, one can see that, in terms of C-index, the proposed ELMCoxBAR outperforms other survival models on EMTAB386, GSE14764, GSE49997, metabric, NSBCD, and TCGAmirna datasets either by a noticeable margin or in most cases. All compared models obtains similar results on the remaining three datasets.

Figure 8 reports the performance of ELMCoxBAR, RSFL, RSFM, RSFCI, CoxLasso, CoxRidge, and CoxBoost algorithms in term of IBS on 100 runs of all these benchmark datasets.

According to Figure 8, ELMCoxBAR is superior to other compared models on DrAsGiven, EMTAB386, GSE14764, GSE32062, LungBeer, metabric, and TCGAmirna datasets in most cases in terms of IBS. ELMCoxBAR and other models are at par on the remaining two datasets.

From the results of Figures 7 and 8, we can see that ELMCoxBAR performs best on EMTAB386, GSE14764, metabric, and TCGAmirna datasets in terms of both C-index and IBS.

Moreover, if we plot the estimated hazard functions with respect to time[58] for all benchmark datasets (Figure 9), we can observe that all the above four datasets (EMTAB386, GSE14764, metabric, and TCGAmirna) have a roughly monotone increasing hazard function. This finding confirms the assertion made in the simulation study that ELMCoxBAR works best with an increase hazard rate.

#### 5.2.2 | Computing time comparison results—

Here, we compare the running time of our proposed method, ELMCoxBAR, with other competing RSFL, RSFM, RSFCI, CoxLasso, CoxRidge, and CoxBoost algorithms on GSE32062 dataset, which is the most challenging dataset used here in terms of both sample size and dimensionality.

All calculations were carried out on a system with an Intel i5–5200 2.2 GHz quarcore processor and 8 GB of memory. The computation was repeated 100 times, and for a fair comparison and to exploit the parallel processing feature of the “ranger” R package, we set “num.threads = 2” for all three random forest approaches (RSFL, RSFM, and RSFCI). All other setting of all the algorithms remain the same as those used in the above performance comparisons experiments.

Training time for all seven survival models is reported in Table 2.

Figure 10 shows the total running time of both training and testing (prediction) for all models.

From Table 2 and Figure 10, we can see that ELMCoxBAR keeps the fast learning speed feature of ELM and is significantly faster than all the other models. RSFM takes the second,

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

followed by CoxBoost and RSFL. RSFCI, CoxLasso, and CoxRidge take rather long time on the current benchmark dataset.

## 6 | DISCUSSION

We have proposed a new algorithm called ELMCoxBAR, where a popular machine learning algorithm, ie, kernel ELM, has been extended for survival analysis of high-dimensional right-censored data. We have incorporated a Cox model with a broken adaptive ridge (CoxBAR) penalization into ELM, which has some favorable properties for survival prediction than traditional $L_1$ or $L_2$ penalization methods. The CoxBAR estimator approximates an $L_0$ penalized Cox Regression with desirable predictive and grouping properties. More importantly, CoxBAR does not require data-specific tuning parameter. These characteristics lead to significant improvement on kernel ELM's performance especially when an increase hazard rate is present.

In addition, the proposed ELMCoxBAR algorithm enjoys the major advantages of ELM such as high training efficiency and an easy implementation for survival analysis. Experimental results show that ELMCoxBAR is competitive with other popular survival models and it requires significantly less training time when compared to RSFL, RSFM, RSFCI, CoxLasso, and Cox Ridge on the high or ultra-high-dimensional survival prediction problems. The proposed ELMCoxBAR algorithm is implemented in the R programming language and available in the ELMSurv R package on CRAN. In this research, we have compared the performance of ELMCoxBAR with six popular survival models, and we evaluated their capability in predicting patient survival in high-dimensional right-censored data. RSFL, RSFM, RSFCI, CoxLasso, CoxRidge, and CoxBoost are highly accurate survival models and find great applications in biostatistics and biomedicine fields but experiment results on extensive simulated scenarios and a variety of real survival datasets have shown that they can be overtaken by the proposed ELMCoxBAR in terms of both and C-index and IBS metrics.

In our simulation study and real data experiments, ELMCoxBAR consistently performs best when the kernel type is correctly specified and obtains comparative results when the underlying kernel is misspecified. The superiority of ELMCoxBAR is dominant for these datasets with increase hazard rates and ultra-high dimensionality.

In the experiments, we also justify the choice of a linear kernel for ELMCoxBAR in high-dimensional setting. Quite a few kernel options, such as the linear kernel, the RBF kernel, the sigmoid kernel, are available for a kernel ELM. However, when the number of covariates is very large or if the computational cost is a concern, a nonlinear mapping such as the RBF kernel does not promise an improved performance.[39,40] Moreover, as discussed in the simulated study, the RBF kernel is sensitive to the choice of kernel parameter. Hence, in real applications, one may just use the linear kernel. Another parameter that controls the iteration times in estimating model coefficients seems to be rather insensitive to the ELMCoxBAR's predictive ability.

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

The proposed ELMCoxBAR method also has some limitations. First, ELMCoxBAR is currently implemented using a single kernel function. It is well known that the performances of any kernel-based method may depend on the kernel it uses.[59] How to construct an effective kernel function that adapts to real applications needs further study. A promising approach is to consider multiple kernel learning that is capable of selecting an optimal kernel and parameters from a larger set of kernels.[60] Second, for illustration purpose, we only extend the simplest ELM algorithm for SFLNs while other more complex ELM algorithms for MLP[61] or deep neural networks[62] have just been proposed.

