New Variations of Random Survival Forests and Applications to Age-Related Disease Data

Tossapol Pomsuwan and Alex A. Freitas
tp346@kent.ac.uk, a.a.freitas@kent.ac.uk
School of Computing
University of Kent
Canterbury, UK

***Abstract*—This work addresses a type of survival prediction (or survival analysis) problem, where the goal is to predict the time passed until an individual is diagnosed with a certain age-related disease. Survival prediction is more challenging than standard regression because the former involves censored data, i.e. individuals who have not been diagnosed with the disease yet. Random Survival Forests (RSFs) are a powerful type of Random Forest algorithm developed specifically for survival analysis. In this work we investigate new variations of RSFs, namely variations in the node-splitting criterion and the leaf-node-prediction criterion. Results of experiments on 10 real-world survival prediction problems show that, although the variations in node-splitting criteria did not lead to significant differences in predictive performance, RSFs with a new proposed leaf-node-prediction criterion had significantly better predictive performance than standard RSFs.**

***Index Terms*—Random forest, Regression tree, Survival analysis, Censored data, Age-related diseases**

I. Introduction

Survival analysis consists of a set of statistical or machine learning methods concerned with analysing the time until the occurrence of an event of interest [1]. The occurrence of such event is often referred to as a “failure” in the literature, but the event of interest can be a “failure” or a “success”, or any other type of event of interest. To some extent, survival analysis is similar to linear regression analysis where the prediction of the value of a target (response) variable is computed from a set of features, since the time to an event occurrence can be considered a numerical target variable to be predicted. The important difference, however, is the presence of data censoring, i.e. uncertainty about whether an event has occurred, which cannot be effectively handled by traditional linear regression methods.

Consider, for example, Random Forests (RFs) for regression [2], which is the basis for Random Survival Forests (discussed later). RFs are a powerful type of technique which tends to obtain high predictive performance in regression tasks in general, using the power of an ensemble of decision trees to make more robust predictions. By default, RFs use the Root-Mean-Square-Error (RMSE) measure as the node-splitting criterion [2], which cannot cope with data censoring. Hence, some studies developed pre-processing methods to cope with data censorship before applying the algorithm in the model-training step. The complexity of these methods varies from the simplest approach of removing the censored data [3], [4] to more sophisticated statistical approaches, for example, computing Inverse Probability of Censoring (IPC) Weights [5], [6] and Pseudo Survival Values [7].

However, the most popular variation of RFs for survival analysis is the Random Survival Forests (RSF) algorithm [8], [9]. The RSF algorithm learns an ensemble of survival trees (decision trees adapted to survival analysis problems, considering censored data), and has been shown to outperform several methods in survival analysis [10]. Compared to the classical RF [2], RSF employs some statistical techniques which enable it to cope with the censoring issue. Therefore, in this paper we develop new variants of RSFs based around these techniques. More precisely, we focus on variants that modify two key components of RSFs, namely the node-splitting criterion and the leaf-node-prediction criterion, as follows.

First, regarding the node-splitting criterion, we investigate replacing the Log-rank test (the criterion in standard RSFs) by the Wilcoxon and Tarone-Ware tests. The latter two tests were proposed in [11] as alternative node-splitting criteria for learning a single survival tree, rather than for learning a RSF (as an ensemble of survival trees). Recent survival analysis studies employing the RSF algorithm are still using the Log-rank criterion in general, see e.g. [12]–[15]. Studies considering different node-splitting criteria in RSFs are relatively rare, and they have tended to focus on criteria other than the Wilcoxon and Tarone-ware ones. In particular, references [8] and [16] considered not only the standard Log-Rank test criterion, but also other node-splitting criteria, like conservation of events, the Log-Rank score and even random. In addition, reference [17] introduced oblique splits, applying the Log-Rank test to regularized Cox Proportional hazards (PH) models; and reference [18] proposed an AUC-based node-splitting criteria involving the well-known C-index. However, to the best of our knowledge, no previous work on RSFs has considered the Wilcoxon and Tarone-Ware node-splitting criteria which are investigated in this work.

Regarding the leaf-node-prediction criterion, we investigate replacing the criterion used in standard RSFs (the ensemble Cumulative Hazard Function, described later) by a new proposed criterion, which is a more direct and simpler estimate of the survival time for each individual, directly based on

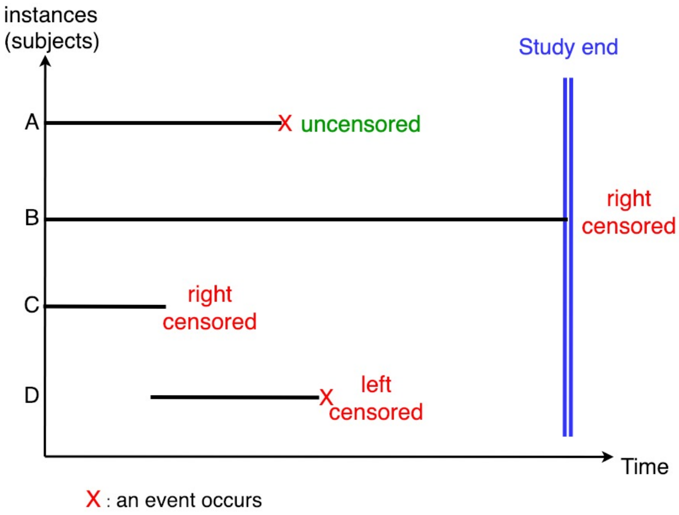


Fig. 1: Diagrammatic representation of uncensored and censored instances in survival analysis. “X” denotes an occurrence of the event of interest.

the mean of the target variable, but taking into account the presence of censored data.

In order to evaluate the predictive performance of the proposed variants of RSFs, we use datasets derived from the English Longitudinal Study of Ageing (ELSA) [19] and Survey of Health, Ageing and Retirement in Europe (SHARE) [20], [21] – surveys of ageing and quality of life among people aged 50 and over. This paper focuses on the biomedical data from these surveys, such as the results of blood tests and other data collected by nurses, and information about the subjects' age-related diseases.

This paper is organised as follows. Section II reviews background on survival analysis. Section III describes the proposed variants of RSF. Section IV describes the experimental methodology. Section V reports experimental results, and Section VI presents the conclusions.

