A Survival Ensemble of Extreme Learning Machine

Hong Wang[a,b], Jianxin Wang[a,*], Lifeng Zhou[c]

[a] *School of Information Sciences and Engineering, Central South University, China*
[b] *School of Mathematics and Statistics, Central South University, China*
[c] *School of Economics and Management, Changsha University, China*

Abstract

Due to the fast learning speed, simplicity of implementation and minimal human intervention, extreme learning machine has received considerable attentions recently, mostly from the machine learning community. Generally, extreme learning machine and its various variants focus on classification and regression problems. Its potential application in analyzing censored time-to-event data is yet to be verified. In this study, we present an extreme learning machine ensemble to model right-censored survival data by combining the Buckley-James transformation and the random forest framework. According to experimental and statistical analysis results, we show that the proposed model outperforms popular survival models such as random survival forest, Cox proportional hazard models on well-known low-dimensional and high-dimensional benchmark datasets in terms of both prediction accuracy and time efficiency.

Keywords: survival ensemble, extreme learning machine, time-to-event data, censored data, Buckley-James transformation

1. Introduction

Survival analysis focuses on modelling time-to-event data which are ubiquitous in the fields of biomedical sciences, health-care industry and financial economics. One of the major challenging problems when dealing with such

*Corresponding Author. School of Information Sciences and Engineering, Central South University, China

Email addresses: wanghong@ucla.edu (Hong Wang), jxwang@csu.edu.cn (Jianxin Wang), lfzhou@csu.edu.cn (Lifeng Zhou)

survival data is that the event of interest (death or occurrence of a disease) may not always be observed due to various reasons such as patients' withdrawals or loss of contact. These incomplete observations of event times give rise to the so-called censored data problem. This problem makes modelling time to event data more complicated compared to standard regression approaches. Parametric models(Weibull, Gamma, etc) or semi-parametric models such as Cox-proportional hazards models [1, 2] and their variants [3, 4, 5], could be useful and have been discussed in details in the literature. Usually, a partial likelihood approach is generally adopted to approximate the full likelihood based on different assumptions on the survival data. For example, in the most prevalent Cox model, the effect of the covariates with respect to the hazard rate is assumed to be multiplicative and there is a constant hazard ratio over time. These underlying assumptions, however, are not easy to satisfy and/or hard to verify in practice. Therefore, non-parametric models including neural networks [6, 7], survival trees [8, 9], survival ensembles [10], survival forests [11], and smoothing splines boosting [12] are developed to relax or remove underlying restrictive assumptions.

Neural networks have a recognized strength in dealing with complex interactions between covariates in classification and regression scenarios. However, the application of neural networks to survival analysis requires necessary transformation of the training data and subtle modifications to the network structures, both of which are non-trivial. In early attempts of incorporating neural network into survival analysis, the survival time was supplied to the neural network as an additional covariate [13] or the outcome was coded as an uncensored discrete variable with all censored samples omitted [14]. In [15], a piecewise constant hazard approach was proposed in which survival times are grouped into time intervals and the hazard is assumed to be constant for each interval. With a hidden layer of nodes and a proper activation function, this approach also supports non-linear effects. In [6], a Cox-like model in which a neural network output is used in place of usual linear combinations of covariates. This method keeps the proportional hazard nature of the Cox model while providing the ability to model nonlinear interactions. A so-called partial logistic regression model was later developed in [7], in which the time interval is treated as an input variable in a standard feed forward network, and conditional failure probabilities are estimated by smoothed discrete hazards. Several extensions to this approach have been proposed in [16, 17, 18, 19]. A neural network approach based on imputation of survival times via the Buckley-James estimator is discussed in [20].

A simulation study in [20] also revealed that the Buckley-James imputation based neural networks performed as well as Cox-neural networks [6] in most cases.

Regardless of what strategy is used to extend neural networks to accommodate right-censored data, most survival neural network methods only consider the standard single-hidden layer or multi-hidden layer feedforward neural networks, the bottleneck of which is their slow model training speed.

In this article, we want to explore the plausibility of extending the extreme learning machine (ELM) [21, 22], an emerging fast classification and regression learning algorithm for single-hidden layer feedforward neural networks (SLFN), to analysis of right-censored survival data. The main concept behind the ELM is the replacement of a computation-intensive procedure of finding the input weights and bias values of the hidden layer by just random initializations. The subsequent output weights of the network can be calculated analytically and efficiently using a least square approach and this usually implies a fast model training speed. Given enough hidden neurons, ELM is proven to be a universal function approximator [23].

Before applying ELM to censored survival data, two vital issues have to be properly addressed. First, ELM itself does not handle censored survival times and simple exclusion of censored observations from training data will result in significant biases in event predictions. Second, ELM is somewhat sensitive to random initialization of input-layer weights and hidden-layer biases, and this might will incur unstable predictions [24]. In this research, we deal with the first issue by replacing the survival times of censored observations with surrogate values using the Buckley-James estimator [25, 26], which is a censoring unbiased transformation in nature. For the second issue, we will adopt a well-established random forest ensemble learning framework [27] which is most effective when the base learner is unstable. In our approach, the base learners in the original random forest is changed from decision trees to ELM neural networks.

The rest of the paper is organized as follows. Section 2 overviews the Buckley-James estimator, extreme learning machine and random forest, and then we propose a novel survival neural network ensemble using extreme learning machine in Section 3. Experimental setup and result analysis are described in Section 4. Finally, in Section 5 we conclude the paper.

2. Preliminaries

In this section, we briefly describe the Buckley-James estimator, extreme learning machine, and random forest. In the next section, we shall develop a novel survival neural network ensemble algorithm using the random forest framework.

2.1. The Buckley-James Estimator

Suppose that we have a training data D of n observations and sample covariates $\mathbf{x}$ are p-dimensional vectors namely, $\mathbf{x}_i = (x_{i1}, x_{i2}, \cdots, x_{ip}), i \in 1, 2, \cdots, n$. The Buckley-James estimator [25] assumes that the transformed survival time (e.g. a monotone transformation such as the logarithm transform) T_i follows a linear regression

$$T_i = \alpha + \mathbf{x}_i\beta + \epsilon_i, \quad i = 1, \cdots, n \tag{1}$$

where ϵ_i is the i.i.d error term with $E(\epsilon_i) = 0$ and $Var(\epsilon_i) = \sigma^2$. For simplicity, we can absorb the unknown intercept α into ϵ_i and a new term would be $\xi_i = \alpha + \epsilon_i$. Consequently, the above model (1) could be reformulated as

$$T_i = \mathbf{x}_i\beta + \xi_i, \quad i = 1, \cdots, n \tag{2}$$

If there were no censoring, parameters of the above model could be estimated via an ordinary least square approach or its regularized extensions. However, in many cases, only censored observations from Y are available. In the case of right-censored data, we can only observe $(Y_i, \delta_i, \mathbf{x}_i)$, where $Y_i = min(T_i, C_i)$, C_i is the transformed censoring time and $\delta_i = I(T_i \leq C_i)$, the censoring indicator. And, in the presence of censoring, the usual least square approach is not applicable. Buckley and James [25] proposed to approximate those censored survival times by their conditional expectations and define the newly imputed survival times as

$$Y_i^* = Y_i\delta_i + E(T_i|T_i > Y_i, \mathbf{x}_i)(1 - \delta_i), \quad i = 1, \cdots, n \tag{3}$$