Besides ELMCoxBAR, two recently proposed RSF models, ie, RSFM and RSFCI, also stand out with competitive predictive performance. As illustrated by the time efficiency results, due to heavy computation incurred in computing the C-index splitting statistic, RSFCI is time consuming for large samples and that is the major reason why it is not recommend for large-scale survival study by its inventors.[45] Experimental results have also shown that traditional penalized Cox models (CoxLasso and CoxRidge) are quite competitive if a decrease hazard rate is detected. However, cross-validation is usually required to obtain an optimal parameter estimate and consequently much training time is need. This situation may become worse in big survival datasets with a large sample size and/or ultra-high dimensionality. we notice that in terms of model execution time, RSFM is the most similar model to ELMCoxBAR and could be an alternative when ELMCoxBAR is not available.

## ACKNOWLEDGEMENTS

The research of Hong Wang was partly supported by the National Social Science Foundation of China (17BTJ019). The research of Gang Li was partly supported by the National Institute of Health (grants P30 CA-16042, UL1TR000124-02, and P01AT003960). The authors want to thank all reviewers and the associate editor for their valuable and constructive comments, which greatly improve the quality of this paper.

Funding information

National Social Science Foundation of China, Grant/Award Number: 17BTJ019; National Institute of Health, Grant/Award Number: P30 CA-16042, UL1TR000124-02, and P01AT003960

## REFERENCES

1. Tibshirani R The Lasso method for variable selection in the Cox model. StatMed 1997;16(4):385–395.
2. Gui J, Li H. Penalized Cox regression analysis in the high-dimensional and low-sample size settings, with applications to microarray gene expression data. Bioinformatics 2005;21(13):3001–3008. [PubMed: 15814556]
3. Cai T, Huang J, Tian L. Regularized estimation for the accelerated failure time model. Biometrics 2009;65:394–404. [PubMed: 18573133]
4. Benner A, Zucknick M, Hielscher T, Ittrich C, Mansmann U. High-dimensional Cox models: the choice of penalty as part of the model building process. BiomJ 2010;52(1):50–69. [PubMed: 20166132]
5. Simon N, Friedman JH, Hastie T, Tibshirani R. Regularization paths for Cox's proportional hazards model via coordinate descent. J Stat Softw 2011;39(5):1–13.
6. Mittal S, Madigan D, Burd R, Suchard M. High-dimensional, massive sample-size Cox proportional hazards regression for survival analysis. Biostatistics 2014;15(2):207–221. [PubMed: 24096388]

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

7. Liu W, Ji Z, He S, Zhu Z. Survival analysis of gene expression data using PSO based radial basis function networks. Paper presented at: 2012 IEEE Congress on Evolutionary Computation (CEC); 2012; Brisbane, Australia.

8. Hess M, Lenz S, Blatte TJ, Bullinger L, Binder H. Partitioned learning of deep Boltzmann machines for SNP data. Bioinformatics 2017;33(20):3173–3180. [PubMed: 28655145]

9. Yousefi S, Amrollahi F, Amgad M, et al. Predicting clinical outcomes from large scale cancer genomic profiles with deep survival models. Sci Rep 2017;7: article no. 11707.

10. Ebell MH. Artificial neural networks for predicting failure to survive following in-hospital cardiopulmonary resuscitation. J Fam Pract 1993;36(3):297–304. [PubMed: 8454976]

11. Davis G, Lowell W, Davis G. A neural network that predicts psychiatric length of stay. MD Comput ComputMed Pract 1993;10(2):87–92.

12. Tu JV, Guerriere MR. Use of a neural network as a predictive instrument for length of stay in the intensive care unit following cardiac surgery. Comput Biomed Res 1993;26(3):220–229. [PubMed: 8325002]

13. Ravdin PM, Clark GM. A practical application of neural network analysis for predicting outcome of individual breast cancer patients. Breast Cancer Res Treat 1992;22(3):285–293. [PubMed: 1391994]

14. Liestbl K, Andersen PK, Andersen U. Survival analysis and neural nets. StatMed 1994;13(12): 1189–1200.

15. Biganzoli E, Boracchi P, Mariani L, Marubini E. Feed forward neural networks for the analysis of censored survival data: a partial logistic regression approach. Stat Med 1998;17(10):1169–1186. [PubMed: 9618776]

16. Biganzoli E, Boracchi P, Marubini E. A general framework for neural network models on censored survival data. Neural Netw 2002;15(2):209–218. [PubMed: 12022509]

17. Biganzoli EM, Boracchi P, Ambrogi F, Marubini E. Artificial neural network for the joint modelling of discrete cause-specific hazards. ArtifIntell Med 2006;37(2):119–130.

18. Ambrogi F, Lama N, Boracchi P, Biganzoli E. Selection of artificial neural network models for survival analysis with genetic algorithms. Comput Stat Data Anal 2007;52:30–42.

19. Lisboa PJ, Etchells TA, Jarman IH, et al. Time-to-event analysis with artificial neural networks: an integrated analytical and rule-based study for breast cancer. Neural Netw 2008;21(2):414–426. [PubMed: 18304780]

20. Lisboa PJG, Etchells TA, Jarman IH, et al. Partial logistic artificial neural network for competing risks regularized with automatic relevance determination. IEEE Trans Neural Netw 2009;20(9): 1403–1416. [PubMed: 19628458]

21. Faraggi D, Simon R. A neural network model for survival data. StatMed 1995;14(1):73–82.

22. Faraggi D, Simon R. Bayesian neural network models for censored data. StatMed 1997;39:519–532.

23. Sato F, Shimada Y, Selaru FM, et al. Prediction of survival in patients with esophageal carcinoma using artificial neural networks. Cancer 2005;103(8):1596–1605. [PubMed: 15751017]

24. Katzman J, Shaham U, Bates J, Cloninger A, Jiang T, Kluger Y. Deep survival: a deep Cox proportional hazards network arXiv preprint arXiv:1606.00931; 2016.

