A Simulation-based comparison of the predictive accuracy of the random survival forest and the lasso-regularized Cox Model in Survival Analysis

---

Willem Van Der Merwe

Supervisor(s):
Dr. Alphonce Bere

A research report submitted in partial fulfillment of the requirements for the degree of Master of Science, Artificial Intelligence

in the

Faculty of Science
University of the Witwatersrand, Johannesburg

10 September 2024

Declaration

I, Willem Van Der Merwe, declare that this report is my own, unaided work. It is being submitted for the degree of Master of Science, Artificial Intelligence at the University of the Witwatersrand, Johannesburg. It has not been submitted for any degree or examination at any other university.

Willem Van Der Merwe

10 September 2024

Abstract

This report provides a simulation study between the Random Survival Forest and the Cox Proportional Hazards Model using proven frameworks in the field. Despite the widespread application of survival models within traditional and machine learning contexts for predicting event times, discrepancies in method implementation and evaluation can lead to biased outcomes and limit the generalizability of findings. By adhering to the ADMEP framework and employing advanced data-generating mechanisms, this research seeks to provide empirical insights into the models' behavior and effectiveness under varied conditions. I aim to enhance the reliability of survival predictions and contribute to methodological standards in survival analysis, addressing the prevalent issues of inadequate design, reporting, and the need for methodological rigor in simulation studies.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Survival analysis is used to examine the time until the occurrence of an event, like disease relapse. A major challenge in this area is handling censored data, where the event information is incomplete. Censoring can be of different types; right-censored data is when the event has not occurred by the end of the observation period, left-censored data is when the event occurred before the study began, and interval-censored data is when the event occurred between two observed times [1]. To analyze such data, statistical methods have been developed.

Non-parametric methods like the Kaplan-Meier estimator and the Logrank test do not assume any specific distribution for the time-to-event data, making them robust against mis-specifications of the event-time distribution [1]. Parametric methods like the Exponential and Weibull models assume a known distribution that models the time-to-event data [1]. They are typically more precise, at the risk of introducing bias when the assumed distribution is wrong.

In survival analysis, understanding the survival function $S(t)$ is crucial. The Survival Function is the statistical representation that outputs survival probabilities at various time points for given covariates (x). It quantifies the probability of an event not occurring by a certain time t, effectively illustrating how the likelihood of survival decreases over time. This function is central in studies that aim to compare the effectiveness of treatments under different conditions by analyzing how quickly the events occur, such as comparing the onset times of motion sickness under varying experimental conditions.

The Proportional Hazards Model, which can be used in both semi-parametric (Cox model) and parametric forms, is employed to estimate the hazard ratio, which is a measure of effect size regarding the time to event. An example is, that studies may compare the time until the onset of motion sickness under different conditions to assess treatment effectiveness. It is important to understand the context of discrete and continuous time models; discrete models address survival data that is categorical or not continuously distributed while the continuous model proposed by Cox, utilizes continuous data to model hazard functions [14]. For this continuous model, estimation is based on maximizing the conditional likelihood across observed failure times, while the discrete model uses a logistic framework for estimation, treating survival as a sequence of binary outcomes.

Traditional statistical methods require explicit programming and often suffer from user bias in variable selection, whereas Machine Learning (ML) operates under a paradigm where algorithms autonomously identify patterns in large data sets, which potentially increases accuracy and efficiency [24]. Shown below is a review [30] of methods broadly utilized in survival analysis studies, Figure 1.1.

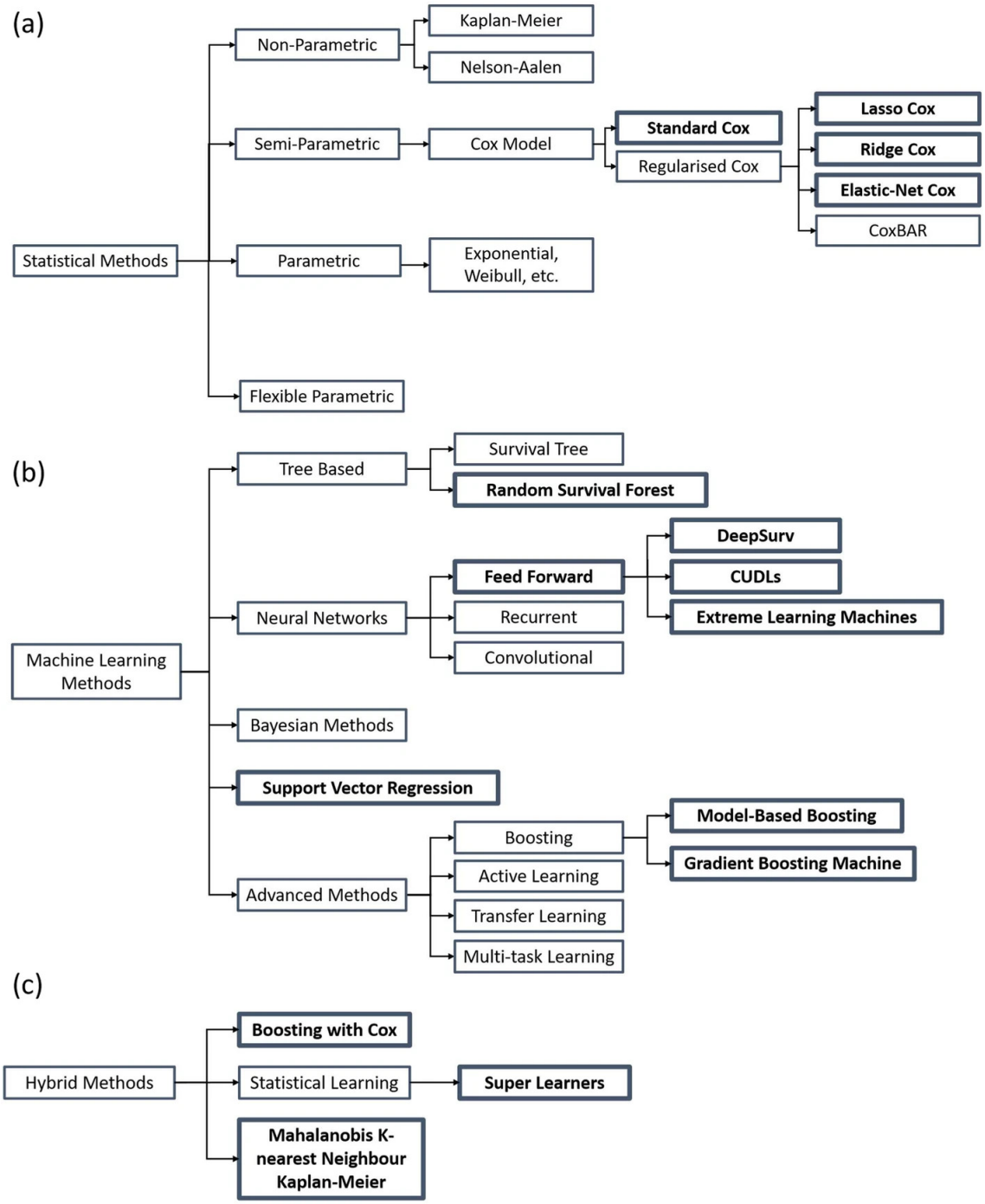


FIGURE 1.1: Shows the breakdown of methods analysis preformed by [30] during a method review. The Study ran a literature selection process based on qualitative and quantitative metrics of methodology used in studies.

Simulation studies are a crucial statistical tool used for evaluating and comparing different statistical methods, particularly when analytic solutions are hard or impossible to achieve [18]. These studies generate data through pseudo-random sampling from known probability distributions, enabling researchers to empirically test the behavior of statistical methods under varied scenarios. Common uses include validating new statistical methods, ensuring accuracy in mathematical models and code, and comparing the effectiveness of various approaches.

Particularly in medical statistics, simulation studies help in designing experiments, determining sample sizes, and estimating power under specific assumptions about data generation [18]. Despite their widespread use, many statisticians face challenges in properly conducting simulation studies due to a lack of understanding and experience [18].

With this research I present a comprehensive comparative study pipeline of the survival models applied to simulated survival data, highlighting their operations, challenges, and implications.

1.2 Problem Statement

Many studies have compared machine learning with traditional statistics, yet comprehensive simulation-based comparisons are scarce. This gap may lead to biases and sometimes questionable practices, affecting the validity of findings.

1.3 Research Aims and Objectives

1.3.1 Research Aims

Perform a comparative analysis of survival models using both simulated and real datasets to identify model robustness and effectiveness, adhering to formal frameworks [18] and avoiding common pitfalls outlined in the literature [21].

1.3.2 Objectives

1. Source a Practical Dataset: Acquire a dataset with clear constraints and features with relevant survival data. This dataset should comply with standards [35].

2. Run the comparative study for the survival models while adhering to the ADMEP design [18]

   (a) Apply Data Generating Methods: Utilise standard libraries to generate simulated data that closely replicate the statistical properties of the real dataset.

   (b) Random Survival Forest Model: Develop and apply this model using both the real and simulated datasets.

   (c) Lasso Regularized Cox Proportional Hazards Model: Similarly, develop and apply this model with both datasets.

   (d) Evaluate and Visualise Predictions: Use common survival analysis metrics for evaluation and employ visualization tools from survival libraries to illustrate the results effectively.

1.4 Limitations

1. **Innovation vs. Application:** This research does not venture to innovate on the algorithmic core of these methods. The primary focus is on the application and evaluation of established survival analysis methods and their existing extensions as documented in the literature. I seek to implement and test these pre-existing models in a new dataset context, thereby contributing to empirical evidence and practical applications rather than theoretical advancements.

2. **Redundancy in Literature:** Furthermore, comprehensive comparative studies like those conducted by, [16] [30] have already evaluated these methods extensively. These studies provide a solid foundation of knowledge regarding the performance and limitations of traditional and modified survival analysis models across various types of data.

1.5 Overview

In addressing the noted shortcomings in comparative simulation studies, this literature review methodically examines simulation work in segments relevant to each section of the study. I begin with an overview of the Cox method and its various extensions, illustrating how these foundational techniques are implemented. Following that, I explore proofs and extensions of the Lasso method, which builds on the base Cox method, enhancing its predictive power and flexibility. The discussion then moves to Random Survival Forests (RSF), detailing recent advancements in RSF algorithms that provide a solid reference for current implementations. Two comparative studies are highlighted; these utilize simulations to evaluate the methods mentioned above, offering insights into their practical applications and effectiveness. Finally, the last sections categorize the literature into subgroups that align with the specific components of the proposed research framework [21] [18], facilitating easy reference and integration into the research design and methodology in Chapter 2, ensuring a coherent and structured approach to applying these methods in this proposed study.