For uncensored observations, $\delta_i = 1$ and $Y_i^* = T_i$; for censored observations, $\delta_i = 0$ and $Y_i^* = E(T_i|T_i > Y_i, \mathbf{x}_i)$. Hence, it is easy to verify that $E(Y_i^*) = E(T_i)$. The Buckley-James estimator calculates the conditional expectation given the censored survival time and the corresponding covariates by

$$
\begin{aligned}
E(T_i|T_i > Y_i, \mathbf{x}_i) &= E(\mathbf{x}_i\beta + \xi_i|\mathbf{x}_i\beta + \xi_i > Y_i) \\
&= \mathbf{x}_i\beta + E(\xi_i|\mathbf{x}_i\beta + \xi_i > Y_i) \\
&= \mathbf{x}_i\beta + E(\xi_i|\xi_i > Y_i - \mathbf{x}_i\beta) \\
&= \mathbf{x}_i\beta + \int_{Y_i-\mathbf{x}_i\beta}^{\infty} \frac{\xi dF(\xi)}{1 - F(Y_i - \mathbf{x}_i\beta)}
\end{aligned} \tag{4}
$$

where $F(\xi)$ is an estimator of the distribution function (e.g. the Kaplan-Meier estimator [28] $\hat{F}$) of ξ. Then, we have

$$
Y_i^* = Y_i\delta_i + (1 - \delta_i)\left(\mathbf{x}_i\beta + \frac{\sum\limits_{\xi_j > \xi_i} s_j\xi_j}{1 - F(\xi_i)}, \right), \quad i = 1, \cdots, n \tag{5}
$$

where s_j are steps of the estimated function $\hat{F}$. The unknown coefficients β in equation (5) can be computed through a straightforward iterative procedure. And in case of a high dimensional p, a regularized technique with the elastic net penalty proposed in [26] can be adopted.

Now that all survival times are "available" via an imputation of the Buckley-James estimator, in the following subsection, we will briefly describe how a state-of-the-art extreme learning machine neural networks with a fast learning speed works.

2.2. Extreme learning machine

Given n observations $(\mathbf{x}_i, y_i), i \in 1, 2, \cdots, n$, where $\mathbf{x}_i$ is the same as defined above and y_i is the imputed survival time with the corresponding inverse transformation, e.g. $y_i = exp(Y_i^*)$ in case of the logarithm transform. A single hidden layer feedforward network (SLFN) with L hidden nodes and an activation function $g(x)$ can be formulated as

$$
f_L(\mathbf{x}) = \sum_{j=1}^{L} w_j \cdot g(r_j \cdot \mathbf{x} + b_j) \tag{6}
$$

where $w_j (j \in 1, 2, \cdots, L)$ is the vector of output weights between the hidden layer and the output node, $r_j (j \in 1, 2, \cdots, L)$ is the set of weights connecting the input vector $\mathbf{x}$ to the hidden layer and b_j is the related bias term. In [21], a simple learning method called extreme learning machine (ELM) was proposed

to deal with the above single-hidden layer feedforward neural networks. In ELM, both r_j and b_j in equation (6) are random assigned values, namely, the original p-dimensional covariates space is mapping into an L-dimensional one through a random matrix. Denote the feature mapping $g(r_j \cdot \mathbf{x} + b_j)$ by $h_j(\mathbf{x})$, equation (6) can be simplified as

$$f_L(\mathbf{x}) = \sum_{j=1}^{L} w_j \cdot h_j(\mathbf{x}) = \mathbf{h}(\mathbf{x})\mathbf{w} \tag{7}$$

where $\mathbf{h}(\mathbf{x}) = [h_1(\mathbf{x}), \cdots, h_L(\mathbf{x})]$ and $\mathbf{w} = [w_1, \cdots, w_L]$. Consequently, output weights $\mathbf{w}$ can be analytically calculated by

$$\mathbf{w} = \mathbf{H}^{\dagger}\mathbf{y} \tag{8}$$

where $\mathbf{H}^{\dagger}$ is the Moore-Penrose generalized inverse of the hidden layer output matrix $\mathbf{H} = [\mathbf{h}^T(\mathbf{x}_1), \cdots, \mathbf{h}^T(\mathbf{x}_n)]$ and $\mathbf{y} = [y_1, \cdots, y_n]^T$. To make the solution of $\mathbf{w}$ more stable, a positive value can be added to the diagonal of $\mathbf{H}^T\mathbf{H}$ or $\mathbf{H}\mathbf{H}^T$, and the resultant solution becomes:

$$\mathbf{w} = \left(\frac{I}{C} + \mathbf{H}\mathbf{H}^T\right)^{-1}\mathbf{H}^T\mathbf{y} \tag{9}$$

If no mapping $\mathbf{h}(\mathbf{x})$ is specified, we can also define a kernel matrix for ELM as was done in [22]:

$$\Omega_{ELM} = \mathbf{H}\mathbf{H}^T : \Omega_{ELM_{i,j}} = \mathbf{h}(\mathbf{x}_i) \cdot \mathbf{h}(\mathbf{x}_j) = K(\mathbf{x}_i, \mathbf{x}_j) \tag{10}$$

Thus, if a kernel matrix (e.g. $K(\mu, \nu) = \exp(-\gamma\|\mu - \nu\|^2)$) is adopted, the output function of ELM can be written as

$$f(\mathbf{x}) = \begin{bmatrix} K(\mathbf{x}, \mathbf{x}_1) \\ K(\mathbf{x}, \mathbf{x}_2) \\ K(\mathbf{x}, \mathbf{x}_n) \end{bmatrix}^T \left(\frac{I}{C} + \Omega_{ELM}\right)^{-1}\mathbf{y} \tag{11}$$

Similar to all other kernel learning methods, users do not have to worry about the feature mapping function $\mathbf{h}(\mathbf{x})$. And, another advantage of usage of kernel matrix in ELM is that the number of hidden nodes L needs not be provided. Hereafter, we will use the kernel matrix version of ELM in the proposed approach.

2.3. Random forest

Random forest [27] is an ensemble approach combining bagging [29] and random subspace [30] techniques. In random forest, two kinds of randomizations are realized. First, a randomly generated bootstrap sample of the data is applied to grow a base learner called CART decision tree [31]. Second, when building the decision tree, a random selected subset of m covariates is chosen as split candidates from the full set of p covariates and typically we set $m = \sqrt{p}$. The above procedure is iterated a number of times to obtain a forest of decision trees. The finally decision is made by a simple average of all these highly uncorrelated decision trees.

Considerable empirical evidence has shown that random forest can approximate a variety classes of functions while maintaining a low generalization error and is regarded as one of state-of-the-art learning methods to date [32, 33]. In the proposed approach, we will also consider bagging and random subspace-the two primary resampling methods of random forest in building our ELM ensemble.