25. LeCun Y, Bengio Y, Hinton G. Deep learning. Nature 2015;521(7553):436–444. [PubMed: 26017442]

26. Gu J, Wang Z, Kuen J, et al. Recent advances in convolutional neural networks arXiv preprint arXiv:1512.07108; 2015.

27. Angelov P, Sperduti A. Challenges in deep learning. In: Proceedings of the 24th European Symposium on Artificial Neural Networks (ESANN); 2016; Bruges, Belgium.

28. Huang GB, Zhu QY, Siew CK. Extreme learning machine: a new learning scheme of feedforward neural networks. In: Proceedings of the 2004 IEEE International Joint Conference on Neural Networks; 2004; Budapest, Hungary.

29. Park J, Sandberg IW. Universal approximation using radial-basis-function networks. Neural Comput 1991;3(2):246–257.

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

30. Leshno M, Lin VY, Pinkus A, Schocken S. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. Neural Netw 1993;6(6):861–867.

31. Huang GB, Zhu QY, Siew CK. Extreme learning machine: theory and applications. Neurocomputing 2006;70(1):489–501.

32. Kawaguchi ES, Suchard MA, Liu Z, Li G. Scalable sparse Cox's regression for large-scale survival data via broken adaptive ridge arXiv preprint arXiv:1712.00561; 2017.

33. Huang GB, Chen L, Siew CK. Universal approximation using incremental constructive feedforward networks with random hidden nodes. IEEE Trans Neural Netw 2006;17(4):879–892. [PubMed: 16856652]

34. Huang GB, Zhou H, Ding X, Zhang R. Extreme learning machine for regression and multiclass classification. IEEE Trans SystMan Cybern B Cybern 2012;42(2):513–529.

35. Deng WY, Ong YS, Zheng QH. A fast reduced kernel extreme learning machine. Neural Netw 2016;76:29–38. [PubMed: 26829605]

36. David CR. Regression models and life tables (with discussion). J Roy Stat Soc 1972;34:187–220.

37. Li H, Luan Y. Kernel Cox regression models for linking gene expression profiles to censored survival data. Pac Symp Biocomput 2002;8:65–76.

38. Frommlet F, Nuel G. An adaptive ridge procedure for L0 regularization. PloS One 2016;11(2):e0148620. [PubMed: 26849123]

39. Hsu C-W, Chang C-C, Lin C-J. A practical guide to support vector classification Taipei, Taiwan: National Taiwan University; 2016 http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf

40. Zong W, Huang G-B. Learning to rank with extreme learning machine. Neural Process Lett 2014;39(2):155–166.

41. Graf E, Schmoor C, Sauerbrei W, Schumacher M. Assessment and comparison of prognostic classification schemes for survival data. Stat Med 1999;18(17–18):2529–2545. [PubMed: 10474158]

42. Harrell FE, Lee KL, Mark DB. Tutorial in biostatistics multivariable prognostic models: issues in developing models, evaluating assumptions and adequacy, and measuring and reducing errors. StatMed 1996;15:361–387.

43. Breiman L Random forests. Mach Learn 2001;45(1):5–32.

44. Ishwaran H, Kogalur UB, Blackstone EH, Lauer MS. Random survival forests. Ann Appl Stat 2008;2(3):841–860.

45. Schmid M, Wright MN, Ziegler A. On the use of Harrell's C for clinical risk prediction via random survival forests. Expert Syst Appl 2016;63:450–459.

46. Wright MN, Dankowski T, Ziegler A. Unbiased split variable selection for random survival forests using maximally selected rank statistics. Stat Med 2017;36(8):1272–1284. [PubMed: 28088842]

47. Binder H, Schumacher M. Allowing for mandatory covariates in boosting estimation of sparse high-dimensional survival models. BMC Bioinformatics 2008;9(1):14. [PubMed: 18186927]

48. Jian Y, Huang D, Yan J, et al. A novel extreme learning machine classification model for e-nose application based on the multiple kernel approach. Sensors 2017;17(6):1434.

49. Dressman HK, Berchuck A, Chan G, et al. An integrated genomic-based approach to individualized treatment of patients with advanced-stage ovarian cancer. J Clin Oncol 2007;25(5): 517–525. [PubMed: 17290060]

50. Bentink S, Haibe-Kains B, Risch T, et al. Angiogenic mRNA and microRNA gene expression signature predicts a novel subtype of serous ovarian cancer. PloS One 2012;7(2):e30269. [PubMed: 22348002]

51. Denkert C, Budczies J, Darb-Esfahani S, et al. A prognostic gene expression index in ovarian cancer—Validation across different independent data sets. J Pathol 2009;218(2):273–280. [PubMed: 19294737]

52. Yoshihara K, Tsunoda T, Shigemizu D, et al. High-risk ovarian cancer based on 126-gene expression signature is uniquely characterized by downregulation of antigen presentation pathway. Clin Cancer Res 2012 10.1158/1078-0432.CCR-11-2725

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

53. Pils D, Hager G, Tong D, et al. Validating the impact of a molecular subtype in ovarian cancer on outcomes: a study of the OVCAD Consortium. Cancer Sci 2012;103(7):1334–1341. [PubMed: 22497737]

54. Beer DG, Kardia SL, Huang C-C, et al. Gene-expression profiles predict survival of patients with lung adenocarcinoma. Nature Med 2002;8(8).

55. Curtis C, Shah SP, Chin S-F, et al. The genomic and transcriptomic architecture of 2,000 breast tumours reveals novel subgroups. Nature 2012;486(7403):346–352 [PubMed: 22522925]

56. Sorlie T, Tibshirani R, Parker J, et al. Repeated observation of breast tumor subtypes in independent gene expression data sets. ProcNatl Acad Sci USA 2003;100(14):8418–8423.