Chapter 2

Literature review

2.1 Important issues in comparative simulation studies

[24] Show that the literature on ML in orthopedics is predominantly composed of preliminary studies, and frequently lacks depth in addressing complex ML concepts and often falls short in providing comprehensive method specification for result interpretation. [24] Continues to explain that deep Learning is a prominent subset of ML, and utilizes neural networks to process both structured and unstructured data, enhancing the capability to handle diverse data types like images and texts. Similarly [30] show out of their methodical study selection process that only a handful of studies have attempted such comparisons at an acceptable standard, while most studies focus predominantly on machine learning techniques neglecting the broader spectrum of statistical methods.

Furthermore [30] point out authors often omit interaction terms and non-linear covariate effects which are essential components for enhancing model robustness and accuracy. The predominance of studies failed to relax the proportional hazards assumption which underscores a critical oversight in adapting models to more complex datasets. Finally and more broadly [30] show that there is a need for comprehensive methodological improvements and enhanced reporting standards to ensure reproducibility and a fair assessment of method capabilities.

Common issues include inadequate design and reporting that lead to uncritical acceptance of results. This lack of rigor can result in misleading conclusions, for

example, the variability introduced by different sets of random numbers in Monte Carlo simulations that are sometimes ignored. [30] Found a notable scarcity of quality comparative research between statistical and machine learning methods. Predominantly, these studies focus on machine learning techniques while traditional statistical methods are often neglected. For instance, it was common for some authors to overlook the inclusion of interaction terms and non-linear covariate effects in the Cox model as well as time-dependent effects which are key elements for effectively handling complex datasets. The reporting standards of the reviewed studies were also generally poor. Important details such as data-generating mechanisms (DGMs), estimands, and method implementations are frequently underreported, which impedes the reproducibility of the research and the ability to conduct fair comparisons between methods. [30] also pointed out that a significant bias could be observed in the selection of DGMs, which tend to favor machine-learning approaches, especially in scenarios where the number of variables exceeds the sample size. This predisposition can lead to biased results unless the study incorporates specific statistical variable selection techniques that are suited for high-dimensional data. Additionally, the prevalent use of the C-index as the sole performance metric, without accounting for calibration is noted. By relying solely on this metric results analysis may not provide a complete picture of the model's predictive accuracy over time, particularly when the proportional hazards assumption is not valid.

Finally, [30] exclaim that there is a concerning lack of expertise in implementing complex statistical methods thoroughly. This deficiency often results in potentially misleading outcomes that do not genuinely reflect the true performance capabilities of the methods being compared. The findings underscore the need for improved methodological rigor and enhanced collaboration among researchers to ensure that both statistical and machine learning methods are implemented to their full potential and evaluated fairly. As a framework [21] formalizes the use of indicators, defined as "questionable research practices" (QRP) which indicate faulty research methods used widely throughout simulation studies, which should get the necessary attention. The QRP's are categorized during phases of comparative simulation, namely the design phase, execution phase, and reporting phase this is shown in Figure 2.1. [21] Labels specific components of simulation studies, an example being D1 which references "the data-generating process", and cross correlates with

other components to define QRP occurrence and relationships.

Tag	Related	Type of QRP
Design		
D1	E1, R1	Not/vaguely defining objectives of simulation study
D2	E2, R1	Not/vaguely defining data-generating process
D3	E3, E4, R1	Not/vaguely defining which methods will be compared and how their parameters are specified
D4	E1, E5, R1	Not/vaguely defining estimands of interest
D5	E1, E5, R1	Not/vaguely defining evaluation criteria
D6	E6, R1	Not/vaguely defining how to handle missing values (for example, due to non-convergence of methods)
D7	E7, E8, R3	Not justifying number of simulations
Execution		
E1	D1, R2	Changing objective of the study to achieve desired outcomes
E2	D2, R2	Adapting data-generating process to achieve desired outcomes
E3	D3, R2	Adding/removing comparison methods to achieve desired outcomes
E4	D3, R2	Selective tuning of method hyperparameters to achieve desired outcomes
E5	D4, D5, R2	Choosing evaluation criteria to achieve desired outcomes
E6	D6, R2	Adapting inclusion/exclusion/imputation rules to achieve desired outcomes
E7	D7, R3	Choosing number of simulations to achieve desired outcomes
E8	D7, R3	Choosing random number generator seed to achieve desired outcomes
Reporting		
R1	D1–D6	Justifying design decisions which lead to desired outcomes *post hoc*
R2	E1–E6	Selective reporting of results from simulations that lead to desired outcomes
R3	D7, E7, E8	Failing to report Monte Carlo uncertainty
R4		Failing to assure computational reproducibility (for example, not sharing code and sufficient details about computing environment)
R5		Failing to assure replicability (for example, not sufficiently reporting design and execution methodology)

FIGURE 2.1: QRP Summary [21]

2.2 Models

2.2.1 Cox's Proportional Hazards Model

In the seminal work by [2], the Cox model is introduced as an extension to prior work formalized as the Kaplan-Meier estimator, by exploring time-to-event data (life tables). The major benefit is that it addresses censored data, which is a known concept in survival analysis, that there is missing information within the data, specifically, event occurrence without observation on a continuous time scale. Hazard is the estimated conditional probabilities, in line with the observed conditional frequencies of events or simply the risk of event occurrence at a specific time.

Time-Dependant Covariates

The Cox model incorporates both time-independent and time-dependent covariates. [14] Time-dependent covariates can change over the time, such as $Z_2(t) = Z_1 t^*$. This flexibility allows the model to handle scenarios where hazards are not proportional, which extends its applicability. Relative risk is represented as $\exp Z(t)'$, showing how risk changes with time and covariate values, and depending on the

coefficients, the relative risk in a treatment group can increase, decrease, or remain constant over time. [14] External covariates are variables that are independent of the subject's survival process. Whereas internal covariates are variables that might influence and be influenced by survival. [14] Different approaches for modeling survivor functions are required for external and internal covariates due to their nature.

[36] Proposes a step-by-step development and testing of time dependence in the Cox model, with emphasis on methods and critical formulas to highlight key concepts in understanding and applying time-dependent effects in survival analysis. [36] Point out how the impact of variables can change over time, which is critical for understanding complex dynamics in data that cannot be captured by static models. Step 1 includes the selection and justification of the appropriate survival time variable for use in the Cox model. Accurate identification of the "at risk" period is crucial for defining when subjects are susceptible to the event of interest. The measurement scale (e.g., years, and months) should match the scale of independent variables. It is noted about 50% of the reviewed studies properly justified their choice of survival time variable, highlighting the need for a clear explanation. Step 2 is for developing time-dependent moderation hypotheses based on a priori theory-building. It is a good indication of the importance of time dependence as it directly shows effects between [36] the three types of time-dependent effects are type 1; both main and moderation effects are significant and in the same direction, thereby strengthening each other. Type 2; main and moderation effects are significant but in opposite directions, thereby weakening the main effect. Type 3; only the moderation effect is significant with no main effect, showing causality only in extended survival times. Step 3, tests for the proportional hazards assumption, using graphical methods like log-log survival curves. [36] This however is subjective and lacks statistical tests, problematic with continuous variables. Another approach is to approximate goodness-of-fit by using the Schoenfeld residuals test, which can detect non-zero slopes in survival time against scaled Schoenfeld residuals to check PH assumption violations. Step 4: Shows the extended Cox Model by integrating time-dependence detected from PH tests and adding interaction terms to the model. Interaction terms are the product of time and independent variables to handle non-proportional hazards. The extended Cox model

equation: $h(t, X) = h_0(t).\exp b_1 x + b_2 xt + b_3 z$. Lastly, Step 5 is to interpret the effects of the time dependence integration by computing hazard ratios over time to understand changes in effects due to survival time, using the extended model. [36] This allows the use of model results to further develop or adjust theoretical assumptions based on observed data. Hazard Ratio calculation for time-dependence, $\hat{HR} = \exp b_1 + b_2 t$

Likelihood Function

The various approaches to the Cox model handle data ties and time-dependent covariates differently, [14] recommended specific methods based on the data structure (e.g., number of ties). The concept of partial likelihood is particularly important as it provides a way to focus on relevant factors in the presence of complex data types, enhancing both the theoretical understanding and practical application of the Cox model. The marginal likelihood approach [14] (Kalbfleisch & Prentice, 1973), was developed for both uncensored and censored data. In uncensored scenarios, it treats the ranks of data points as arising from the marginal distribution, leading to the Cox likelihood. It allows for a statistical handling of tied data points by breaking ties in all possible ways, which though accurate, is computationally intensive. [14] Breslow's step function approach (1974), assumes a step function for the baseline hazard with changes at observed failure times. Simplifies computations but is less theoretically satisfying as the model depends on the data itself. This approach is particularly useful for handling time-dependent covariates. [14] Bailey's non-parametric approach (1984), which uses a nonparametric maximization of the full likelihood. This provides estimators for regression coefficients and survival probabilities similar to those in previous methods. Finally the partial likelihood [14] (Cox, 1975), uses partial likelihood for estimation, separating the effect of nuisance parameters. This approach simplifies the computational process and can isolate useful data from noise.

2.2.2 Lasso Regularisation For Variable Selection

By penalizing the sum of the absolute values of the model parameters, the LASSO method encourages models with fewer parameters. This can lead to the exclusion of some variables entirely if their effect is not strong enough to justify a larger coefficient size given the regularisation penalty. LASSO can incorrectly include or exclude important variables, known as false discoveries. Enhanced methods like Adaptive LASSO, and Stability Selection are used to improve variable selection accuracy. The choice of λ affects the sparsity of the resulting model; too large a λ might shrink all coefficients to zero. The λ parameter is often chosen via cross-validation by optimizing some criterion (e.g., AIC, BIC, MSE). [6] Popular extensions of the lasso method includeded in Table 2.1.