3. A Survival Ensemble of Extreme Learning Machine

Our proposed method addresses the censored data problem in survival analysis by uncensoring the survival times via the Buckley-James estimator and consequently a state-of-the-art machine learning algorithm with high prediction accuracy such as extreme learning machine can be applied. However, the randomly choice of parameters in hidden layer or the kernel matrix [34] of ELM might lead to unstable predictions. Hence, to improve the prediction accuracy of such a highly accurate yet unstable model, an ensemble learning framework would be a natural choice.

As is known, the success of an ensemble method lies in the diversity among all the base learners[35], thus in the proposed method, the most popular methods to achieve diversity from data such as bagging and random subspace are applied. More diversity is introduced through imputation of the censored observations via the Buckley-James estimator. In our approach, only a subset of covariates are considered in estimating the censored survival times for each base kernel ELM. The fact that different estimates might be made to the same censored training sample actually diversify the training data. In fact, according to the study of the DEcoratE (Diverse Ensemble Creation by Oppositional Relabeling of Artificial Training Examples) algorithm [36], a small portion of artificially generated wrong observations would generate a more

diverse ensemble. In this sense, even wrong predictions occasionally made by the Buckley-James estimator could improve the ensemble's performance.

The pseudo-code of the proposed survival ensemble of extreme learning machine (SE-ELM) algorithm is presented in Algorithm 1:

Algorithm 1 Survival Ensemble of Extreme Learning Machine

Given:
  Training data: $D = (\tau_i, \delta_i, \mathbf{x}_i), i = 1, \cdots, n$, where $\mathbf{x}_i$ is p-dimensional
  Testing data: The p-dimensional $\mathcal{X}_j, j = 1, \cdots, k$
  m: how many covariates used to train a base ELM
  L: Ensemble size
procedure SE-ELM(Training)
  while i in $1 : L$ **do**
    Randomly select m covariates from p-dimensional data, where $m = \lceil \sqrt{p} \rceil$ for simplicity.
    Save the corresponding covariate indexes into array $\mathcal{A}[i]$.
    Generate a bootstrap sample $D^b = (\tau^b, \delta^b, \mathbf{x}^{mb})$ with m selected covariates from D.
    Impute the censored survival times via the Buckley-James estimator to obtain a new dataset $D^* = (y, \mathbf{x}^{mb})$.
    Using D^* as the training set, train a base kernel ELM C_i.
  end while
  return The survival ensemble $C = \{C_i\}$
end procedure
procedure SE-ELM(Testing)
  while i in $1 : L$ **do**
    Select the same m covariates from all p covariates as stored in $\mathcal{A}[i]$ and denote the test samples by $\mathcal{X}^m$.
    Predict the above test samples $\mathcal{X}_j^m, j = 1, \cdots, k$ with C_i.
  end while
  Hence, the predicted survival time for the j-th observation is

$$\hat{t}_j = \frac{1}{L} \sum_{C_i \in C} C_i(\mathcal{X}_j^m)$$

end procedure

Since the "while" parts in both training and testing phases in Algorithm

1 can be executed concurrently, thus in case of big survival data, SE-ELM can be trained on a multi-core CPU or computer clusters in parallel to save time.

4. Results & Discussion

In this section, we first describe the datasets used in the experiments and then we present the performance metrics and statistical tests we have adopted. Finally, we give and discuss the experimental results.

4.1. Datasets

In this study, we want to test our SE-ELM algorithm's performance on well-known benchmark survival datasets extensively analyzed in the statistical literature. Both low-dimensional and high-dimensional survival datasets are considered in this study to show the effectiveness of the proposed algorithm.

Table 1 shows the characteristics of the total 6 low dimensional datasets used in the experiments.

Table 1: Descriptions of the low dimensional datasets

Dataset	Sample size	#Covariates	Censoring rate	Data source
Burn	154	15	69%	iBST
Lung	167	8	28%	survival
Myeloma	167	8	28%	survival
PBC	276	17	60%	randomForestSRC
StageC	134	6	64%	rpart
WPBC	194	32	76%	TH.data

All these low dimensional datasets are public available through their source R packages on CRAN (https://cran.r-project.org/) and their censoring rates vary from 28 % to 76%.

Table 2 shows the characteristics of the total 6 high dimensional datasets used in the experiments.

All these high dimensional datasets are public available through their R packages on Bioconductor (https://www.bioconductor.org/) or through the given web addresses. The dimension to sample size ratio of each dataset ranges from 725 to about 31 and the related censoring rate varies from 43 % to 82%.

Table 2: Descriptions of the high dimensional datasets

Dataset	Sample size	#Covariates	Censoring rate	Data source
DLBCL	240	7401	43%	http://user.it.uu.se/~liuya610/
LungBeer	86	7131	72%	http://user.it.uu.se/~liuya610/
TransBig	196	22292	68%	breastCancerTRANSBIG
UNT	62	44935	82%	breastCancerUNT
UPP	197	44938	78%	breastCancerUPP
VDX	197	22291	64%	breastCancerVDX

4.2. Performance metric and statistical tests

To measure the predictive accuracy of survival models, Harrell' s concordance index (C-index) [37], which measures the relative risks between patients, is adopted as our evaluation metric. The C-index is defined as the ratio of the number of concordant predictions(survival times or survival probabilities) over the number of possible pairs of observed survival times. Note that, C-index = 1 indicates the model has a perfect prediction, and C-index = 0.5 implies that the model is as good as a random predictor. Usually, a larger C-index implies a better performance of the model.

All the experiments results obtained are based on the 5-2 cross-validation procedure suggested by [38]. To test whether a model performs significantly better than other models, we mainly use two types of statistic tests: the non-parametric Friedman test and the Nemenyi post-hoc test [39]. If the p-value of the Friedman test is less than a threshold (say, 0.05), the null hypothesis that there is no significant difference among the compared survival models can be rejected and a Nemenyi post-hoc test can be adopted to find where the differences lie. If needed, a third Wilcoxon signed-rank pairwise test will also be applied to check the significance of the difference between two models.

4.3. Performance Comparison results

The proposed SE-ELM method is implemented in the R program language using RCpp and related packages. In the implementation of kernel matrix, the radial basis function (RBF) kernel is generally a first choice among the four basic kernels, i.e. linear, polynomial, RBF and sigmoid kernels. However, in case of high dimensional data, the linear kernel is often preferred as it can achieve a comparable performance to that of the RBF kernel with a much less training time [40]. Hence, to make our proposed method suitable for both low and high dimensional settings, a linear kernel matrix is chosen for SE-ELM in the following experiments.

In all experiments, calculation of C-index values, the Friedman test, the Nemenyi test and the Wilcoxon test are carried out using the "concordance.index" function from the "survcomp" R package, "friedman.test" function from the R "stat" package [41], the "posthoc.friedman.nemenyi.test" function from the R "PMCMR" package [42], and the "wilcox.test" from the R "base" package, respectively.