57. Cancer Genome Atlas Research Network, et al. Integrated genomic analyses of ovarian carcinoma. Nature 2011;474(7353):609–615. [PubMed: 21720365]

58. Muller H-G, Wang J-L. Hazard rate estimation under random censoring with varying kernels and bandwidths. Biometrics 1994;50:61–76. [PubMed: 8086616]

59. Cortes C, Vapnik V. Support-vector networks. Machine Learn 1995;20(3):273–297.

60. Liu X, Wang L, Huang G-B, Zhang J, Yin J. Multiple kernel extreme learning machine. Neurocomputing 2015;149:253–264.

61. Tang J, Deng C, Huang G-B. Extreme learning machine for multilayer perceptron. IEEE Trans Neural Netw Learn Syst 2016;27(4):809–821. [PubMed: 25966483]

62. Ding S, Guo L, Hou Y. Extreme learning machine with kernel model based on deep learning. Neural Comput Appl 2017;28(8):1975–1984.

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

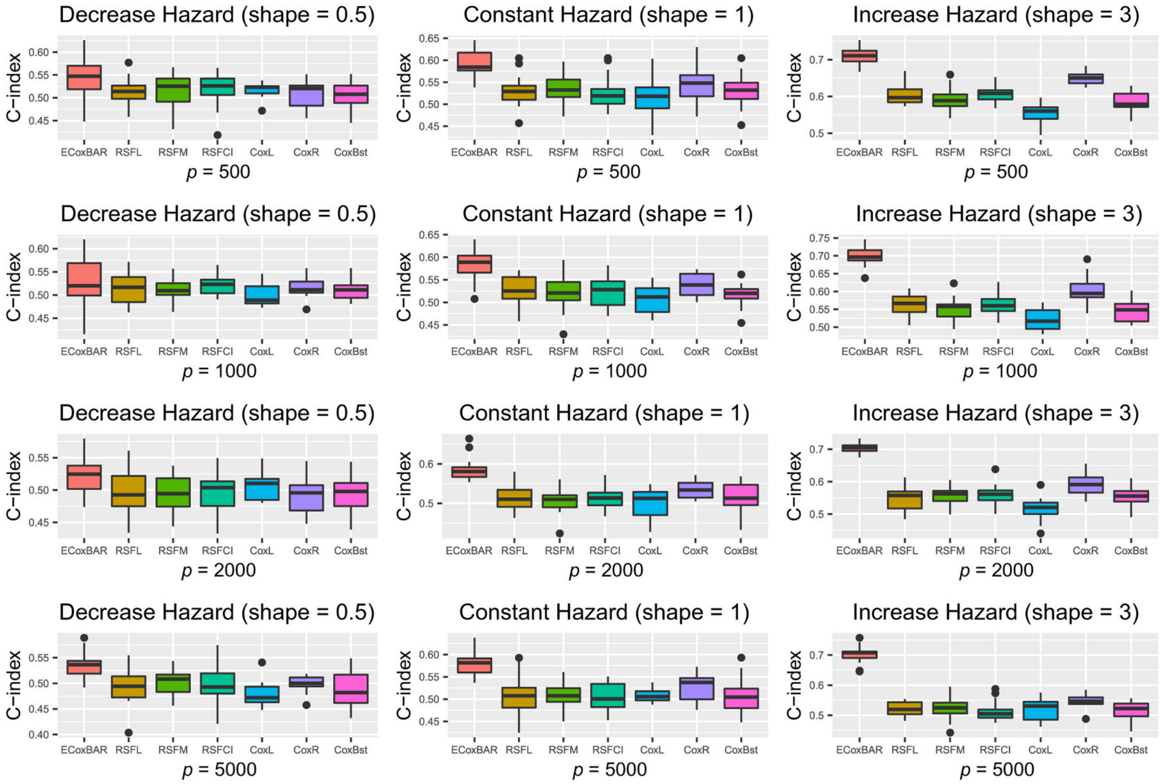

**FIGURE 1.**

Predictive performance in terms of C-index when $p$=500,1000,2000,5000 [Colour figure can be viewed at wileyonlinelibrary.com]