Method	Library	Description
LASSO	glmnet	Regularized regression encourage sparse solutions by adding a penalty proportional to the absolute value of the coefficients.
SCAD	ncvreg	Non-convex penalty that encourages sparsity without overly penalizing large coefficients, more continuity in coefficient estimation compared to Lasso
Adaptive Lasso	adapl & glmnet	weights penalties based on initial estimates to improve consistency.
Dantzig Selector	Dant & flare	Ensures residuals are small and the solution is sparse, focussing on covariate selection accuracy
Relaxed Lasso	relaxl & relaxo	Combines Lasso solution with unpenalized least squares, reducing bias and variability
Square-root Lasso	sqrtl & flare	Modification of lasso stabilizing noise level variability
Scaled Lasso	Scail & scalreg	Adjusts the penalty term dynamically based on residual variance, improving error rate and variable selection

TABLE 2.1: Libraries illustrating Lasso implementations. [6]

Adaptive Lasso

By using weights that are inversely proportional to the magnitude of initial estimates, [38] Adaptive Lasso can differentiate more effectively between relevant and irrelevant predictors, for variable selection. Regularisation helps prevent overfitting, a common issue in models trained on high-dimensional data. [38] By penalizing the sum of the absolute values of the coefficients, the Adaptive Lasso ensures that the model generalizes well to unseen data. Compared to the standard Lasso, the adaptive version reduces the bias in the estimation of large coefficients, which is beneficial when true model coefficients vary in size.

$$\min \left[-\ell(\beta) + \lambda \sum_{j=1}^{p} \frac{|\beta_j|}{|\beta_j|^{\gamma}} \right] \tag{2.1}$$

The Adaptive Lasso adds a penalty that adjusts according to the initial estimates of the coefficients. This penalization mechanism performs two critical roles [38], Shrinkage; coefficients estimated to be small by the initial model are shrunk towards zero more aggressively, reducing the model's complexity and enhancing interpretability, Selection; larger coefficients (i.e., those considered more significant in the initial model) are penalized less, allowing them to stand out in the final model, thus maintaining their impact on the model's predictions. Each coefficient is updated in turn, optimizing the objective function concerning one β while keeping the others fixed [38]. The algorithm iterates over all coefficients repeatedly until convergence is achieved, usually defined by a small change in the value of the objective function

Outcome Adaptive Lasso

The Outcome-adaptive Lasso (OALasso) [29] modifies the standard Lasso penalty by weighting the regularisation of each coefficient according to its association with the outcome variable. This is intended to handle situations common in causal inference where the goal is not just prediction but understanding which variables causally affect the outcome. The OALasso minimization [29] problem is formulated as:

$$\min_{\beta} \left\{ \frac{1}{2n} \sum_{i=1}^{n} (y_i - x_i^T \beta)^2 + \lambda \sum_{j=1}^{p} w_j |\beta_j| \right\} \tag{2.2}$$

Where w_j are weights that are inversely proportional to the absolute values of the estimated coefficients from a preliminary unpenalized regression on the outcome. This weighting scheme is calculated as follows:

$$W_j = \frac{1}{|\hat{\beta}_j^{OLS}|^{\gamma}} \tag{2.3}$$

The term $\hat{\beta}_j^{OLS}$ is the ordinary least squares estimates for each predictor. γ is a tuning parameter that determines how the weights decay; commonly set to values like 0.5 or 1 depending on the desired sensitivity. The penalty weights w_j ensure that predictors with smaller absolute coefficients in a simple OLS regression on the outcome are penalized more heavily, under the assumption that they are less likely to be causally related to the outcome. By focusing the regularisation in this way, [29] OALasso aims to retain variables in the model that are more likely to be true causal factors rather than merely correlated with the outcome. The outcome-adaptive weighting mechanism can be justified theoretically by considering the bias-variance tradeoff and the properties of estimators in high-dimensional settings. Predictors with large coefficients are less likely to be due to random fluctuations in the data; hence, reducing their penalty helps to reduce bias without a substantial increase in variance. The λ and γ parameters must be carefully tuned, often via "cross-validation", to balance complexity for the model against the risk of overfitting. This method is more computationally intensive than standard Lasso due to the need for preliminary OLS estimation and weight calculation, OALasso can be implemented efficiently using iterative algorithms [29] similar to those used for other Lasso variations.

2.2.3 Random Survival Forest

Survival Trees as the Foundation of RSF

The Random Survival Forest (RSF) builds upon the concept of survival trees, which are a variant of decision trees specifically adapted for survival analysis. In survival trees, nodes are split based on criteria that consider the time-to-event data and censoring. Unlike traditional decision trees that focus on classification or regression, survival trees use criteria such as the log-rank test to evaluate potential splits. This approach ensures that the survival differences between the resulting groups are

maximized, effectively capturing the underlying structure of the data.

To reduce tree correlation and prevent overfitting [23], two main mechanisms are used namely bagging (Bootstrap Aggregating) and random feature selection at each split. Tuning common parameters like the number of trees (ntrees), the number of features (mtry) considered at each split, and the minimum sample size per node (nmin), is critical for optimizing random forest performance. A benefit of the model is its ability to capture survival functions for an individual in the distribution by estimating its survival function across all trees where the individual is captured in terminal nodes.

A critical aspect of survival trees is the handling of censored data. Censoring occurs when the exact time of the event (e.g., death, failure) is unknown for some subjects. The RSF algorithm manages this by considering both the event occurrence and the censoring information during the node-splitting process, which allows for the accurate estimation of survival functions even in the presence of incomplete data.

A benefit of the model is well suited for high dimensional data because of the random subset selection process, which helps mitigate overfitting. Due to the permutative nature of the ensemble bound to the brevity of the underlying data distribution, the model is computationally demanding [23], and although the model can yield variable information, it might be difficult to interpret the final resulting model, because the correlated variables don't necessarily account for mutual information between.

It is important to note here that [10] puts forward an approach to deal with missing data, outlining the short-comings of prior methods like replacing missing values with distribution medians, and for categorical data replacing with most frequent occurrences. The method is called adaptive tree imputation and relies on the OOB data set to determine missing data, for both continuous or integer values. This method is a part of the model and deals with censoring implicitly, which is different from external simulation and imputation methods.

Splitting Criteria

The performance of an RSF model heavily depends on the choice of splitting criteria. The log-rank test is the most widely used splitting criterion in survival analysis [34]. It evaluates potential splits based on the difference in survival between the groups created by the split. The log-rank statistic is given by:

$$\text{Log-Rank Statistic} = \frac{\sum_{i=1}^{n}(O_i - E_i)}{\sqrt{\sum_{i=1}^{n} V_i}}$$

where O_i and E_i are the observed and expected number of events, respectively, and V_i is the variance. This test helps in identifying the split that maximizes the difference in survival between the resulting groups.

While the traditional log-rank test is widely used, several advanced splitting criteria have been developed to improve the model's accuracy, particularly in high-dimensional settings [34]. For example, the AUC (Area Under the Curve) splitting criterion and C-index splitting have been introduced to better capture the relationship between covariates and survival times. These criteria aim to maximize the discrimination between different survival outcomes, which can be especially useful when dealing with complex, noisy datasets.

Moreover, the introduction of L1 splitting, which leverages the Kaplan-Meier estimator to measure differences between survival curves, provides an alternative that can be more robust in certain scenarios, such as when the proportional hazards assumption is violated [34]. These advanced splitting techniques enhance the flexibility and applicability of RSF in diverse research contexts, making it a powerful tool for survival analysis.

Handling High-Dimensional Data

One of the significant advantages of RSF over traditional survival models, like the Cox Proportional Hazards model, is its ability to handle high-dimensional data effectively [34]. The random selection of features at each split (random subspace method) helps mitigate the risk of overfitting, which is a common challenge when

the number of predictors far exceeds the number of observations. This characteristic makes RSF particularly well-suited for modern biomedical datasets, where high-dimensional genomic or imaging data are common.

RSF's capability to capture complex interactions between variables without requiring explicit model specifications further adds to its utility in high-dimensional settings. This non-parametric nature allows RSF to uncover intricate patterns that might be missed by more rigid, parametric models.

Recent Extensions and Improvements

Recent advancements in RSF have focused on extending its applicability and improving its computational efficiency. The development of Oblique Random Survival Forests (ORSF) is a notable example [11]. ORSFs use oblique splits, which are linear combinations of features, rather than axis-aligned splits that use a single feature at a time. This approach allows ORSFs to handle correlated predictors more effectively and can lead to improved prediction accuracy. However, the increased computational complexity of ORSFs has led to the development of specialized algorithms, such as the use of Newton-Raphson scoring, to reduce the time required for model training.

Another area of improvement is in variable importance measures[11]. Traditional permutation-based methods may not be as effective for oblique splits due to the linear combinations of features involved. Innovations like Negation Variable Importance (Negation VI) have been proposed to address this, offering more accurate assessments of feature importance in models with complex interactions.

These advancements are particularly relevant for applications involving high-dimensional, correlated data, where traditional RSF methods might struggle. The availability of these new techniques in R packages, such as 'aorsf,' has made these cutting-edge methods more accessible to researchers and practitioners.

Learner Class	Software	Learners	Description
Random Survival Forests			

Axis based	`RandomForestSRC`
	`ranger`
	`party`
	`rotsf`
	`rsfse`
		`rsf-standard`
		`rsf-extratrees`
		`cif-standard`
		`cif-rotate`
		`cif-spacextend`
			`rsf-standard` grows survival trees following Leo Breiman's original random forest algorithm with variables and cut-points selected to maximize a log-rank statistic. `rsf-extratrees` grows survival trees with randomly selected features and cut-points. `cif-standard` uses the framework of conditional inference to grow survival trees. `cif-rotate` extends `cif-standard` by applying principal component analysis to random subsets of data prior to growing each survival tree. `cif-spacextend` derives new predictors for each tree in the ensemble, separately.

Oblique	`obliqueRSF`
	`aorsf`
		`obliqueRSF-net`
		`aorsf-net`
		`aorsf-fast`
		`aorsf-cph`
		`aorsf-extratrees`
			Oblique survival trees following Leo Breiman's random forest algorithm. Linear combinations of inputs are derived using `glmnet` in `obliqueRSF-net` and `aorsf-net`, using Newton Raphson scoring for the Cox partial likelihood function in `aorsf-fast` (1 iteration of scoring) and `aorsf-cph` (up to 20 iterations), and chosen randomly from a uniform distribution in `aorsf-extratrees`. Cut-points are selected from 5 randomly selected candidates to maximize a log-rank statistic.