II. Background

A. Censoring in Survival Analysis

As mentioned earlier, data censoring involves uncertainty about whether an event has occurred, which cannot be effectively handled by traditional linear regression methods. Hence, survival analysis methods have been developed to effectively cope with data censoring.

There are two main types of censoring, namely right censoring and left censoring [1]. The first and most common one is right censoring, where no event of interest occurred for a subject during the period in which he/she was observed in the study. There are essentially two reasons for the occurrence of right censoring. First, the patient was observed until the end of the study, and no event of interest occurred until that time (instance B in Fig.1). Second, the subject dropped out of the study or was lost to follow up before its end and no event of interest occurred before the drop out (instance C in Fig.1). Note that, in both cases of right censoring, the last observed time for a subject is a lower bound for the unknown event occurrence time. Left censoring occurs when a subject enters the study after its start (i.e., she/he starts to be observed after the start of the study) (instance D in Fig.1), so that we lack information about whether or not an event of interest occurred for the subject between the start of the study and the time the subject joined the study.

We focus on right-censoring (as opposed to left-censoring), which is generally encountered in medical research [1], such as the prediction of cancer survival [22]. More specifically, it is also a common type of censoring in the datasets used in our experiments, described later; since, a patient is often either lost to follow up before the end of the study or does not experience the event during the study.

B. Random Target Imputation Forests

Recall that standard random forests cannot cope with censored training data, where the target variable is a lower bound for the unknown survival time. To bypass this limitation, reference [23] proposed Random Target-Imputation Forests (RTIF). RTIF first calculates (based on the training data) an upper bound for the target variable for each censored instance. Then, when generating the bootstrap samples for learning the trees in the forest, for each censored instance in each bootstrap sample, the censored value of the target variable in that instance is replaced by a randomly generated value between the lower and upper bounds for that instance. Then, a standard regression tree method is used for learning each tree. RTIF stochastically imputes the values of the target variable for censored instances by using instance-specific lower and upper bounds. That is, for each instance (subject), lower and upper bounds for the target variable (an age-related disease) are calculated based on the data, and then, before learning each decision tree in the random forest, the censored values of the target variable in the training set for that tree are imputed with a randomly generated value within those lower bounds. Note that the generated target values of an instance will tend to vary across the bootstrap samples containing that instance, increasing the diversity of the trees in the forest (contributing to the forest's robustness).

In [23], RTIF was compared against a well-known random forest variation for survival prediction, named Survival Ensemble [24], which was also used in [6], [8], [14]. The results in [23] showed that RTIF significantly outperformed Survival Ensembles, so in this work we use RTIF as a baseline method in our experiments.

C. Nelson-Aalen Estimates of Cumulative Hazard Function

The Cumulative Hazard Function (CHF) defines the ratio of occurrence of the event of interest given that subjects survive past a certain amount of time. We review the CHF here since it is an important component of Random Survival Forests, the target type of algorithm in this work, as discussed later. The CHF is another measure of the population's survival

TABLE I: An example of the Nelson-Aalen estimates for a CHF

Time (t)	Risk set (n_t))	Failed (m_t)	Censored (c_t)	$H(t)$
0	1000	0	0	0
2	1000	90	10	$\frac{90}{1000} = 0.09$
3	900	300	100	$0.09 + \frac{300}{900} = 0.42$
7	500	250	50	$0.42 + \frac{250}{500} = 0.92$
9	200	50	50	$0.92 + \frac{50}{200} = 1.17$
10	100	10	90	$1.17 + \frac{10}{100} = 1.27$

distribution against time. Whilst the well-known Kaplan-Meier estimator analyses the survival distribution of the population through their survival function, the Nelson-Aalen estimator is its counterpart, analysing the survival distribution of the population through their CHF [25].

The Nelson-Aalen estimate of the Cumulative Hazard Function for the event of interest at a time point t, denoted by $H(t)$, is given by Equation (1) [25].

$$H(t) = \sum_{j=0}^{t} \left(\frac{m_j}{n_j} \right) \tag{1}$$

Where m_j is the number of failures at time *j* and n_j is the number of subjects in the risk set at time *j*, i.e., the set containing subjects who have survived at least to time *j*.

Table I shows an example of how the Nelson-Aalen method estimates several values of a Cumulative Hazard rate from an example hypothetical dataset of 1000 subjects, with censored data included. Table I has five columns, where

- *t* is the observed failure time;
- n_t is the number of subjects in the risk set at time *t*, i.e., the set containing subjects who have survived at least to time *t*; for each row i where i in [2. . . 6], corresponding to t in $\{2,3,7,9,10\}$, $n_i = n_{i-1} - m_{i-1} - c_{i-1}$;
- m_t is the number of subjects who "failed" at time *t*;
- c_t is the number of subjects who were censored in the time interval starting with time *t* up to but excluding the next failure time.
- $H(t)$ is the ratio estimated by the Cumulative Hazard Function at time *t*. Note that there are five unique failure times by which the table is ordered.

Hence, in the example of Table I, subjects who survived until at least time t = 10 have a CHF of 1.27.

D. Random Survival Forests

We assume that the reader is familiar with the well-known standard Random Forest algorithm for regression [2]. Hence, we focus here on describing mainly the characteristics of the Random Survival Forest (RSF) algorithm that makes it specifically adapted for survival analysis with censored data [8], rather than standard regression. RSF is a powerful technique for learning prediction models from survival data, which learns an ensemble of "survival trees" (as opposed to standard regression trees). It uses the Log–rank test as the node-splitting criterion; this is a non-parametric test designed for comparing the survival distributions between two (or more) groups (in our case, child nodes in a survival tree). It compares the hazard or survival functions at each observed event time. The Log-rank statistics is given by Equation (2):

$$\text{Log–rank statistics } = \frac{(O_i - E_i)^2}{\text{Var}\,(O_i - E_i)} \tag{2}$$

$$O_i - E_i = \sum_{j=1}^{k} (m_{ij} - e_{ij}) \tag{3}$$

$$e_{ij} = \left(\frac{n_{ij}}{n_{1j} + n_{2j}} \right) \times (m_{1j} + m_{2j}) \tag{4}$$

$$\text{Var}\,(O_i - E_i) = \sum_{j=1}^{k} \frac{n_{1j} n_{2j} \left(m_{1j} + m_{2j}\right) \left(n_{1j} + n_{2j} - m_{1j} - m_{2j}\right)}{\left(n_{1j} + n_{2j}\right)^2 \left(n_{1j} + n_{2j} - 1\right)} \tag{5}$$