4.3.1. Performance comparison on low-dimensional data

On low-dimensional datasets, we compare SE-ELM with five popular survival models. The first three models are random survival forests(RSF) [11] with different splitting rules: RSF with log-rank rule(RSFL), RSF with log-rank score rule(RSFLS), and RSF with C-index rule (RSFCI) [43]. The fourth model is generalized boosted model (GBM) [44]. And the fifth model is Cox proportional hazard (Cox) model [1] and comparisons with these models are conducted with corresponding "randomForestSRC", "ranger", "gbm" and "survival" packages in R. For the ease of notation, survival models SE-ELM, RSFL, RSFLS, RSFCI, GBM and Cox are denoted by A, B, C, D, E and F, respectively when necessary. The default settings of all models in corresponding packages are adopted. For ensemble methods, i.e. RSFL, RSFLS, RSFCI and GBM, 500 trees are built.

The following Figure 1 reports the performance of SE-ELM, RSFL, RSFLS, RSFCI, GBM and Cox in term of C-index on all six low-dimensional datasets.

From Figure 1, one may observe that the proposed SE-ELM approach generally outperforms all other competing methods on these benchmark datasets and it is usually more stable than other methods as well.

The Friedman rank sum test statistic on these datasets is 57.706 and it is significant as the corresponding p-value is 3.616e-11. Thus, to find out which pairs of models are significantly different, we can compute the Nemenyi test statistics for different pairs of survival models. Here, we are only concerned with relative performance of the proposed model (A), and thus only Nemenyi test statistics related with model A are calculated.

The corresponding post hoc Nemenyi test results are shown in the following Table 3.

It can be seen that all these p-values are less than 0.05. Thus, in terms of C-index, there exists significant differences between the proposed method and the other four algorithms on these datasets. That is to say, based on

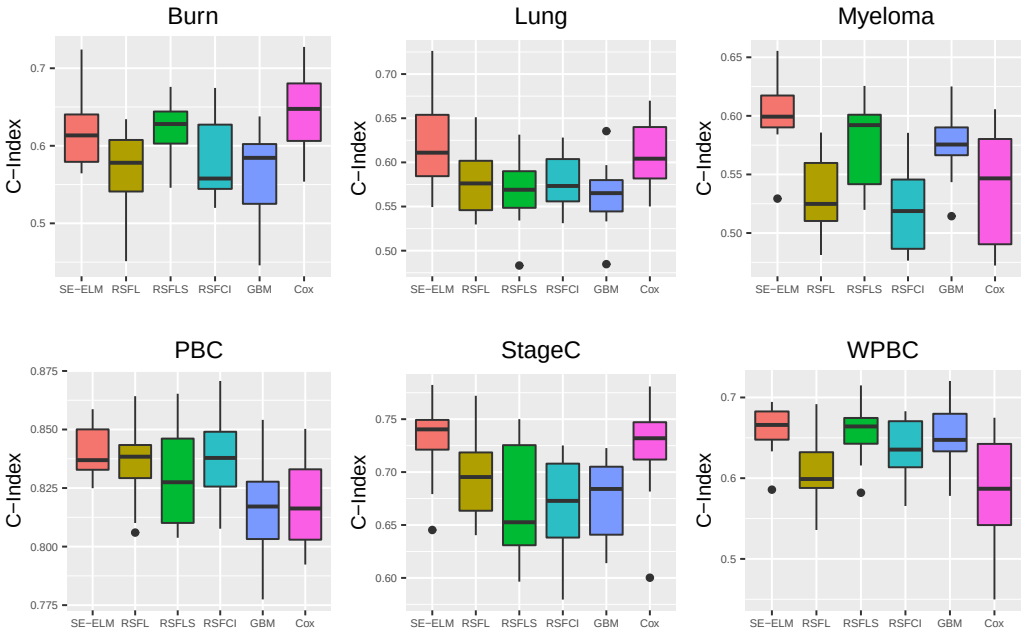

Figure 1: Performance in terms of C-index on low-dimensional data

Table 3: Nemenyi test results on low dimensional datasets

	z_{BA}	z_{CA}	z_{DA}	z_{EA}	z_{FA}
Nemenyi statistic	8.8329	4.6234	7.7977	8.9018	6.0726
p-value	6.32e-09	0.0138	5.24e-07	4.62e-09	2.54e-04

the results on all six low dimensional benchmark datasets, SE-ELM is significantly better than RSFL, RSFLS, RSFCI, GBM and Cox on the whole.

4.3.2. Performance comparison on high-dimensional data

When $p >> n$, the traditional Cox model does not work and in the high-dimensional setting, GBM faces a heavy computation burden and often crashes in normal desktop computers. Consequently, both of them are not unsuitable for high dimensional data. Here, we only keep RSF and consider two regularized Cox proportional hazard models: CoxLasso and CoxRidge for high dimensional comparisons.

Comparisons with regularized Cox model are conducted with "glmnet" packages in R. For ensemble methods, i.e. RSFL, RSFLS and RSFCI, 500

trees are built. For the ease of notation, survival models SE-ELM, RSFL, RSFLS, RSFCI, CoxLasso and CoxRidge are denoted by a, b, c, d, e and f, respectively when necessary.

The following Figure 2 reports the performance of SE-ELM, RSFL, RSFLS, RSFCI, CoxLasso and CoxRidge in term of C-index on all six high dimensional datasets.

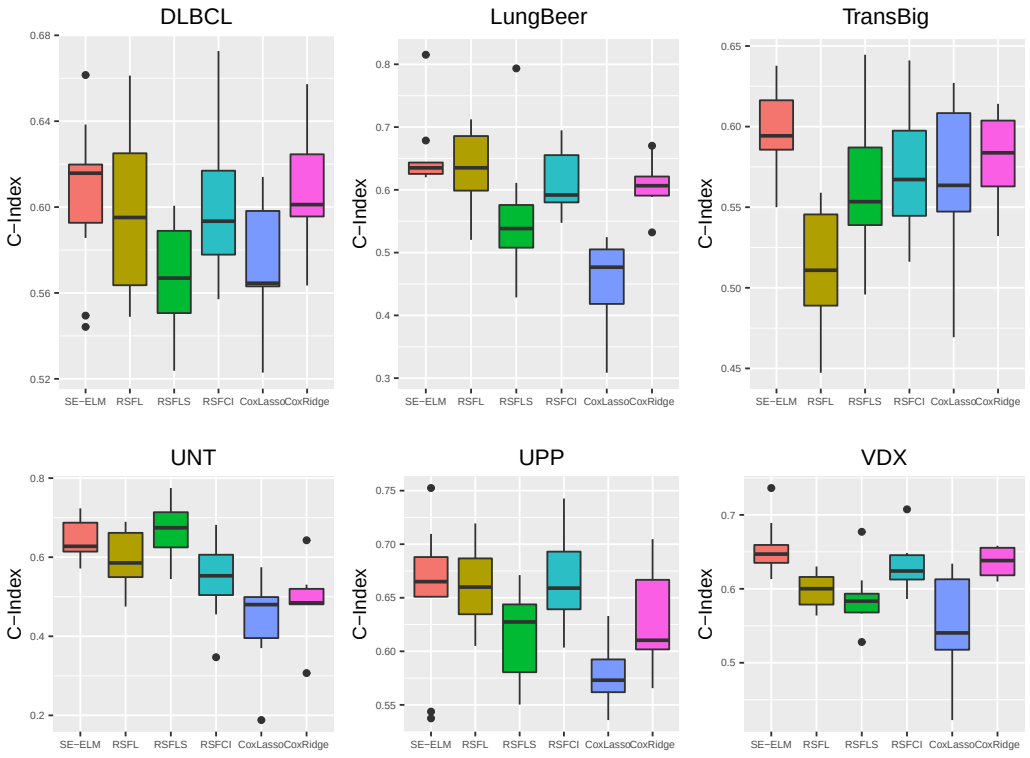

Figure 2: Performance in terms of C-index on high-dimensional data

From Figure 2, one may observe that the proposed SE-ELM approach generally outperforms all other competing methods on these benchmark datasets and it is usually more stable than other methods as well.