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

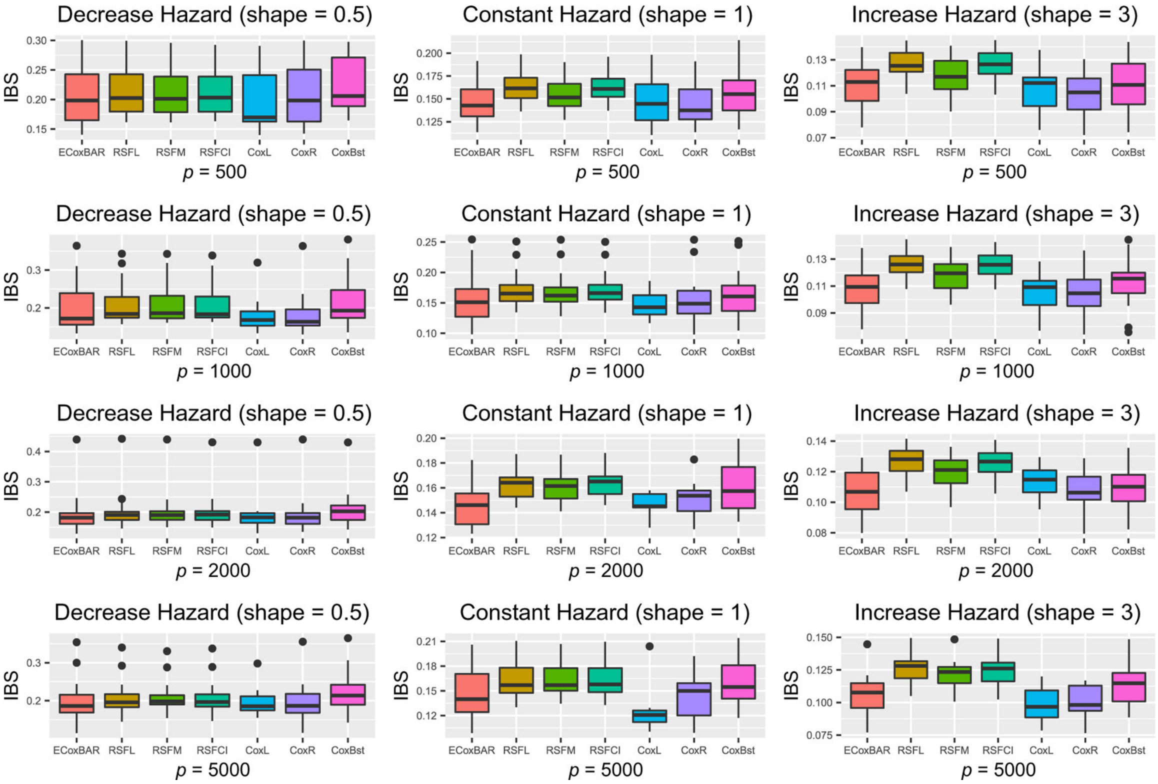

**FIGURE 2.**

Predictive performance in terms of integrated Brier score (IBS) when $p =$ 500,1000,2000,5000 [Colour figure can be viewed at wileyonlinelibrary.com]

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

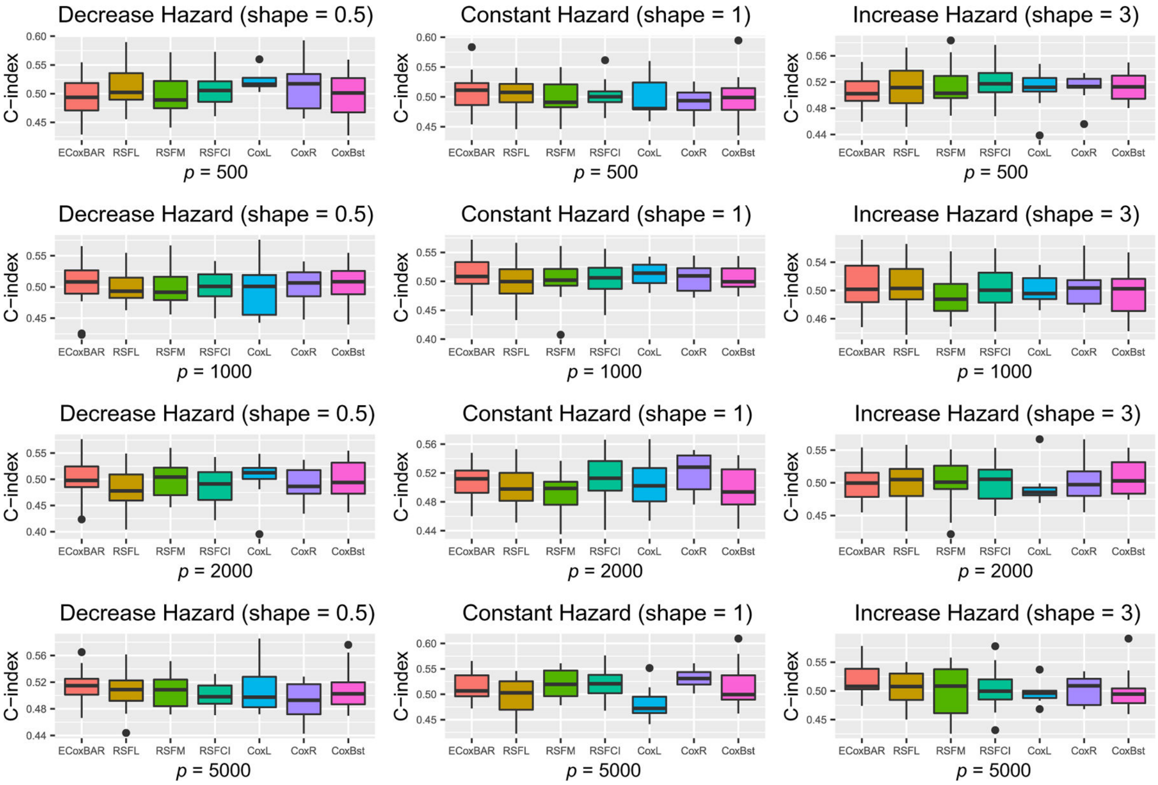

**FIGURE 3.**

Predictive performance in terms of C-index when $p$ =500,100, 2000,5000 [Colour figure can be viewed at wileyonlinelibrary.com]