In Equation (2), O_i is the sum of the number of observed failures in group i across all failure times and E_i is the expected value of the sum of the number of failures in group *i* across all failure times. To compute the Log–rank statistics, we need to calculate the term $O_i - E_i$, which is a measure of the overall differences of the survival or hazard function (curve) over all *k* failure times and is given by Equation (3), where e_{ij} is the expected number of failures for group *i* at the failure time *j*, as shown in Equation (4). *Var(*$O_i - E_i$*)* is the estimated variance, which involves the number of subjects in the risk set in each group (n_{ij}) and the number of failures in each group (m_{ij}) at time *j*. *k* is the number of distinct times of observed failures. The summation is over all distinct failure times. Note that when comparing any pair of survival functions, this calculation will be done for just one of the two groups since the absolute difference is the same for the two groups.

In addition, standard RSFs use a specific type of predicted outcome at their leaf nodes, based on the ensemble Cumulative Hazard Function [8], [9], which was designed to cope with censored data. Hence, this replaces the normal prediction of target values at leaf nodes in random forests for regression (which cannot cope wtih censored data).

The ensemble CHF for a given subject is calculated as follows. First, for each tree in the RSF, the subject's feature values are used to find the leaf node used to predict the survival time for that subject. In each tree, the CHF for that subject is calculated using Equation 1 (the Nelson-Aalen estimate for CHF), setting t to the last observed failure time (so that all failure times are considered in the summation), and calculating the terms m_j and n_j for the j-th failure time based on all the subjects assigned to the same leaf node as the current subject. Finally, the ensemble CHF for a subject is simply the

TABLE II: weights used in Equations (6) and (7) by different Log-rank statistics variants

Test Statistics	Weight
Log-rank	1
Wilcoxon	n
Tarone-Ware	sqrt(n)

arithmetic mean of the CHF for that subject over all trees in the RSF.

III. Proposed Variants of the Random Survival Forests

We propose new variants of the Random Survival Forest (RSF) algorithm, in order to try to improve this type of algorithm's predictive performance. Since RSF is a decision tree-based learning algorithm, we propose modifying the two key components of the algorithm: (1) the node-splitting criterion, and (2) the leaf-node-prediction criterion.

A. Modifying the Node-Splitting Criterion

We propose to replace the log-rank statistics, the default node-splitting criterion used in RSF, with its weighted versions, replacing the $O_i - E_i$ term in the numerator and the denominator of Equation (2) by Equations (6) and (7), respectively. Note that Equations (6) and (7) have weights w_j and w_j^2, respectively, multiplying the term within the scope of the summation symbol. Hence, the effect of using the weighted Equations (6) and (7) to implement Equation (2) will depend on how those weights are determined. In this work, the weight w_j is varied according to the Log-rank variants in Table II, including the Wilcoxon and Tarone-Ware criteria, where *n* is the number of subjects in the current risk set $(n_{1j} + n_{2j})$ and *sqrt(n)* is the square root of *n*.

$$O_i - E_i = \sum_{j=1}^{k} w_j \left(m_{ij} - e_{ij}\right) \tag{6}$$

$$\mathrm{Var}\left(O_i - E_i\right) =$$

$$\sum_{j=1}^{k} w_j^2 \frac{n_{1j} n_{2j} \left(m_{1j} + m_{2j}\right)\left(n_{1j} + n_{2j} - m_{1j} - m_{2j}\right)}{\left(n_{1j} + n_{2j}\right)^2 \left(n_{1j} + n_{2j} - 1\right)} \tag{7}$$

Note that in Equations (6) and (7) the summation is performed over the k distinct failure times. Hence, the original Log-rank node-splitting criterion assigns the same importance (weight 1) to all failure times, whilst the Wilcoxon and Tarone-Ware node-splitting criteria emphasize earlier failure times, since the value of n (the size of the risk set) tends to be greater in earlier failure times.

B. Modifying the Leaf-Node-Prediction Criterion

As discussed earlier, in the standard RSF algorithm, the prediction made by the leaf nodes of the trees for the current instance (subject) being classified is the value of the ensemble Cumulative Hazard Function (CHF) for that instance, which is the average of the CHF values over all trees in the forest. Note that a CHF value is essentially a sum of the "failure rates" across all observed failure times, but it was not designed to directly answer the fundamental question about how long a subject will "survive", i.e. how much time will pass until the event of interest occurs for a given subject.

Therefore, we propose a leaf-node-prediction criterion that is inspired by the standard Random Forest algorithm for regression (rather than for survival analysis), where the value predicted at a leaf node is an estimate of the mean of the target variable over the instances at that leaf node.

However, the standard Random Forest algorithm for regression cannot cope with censored data. Therefore, we propose a variation of the Random Survival Forest algorithm where, at each leaf node in a decision tree, the value predicted at that leaf node will be an estimate of the mean survival time of the instances at that leaf node by taking into account censored data.

We make the following two assumptions:

- the hazard rate is constant throughout the study — i.e., a person's chance of experiencing the event of interest does not change with time (a strong assumption); and
- the censoring is non-informative — i.e., the time when an instance is censored is independent of its "failure" time, or in short, instances are censored at random (a common assumption in the survival analysis literature).

At first glance, the constant hazard rate assumption would seem unlikely to be satisfied in our datasets of age-related diseases, since the time passed until the diagnosis of an age-related disease (our "survival time") tends to be smaller for older subjects. However, in our datasets this age effect is relatively small in general, and so that assumption can still be used to produce reasonable estimates of survival time in practice, as shown next.