We also conduct the Friedman test on these high dimensional results and the corresponding statistic is 47.582 with a p-value of 4.323e-09. In the same way, the post hoc Nemenyi test is applied and the corresponding results are shown in the following Table 4.

It can be seen that all p-vales except the Nemenyi test between model a

Table 4: Nemenyi test results on low dimensional datasets

	z_{ba}	z_{ca}	z_{da}	z_{ea}	z_{fa}
Nemenyi statistic	5.5550	7.5217	4.3129	5.1065	0.6901
p-value	0.0012	1.56e-6	0.0278	0.0041	0.9966

and model f are less than 0.05. Thus, in terms of C-index, there exists significant differences between SE-ELM and RSFL, RSFLS, RSFCI, CoxLasso algorithms on these datasets.

Tough SE-ELM beats CoxRidge in term of average of mean C-index values($Mean_{SEELM}$=0.6358 and $Mean_{SEELM}$= 0.5954) and mean average rank($Rank_{SEELM}$=2.07 and $Rank_{CoxRidge}$=3.51) across all datasets, the difference between SE-ELM and CoxRidge is not significantly different according to the Nemenyi statistic and we need to make a further comparison. Consequently, a pairwise comparison between SE-ELM and CoxRidge is carried out using Wilcoxon rank sum test. The test result rejects the hypothesis of equivalence with low p-values(p_{fa}=0.0007444). Thus, SE-ELM is also significantly better than CoxRidge on these datasets.

In other words, from the performance on these high dimensional datasets, SE-ELM is significantly better than RSFL, RSFLS, RSFCI, CoxLasso and CoxRidge in terms of C-index.

4.4. Computation time

To illustrate the time efficiency of the proposed method, we also compare SE-ELM with other survival models in the execution times of both training and prediction. We conduct our experiments on a 64-bit Windows 7 system with a Intel Core i5-5200U Dual-Core 2.20GHz CPU and 8G RAM.

For simplicity, we only consider the most challenging UPP data with the largest dimension among all benchmark datasets. We use a popular and accurate R "microbenchmark" package [45] to benchmark the running time of all compared models and all the comparisons are evaluated 100 times. Figure 3 presents the corresponding computation time of all six models.

From Figure 3, one can clearly see that in terms of computation time, the proposed SE-ELM approach takes the first place with a mean running time of about 3.8 seconds; RSFL takes the second place with a mean execution time of 14.1 seconds; CoxRidge ranks the last and takes a mean computation time of 118.9 seconds. Hence, compared with other popular high dimensional

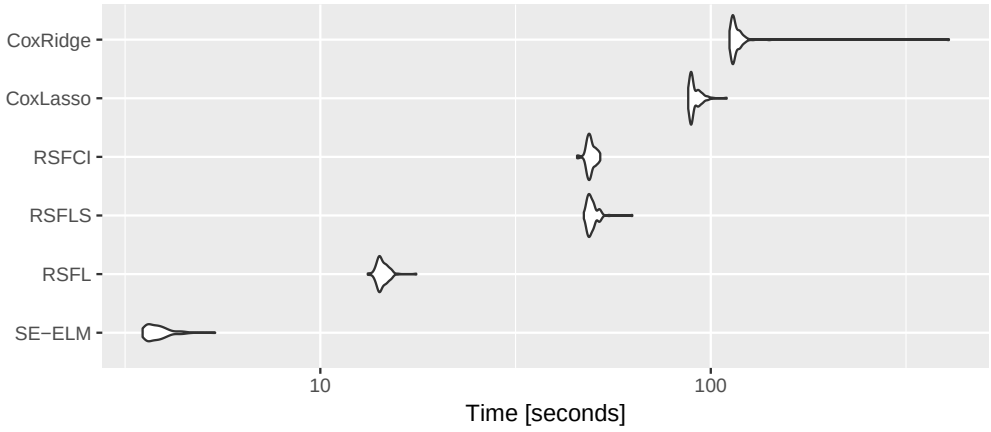

Figure 3: Comparison of computation time on UPP dataset

models, the proposed method takes much less time and generally obtains the best performance.

4.5. Parameter sensitivity

In addition to the above comparison experiments, we also want to examine the effects of parameter selection on the prediction accuracy of SE-ELM with different C, L and m on both low and high dimensional datasets.

Here, C refers to the user-specified regularization parameter in equation (9) which provides a tradeoff between the training error and the generalization error of a base ELM model, L, the ensemble size and m, the number of randomly selected covariates (random subspace) used to train each base model. For similar reasons stated in the comparison experiments, only the linear kernel matrix is considered for SE-ELM in later experiments.

4.5.1. The choice of regularization parameter C

First, we want to examine the sensitivity of SE-ELM on the regularization parameter C with values ranging from 2^{-15} to 2^{15} on both low and high dimensional data.

From Figure 4, one may observe that on low dimensional data, the performance of SE-ELM is rather poor when $C < 2^{-5}$. There exists a steady increase in performance when C increases from 2^{-5} to 1. However, when C is greater than 2^2, SE-ELM seems insensitive to the changes of C values. These results agree with the results obtained for regularized ELM in the regression