Boosting ensembles			
Trees	`xgboost`	`xgboost-cox` `xgboost-aft`	`xgboost-cox` maximizes the Cox partial likelihood function, whereas `xgboost-aft` maximizes the accelerated failure time likelihood function. Nested cross validation (5 folds) is applied to tune the number of trees grown, the minimum number of observations in a leaf node was 10, the maximum depth of trees was 6, and $\sqrt{p}$ variables were considered randomly for each tree split, where p is the total number of predictors.
Regression models			
Cox Net	`glmnet`	`glmnet-cox`	The Cox proportional hazards model is fit using an elastic net penalty. Nested cross validation (5 folds) is applied to tune penalty terms.
Neural networks			
Cox Time	`survivalmodels`	`nn-cox`	A neural network based on the proportional hazards model with time-varying effects. Nested cross-validation was applied to select the number of layers (from 1 to 8), the number of nodes in each layer (from $\sqrt{p}/2$ to $\sqrt{p}$), and the number of epochs to complete (up to 500). A drop-out rate of 10% was applied during training.

FIGURE 2.2: [11] Shows available packages based on model types for random survival forests.

2.2.4 Applied example of these models in a simulation studies

[16] Evaluated and compared the effectiveness of Cox regression analysis (CRA) and random survival forests (RSF) through both simulated and actual breast cancer data scenarios. Initially, the study utilized Monte Carlo simulations to assess how both methods performed across various sample sizes, specifically observing their performance metrics based on Harrell's concordance index. The results indicated that CRA consistently outperformed RSF under simulation conditions, particularly when using the concordance index for evaluation. Following the simulations, the methods were applied to a real dataset comprising 279 breast cancer patients to identify major risk factors influencing disease-free survival (DFS). In this practical application, RSF slightly edged out other methods, offering marginally better performance according to the concordance index when using the approximate log-rank splitting rule, compared to the other log-rank rules.

[16] CRA was noted for its predictive accuracy across different sample sizes, making it suitable for a broad range of survival data applications. Conversely, RSF was

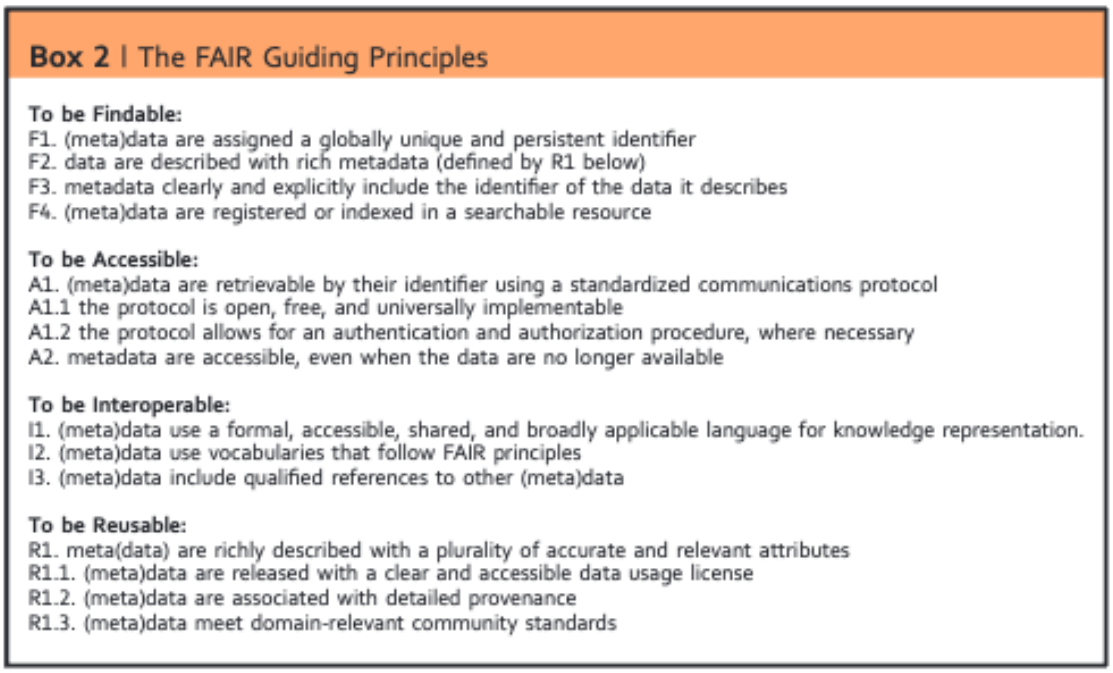
recommended for its interpretative power, especially beneficial in handling complex datasets where multiple survival trees are analyzed.

Furthermore, [15] shows a comparison between a machine learning method, termed survival neural networks (SNNs) and compared it with the Cox proportional hazards model, using clinical trial data for survival outcomes. The models are formulated subject to the European Osteosarcoma intergroup trial data, which is used as the foundation for the synthetic data generation that would ultimately be used for simulation training. The original dataset contains various instances of censoring, and the authors approach this issue, by segmenting the datasets into samples with degrees of censoring present (20%, 40%, 61%, 80%), after which data imputation techniques such as the inverse probability weighting, censoring method (IPW), was used, which is based on calibration procedures outlined in the paper, to ensure the synthetic data retains the statistical properties of the original clinical data.

Accuracy (Brier) interpreted in continuous form as the integrated Brier score for prediction error over the total period, and lastly Miscalibration (mean squared error) for censored groups. [15] The results indicated comparable predictive performance but highlighted a lack of accuracy for calibration measures with SNNs. The authors point out that although machine learning techniques are attractive for survival analysis scenarios because of the ability to model interactions and nonlinearities with a no-assumption approach, the robustness of the Cox model, regarding ease of implementation as well as interpretability of covariates makes it formidable in situations where limited sample sizes and variables are available. The paper ties in nicely with the other literature in support of the need for clear and better implementation of calibration metrics specifically with machine learning models, and caution against indiscriminate application of these models.

2.3 Data Generating Mechanisms For Simulation

Methods for extrapolating missing data, speaks to the key feature of survival analysis and the difficult problem of dealing with censored data. Several different



Box 2 | The FAIR Guiding Principles

To be Findable:
F1. (meta)data are assigned a globally unique and persistent identifier
F2. data are described with rich metadata (defined by R1 below)
F3. metadata clearly and explicitly include the identifier of the data it describes
F4. (meta)data are registered or indexed in a searchable resource

To be Accessible:
A1. (meta)data are retrievable by their identifier using a standardized communications protocol
A1.1 the protocol is open, free, and universally implementable
A1.2 the protocol allows for an authentication and authorization procedure, where necessary
A2. metadata are accessible, even when the data are no longer available

To be Interoperable:
I1. (meta)data use a formal, accessible, shared, and broadly applicable language for knowledge representation.
I2. (meta)data use vocabularies that follow FAIR principles
I3. (meta)data include qualified references to other (meta)data

To be Reusable:
R1. meta(data) are richly described with a plurality of accurate and relevant attributes
R1.1. (meta)data are released with a clear and accessible data usage license
R1.2. (meta)data are associated with detailed provenance
R1.3. (meta)data meet domain-relevant community standards

FIGURE 2.3: [35] Fair principles summary

methodologies and considerations exist outlining how to impute, simulate, and generate data, for different settings of survival data.

2.3.1 Data Simulation Methods

[32] Shows that there is a scarcity of publicly available real-world datasets for benchmarking models in oncology, particularly those that comply with the FAIR Data Principles. This significantly limits the ability to conduct model comparisons using real patient data, which is crucial for ensuring the models' applicability and robustness in real-world scenarios. The challenge of simulating realistic survival data for model comparison in situations where real-world data is not available or is incomplete. This is particularly relevant for ensuring that simulated data can reliably mimic real-world outcomes to validate model performance effectively. [32] proposes several methods for dealing with data simulation.

2.3.2 Data Imputation Methods

Data imputation is crucial for addressing issues arising from censored data by replacing missing values with values that resemble others in the distribution. Censoring happens because of factors like; the end of the observation period, loss before observation, or discontinuation of study participation. This often prevents the collection of complete data on the time until an event of interest occurs and so [12] classifies these scenarios into censoring classes; censored at random as well as censored not at random (CAR & CNAR). Data imputation is relevant to these contexts to correct for the potential biases introduced by censoring, especially when it is informative or non-random. [12] Shows the Cox Proportional Hazards model assumes noninformative censoring (CAR) for its analysis. In cases where this assumption might not hold due to practical reasons, such as decisions influenced by patient or physician preferences, they point to data imputation under CNAR assumptions as a way to account for potential biases.

Under the CAR assumption [12], the hazard function after censoring is assumed to be the same as if the subject had not been censored, conditional on the covariates. This reflects the assumption that the censoring is non-informative regarding the survival probability. This means that the survival and hazard functions do not need special adjustments beyond the point of censoring other than ensuring that the analysis correctly accounts for the time of censoring. For CNAR, [12] the assumption is that the hazard of having an event after censoring may differ from that before censoring due to the censoring being potentially dependent on unobserved variables affecting the hazard. Mathematically, this means the post-censoring hazard function cannot simply extend the pre-censoring model. In practical terms, the CNAR model needs to be specifically formulated to reflect how the hazard might increase or decrease post-censoring due to factors related to why the censoring occurred. This could involve modifying the functional form of λ_{post} or using additional data and techniques to estimate it. [12] Shows 4 methods for handling data imputation under the CNAR and CAR assumptions.

2.3.3 Synthetic Data Generation Methods

Machine learning methods for simulation and data generation have risen in popularity recently, [20] shows multiple methods combining statistical imputation methods into machine learning architectures. Specifically [20] extend prior work for survival analysis to generate synthetic data using a Conditional Generative Adversarial Network (GAN) framework. The process integrates various components to handle different data types and ensure a realistic simulation of survival times based on censoring and event data.

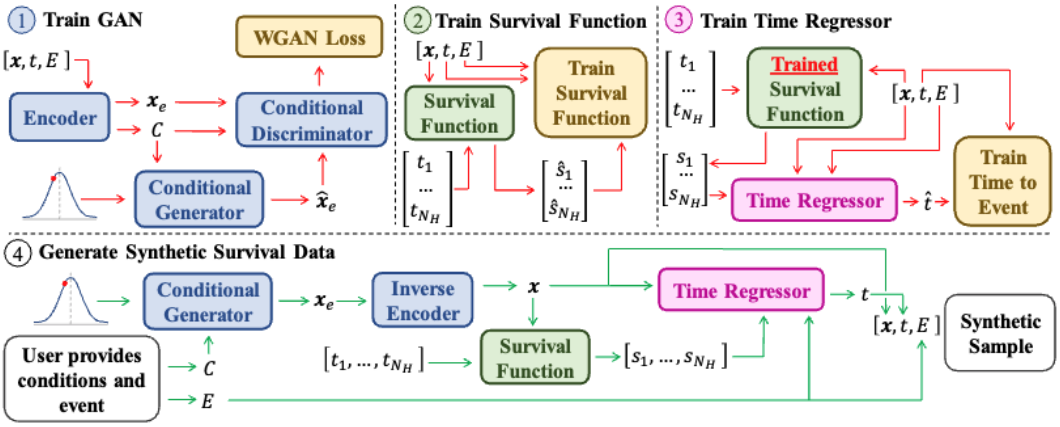


FIGURE 2.4: [20] Survival GAN architecture

The tabular encoder converts continuous features into a format suitable for the GAN using a Gaussian Mixture Model (GMM) [20]; each feature is represented by its GMM component and deviation from the component mean. Categorical features are directly converted into one-hot vectors. The Encoder simplifies the full tabular encoding to just the one-hot vector component, representing condition C for the generator. For the Survival GAN architecture, they used models like DeepHit [20] to estimate probabilities. The Time Regressor component predicts the actual time of an event or censoring based on the outputs of the survival function and the event type (E). This component can utilize various regression models, such as XGBoost [20].