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

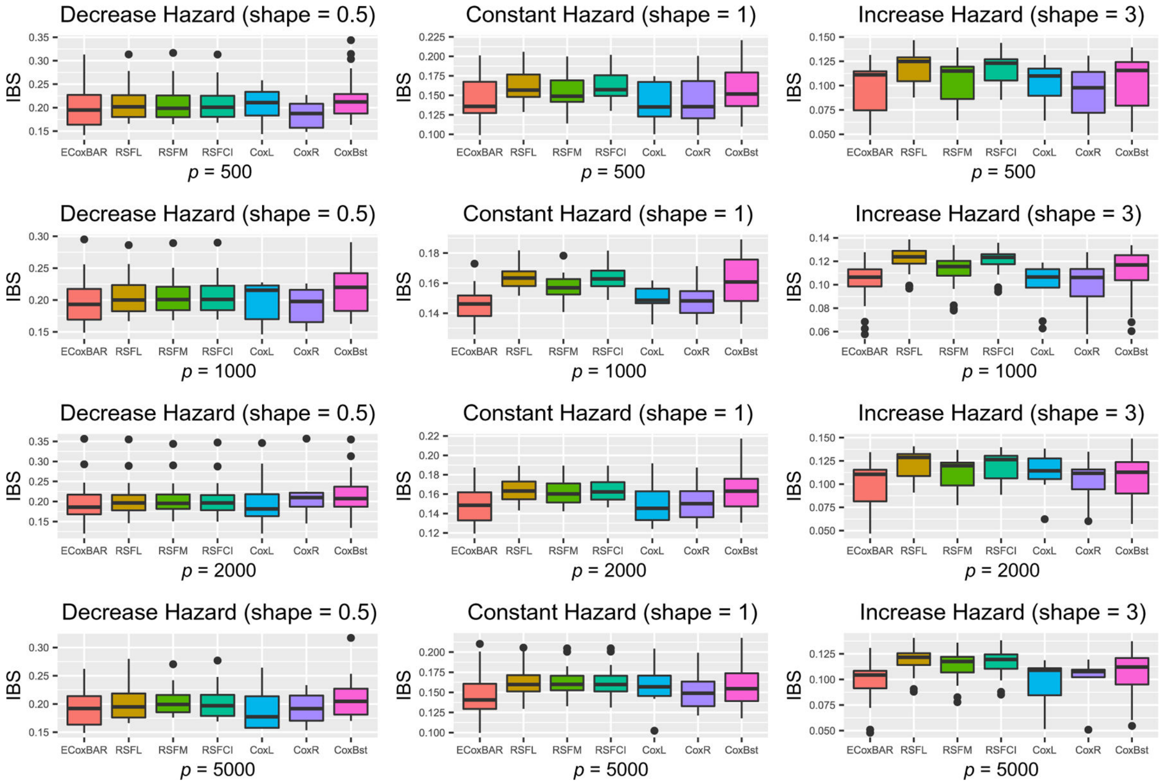

**FIGURE 4.**

Predictive performance in terms of integrated Brier score (IBS) when $p = 500, 1000, 2000, 5000$ [Colour figure can be viewed at wileyonlinelibrary.com]

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

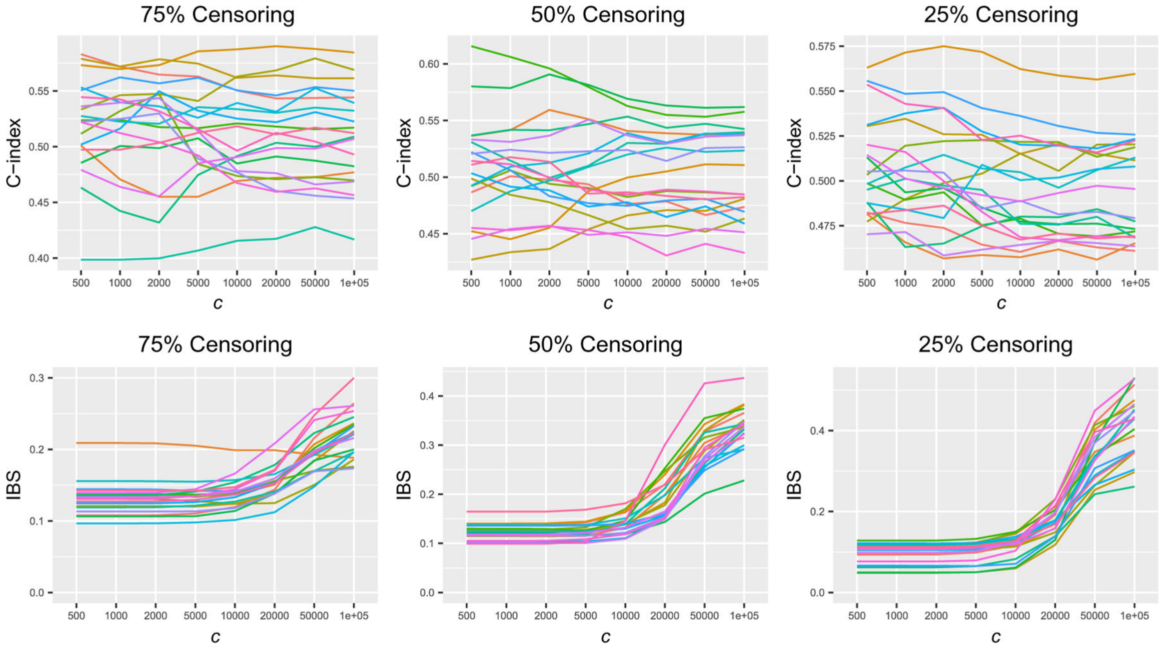

**FIGURE 5.**

Sensitivity of parameter *c* in ELMCoxBAR with radial basis function kernels. Each line represents results from a simulated dataset [Colour figure can be viewed at wileyonlinelibrary.com]