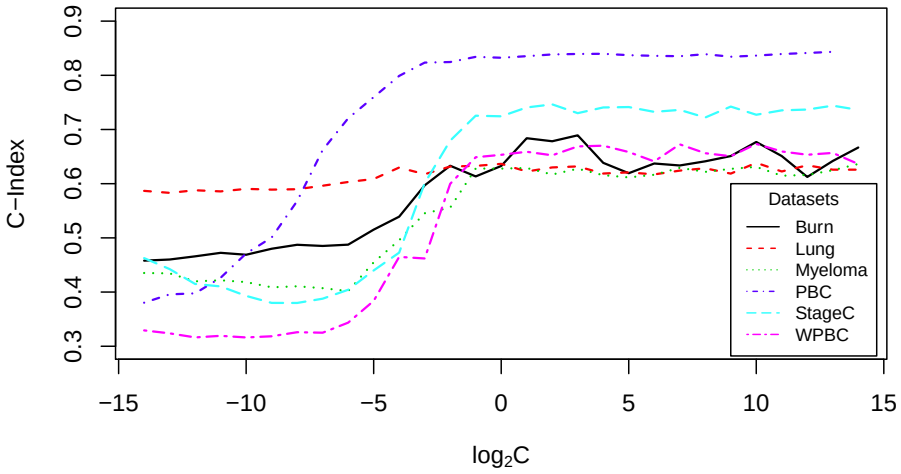

Figure 4: Performance with respect to C on low-dimensional datasets

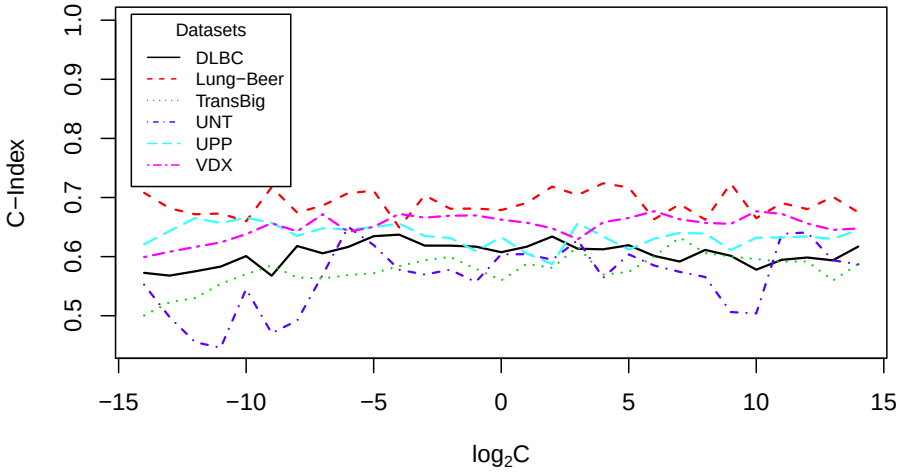

Figure 5: Performance with respect to C on high-dimensional datasets

context [46], i.e., the model's performance improves with the increase in C and keeps stable when C is above a certain threshold.

On the other hand, things are quite different on the high dimensional datasets. From Figure 5, we can see that when $C < 2^{-8}$, the performance of SE-ELM improves slowly but steadily with the increase of C except for the

UNT dataset. However, when $C > 2^{-8}$, SE-ELM seems rather insensitive to changes of C. When $2^{10} < C < 2^{14}$, SE-ELM often gets the best performance.

Clearly, SE-ELM is more sensitive to C with small values on low dimensional datasets. And in both low dimensional and high dimensional settings, lower values of C implies higher possibilities of underfitting. Based on the above findings, we suggest to set the default value of C in SE-ELM to 10000 to deal with both low and high dimensional data. We do not recommend an value much higher than 10000, as it may incur overfitting or an ill-conditioned kernel matrix.

4.5.2. The choice of ensemble size L

Next, we want to examine the influence of the ensemble size L with values ranging from 5 to 200 with a step size of 5 on SE-ELM on both low and high dimensional data.

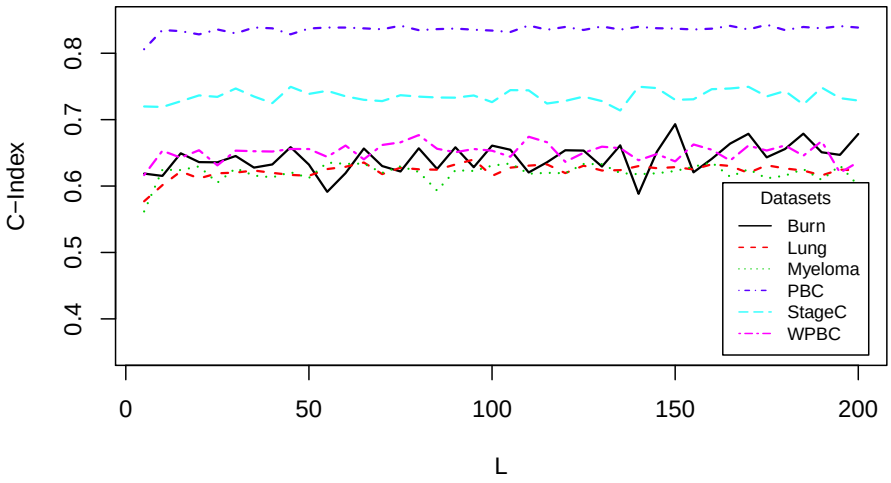

Figure 6: Performance with respect to L on low-dimensional datasets

From Figure 6, we can observe that SE-ELM is not sensitive to L when $L > 10$ on low dimensional datasets. While from Figure 7, the performance of SE-ELM generally becomes stable when $L > 80$ on high dimensional datasets. And when $L < 10$, SE-ELM usually has significantly lower C-index values on both low and high dimensional datasets as is clearly demonstrated in Figure 6 and Figure 7. We can also notice that a larger L value results in a higher

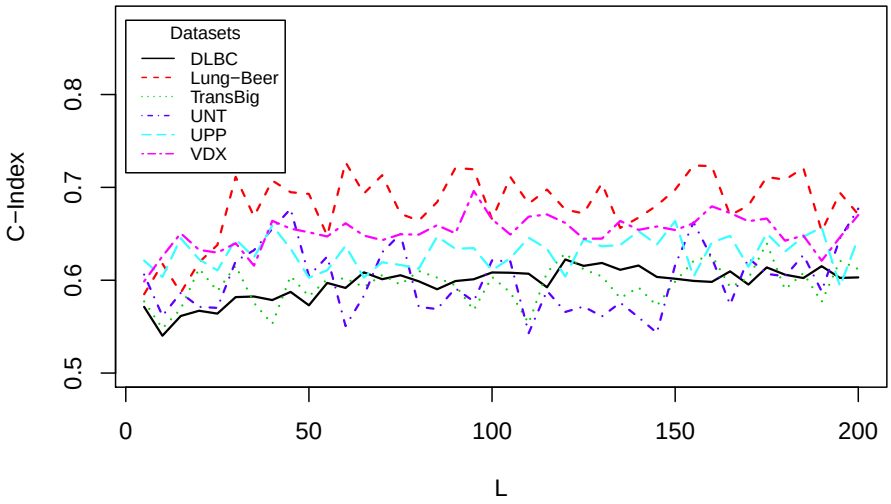

Figure 7: Performance with respect to L on high-dimensional datasets

survival prediction when $10 < L < 80$. And there is no significant difference observed between C-index values when $L \geq 80$.

Genially, a large $L(\geq 80)$ is a safe choice on both low and high dimensional data and usually implies less performance variability. However, a larger L always means a longer model training and predication time. Hence, to make a tradeoff between model efficiency and accuracy, we set $L = 100$ as the default ensemble size.

4.5.3. The choice of random subspace m

Finally, we want to examine the effect of randomly selected covariates m on SE-ELM on both low and high dimensional data. Different from the above two parameters which are generally independent of the training data, the value of m usually depends upon the dimensionality of the data.

For low dimensional datasets, we test the performance of SE-ELM with m values ranging from $min(\sqrt{p}, 3)$ to p in the experiments. In case of high dimensionality, a small m may result in a less accurate base model and $m > 5\sqrt{p}$ may incur a relatively long training time, thus we only test SE-ELM with m values ranging from 50 and $5\sqrt{p}$ for high-dimensional datasets.

Figure 8 and Figure 9 shows the performance of SE-ELM with different values of m on all low and high dimensional datasets, respectively.