During training the loss function used was a Wasserstein GAN [20] with a gradient penalty which ensures stable training of the GAN by adjusting the generator and discriminator losses to minimize the distance between real and generated data distributions while penalizing gradient norms. This allows for user-defined or sampled conditions and events. The GAN uses these to generate encoded covariates, which are then decoded back to their original form. These covariates are input into the survival function and time regressor to finally produce the synthetic time-to-event data. This architecture allows SurvivalGAN to realistically model survival data by accurately simulating the underlying time-to-event dynamics and handling various data imbalances and types.

2.4 Model Evaluation And Result Interpretation

In evaluating the performance of survival prediction models, it is crucial to employ metrics that not only assess accuracy but also ensure fairness by minimizing bias. This involves selecting evaluation measures that effectively balance calibration (the agreement between predicted and observed outcomes) and discrimination (the model's ability to distinguish between different outcomes). [31] Provides a guidelines and comprehensive framework for assessing model performance against true event times, ensuring that the models are both fair and accurate across different scenarios. These are essential for advancing survival analysis in ethically sensitive domains, thereby supporting more reliable and equitable outcomes. Following are some common methods to evaluate model results.

	Distribution		
	Exponential	Weibull	Gompertz
No competing risks			
Density function	$f(t) = \lambda \exp(-\lambda t)$	$f(t) = \lambda \nu t^{\nu-1} \exp(-\lambda t^{\nu})$	$f(t) = \lambda \exp(\alpha t) \exp[\frac{\lambda}{\alpha}\{1 - \exp(\alpha t)\}]$
Hazard function	$h(t) = \lambda \exp(\beta X)$	$h(t) = \lambda \nu t^{\nu-1} \exp(\beta X)$	$h(t) = \lambda \exp(\alpha t) \exp(\beta X)$
Survival function	$S(t) = \exp\{-\lambda \exp(\beta X) t\}$	$S(t) = \exp\{-\lambda \exp(\beta X) t^{\nu}\}$	$S(t) = \exp\left[\frac{\lambda \exp(\beta X)}{\alpha}\{1 - \exp(\alpha t)\}\right]$
Event time	$t = \frac{-\log\{S(t)\}}{\lambda \exp(\beta X)}$	$t = \sqrt[\nu]{\frac{-\log\{S(t)\}}{\lambda \exp(\beta X)}}$	$t = \frac{1}{\alpha}\log\left[1 - \frac{\alpha \log\{S(t)\}}{\lambda \exp(\beta X)}\right]$
Competing risks			
Marginal hazard function	$h_1^{Marg}(t) = \lambda_1 \exp(\beta X)$ $h_2^{Marg}(t) = \lambda_2$	$h_1^{Marg}(t) = \lambda_1 \nu t^{\nu-1} \exp(\beta X)$ $h_2^{Marg}(t) = \lambda_2 \nu t^{\nu-1}$	$h_1^{Marg}(t) = \lambda_1 \exp(\alpha t) \exp(\beta X)$ $h_2^{Marg}(t) = \lambda_2 \exp(\alpha t)$
Joint survival function	$S(t) = \exp[-\{\lambda_1 \exp(\beta X) + \lambda_2\} t]$	$S(t) = \exp[-\{\lambda_1 \exp(\beta X) + \lambda_2\} t^{\nu}]$	$S(t) = \exp\left[\frac{\lambda_1 \exp(\beta X) + \lambda_2}{\alpha}\{1 - \exp(\alpha t)\}\right]$
Event time	$t = \frac{-\log\{S(t)\}}{\lambda_1 \exp(\beta X) + \lambda_2}$	$t = \sqrt[\nu]{\frac{-\log\{S(t)\}}{\lambda_1 \exp(\beta X) + \lambda_2}}$	$t = \frac{1}{\alpha}\log\left[1 - \frac{\alpha \log\{S(t)\}}{\lambda_1 \exp(\beta X) + \lambda_2}\right]$
Frailty model	$t_{ij} = \frac{-\log\{S(t_{ij})\}}{[\lambda_1 \exp(\beta X_{ij}) + \lambda_2] Z_i}$ Z_i is a frailty term	$t_{ij} = \sqrt[\nu]{\frac{-\log\{S(t_{ij})\}}{\{\lambda_1 \exp(\beta X_{ij}) + \lambda_2\} Z_i}}$ Z_i is a frailty term	$t_{ij} = \frac{1}{\alpha}\log\left[1 - \frac{\alpha \log\{S(t_{ij})\}}{\{\lambda_1 \exp(\beta X_{ij}) + \lambda_2\} Z_i}\right]$ Z_i is a frailty term
Probability transform	$t_{ij} = \frac{-\log\{\Phi(y_{ij})\}}{\lambda_1 \exp(\beta X_{ij}) + \lambda_2}$ $\Phi(y_{ij})$ is cumulative normal probability	$t_{ij} = \sqrt[\nu]{\frac{-\log\{\Phi(y_{ij})\}}{\lambda_1 \exp(\beta X_{ij}) + \lambda_2}}$ $\Phi(y_{ij})$ is cumulative normal probability	$t_{ij} = \frac{1}{\alpha}\log\left[1 - \frac{\alpha \log\{\Phi(y_{ij})\}}{\lambda_1 \exp(\beta X_{ij}) + \lambda_2}\right]$ $\Phi(y_{ij})$ is cumulative normal probability
Event Indicator	$D = \begin{cases} 1, & \frac{\lambda_1 \exp(\beta X_{ij})}{\lambda_1 \exp(\beta X_{ij}) + \lambda_2} \\ 2, & \frac{\lambda_2}{\lambda_1 \exp(\beta X_{ij}) + \lambda_2} \end{cases}$	$D = \begin{cases} 1, & \frac{\lambda_1 \exp(\beta X_{ij})}{\lambda_1 \exp(\beta X_{ij}) + \lambda_2} \\ 2, & \frac{\lambda_2}{\lambda_1 \exp(\beta X_{ij}) + \lambda_2} \end{cases}$	$D = \begin{cases} 1, & \frac{\lambda_1 \exp(\beta X_{ij})}{\lambda_1 \exp(\beta X_{ij}) + \lambda_2} \\ 2, & \frac{\lambda_2}{\lambda_1 \exp(\beta X_{ij}) + \lambda_2} \end{cases}$

FIGURE 2.5: [17] Shows simulation formulas under spesific conditions.

2.4.1 C-Index

Variants of the C-index include; the time-independent C-index (Cti) which negates survival probability at some specified period. It assesses the sequence of actual event times matches the predicted times, time-dependent C-index (Ctd) shown by [27] and it accounts for varying amounts of censoring over time. The C-index can be biased upwards with a high level of censoring in the data. This issue is addressed through the Ctd rule, although it is not a proper scoring rule [27]. The C-index, while useful, does not always align with other metrics such as the Mean Absolute Error (MAE). A model can have a high C-index (accurately ranking the order of events) while still having large discrepancies in the actual predicted times of those events.

2.4.2 Brier Score and Integrated Brier Score

The Brier Score and the Integrated Brier score (IBS) are essential metrics used to evaluate the accuracy and reliability of survival models [7]. Both scores measure

the calibration and discrimination capabilities of a model, which are crucial for producing unbiased and precise predictions in survival analysis. Below is a detailed explanation of both metrics. A perfect model, which perfectly predicts whether events happen by time t^* (predicting 1s and 0s accurately), would have a Brier score of 0. A model that always predicts a 50% chance of survival regardless of the actual outcome will have a Brier score of 0.25, representing poor predictive accuracy. IBS is particularly useful for survival prediction models where it is important to assess model performance comprehensively across time rather than at a single time point. It gives an average score that reflects the overall performance of the model across the specified time interval. For censored data, the Inverse Probability Censoring Weight (IPCW) [7] technique is often used in conjunction with IBS to adjust the contributions of censored subjects. This method helps to ensure that the model's performance is not unduly biased by the censoring. IBS is considered a proper scoring rule if the censoring distribution is estimated correctly, meaning it incentivizes truthful forecasting and accurately reflects the model's predictive capabilities. IBS can be particularly impactful in clinical settings where decisions might depend on accurate, time-specific survival probabilities, such as deciding on conservative treatments based on predicted long-term survival chances.

2.4.3 Hosmer-Lemeshow Calibration (1-Calibration)

1-Calibration [7] measures how well the predicted probabilities of an event (e.g., failure, death) occurring by t^* match the actual proportion of those events in the dataset. This test is particularly useful in contexts where predictions need to be reliable at specific critical thresholds. A low value of the Hosmer-Lemeshow statistic [27] suggests that the model's predictions are well-calibrated i.e. the predicted probabilities of survival match the actual rates observed. The statistic follows a chi-squared distribution [27], allowing for the derivation of a p-value to assess the significance of the results. A model is considered well-calibrated at the chosen significance level if the p-value is greater than 0.05.

2.4.4 D-Calibration

D-Calibration measures the consistency of predicted probabilities across a range of outcomes within a dataset. It assesses whether the distribution of predicted probabilities (over time or across conditions) matches the observed distribution of outcomes [7]. Predicted probabilities are checked across a range of values. For each interval [a,b] within the probability range [0,1], the proportion of subjects with predicted probabilities within this range is compared to the actual proportion of events occurring in this interval. The fraction of subjects per interval is expected to match the width of the interval $b - a$. For instance, for the interval [0.1, 0.2], approximately 10% of the subjects should ideally have their predicted probabilities fall within this range if the model is perfectly D-Calibrated [7]. A chi-squared test can be used to assess the uniformity of the distribution of predictions across the intervals, providing a statistical measure of calibration [7].