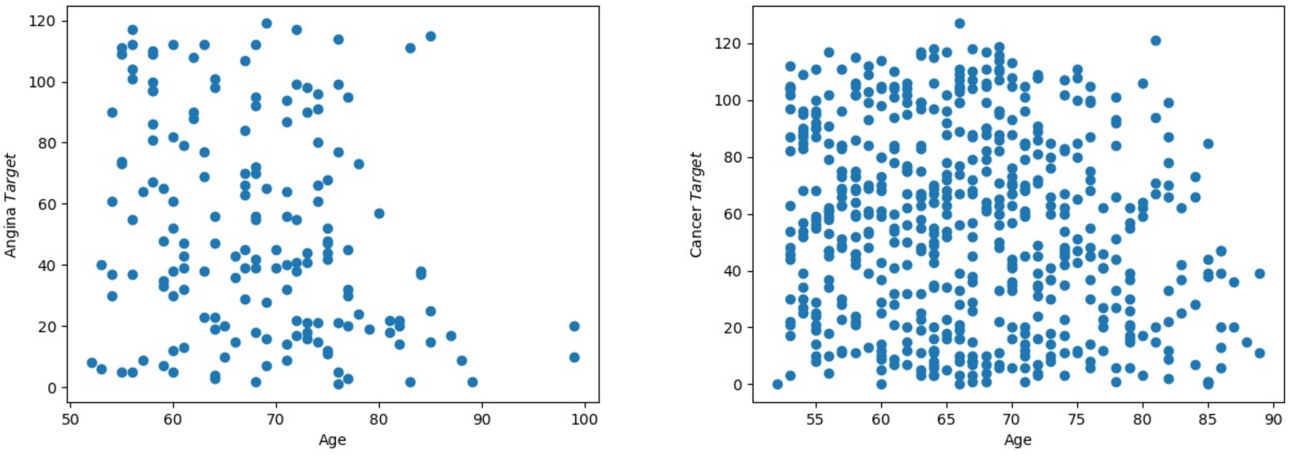
To be precise, we measured the Pearson's correlation coefficient between the age and survival time of uncensored subjects, for each disease (target variable), i.e. for each dataset. Age was measured at the ELSA/SHARE survey's baseline, and survival time is the number of months passed from that baseline time until the diagnosis of a disease. Table III reports these correlations.

As expected, the correlations are negative, since older individuals are more likely to be diagnosed with an age-related disease sooner, resulting in a shorter 'survival time'. However, these correlations are quite weak in general. In addition, Fig 2 shows the scatterplots for two diseases (datasets) as examples: Angina (with the largest negative correlation, -0.251) and Cancer (with the 5th largest negative correlation, -0.164). Note that there is no clear correlation between age and survival time for ages below about 85. The (negative) correlation is clear and strong only for ages above about 85, representing a small minority of subjects in our datasets. Therefore, the constant hazard rate assumption is approximately valid in our datasets in general.

Given the aforementioned assumptions, we can conclude that all instances have identical remaining mean survival time

TABLE III: The correlation coefficients between age and survival time for the uncensored instances, for each dataset (disease).

Dataset	Alzheim.	Angina	Heart attack	Psychiat.	Stroke	Diabetes	Cancer	Arthritis	Any-dise. ELSA	Any-dise. SHARE
Correl. Coeff.	-0.196	-0.251	-0.168	-0.044	-0.163	-0.113	-0.164	-0.081	-0.125	-0.175



(a) Angina dataset

(b) Cancer dataset

Fig. 2: Survival time of uncensored individuals over different ages

μ, regardless of their previously observed survival time t [26]. With these two assumptions, as shown in [26], the estimated mean survival time ($\hat{\mu}$) can be computed as shown in Equation (8):

$$\hat{\mu} = \frac{\sum_{j=1}^{n} t_j + m\hat{\mu}}{n} \tag{8}$$

where t_j is the value of the target variable (survival time) for the j-th individual, n is the total number of individuals (counting both uncensored and censored individuals), and m is the number of censored individuals. Recall that t_j is the true value of survival time if the j-th individual is uncensored, whilst it is a lower bound of the true, unknown survival time for censored individuals. Hence, in Equation (8), the term $\sum_{j=1}^{n} t_j$ is simply the sum of all observed survival times, considering both uncensored and censored individuals, whilst the term $m\hat{\mu}$ adds the total "missing", unobserved survival time associated with all m censored individuals — assuming that each censored individual has a remaining expected survival time (after censorship) of μ — as implied by the above two assumptions [26]. By applying some simple algebraic operations to Equation (8), we derive Equation (9):

$$n\hat{\mu} = \sum_{j=1}^{n} t_j + m\hat{\mu}$$

$$\hat{\mu}(n-m) = \sum_{j=1}^{n} t_j$$

$$\hat{\mu} = \frac{\sum_{j=1}^{n} t_j}{(n-m)} \tag{9}$$

Hence, the estimated mean survival time at each leaf node of the survival trees in a RSF model is computed using Equation (9), where the summation of all survival times, censored and uncensored included, is divided by the number of uncensored instances.

IV. Experimental Methodology

A. Dataset Creation

We created 10 datasets for 10 different age-related diseases (i.e. 10 separate survival prediction problems) from two different surveys, ELSA (English Longitudinal Study of Ageing) [19] and the Survey of Health Ageing and Retirement in Europe (SHARE) [20]. The ELSA survey collected data from English subjects living in private households, aged 50 and over; whilst the SHARE survey collected data from European individuals living in 28 European countries and Israel with the same age range. 9 out of the 10 datasets were constructed from the ELSA data, containing between 3,000 and 7,000 instances (depending on the target variable), with exactly the same 44 predictive features, but different target variables. On the other hand, the dataset constructed from the SHARE data is much larger, containing almost 140,000 instances but only 15 predictive features. In essence, the instances represent individuals (subjects) in these surveys, the target variables represent the 'survival times' (defined more precisely below), and the predictive features represent biomedical information collected by nurses or other relevant characteristics of an individual (age and gender).

As part of the data preparation for the survival prediction task, we create two special types of variables, the target variables and the uncensored status variables. For each dataset, the target variable, representing the 'survival time', contains the time passed (in months) until an individual is diagnosed with a certain disease (for 8 datasets) or any of several diseases (for two datasets); and the uncensored status variable indicates whether or not we know the "survival time" of an individual. Note that the target variables and uncensored status variables come into pairs, one pair for each dataset (for each target disease).

B. Predictive Performance Measure

The predictive performance of the survival models was estimated by the concordance index (C-index), which is a measure accounting for censored data, and is probably the most used measure of performance for survival prediction tasks. The C-index can be interpreted as the probability of correctly ordering the predicted survival values for a randomly chosen pair of subjects whose actual survival times are different.