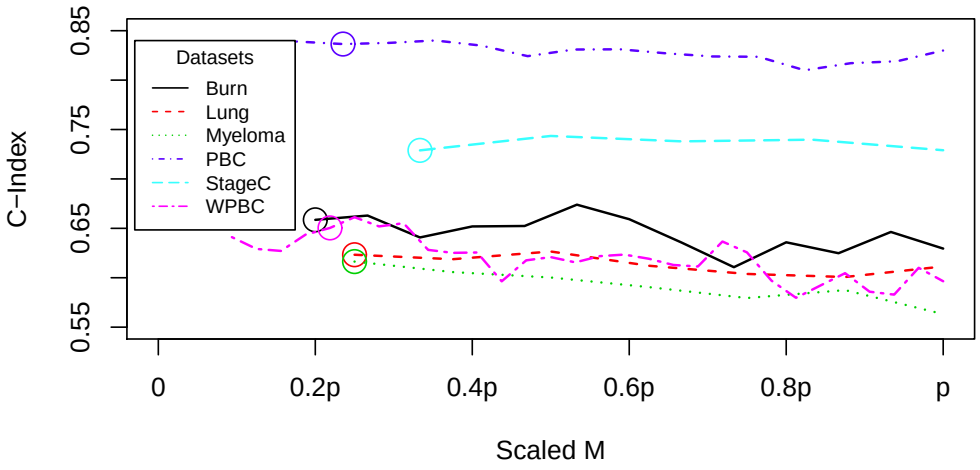

Figure 8: Performance with respect to m on low-dimensional datasets

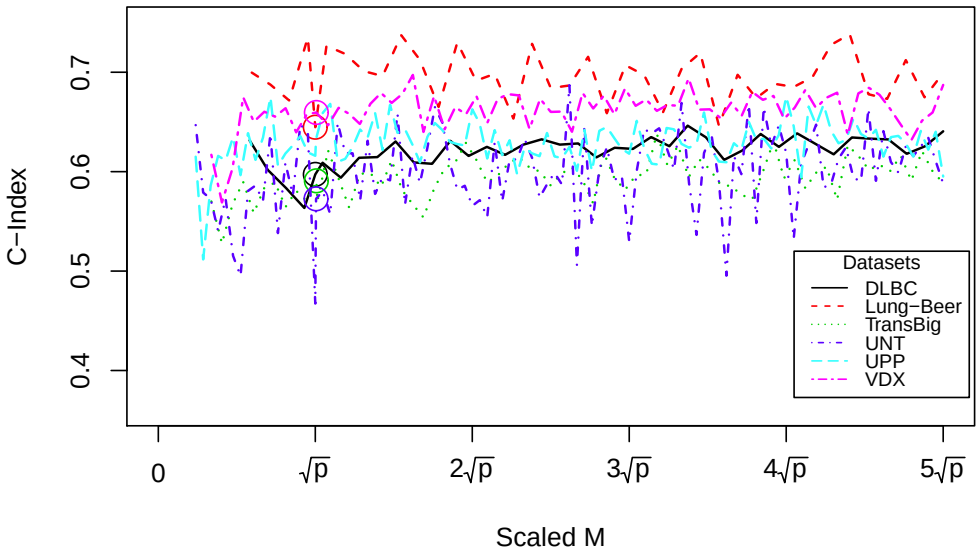

Figure 9: Performance with respect to m on high-dimensional datasets

As the dimensions of all datasets vary, all m values (on the x-axis) in both plots are scaled to range from 0 to p (low-dimensional) or $5\sqrt{p}$ (high-dimensional). Performance of the default values($m = \lceil \sqrt{p} \rceil$) of m on each

datasets is indicated by a circle.

From Figure 8, it is observed that SE-ELM is insensitive to the choice of m on low dimensional data except for the Myeloma dataset where the prediction accuracy is on the slowly decrease with the increase of m. It is also found that except for the values at the very beginning($m < \sqrt{p}$), no significant differences are observed for SE-ELM with different values of m on high dimensional data (Figure 9). The default values of $m = \sqrt{p}$ on all low and high dimensional datasets general yield fairly good results, though they are not optimal in most cases.

As we have shown in the above comparison experiments, SE-ELM has outperformed other popular survival models for these non-optimal values. As these values work well across all datasets, we may conclude that using these values as the default parameters have the potential to perform well in future study. Of course, one can use grid search or other cross-validation techniques to tune these parameters on particular cases for a higher model performance.

5. Conclusion

In conclusion, we have demonstrated how extreme learning machine can be applied to modelling right-censored survival data and have shown the superiority of the proposed non-parametric SE-ELM method to popular survival models in predictive capability on both high and low dimensional benchmark datasets. Furthermore, the fast training speed plus the parallel structure makes SE-ELM a very competitive method in deal with big survival data. We have also developed a R SE-ELM package called "ELMSurv" which will be soon available on CRAN or sent upon request.

In this research, only the standard Buckley-James estimator is used for imputation of the survival times. One may use other censoring unbiased transformation methods as well. Similar to [6], ELM could also be used as a Cox-like model in which the ELM output might be used in place of usual linear combinations of covariates. Besides the popular random forest framework adopted in SE-ELM, other ensemble methods using ELM for classification and regression tasks [47, 48, 49] could also be adapted to the survival analysis context. We leave the formal investigation of these aspects to a future study.

Acknowledgement

This work was supported in part by National Social Science Foundation of China (17BTJ019), Social Science Foundation for Young Scholars of Ministry of Education of China(15YJCZH166), Hunan Provincial Social Science Foundation of China(16YBA367), the scholarship from China Scholarship Council(CSC201606375129), China Postdoctoral Science Foundation (2017M612574) and Postgraduates Education Reform Fund(2016JGB25) at Central South University, China.

The funders had no role in the preparation of this article.

References

[1] C. R. David, Regression models and life tables (with discussion), Journal of the Royal Statistical Society 34 (1972) 187–220.

[2] D. R. Cox, D. Oakes, Analysis of survival data, Vol. 21, CRC Press, 1984.

[3] R. Tibshirani, et al., The lasso method for variable selection in the cox model, Statistics in medicine 16 (4) (1997) 385–395.

[4] J. Gui, H. Li, Penalized cox regression analysis in the high-dimensional and low-sample size settings, with applications to microarray gene expression data, Bioinformatics 21 (13) (2005) 3001–3008.

[5] N. Simon, J. H. Friedman, T. Hastie, R. Tibshirani, Regularization paths for cox's proportional hazards model via coordinate descent, Journal of Statistical Software 39 (5) (2011) 1–13.

[6] D. Faraggi, R. Simon, A neural network model for survival data, Statistics in medicine 14 (1) (1995) 73–82.

[7] E. Biganzoli, P. Boracchi, L. Mariani, E. Marubini, Feed forward neural networks for the analysis of censored survival data: a partial logistic regression approach, Statistics in medicine 17 (10) (1998) 1169–1186.

[8] M. LeBlanc, J. Crowley, A review of tree-based prognostic models, in: Recent Advances in Clinical Trial Design and Analysis, Springer, 1995, pp. 113–124.

[9] I. Bou-Hamad, D. Larocque, H. Ben-Ameur, A review of survival trees, Statistics surveys 5 (2011) 44–71.

[10] T. Hothorn, P. Bhlmann, S. Dudoit, A. Molinaro, M. J. Van Der Laan, Survival ensembles, Biostatistics 7 (3) (2006) 355–373.