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

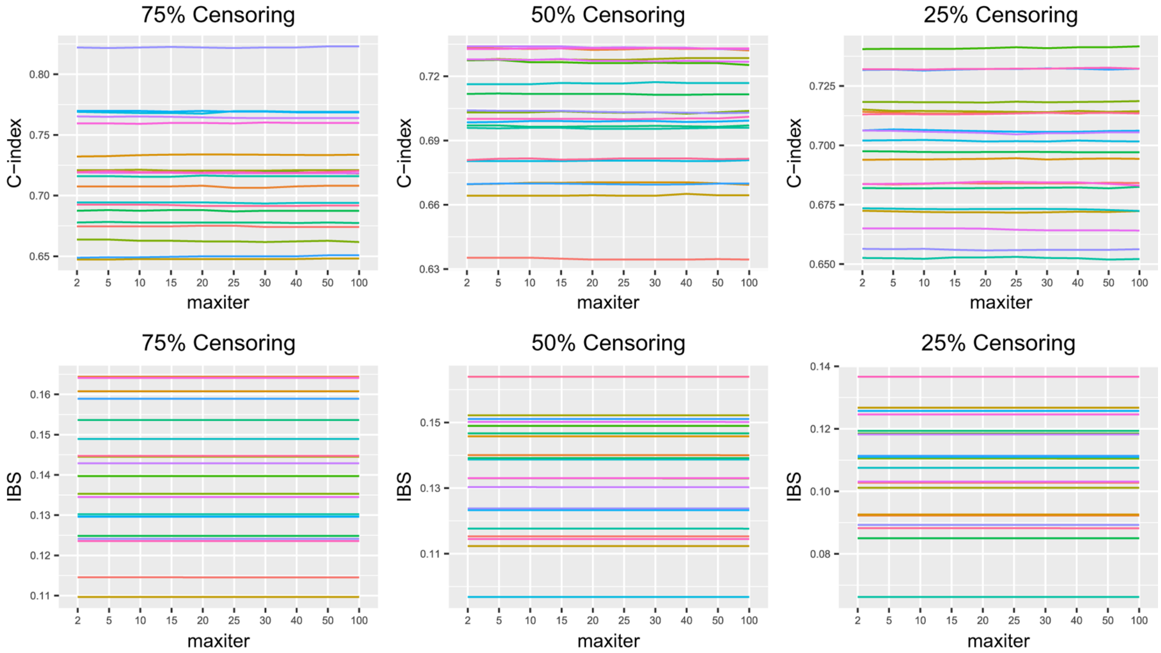

**FIGURE 6.**

Sensitivity of parameter *maxiter* in ELMCoxBAR with radial basis function kernels. Each line represents results from a simulated dataset. IBF, integrated Brier score [Colour figure can be viewed at wileyonlinelibrary.com]

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

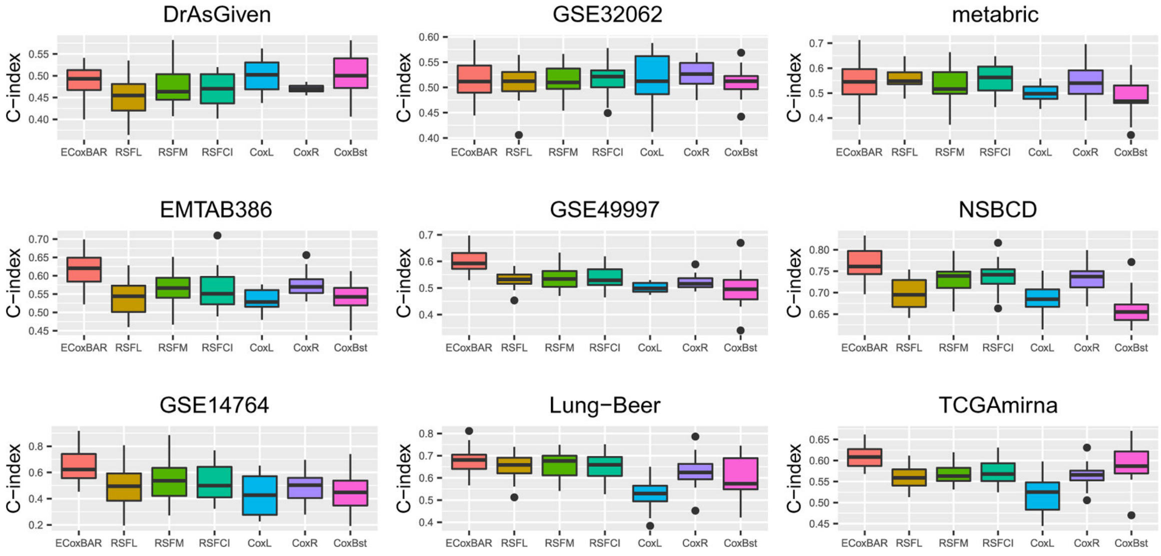

**FIGURE 7.**

Boxplots of performance in terms of C-index [Colour figure can be viewed at wileyonlinelibrary.com]

Author Manuscript
Author Manuscript
Author Manuscript
Author Manuscript

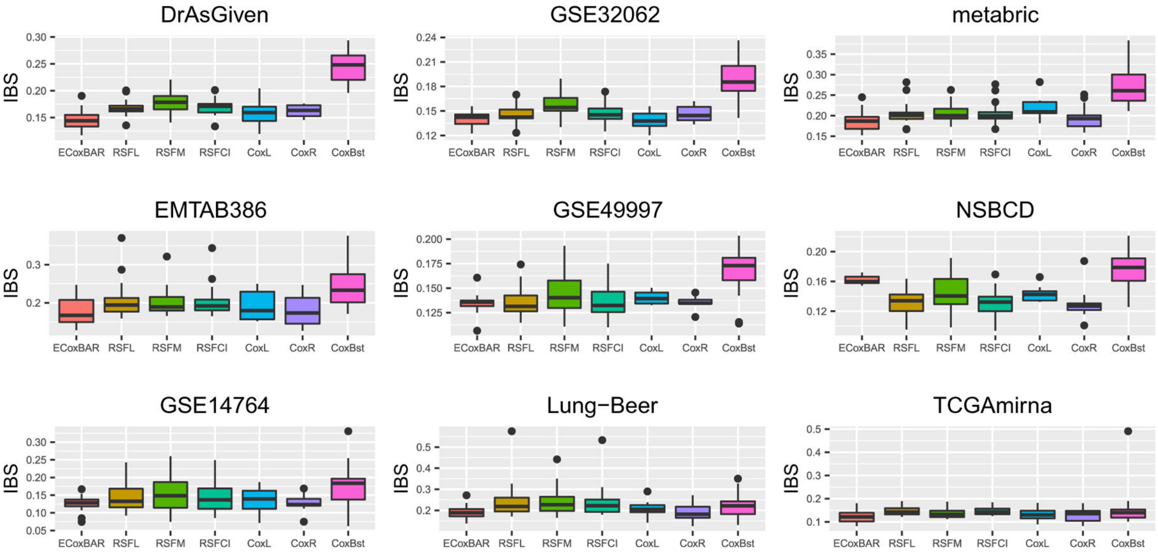

**FIGURE 8.**

Boxplots of performance in terms of integrated Brier score (IBS) [Colour figure can be viewed at wileyonlinelibrary.com]