As described in [27], the C-index can be adapted for censored data by considering the concordance of actual survival times versus predicted survival times among pairs of subjects whose survival outcomes can be ordered with respect to their survival times, i.e., among pairs where both subjects were observed to experience an event, or one subject was observed to experience an event before the other subject was censored. Note that in this latter case we know that the censored subject survived longer than the subject whose event was observed, so this pair of subjects can be ordered, even though we do not know the precise survival time for the censored subject.

Formally, the C-index is computed by equations (10) and (11), where $\hat{T}_i$ and T_i denote the predicted and actual target values ('survival times') of the i-th subject, respectively; and Usable(i,j) returns true if subjects i and j can be ordered with respect to their survival times (as described above) or false otherwise.

$$\text{C-index} = \frac{|\{(i,j)|\ \text{Usable}\ (i,j)\ \text{AND Agreed\_order}\ (i,j)\}|}{|\{(i,j)|\text{Usable}(i,j)\}|} \tag{10}$$

$$\text{Agreed\_order}(i,j) = \begin{cases} \textbf{true}, \text{ if } \widehat{T}_i > \widehat{T}_j \text{ and } T_i > T_j \\ \textbf{true}, \text{ if } \widehat{T}_j > \widehat{T}_i \text{ and } T_j > T_i \\ \textbf{false}, \text{ otherwise} \end{cases} \tag{11}$$

C. Hyper-Parameter Tuning via Nested Cross-Validation

All experiments are performed using a nested cross-validation procedure, where an inner cross-validation performs hyper-parameter tuning and an outer cross-validation estimates the predictive performance of the survival models. That is, for each iteration of the outer cross-validation, each candidate configuration of hyper-parameters for the Random Survival Forest (RSF) algorithm is evaluated via an inner cross-validation using only the training set (i.e. not the test set) of the current outer cross-validation iteration; and the configuration with the highest C-index is chosen as the best configuration for the current outer iteration of cross-validation. Then, the RSF is run with that best configuration using the entire training set to learn the survival model, and finally that model's predictive performance (C-index) is evaluated on the test set of the current outer cross-validation iteration. The result returned by the nested cross-validation is of course the average C-index over all test sets of the outer cross-validation.

We used 5 folds for the inner cross-validation for all datasets, whilst the number of folds for the outer cross-validation was set to 10 in the ELSA datasets and 5 in the SHARE dataset. The latter has fewer folds to save computational time, since the SHARE dataset is much larger than the ELSA datasets.

We tune two hyper-parameters of the RSF algorithm, namely: (a) *mtry*, i.e., the number of features randomly sampled to be used as candidate features for selection at each decision tree node; and (b) *d0*, i.e., the minimum number of uncensored instances required at each leaf node. According to [28], *mtry* has been recognized as the most influential hyper-parameter in general in random forests. In addition, *d0* can be seen as the survival task-related counterpart of the hyper-parameter *node size* in classical random forests. It is considered worth tuning according to, for instance, the experiments in [29] and [30].

We consider three candidate values for *d0*, namely 1, 2 and 3, for all datasets. We consider a different set of candidate *mtry* values for the ELSA datasets and SHARE dataset separately, since there is a difference between their numbers of features, where the former contains 44 and the latter contains 15 predictive features. For ELSA, we specify four candidate values for *mtry* (4, 7, 10, 13). The first two values were calculated from $ceil(ln(44)) = 4$ and $ceil(sqrt(44)) = 7$, which are often considered default functions for specifying the value of mtry in random forests, where *ceil(x)* returns the 'ceiling' of *x*, i.e. the lowest integer that is greater than or equal to *x* (i.e. it rounds *x* up to the nearest integer). Similarly, the set of candidate values for SHARE is (3, 4, 6, 8) since $ceil(ln(15)) = 3$ and $ceil(sqrt(15)) = 4$.

We also tune two hyper-parameters of the RTIF algorithm: (a) *mtry*, with the same aforementioned candidate values used for RSF; and (b) *node size*, analogous to *d0* in RSF, with one difference: *node size* is the minimum number of instances (regardless of their original censorship status) in a leaf node. So, its candidate values, (5, 7, 10), are larger than RSF's candidate *d0* values.

Hence, for all methods (the RSF versions and RTIF), for each dataset, at each iteration of the outer cross-validation, the inner cross-validation is run 12 times on the training set, considering 12 candidate random survival forest configurations (4 candidate *mtry* values times 3 candidate *d0* or *node size* values).

All analyses were performed using Python 3 with the scikit-survival library version 0.14.0 [31], a Python module built on top of the scikit-learn machine learning library [32]. In addition, some parts of the program were written and customised in Cython-code, which played an important role in boosting the performance of RSF due to Python's relatively slow performance. Our program code is made publicly available in the following GitHub link: https://github.com/mastervii/new_variants_of_RSF.

V. Computational Results

A. *Comparing results of RSF variants with different node-splitting criteria and standard leaf-node-prediction criterion*

Table IV reports the C-index values obtained by three different variants of Random Survival Forests (RSF), using three different node-splitting criteria, breaking down by each disease used as the target 'survival' variable (time passed until disease diagnosis). In all these three RSF variants, the leaf nodes compute the CHF as in standard RSFs. In this and the other result tables, the best result (highest C-index) for each dataset is shown in boldface.

The non-parametric Friedman test [33] was used to determine whether or not there is a significant difference between the average ranks of the three RSF variants and the mean rank of 2.0 under the null hypothesis. The calculated value of F_F is 0.278. With 3 variants and 10 datasets, F_F is distributed according to the F distribution with 3 – 1 = 2 and (3-1) x (10-1) = 18 degrees of freedom. The critical value of F(2,18) for α = 0.05 is 3.555. Note that F_F is smaller than the critical value, and so the null hypothesis cannot be rejected. Hence, there is no statistical evidence to support the claim that any of the three RSF variants has better performance than the others.

B. *Comparing results of RSF variants with a new leaf-node-prediction criterion and different node-splitting criteria*

Similarly to the previous comparison, Table V reports the C-index values obtained by three different variants of our proposed Mean-at-Leaf RSF (with a new leaf-node-prediction criterion), using three different node-splitting criteria.