[11] H. Ishwaran, U. B. Kogalur, E. H. Blackstone, M. S. Lauer, Random survival forests, The Annals of Applied Statistics 2 (3) (2008) 841–860.

[12] W. Lu, L. Li, Boosting method for nonlinear transformation models with censored survival data, Biostatistics 9 (4) (2008) 658–667.

[13] P. M. Ravdin, G. M. Clark, A practical application of neural network analysis for predicting outcome of individual breast cancer patients, Breast cancer research and treatment 22 (3) (1992) 285–293.

[14] M. H. Ebell, Artificial neural networks for predicting failure to survive following in-hospital cardiopulmonary resuscitation, Journal of family practice 36 (3) (1993) 297–304.

[15] K. Liestbl, P. K. Andersen, U. Andersen, Survival analysis and neural nets, Statistics in medicine 13 (12) (1994) 1189–1200.

[16] P. J. Lisboa, H. Wong, P. Harris, R. Swindell, A bayesian neural network approach for modelling censored data with an application to prognosis after surgery for breast cancer, Artificial intelligence in medicine 28 (1) (2003) 1–25.

[17] E. M. Biganzoli, P. Boracchi, F. Ambrogi, E. Marubini, Artificial neural network for the joint modelling of discrete cause-specific hazards, Artificial intelligence in medicine 37 (2) (2006) 119–130.

[18] F. Ambrogi, N. Lama, P. Boracchi, E. Biganzoli, Selection of artificial neural network models for survival analysis with genetic algorithms, Computational Statistics & Data Analysis 52 (1) (2007) 30–42.

[19] P. J. Lisboa, T. A. Etchells, I. H. Jarman, M. H. Aung, S. Chabaud, T. Bachelot, D. Perol, T. Gargi, V. Bourdès, S. Bonnevay, et al., Time-to-event analysis with artificial neural networks: An integrated analytical and rule-based study for breast cancer, Neural networks 21 (2) (2008) 414–426.

[20] A. Xiang, P. Lapuerta, A. Ryutov, J. Buckley, S. Azen, Comparison of the performance of neural network methods and cox regression for censored survival data, Computational statistics & data analysis 34 (2) (2000) 243–257.

[21] G.-B. Huang, Q.-Y. Zhu, C.-K. Siew, Extreme learning machine: theory and applications, Neurocomputing 70 (1) (2006) 489–501.

[22] G.-B. Huang, H. Zhou, X. Ding, R. Zhang, Extreme learning machine for regression and multiclass classification, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 42 (2) (2012) 513–529.

[23] G.-B. Huang, L. Chen, C. K. Siew, et al., Universal approximation using incremental constructive feedforward networks with random hidden nodes, IEEE Trans. Neural Networks 17 (4) (2006) 879–892.

[24] Y. Wang, F. Cao, Y. Yuan, A study on effectiveness of extreme learning machine, Neurocomputing 74 (16) (2011) 2483–2490.

[25] J. Buckley, I. James, Linear regression with censored data, Biometrika (1979) 429–436.

[26] S. Wang, B. Nan, J. Zhu, D. G. Beer, Doubly penalized buckley–james method for survival data with high-dimensional covariates, Biometrics 64 (1) (2008) 132–140.

[27] L. Breiman, Random forests, Machine learning 45 (1) (2001) 5–32.

[28] E. L. Kaplan, P. Meier, Nonparametric estimation from incomplete observations, Journal of the American statistical association 53 (282) (1958) 457–481.

[29] L. Breiman, Bagging predictors, Machine learning 24 (2) (1996) 123–140.

[30] T. K. Ho, The random subspace method for constructing decision forests, Pattern Analysis and Machine Intelligence, IEEE Transactions on 20 (8) (1998) 832–844.

[31] L. Breiman, J. Friedman, C. J. Stone, R. A. Olshen, Classification and regression trees, CRC press, 1984.

[32] M. Fernández-Delgado, E. Cernadas, S. Barro, D. Amorim, Do we need hundreds of classifiers to solve real world classification problems, J. Mach. Learn. Res 15 (1) (2014) 3133–3181.

[33] Y. Ren, L. Zhang, P. N. Suganthan, Ensemble classification and regression-recent developments, applications and future directions [review article], IEEE Computational Intelligence Magazine 11 (1) (2016) 41–53.

[34] L. Jia, S. Liao, Accurate probabilistic error bound for eigenvalues of kernel matrix, in: Asian Conference on Machine Learning, Springer, 2009, pp. 162–175.

[35] R. Polikar, Ensemble based systems in decision making, Circuits and Systems Magazine, IEEE 6 (3) (2006) 21–45.

[36] P. Melville, R. J. Mooney, Constructing diverse classifier ensembles using artificial training examples, in: Proceedings of the 18th international joint conference on Artificial intelligence, IJCAI'03, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2003, pp. 505–510.

[37] F. E. Harrell, K. L. Lee, D. B. Mark, Tutorial in biostatistics multivariable prognostic models: issues in developing models, evaluating assumptions and adequacy, and measuring and reducing errors, Statistics in medicine 15 (1996) 361–387.

[38] T. G. Dietterich, Approximate statistical tests for comparing supervised classification learning algorithms, Neural computation 10 (7) (1998) 1895–1923.

[39] J. Demšar, Statistical comparisons of classifiers over multiple data sets, The Journal of Machine Learning Research 7 (2006) 1–30.

[40] C. wei Hsu, C. chung Chang, C. jen Lin, A practical guide to support vector classification, Tech. rep., National Taiwan University (2016). URL `http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf`

[41] R Core Team, R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing, Vienna, Austria (2016). URL `https://www.R-project.org/`

[42] T. Pohlert, The Pairwise Multiple Comparison of Mean Ranks Package (PMCMR), r package (2014).
URL http://CRAN.R-project.org/package=PMCMR

[43] M. Wright, A. Ziegler, ranger: A fast implementation of random forests for high dimensional data in c++ and r, Journal of Statistical Software 77 (1) (2017) 1–17. doi:10.18637/jss.v077.i01.

[44] G. Ridgeway, The gbm package, R Foundation for Statistical Computing, Vienna, Austria.

[45] O. Mersmann, microbenchmark: Accurate Timing Functions, r package version 1.4-2.1 (2015).
URL https://CRAN.R-project.org/package=microbenchmark

[46] W. Deng, Q. Zheng, L. Chen, Regularized extreme learning machine, in: Computational Intelligence and Data Mining, 2009. CIDM'09. IEEE Symposium on, IEEE, 2009, pp. 389–395.

[47] N. Liu, H. Wang, Ensemble based extreme learning machine, IEEE Signal Processing Letters 17 (8) (2010) 754–757.

[48] H.-j. Lu, C.-l. An, E.-H. Zheng, Y. Lu, Dissimilarity based ensemble of extreme learning machine for gene expression data classification, Neurocomputing 128 (2014) 22–30.

[49] Q. Yu, M. Van Heeswijk, Y. Miche, R. Nian, B. He, E. Séverin, A. Lendasse, Ensemble delta test-extreme learning machine (dt-elm) for regression, Neurocomputing 129 (2014) 153–158.