Author Manuscript
Author Manuscript
Author Manuscript
Author Manuscript

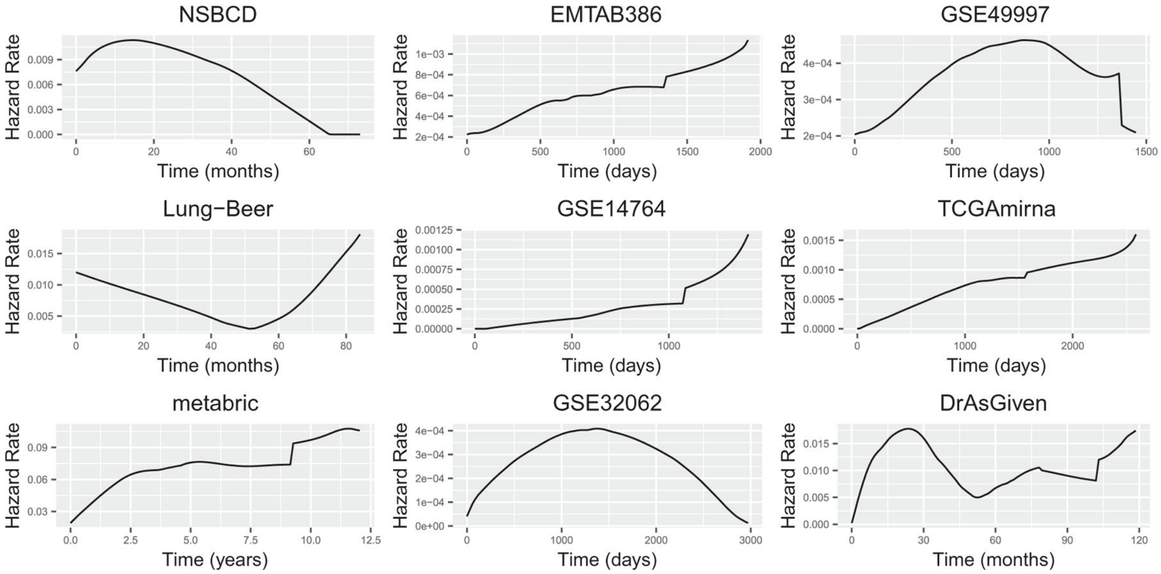

**FIGURE 9.**

Hazard functions over time of all benchmark datasets

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

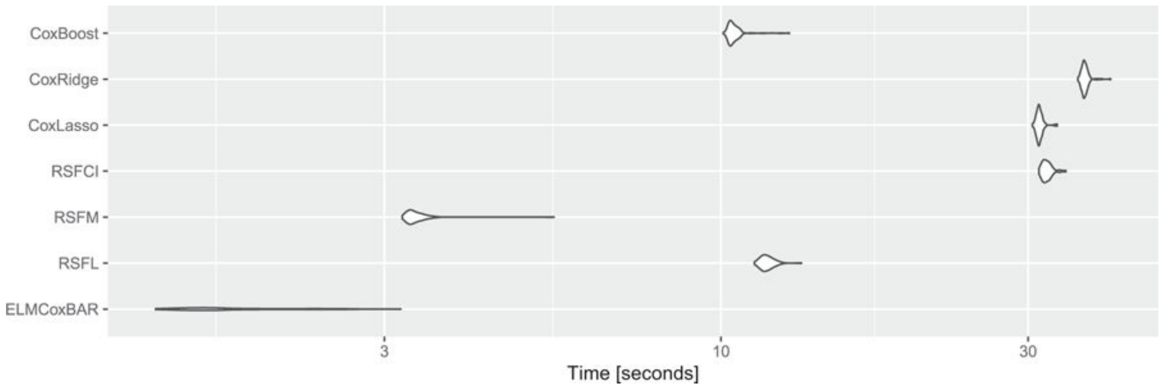

**FIGURE 10.**

Computing time taken for training and testing of all compared models

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

## TABLE 1

Summary of benchmark datasets

| Dataset | Sample Size | Covariates | Censoring Rate |
|---|---|---|---|
| DrAsGiven | 117 | 22122 | 41.88% |
| EMTAB386 | 107 | 10 364 | 44.86% |
| GSE14764 | 42 | 13112 | 78.57% |
| GSE32062 | 204 | 20112 | 56.86% |
| GSE49997 | 193 | 16 057 | 70.98% |
| Lung-Beer | 86 | 7131 | 72.09% |
| metabric | 115 | 2002 | 57.39% |
| NSBCD | 115 | 549 | 66.96% |
| TCGAmirna | 187 | 816 | 32.09% |

Author Manuscript Author Manuscript Author Manuscript Author Manuscript

# TABLE 2

Training time (milliseconds) of all compared models

| | Min | Lower Quartile | Mean | Median | Upper Quartile | Max |
|---|---|---|---|---|---|---|
| ELMCoxBAR | 889.1731 | 935.7729 | 988.7739 | 962.6211 | 1018.655 | 1380.299 |
| RSFL | 10485.6006 | 10 788.1589 | 11021.9918 | 10 999.2022 | 11 199.8 | 11 921.978 |
| RSFM | 2676.0311 | 2782.1645 | 2883.5454 | 2868.3779 | 2989.448 | 3139.939 |
| RSFCI | 30 021.0063 | 30 383.3676 | 30 583.0262 | 30 525.8739 | 30 689.14 | 31 794.927 |
| CoxLasso | 30 176.6343 | 30 771.7029 | 30 994.7853 | 30 957.5572 | 31 184.116 | 32 035.022 |
| CoxRidge | 35 707.3309 | 36 218.0172 | 36 426.4145 | 36 416.9546 | 36 654.161 | 37 205.278 |
| CoxBoost | 9945.9276 | 10 201.1658 | 10 318.3659 | 10 307.159 | 10 416.927 | 10 798.885 |