2.4.5 Scoring Theory

Scoring rules are essential tools in statistics and machine learning for evaluating the accuracy of probabilistic predictions. [37] It is used to measure the quality of predictions by assigning a numerical score based on the probability forecast and the actual outcome. Thus it helps assess how well a model predicts the timing of future events, such as failures or deaths. A proper scoring rule incentivizes truthful forecasting [37], meaning it rewards the forecaster if the predicted distribution closely matches the true distribution of outcomes. A scoring rule is called proper [37] if the expected score is minimized when the prediction model uses the true probability distribution. It is strictly proper if the score is uniquely minimized by the true distribution [37].
Proper Scoring Rule:

$$E[(t,c) \sim (T,C)][S(\hat{F},(z,\delta))] \geq E[(t,c) \sim (T,C)][S(F,(z,\delta))] \tag{2.4}$$

Strictly Proper Scoring Rule:

$$E[(t,c) \sim (T,C)][S(\hat{F},(z,\delta))] > E[(t,c) \sim (T,C)][S(F,(z,\delta))] \quad \text{if} \quad \hat{F} \neq F \tag{2.5}$$

Chapter 3

Research Methodology

3.1 Research design

In this I look at the structural foundation for running the comparative study of the Random Survival Forest [10] and the Cox proportional hazards model [2]. Central to this design is the use of **simulation studies**, to gain empirical insights into the behavior of the statistical methods across various scenarios. As [18] aptly puts it, "Simulation studies are used to obtain empirical results about the behavior of statistical methods in certain scenarios, as opposed to analytic results."

This section focuses on the application of the (ADMEP) framework, formalized by [18]. This framework is the building block of most of the methodical parts later on in the methodology. It served as the backbone of my methodology, allowing me to produce results that "accurately" reflect the performance and effectiveness of these models.

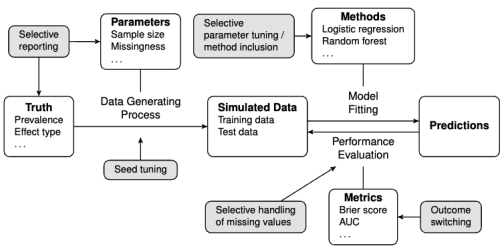


FIGURE 3.1: [21] Shows a common example of a simulation study plan.

Taking inspiration from the the application of ADMEP by [21], the following sections define methods relevant in each section of the framework for later use in the methodology:

3.1.1 Aims

- To evaluate how effectively the Random Survival Forest and Lasso-Cox models predict survival outcomes.
- To understand model behavior under varying conditions of data complexity and censoring rates.
- To understand the effects of data and simulation methods.

3.1.2 Data-generating mechanisms

- Simulate datasets with packages like [3] to introduce complexities like varying censoring rates and non-linear effects.
- Introduce multicollinearity and various interaction effects to challenge the models' assumptions and robustness.

Missing Data Imputation

As discussed in 2.3.2 we look at the methods for producing missing value imputation under the CAR and CNAR assumptions.

Delta-adjusted Method [12]

$$\lambda_{post}(t|Z_i, X_i(t)) = \lambda_{\varnothing}(t|Z_i, X_i(t)) = \lambda_0(t)\exp^{(1-\varnothing)\beta Z_i + \alpha X_i(t)}, t > C_i \tag{3.1}$$

The sensitivity parameter ($\varnothing$) represents a discounted proportion of the log-hazard ration β under the CNAR assumption. The method assumes that the hazard of having an event for censored subjects is multiplicatively decreased by a factor depending on the $\varnothing$. The term $(1-\varnothing)\beta$ suggests a reduced effect of the treatment on the hazard rate post-censoring.

Tipping point Analysis [12]
This method is used to identify how much the assumption of noninformative censoring (CAR) would need to be violated for the study results to become statistically nonsignificant. It uses the delta-adjusted method's parameter $\varnothing$ to calculate the

minimum shift in β (log HR) needed to nullify the treatment effect.

Jump to Reference [12]

$$\lambda_{post}(t|Z_i, X_i(t)) = \lambda_{J2R}(t|Z_i, X_i(t)) = \lambda_{ref}(t|X_i(t)) = \lambda_0(t)\exp^{\alpha X_i(t)}, t > C_i \quad (3.2)$$

This Method assumes that once censored, the hazard rate for a subject from the treatment group immediately aligns with that of the reference group, completely disregarding any residual effects of the treatment past the point of censoring.

Copy Reference [12]

$$\lambda_{CR}(t|Z_i = 1, X_i(t)) = \lambda(t|Z_i = 0, X_i(t)) = \lambda_0(t)\exp^{\alpha X_i(t)} \text{ for all}, t \quad (3.3)$$

This method is conservative but less so than J2R in that it assumes that if a subject in the treatment group is censored, their hazard function matches that of the reference group for the remainder of the study. It does not account for any time-specific variations in hazard that might have been influenced by the treatment before censoring.

Censored at Random (CAR) [12]
Assumes noninformative censoring where the post-censoring hazard (λ_{post}) is identical to the pre-censoring hazard, simplifying the imputation process.

Hazard post CAR:

$$\lambda_{post}(t|Z_i, X_i(t)) = \lambda_0(t)\exp^{\beta Z_i + \alpha X_i(t)}, t > C_i \quad (3.4)$$

Hazard post CNAR:

$$\lambda_{post}(t|Z_i, X_i(t)) \neq \lambda_0(t)\exp^{\beta Z_i + \alpha X_i(t)}, t > C_i \quad (3.5)$$

Data Simulation

Parametric Distribution
This method is used to simulate survival times using predefined parametric distributions. The fit of each distribution to the data is tested using the one-sample Cramér-von Mises (CVM) test [32]. The distribution that shows the least deviation

from the data (highest p-value in the CVM test) is chosen for simulation. Parameters for these distributions are estimated using Maximum Likelihood Estimation (MLE). An Example of mixed distribution:

$$f(x) = 0.2.f_{Weibull}(\alpha, \lambda) + 0.8.f_{norm}(\mu, \sigma)(x) \tag{3.6}$$

Where $f_{Weibull(\alpha,\lambda)}$ and $f_{norm}(\mu, \sigma)$ are the density functions for the Weibull and normal distributions, respectively.

Kernel Density Estimation (KDE)
Used to estimate by utilizing the density function of the data for simulation. The density function is estimated using the kdensity function from the [32] R-package, which employs a Gaussian kernel. The Accept-reject method is used to generate random values that follow the estimated density function. Draw (X, U) from the joint distribution (X, U) $\{(x, u) : 0 < u < f(x)\}$ where X is a random variable following the estimated density f, and U is uniformly distributed between 0 and $f(x)$. If $u_i < f(x_i)$ for a sampled (X, U), x_i is accepted as a realization from the density function f

Case Resampling
Simulate data by resampling observed data points with replacement. Directly resample observations (ti, di) from the existing dataset of observed times and censoring indicators. $(t_i, d_i)^*$ are drawn with replacement from $\{(t_1, d_1) \dots (t_n, d_n)\}$

Conditional Bootstrapping
To simulate data using bootstrapping that accounts for censoring, this helps sample censor times. For censored observations, censoring times are carried over to the simulated data. For uncensored observations, new censoring times are sampled based on the conditional distribution of censoring times given they are greater than the observed event time. Censoring Time for censored data: $c_i^* = t_i$ for censored observation. Censoring Time for uncensored data: c_i^* is sampled from $G_i(c) = \frac{G(c)-G(t_i)}{1-G(t_i)}$ where G is the distribution function of censoring times and t_i is the observed event time. Event times t_{oi}^* are sampled from the observed event times with replacement. [32] Show that the reconstruction of reliable benchmark data sets

is meticulously reconstructed from published Kaplan-Meier plots and other vital statistics like hazard ratios and p-values from log-rank tests. The data sets, particularly those with non-crossing survival curves, demonstrate high fidelity to the original data, proving them to be excellent resources for subsequent model evaluations.

Synthetic Data Generation

[20] Employ several metrics to assess synthetic survival data according to the real dataset. They help identify and correct biases in synthetic data to better align with real-world data, enhancing the credibility of survival analysis models. The optimism metric assesses whether the synthetic data is over-optimistic or over-pessimistic compared to real data by comparing the expected lifetimes derived from Kaplan-Meier (KM) plots.

$$\text{Optimism} = \frac{1}{T}\int_0^T (S_{Syn}(t) - S_{Real}(t))dt \tag{3.7}$$

Where $S_{Syn}(t) and S_{Real}(t)$ are the synthetic and real Kaplan-Meier survival functions respectively, and T is the latest time point available. The optimism metric ranges between -1 and 1, where values closer to 0 indicate accurate lifetime predictions, positive values suggest over-optimism, and negative values indicate over-pessimism. The short-sightedness metric quantifies how much earlier synthetic data, discontinues providing predictions compared to real data, reflecting potential censorship in the synthetic modeling.

$$\text{Short-Sightedness} = \frac{T_{Real} - T_{Syn}}{T_{Real}} \tag{3.8}$$

Here, $T_{Syn} and T_{Real}$ are the end times of the synthetic and real Kaplan-Meier plots, respectively. This metric ranges from 0 to 1, where 0 indicates no censorship issues and 1 indicates complete short-sightedness in the synthetic data predictions. Lastly, the Kaplan-Meier divergence measures the overall divergence between the synthetic and real KM curves across the observed time, providing a comprehensive

measure of similarity between the two datasets.

$$\text{KM Divergence} = \frac{1}{T}\int_0^T |S_{Syn}(t) - S_{Real}(t)|dt \tag{3.9}$$

This formula calculates the mean absolute difference between the synthetic and real survival functions over time, scaled by the total duration observed. The KM divergence values range from 0 to 1, where 0 indicates perfect matching KM curves and 1 represents the maximum possible difference.

3.1.3 Methods

- Random Survival Forest: Implement using a combination of proposed packages in 3.3, tuning tree-related parameters.
- Lasso-Cox Model: Use a combination of the proposed packages in 3.3 for implementing Cox regression with Lasso regularization, optimizing the regularization strength and model complexity.
- Explore integration effects from the different packages.