Again, after applying the Friedman test and calculating the value of F_F which is 0.512 in this case, the null hypothesis cannot be rejected. Hence, there is no statistical evidence to support the claim that any of the three variants of the Mean-at-Leaf RSF has better performance than the others.

C. *Comparing results of the best RSF for each type of leaf-node-prediction criterion*

The previous two subsections showed that the choice of node-splitting criterion does not significantly affect the C-index values, both when using the standard leaf-node-prediction criterion and when using the new leaf-node-prediction criterion (Mean-at-Leaf). This subsection investigates the complementary issue of whether the choice of leaf-node-prediction criterion affects the C-index, by comparing the best RSF variant from Table IV (with a standard leaf-node-prediction criterion) against the best RSF variant from Table V (with the new leaf-node-prediction criterion).

As can be observed in Table IV, both the Log-rank and the Wilcoxon RSF variants were tied in terms of achieving the highest C-index (boldfaced values) in 4 out 10 datasets, whilst the Tarone-Ware RSF achieved the highest C-index in only two datasets. In addition, the last row of the table shows the average rank for each variant — recall that the smaller the average rank, the better (higher) the C-index value of a RSF variant across all datasets in general. Both the original (Log-rank) RSF and the Wilcoxon RSF were tied with the best average rank (1.9), whilst the Tarone-Ware RSF had a slightly worst average rank (2.2). Hence, the Log-Rank and Wilcoxon criteria were tied as winners, and we chose the Log-Rank variant as the representative "best" variant of this table because it is the default variant, much more used in practice than the Wilcoxon variant.

In Table V, the winner was clearly the Log-rank variant of Mean-at-Leaf RSF, which has both the highest number (6) of wins and the best (smallest) average rank, 1.8.

Table VI shows the C-index values obtained by RTIF [23], the original RSF with standard leaf-node prediction and Log-Rank [8], and the proposed Mean-at-Leaf RSF variant with Log-Rank, for each disease used as the target 'survival time' variable. In this table, the uncensoring ratio is the ratio of the number of uncensored instances over the total number of (censored or uncensored) instances. Note that most datasets have small uncensoring ratios, representing challenging problems. In terms of C-index values, the proposed Mean-at-Leaf RSF outperformed the two other methods in 9 out of 10 datasets. Moreover, the average rank of Mean-at-Leaf RSF (1.2) is much lower (better) than that of RTIF (2.7) and the original RSF (2.1).

The non-parametric Friedman test [33] was used to determine whether or not there is a significant difference between the average ranks of the three methods and the mean rank of 2.0 under the null hypothesis. The calculated value of F_F is 11.93. With 3 methods and 10 datasets, F_F is distributed according to the F distribution with 3 – 1 = 2 and (3-1) x (10-1) = 18 degrees of freedom. The critical value of F(2,18) for α = 0.05 is 3.555. F_F is greater than the critical value, and so the null hypothesis is rejected (p-value = 0.003) at the conventional significance level of α = 0.05. Hence, there is a statistically significant difference between the performances of the three methods as a whole. Therefore, we proceed with the Holm post-hoc test [33], which compares the average rank of the best (control) method, viz. Mean-at-Leaf RSF, against each of the other two methods, by adjusting the significance level of α = 0.05 to compensate for multiple comparisons. The results were that Mean-at-Leaf RSF significantly outperformed both RTIF (p-value = 0.0004, smaller than adjusted α = 0.025) and the original RSF (p-value = 0.0221, smaller than α = 0.05).

TABLE IV: C-index values of three RSF variants with different node-splitting criteria. All RSF variants had their *mtry* and *d0* hyper-parameters tuned via nested cross-validation

Dataset (Disease)	RSF Log-rank	RSF Wilcoxon	RSF Tarone-Ware
Alzheimer	0.7725	**0.776**	0.7736
Angina	**0.6018**	0.6002	0.6013
HeartAtt	0.6351	**0.6370**	0.6343
Psychiatric	0.5372	0.541	**0.5462**
Stroke	**0.6373**	0.635	0.6335
Diabetes	0.7527	**0.7542**	0.7516
Cancer	**0.5473**	0.5436	0.5382
Arthritis	0.5340	**0.5383**	0.5380
Any-disease (ELSA)	0.5426	0.5425	**0.5455**
Any-disease (SHARE)	**0.6890**	0.6882	0.6883
Average Rank	**1.90**	**1.90**	2.20

TABLE V: C-index values of three RSF variants of Mean-at-leaf with different node-splitting criteria. All RSF variants had their *mtry* and *d0* hyper-parameters tuned via nested cross-validation

Dataset (Disease)	RSF Mean-at-leaf Log-rank	RSF Mean-at-leaf Wilcoxon	RSF Mean-at-leaf Tarone-Ware
Alzheimer	0.7564	**0.7738**	0.7576
Angina	**0.6085**	0.6054	0.6073
HeartAtt	0.651	0.6537	**0.6538**
Psychiatric	**0.5596**	0.5539	0.557
Stroke	**0.6425**	0.6424	0.6417
Diabetes	0.7594	**0.7641**	0.7611
Cancer	**0.5553**	0.5476	0.5532
Arthritis	**0.5462**	0.5423	0.5458
Any-disease (ELSA)	**0.5616**	0.5578	0.5583
Any-disease (SHARE)	0.7104	**0.7126**	**0.7126**
Average Rank	**1.80**	2.25	1.95

TABLE VI: C-index values of RTIF, standard RSF and the proposed mean-at-leaf RSF (Log-rank), all with hyper-parameters tuned via nested cross-validation

Dataset		RTIF [23]	RSF [8]	RSF Mean-at-Leaf (Log-rank)
Disease	uncensoring ratio			
Alzheimer	69/6825 (1.0%)	**0.7742**	0.7725	0.7564
Angina	165/6488 (2.5%)	0.5723	0.6018	**0.6085**
HeartAttack	186/6607 (2.8%)	0.6228	0.6351	**0.651**
Psychiatric	219/5972 (3.5%)	0.4692	0.5372	**0.5596**
Stroke	270/6632 (4.1%)	0.6366	0.6373	**0.6425**
Diabetes	416/6500 (6.4%)	0.7443	0.7527	**0.7594**
Cancer	562/6386 (8.8%)	0.5135	0.5473	**0.5553**
Arthritis	784/4276 (18.3%)	0.5078	0.5340	**0.5462**
Any-disease (ELSA)	979/3280 (29.8%)	0.5384	0.5426	**0.5616**
Any-disease (SHARE)	101300/139522 (72.6%)	0.7061	0.6890	**0.7104**
Average Rank		2.70	2.10	**1.20**

D. Identifying the top-ranked features for survival prediction

Although the model learned by a RSF algorithm is not directly interpretable, since there are too many survival trees for human interpretation, we can use a feature importance measure to identify the most important features in a RSF model learned from the data, i.e., the features that most influence the predictions of the RSF model. This feature-importance analysis can highlight general trends in the learned models and can be useful to better understand the relationships between some features and the target variable – i.e., the time passed until the diagnosis of an age-related disease.