Cox Proportional Hazards Model

The proposition consists of covariates, known as attributes regarding a unit in a distribution of data, which is associated with a coefficient β scaling the impact of said covariates; this product is then bound by the baseline hazard $h_0(t)$.

$$h(t|X) = h_0(t)\exp(\beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n X_n) \tag{3.10}$$

An assumption of the Cox model is the proportional hazards assumption, suggesting that the hazard ratios for different covariates remain constant over time we see this for two events observations. This is an operational Assumption and a limitation as this is not always true for survival data.

$$\frac{h(t|X_1)}{h(t|X_2)} = \frac{h_0(t)\exp(\beta^T X_1)}{h_0(t)\exp(\beta^T X_2)} = \frac{\exp(\beta^T X_1)}{\exp(\beta^T X_2)} = \exp(\beta^T(X_1 - X_2)) \tag{3.11}$$

The model can handle censoring, by adjusting the likelihood function for observations where event occurrence did not happen in a particular continuous time frame, and by maximizing the likelihood of all observed events, it is possible to estimate the coefficients that could work the best under the Cox formulation.

$$L(\beta) = \prod_{i:\delta_i=1} \frac{\exp(\beta^T X_i)}{\sum_{j \in R(t_i)} \exp(\beta^T X_j)} \tag{3.12}$$

Lasso Regularisation

The Lasso technique [33], a regression analysis method introduced to address specific limitations of Ordinary Least Squares (OLS) estimation, is particularly beneficial in scenarios with a large number of predictors or high collinearity among them which would mean that models could produce inflated variance scores, and cause impaired interpretability. Lasso optimizes prediction accuracy and enhances model interpretability by employing a shrinkage process that can set certain coefficients to zero effectively, thus performing variable selection.

$$\hat{\beta}^{lasso} = \arg\min_{\beta} \left\{ \frac{1}{2N} \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} X_{ij}\beta_j)^2 \right\} \tag{3.13}$$

bound by,

$$\sum_{j=1}^{p} |\beta_j| \leq t. \tag{3.14}$$

[6] The inclusion of the regularisation term, is crucial as it allows for the reduction of model complexity by penalizing the magnitude of the coefficients, which promotes sparser models. This sparsity is instrumental in enhancing interpretability by isolating only the most significant predictors that contribute to the dependent variable. The objective function of LASSO is convex for l_1 norm, which simplifies finding the global minimum. Multiple optimal solutions might exist, especially when the number of predictors p exceeds the number of observations n. [6] LASSO introduces bias in the coefficients to achieve lower variance and better model parsimony. Consistent under conditions like the irrepresentable condition, crucial for variable selection. Bias Correction achieved via,

$$\hat{\beta}_j^{LASSO} = sign(\hat{\beta}_j^{OLS})(|\hat{\beta}_j^{OLS}| - \lambda)_+ \tag{3.15}$$

Random Survival Forest

Random survival forests [10] are an extension of random forests, which can handle right-censored data and aim to estimate the appropriate survival function. Consisting of an ensemble of trees [10], which are grown from a bootstrap sample, and each node of underlying trees, consists of specific covariates due to a random selection of features for splits in each tree.

Per node splitting criteria are conditional to survival time and censoring, whereby node "impurity" [10] is determined by the survival differences. Methods like log-rank, conservation of events splitting rule, and random log-rank are used. Terminal nodes are the result of saturated splitting criteria, with each endpoint having d-dimensional covariates of the individuals encapsulated.

A key component of the model is the conservation of events principle, which is used to define a type of predicted outcome, namely ensemble mortality, which is derived from the cumulative hazard function (CHF) using the Nelson-Aalen estimator in the original paper by [10].

Terminal nodes or nodes at the end of tree branches all share the estimated hazard function. Another key concept is the out-of-bag (OOB) samples which act as a validation subset [10].

The OOB error is calculated on the ensemble survival function of the observed data using metrics like concordance (C-index). Used for estimating prediction error and model performance without a separate test data set. Each tree's error is calculated using data not included in its training set (out-of-bag). Prediction error metrics, like the concordance index which calculates the permissible pairs per node and OOB prediction error, are used for accuracy metrics.

$$\hat{\Lambda}_{oob}(t, x_i) = \frac{1}{\sum_{b=1}^{B} I(x_i \in L_{oob}^b)} \sum_{b=1}^{B} I(x_i \in L_{oob}^b) \hat{\Lambda}_b(t, x_i) \tag{3.16}$$

3.1.4 Estimands

- Focus on estimations, like survival functions, hazard ratios, etc. for both models and survival probabilities at specified time points.
- Bootstrap samples to build confidence intervals for key estimands, ensuring the robustness of the findings.
- Analyze feature importance in RSF and assess how Lasso regularization affects the selection of covariates in the Cox model.

Lasso Penalised Cox Proportional Hazards

The log partial likelihood for the Cox model, as given by:

$$l(\beta) = \sum_{i:\delta_i=1} \left[\beta^T x_i - \log \sum_{j:t_j \geq t_i} \exp(\beta^T x_j) \right] \tag{3.17}$$

Which provides a measure of how well the model's predicted hazards match the observed data. Here, δ indicates whether an event (e.g., failure, death) was observed at time t_i.

Random Survival Forests

Random forests are adapted for survival analysis by modifying how predictions are aggregated to handle censored data effectively. Forest survival function estimator:

$$\hat{\Lambda}(t, x_0) = \frac{1}{B} \sum_{b=1}^{B} \hat{\Lambda}_b(t, x_0) \tag{3.18}$$

Survival function:

$$S(t, x_0) = \exp^{-\hat{\Lambda}(t, x_0)} \tag{3.19}$$

Variable importance (VI) [23] in random forests is used to rank variables based on their contribution to prediction accuracy. It is assessed by looking at each predictor variable in the sample and assessing the impact on prediction error, an increase in error indicating importance. Different random forest settings can yield different importance rankings due to the model's sensitivity to the configuration. VI is shown

by [23]:

$$VI(j) = \frac{1}{B}\sum_{b=1}^{B}(\frac{Err(j)_b}{Err_b} - 1) \tag{3.20}$$

VI Adaptation:

$$VI(j) = \frac{1}{B}\sum_{b=1}^{B}(\frac{Err(j|Z)_b}{Err_b} - 1) \tag{3.21}$$

This adaptation 3.21 addresses correlations [23] by conditioning on other related variables. This accounts for the influence of correlated predictors by adjusting the variable importance calculation.

3.1.5 Performance measures

- Use statistical tests, like the C-index, and integrated Brier score to compare the RSF and Lasso-Cox model's performance metrics, like predictive accuracy, discrimination, and calibration across time.
- Conduct sensitivity analysis to explore the impact of model parameters on their performance.
- Apply graphical methods like Kaplan-Meier curves to visualize survival estimates against actual outcomes.

C-Index

The C-index [27], serves as a crucial statistical measure in evaluating the effectiveness of survival models. This index assesses how well these models predict the sequence of patient outcomes based on their respective risk assessments. It is defined as the ratio of pairs of subjects (deemed "comparable") for which the model's predictions and actual outcomes are consistent in terms of event sequence. "Comparable" refers to pairs where the event sequence, i.e., who encountered the event first, can be established.

$$C\text{-index} = \frac{\sum_{i,j}\mathbf{1}(t_i < t_j)\cdot\mathbf{1}(\eta_i > \eta_j)\cdot\delta_i}{\sum_{i,j}\mathbf{1}(t_i < t_j)\cdot\delta_i} \tag{3.22}$$

In this formula, μ_i is the risk score for unit i, and δ_i indicates whether an event occurred (1 if it did, 0 otherwise)

Brier Score and Integrated Brier Score

Brier score is a measure used to evaluate the accuracy of probabilistic predictions. It is essentially the mean squared error [27] between the observed outcomes and the predicted probabilities at a specific time t^*.

$$BS_{t^*}(VU, \hat{S}(t^*|\cdot)) = \frac{1}{|VU|} \sum_{[\tilde{x}_i, d_i] \in VU} (I[d_i \leq t^*] - \hat{S}(t^*|\tilde{x}_i))^2 \tag{3.23}$$

Where, VU is the validation set, $\hat{S}(t^*|\cdot))$ is the predicted survival probability at t^* and $I[d_i \leq t^*]$ is the event indicator.

Integrated Brier Score (IBS) extends the Brier score across a range of times, providing a measure of model accuracy over time. In other words, it is the expectation of the Brier scores calculated at each time point within a specified interval.

$$IBS(\tau, VU, \hat{S}(\cdot|\cdot)) = \frac{1}{\tau} \int_0^\tau BS_t(VU, \hat{S}(t|\cdot))\, dt \tag{3.24}$$

Where, τ is the biggest event period, BS_t is the brier score at time t.

1-Calibration

1-Calibration, also known as Hosmer-Lemeshow calibration, is a test used to evaluate model calibration at a specific time point t^* [7]. It works by sorting all subjects for the predicted probabilities at time t^*. These probabilities are then grouped into a predefined number of groups or bins (typically 10). For each bin, number of events is determined based on the predicted probabilities, and this is compared to the actual number of events that occurred [27].

$$\text{HL}(VU, \hat{S}(t^*|\cdot)) = \sum_{j=1}^{B} \frac{(O_j - n_j \bar{p}_j)^2}{n_j \bar{p}_j (1 - \bar{p}_j)} \tag{3.25}$$

Where, B is the number of bins, O_j is the observed number of events in bin j, $\hat{p}_j$ is the average predicted probability in bin j.

D-Calibration

D-Calibration [7] extends the concept of 1-calibration over a range of time points or across different distributions of time points, providing a more comprehensive measure of a model's accuracy.

Mean Absolute Error

Mean Absolute Error (MAE) (Uncensored) is the simplest form of MAE variants indicated by [27], it is calculated by taking the non negative value difference between the predicted and actual event times for uncensored subjects only. It does not consider censored data, which may introduce bias when censoring rate is frequent.

$$RMAE(\hat{t}_i, t_i, \delta_i = 1) = |t_i - \hat{t}_i| \tag{3.26}$$

MAE-Hinge variant [27] considers only the cases where the estimated time is sooner than the actual cesonsoring time. It is somewhat optimistic as it assigns zero error to predictions that are later than or equal to the censoring time. Applied when the actual event time is censored $\delta(i = 0)$
"MAE-Margin" [27] uses a "margin time" for induvivual censored units, estimated using a non-parametric method (Kaplan-Meier estimator). This "margin time" is treated as an adjusted event time, creating a more informed guess for censored individuals.

$$\text{Error for censored subjects:} \quad \omega_i[(1 - \delta_i) \cdot \text{emargin}(t_i) + \delta_i \cdot t_i] - \hat{t}_i \tag{3.27}$$

[27] shows that $w_i = 1 - S_{KM}(t_i)$ confidence weight based on the Kaplan-Meier estimator $emargin(t_i) = t_i + \int_{t_i}^{\infty} S_{KM}(D)(t)/S_{KM}(D)(t_i)$ margin time.