Hence, we report the most important features in the survival models learned by the Mean-at-Leaf RSF algorithm, which was the algorithm with the best overall predictive performance in the experiments. To identify the most important features for each dataset, we first computed the importance of each feature in the learned RSF model, using the well-known "permutation feature importance" measure [34]. This measure essentially quantifies the decrease in the C-index of the learned RSF model when a single feature has its values randomly shuffled. The permutation importance measure was computed by ELI5, a Python package built on top of scikit-learn.

TABLE VII: The 8 features which appear most often in the sets of top-10 features in the RSF models learned from the ELSA datasets

Feature	count	Alzheimer	Angina	HeartAtt	Psychiatric	Stroke	Diabetes	Cancer	Arthritis	Any-disease (ELSA)
mmgsn_me	7	V	V	V	V	V		V	V	
confage	6	V	V	V	V	V		V		
mmrroc	5	V	V	V		V		V		
wtval	5	V					V	V	V	V
hdl	5			V	V	V	V		V	
scako	4		V		V	V	V			
smokerstat	4		V	V	V		V			
indsex	4			V	V				V	V

TABLE VIII: The 4 most important features in the RSF model learned from the SHARE dataset

Rank	Feature	Description
1	age	Age at interview (in years)
2	mobilityind	Mobility index (high: has difficulties)
3	bmi	Body mass index
4	lgmuscle	Large muscle index (high: has difficulties)

We report two sets of most important features, one for the ELSA datasets and another for the SHARE dataset, as follows. First, recall that the 9 ELSA datasets share the same set of 44 predictive features – those datasets differ in their target variables (age-related diseases). Hence, we identify the top-8 ranked features across the RSF models learned for those 9 datasets as a whole, which allows us identify the most predictive features for multiple age-related diseases at the same time, which is useful to study the ageing process as a whole, from a more systemic perspective. Second, in the case of the SHARE dataset, since this dataset has only 15 features, we simply report the top-4 ranked features in the RSF model learned from this dataset.

Table VII shows the 8 top-ranked features in the models learned from the 9 ELSA datasets. To identify these features, we first ranked the features in decreasing order of the permutation feature importance in each learned RSF model (i.e. for each dataset). Then, we computed the frequency of occurrence of each feature in the sets of top-10 features for those 9 datasets, and ranked the features in decreasing order of that frequency. That frequency is shown in the column "count" in Table VII, and the following columns show precisely for which datasets (i.e., diseases) the feature was among the top-10 features in the learned RSF model.

The top-8 features in Table VII can be described as follows [19]: mmgsn_me is the grip strength (Kg) of the non-dominant hand, confage is the subject's age when the data were collected, mmrroc is the outcome of chair-rise tests, wtval is the subject's valid weight (Kg), hdl is the blood HDL (High-Density Lipoprotein) level, scako measures how often the subject had an alcoholic drink during the last 12 months, smokerstat is the present or past smoker status, and indsex is the gender.

In addition, Table VIII reports the 4 top-ranked features in the RSF model learned from the SHARE dataset.

As expected, Age is one of the top-ranked features, with ranks 2 and 1 in the ELSA and SHARE datasets, respectively. Interestingly, several of the top-ranked features for both types of dataset are not standard biomarkers of specific diseases, but rather reflect the level of frailty of individuals, like mmgsn_me and mmrroc for ELSA datasets and mobilityind for the SHARE dataset. Out of the several blood test results used as features in the ELSA datasets, only the HDL ("good cholesterol") level is among the top-8 features in Table VII.

VI. Conclusions

We have proposed two types of variations in Random Survival Forests (RSFs), namely: (a) modifying the node-splitting criterion, more precisely replacing the standard Log-rank test by the Wilcoxon and Tarone-Ware tests; and (b) modifying the leaf-node-prediction criterion, more precisely replacing the ensemble Cumulative Hazard Function (CHF) by a more direct and simpler estimate of the survival time for each subject, directly based on the mean of the target variable, but taking into account the presence of censored data.

We have evaluated the proposed RSF variants on 10 survival prediction problems, involving the prediction of the time passed until an individual is diagnosed with a given age-related disease or any of a set of age-related diseases.

We first compared the effectiveness of the three aforementioned node-splitting criteria. The experimental results have shown that the standard Log-rank and the Wilcoxon criteria achieved the joint best predictive performance when the RSF used the standard leaf-node-prediction criterion; whilst the standard Log-Rank criterion achieved the best performance when the RSF used the new proposed leaf-node-prediction criterion. In both cases, however, there was no statistically significant difference in the predictive performance of the 3 RFS variants using different node-splitting criteria.

Next, we compared the predictive performance of two RSF variants using the same standard (default) Log-Rank criterion but two different leaf-node-prediction criteria: the standard CHF-based one and the new proposed leaf-node-prediction criterion. The experimental results have shown that the proposed leaf-node-prediction criterion has led to a statistically significant improvement of predictive performance, across the 10 survival prediction problems.

Future work could involve evaluating other node-splitting criteria or developing other leaf-node prediction criteria for RSFs, to try to further improve their effectiveness.

References

[1] David G Kleinbaum and Mitchel Klein. *Survival Analysis: A Self-Learning Text, Third Edition*. 700 pages, Springer, 2012.

[2] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[3] Dursun Delen, Glenn Walker, and Amit Kadam. Predicting breast cancer survivability: A comparison of three data mining methods. *Artificial Intelligence in Medicine*, 34(2):113–127, 2005.

[4] Evangelia I Zacharaki, N Morita, et al. Survival analysis of patients with high-grade gliomas based on data mining of imaging variables. *American Journal of Neuroradiology*, 33(6):1065–1071, 2012.

[5] Torsten Hothorn, Peter Buhlmann, Sandrine Dudoit, Annette Molinaro, and Mark J. Van Der Laan. Survival ensembles. *Biostatistics*, 7(3):355–373, 2006.

[6] David M Vock, Julian Wolfson, et al. Adapting machine learning techniques to censored time-to-event health record data. *Journal of Biomedical Informatics*, 61:119–31, 2016.

[7] Per Kragh Andersen, Mette Gerster Hansen, and John P Klein. Regression analysis of restricted mean survival time based on pseudo-observations. *Lifetime Data Analysis*, 10(4):335–350, 2004.

[8] Hemant Ishwaran, Udaya B. Kogalur, Eugene H. Blackstone, and Michael S. Lauer. Random survival forests. *Annals of Applied Statistics*, 2(3):841–860, 2008.

[9] Hong Wang and Gang Li. A selective review on random survival forests for high dimensional data. *Quantitative Bio-science*, 36(2):85, 2017.

[10] Dilusha Weeraddana, Sudaraka MallawaArachchi, Tharindu Warnakula, Zhidong Li, and Yang Wang. Long-term pipeline failure prediction using nonparametric survival analysis. *arXiv preprint arXiv:2011.08671*, 2020.

[11] Mark Robert Segal. Regression Trees for Censored Data. *Biometrics*, 44(1):35, mar 1988.

[12] Naz Gul, Nosheen Faiz, Dan Brawn, Rafal Kulakowski, Zardad Khan, and Berthold Lausen. Optimal survival trees ensemble. *arXiv preprint arXiv:2005.09043*, 2020.

[13] Lev V Utkin, Andrei V Konstantinov, Viacheslav S Chukanov, Mikhail V Kots, Mikhail A Ryabinin, and Anna A Meldo. A weighted random survival forest. *Knowledge-Based Systems*, 177:136–144, 2019.

[14] Nikolaj Tollenaar and Peter G.M. Van Der Heijden. Optimizing predictive performance of criminal recidivism models using registration data with binary and survival outcomes. *PLoS ONE*, 14(3):e0213245, Mar 2019.

[15] Wei Wang and Wei Liu. Integration of gene interaction information into a reweighted random survival forest approach for accurate survival prediction and survival biomarker discovery. *Scientific Reports*, 8(1), 2018.

[16] Herbert Pang, Stephen L. George, Ken Hui, and Tiejun Tong. Gene selection using iterative feature elimination random forests for survival outcomes. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 9(5):1422–1431, 2012.

[17] Byron C Jaeger, D Leann Long, Dustin M Long, Mario Sims, Jeff M Szychowski, Yuan-I Min, Leslie A Mcclure, George Howard, Noah Simon, et al. Oblique random survival forests. *Annals of Applied Statistics*, 13(3):1847–1883, 2019.

[18] Fabian Eifler. *Introduction of AUC-based splitting criteria to random survival forests*. PhD thesis, 2014.

[19] S. Clemens et al. English Longitudinal Study of Ageing: Waves 0-8 https://www.elsa-project.ac.uk/, 2019.

[20] Axel Börsch-Supan, Martina Brandt, Christian Hunkler, Thorsten Kneip, Julie Korbmacher, Frederic Malter, Barbara Schaan, Stephanie Stuck, and Sabrina Zuber. Data resource profile: The survey of health, ageing and retirement in europe (share). *International Journal of Epidemiology*, 42(4):992–1001, Aug 2013.

[21] Stefan Gruber, Christian Hunkler, and Stephanie Stuck. Generating easyshare: guidelines, structure, content and programming. Technical report, SHARE Working Paper Series 17-2014. Munich, 2014.

[22] Annette M. Molinaro, Sandrine Dudoit, and Mark J. Van Der Laan. Tree-based multivariate regression and density estimation with right-censored data. *Journal of Multivariate Analysis*, 90(1 SPEC. ISS.):154–177, 2004.

[23] Tossapol Pomsuwan and Alex A Freitas. Adapting random forests to cope with heavily censored datasets in survival analysis. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN 2020)*, pages 697–702, 2020.

[24] James M Robins and Andrea Rotnitzky. Recovery of information and adjustment for dependent censoring using surrogate markers. In *AIDS Epidemiology*, pages 297–331. Springer, 1992.

[25] Wayne Nelson. Theory and Applications of Hazard Plotting for Censored Failure Data. *Technometrics*, 14(4):945–966, 1972.

[26] Steve Selvin. *Survival analysis for epidemiologic and medical research*. Cambridge University Press, 2008.

[27] Frank E Harrell, Kerry L Lee, and Daniel B Mark. Tutorial in Biostatistics Multi-variable Prognostic Models: Issues in Developing Models, Evaluating Assumptions and Adequacy,and Measuring and Reducing Errors. *Statistics in Medicine*, 15(4):361–387, 1996.

[28] Philipp Probst, Marvin N. Wright, and Anne Laure Boulesteix. Hyperparameters and tuning strategies for random forest. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 9(3):1–15, 2019.

[29] Yi Lin and Yongho Jeon. Random forests and adaptive nearest neighbors. *Journal of the American Statistical Association*, 101(474):578–590, 2006.

[30] Chip M. Lynch, Behnaz Abdollahi, Joshua D. Fuqua, Alexandra R. de Carlo, James A. Bartholomai, Rayeanne N. Balgemann, Victor H. van Berkel, and Hermann B. Frieboes. Prediction of lung cancer patient survival via supervised machine learning classification techniques. *International Journal of Medical Informatics*, 108(April 2016):1–8, 2017.

[31] Sebastian Pölsterl. scikit-survival: A library for time-to-event analysis built on top of scikit-learn. *Journal of Machine Learning Research*, 21(212):1–6, 2020.

[32] F. Pedregosa, G. Varoquaux, et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[33] Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, 7:1–30, 2006.

[34] André Altmann, Laura Toloşi, Oliver Sander, and Thomas Lengauer. Permutation importance: a corrected feature importance measure. *Bioinformatics*, 26(10):1340–1347, 2010.