"MAE-IPCW-D" ("Inverse Probability Censoring Weight" - Difference) adapts the "IPCW" method to MAE by re-allocating the weights of censored subjects to those with known outcomes, using the estimated time of the event for calculations.
Then, *RMAE-IPCW-T* is calculated similarly to RMAE-IPCW-D, but using eIPCW for censored subjects. MAE-PO (Pseudo-Observation) utilizes pseudo-observations

that estimate the impact of each data point on the overall survival time estimate.

$$e_{pseudo-obs}(t_i, D) = N \cdot \hat{\theta} - (N-1) \cdot \hat{\theta}_{-i} \tag{3.28}$$

θ and $\hat{\theta}$ are unbiased survival time estimators with and without the ii-th subject. The pseudo-observation is treated as an observed event time for MAE calculations.

Logarithmic Score

The Logarithmic Score [37] is used for models that predict the entire probability distribution of the event times. It rewards models that assign higher probabilities to the events that occur. It assesses the logarithm of the predicted probabilities assigned to the true outcome intervals, promoting models that are confident and correct about their predictions.

$$S_{\text{Log}}(\hat{F}, y; \{\zeta_i\}) = -\sum_{i=0}^{B-1} 1(\zeta_i < y \leq \zeta_{i+1}) \log(\hat{F}(\zeta_{i+1}) - \hat{F}(\zeta_i)) \tag{3.29}$$

Cross-Validation

[8] Cross-validation is a resampling technique used to evaluate the generalizability of a statistical model by systematically partitioning the dataset into training and validation subsets. The most granular form, leave-one-out cross-validation (LOOCV), omits one observation at a time from the dataset of size n, fits the model on the remaining $n-1$ observations, and uses the excluded observation to test the model's performance. The process is repeated n times, yielding an average accuracy metric, typically denoted by $\hat{\theta} = \frac{1}{n}\sum_{i=1}^{n} \theta_i$, where θ_i is the performance metric for the i-th iteration. Grouped cross-validation, a more practical alternative, divides the data into k subsets, using $k-1$ subsets for model training and the remaining subset for validation, iterating this process k times. Although cross-validation mitigates some issues of data-splitting, it is computationally expensive, often requiring numerous repetitions (e.g., 200+ iterations) to stabilize accuracy estimates. Moreover, the method may not fully capture the variability of variable selection, as subsets tend to produce similar feature sets. Additionally, cross-validation does not validate the model on the full dataset, which can lead to an underestimation of the model's overall predictive power. To address potential overfitting, a randomization

method can be employed, where the response variable Y is randomly permuted, and the model's predictive accuracy on this randomized data is compared to its performance on the original data, with significant deviations indicating possible overfitting.

3.2 Data

For my research, I utilized SurvSet, an open-source repository specifically designed for time-to-event (T2E) analysis. SurvSet provides a standardized collection of 76 datasets, primarily focused on biomedicine, that are formatted consistently for easy preprocessing and analysis. This makes it an ideal source for benchmarking machine learning (ML) algorithms in survival analysis.

To understand the specifics of each dataset within SurvSet, the repository provides a clear and consistent structure. Each dataset is accompanied by a unique ds_name and metadata that detail its characteristics, such as whether it includes time-dependent covariates, the number of features, and the types of variables (numerical or categorical). The SurvLoader class allows users to easily load any dataset and access this metadata, along with a reference URL that offers additional context and explanations for the columns. This structure ensures that researchers can quickly grasp what each dataset is used for and how it can be applied in survival analysis or machine learning benchmarking.

SurvSet organizes datasets using a SurvLoader class, which simplifies loading and accessing data. Each dataset includes core columns like `pid` (unique ID), `event` (whether the event occurred), and `time` (time to event or censoring). Features are clearly marked as either numerical (num_{}) or categorical (fac_{}), ensuring a smooth integration with various ML algorithms. This consistent structure across all datasets in SurvSet made it my go-to source for accurate and reliable survival analysis in my project.

Furthermore, when selecting the dataset censoring levels, FAIR principles [35], reproducibility by assessing data simulation accuracy similar to the KM-Divergence metric used by [20] are all important considerations. Utilizing this approach voids

the need to carefully assess the ethical implications of using the datasets as these datasets should be under public licensed availability. Lastly, I don't foresee data pre-processing steps, being necessary, as DGM for simulation are commonly categorized as postprocessing [12].

3.3 Methods

3.3.1 Data Pre-Processing and Simulation

Data formatting for simulation models

The preprocessing stage is a critical component of the data pipeline, ensuring that the data is clean, consistent, and ready for the simulation models. The `preprocess` step uses built-in transformations from the scikit-learn library, a foundational component for most of the software used in the study [22]. This is consistent with other studies in the field that rely on scikit-learn for preprocessing of survival data [7]. Specific processing steps tailored to categorical and numerical columns are outlined below:

There are column-specific pipelines for categorical and numerical columns. Categorical features, identified by the prefix `fac_`, are one-hot encoded using the `OneHotEncoder`, while numerical features, identified by the prefix `num_`, are imputed using the median value and then standardized with `StandardScaler`. This multi-step process of encoding and scaling ensures that the data is in a suitable format for the simulation models, similar to methods applied in previous works like [6]. The preprocessing is implemented using `ColumnTransformer`, which allows for the application of different preprocessing steps to different subsets of the columns while also allowing passthrough of untransformed columns [**pedregosa_etal_2011_scikit_learn**].

Simulation

For the simulation, I utilized the Synthcity library [28]. Synthcity is an open-source package designed for generating and evaluating synthetic data across various data modalities, including static data, regular and irregular time series, and censored data. It provides a unified platform for innovative use cases in machine learning

fairness, privacy, and data augmentation. The library was chosen for its comprehensive support of survival data and synthetic data generation, aligning with best practices in survival analysis simulation [20, 32].

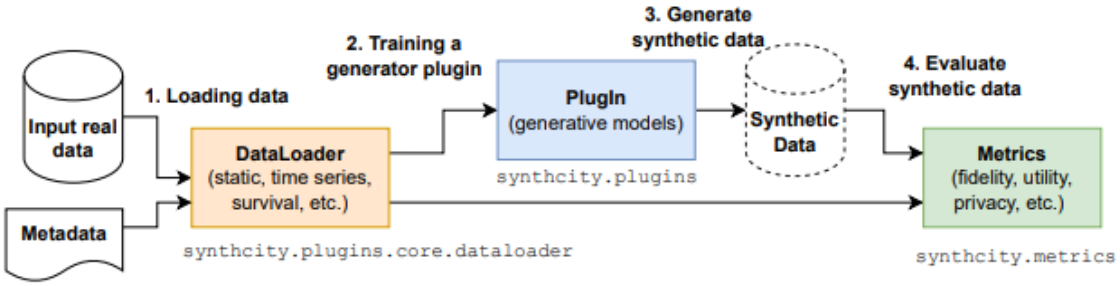


FIGURE 3.2: [28] Simulation Pipeline.

Among the various generative models available in Synthcity, I focused on the use of Survival GAN and Survival VAE. These methods were selected based on previous explorations and their proven ability to handle the complexities of survival data. Survival GAN is particularly effective in generating high-fidelity data with the capability to preserve the underlying distribution [20], while Survival VAE offers a robust framework for capturing the latent structure of the data, making it suitable for the simulation of censored survival datasets [26]. The simulation pipeline is designed to effectively generate synthetic survival data that mirrors real-world datasets, following the structured process provided by Synthcity. This structured approach is supported by [18], which emphasizes the importance of realistic simulation models for validating survival analysis methods.

Data Loading. [28] The first step in the pipeline involves loading the real survival datasets using the `DataLoader` class from Synthcity. The `DataLoader` processes the survival times, censoring indicators, and covariates. This step ensures that the datasets are correctly structured for modeling, as recommended by [14]. Additionally, the `DataLoader` provides validation for censored observations, ensuring that the survival times and event indicators are correctly aligned for downstream modeling [9]. Metadata, such as identifying sensitive features or specifying outcome variables, can also be included for more nuanced data analysis.

Model Training. The `Plugin` class is used to instantiate these models, allowing for straightforward training and application. The training is conducted using the `fit()` method of the `Plugin` class, consistent with practices seen in [10].

Synthetic Data Generation. Once the models are trained, the `generate()` method of the `Plugin` class is used to produce synthetic datasets. This design aligns with the methods proposed in [32] and provides flexibility in generating datasets that reflect various real-world conditions.

Evaluation of Synthetic Data. The final component of the pipeline involves evaluating the synthetic data using the `Metrics` class provided by Synthcity. The design focuses on several key evaluation aspects, as highlighted in [18]:

- Fidelity Metrics: The `Metrics` class is utilized to assess how closely the synthetic data resembles the original data. Metrics such as distributional similarity and survival curve alignment are critical for determining data fidelity [17].
- Statistical Evaluation: The utility of the synthetic data for downstream tasks is tested by applying standard statistical models to the generated data and test performance [27].
- Privacy Considerations: Privacy is a major concern in synthetic data generation. The `Metrics` class includes tools to evaluate the privacy of the synthetic data, such as assessing the risk of re-identification or leakage of sensitive information [35].

Data processing

Stratified Train-Test Split. The process of splitting the data into training and testing sets is crucial for ensuring the validity and reliability of the model evaluation.

- Stratification Based on Event and Time Binning: The function creates a temporary stratification label that combines the event indicator with a discretized version of the time variable. This ensures that the train and test sets maintain similar distributions with respect to the key features, particularly in datasets where time-dependent covariates are present [36].

- Handling Time-Dependent Features: For datasets with time-dependent features, the function includes an additional layer of stratification by incorporating a censoring indicator. This stratification approach ensures that the complex relationships between time, event occurrence, and censoring are preserved across the split [9].
- Robust Splitting Strategy: The function first attempts a stratified split, preserving the distribution of the stratification label across the training and testing sets. If the stratified split is not feasible (e.g., due to insufficient data points in some strata), the function falls back to a random split to ensure the process is completed without errors [22].
- Clean-Up and Return: After performing the split, the temporary stratification labels are removed from the resulting datasets to prevent any unintended impact on subsequent analyses. The function returns the training and testing datasets, ready for further processing or model training [22].

3.3.2 Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) is a crucial first step in understanding the structure of survival data, identifying key features, and detecting potential issues like outliers or multicollinearity, which may affect model performance. In alignment with best practices outlined by [8], the goal of EDA in this context is to ensure that the subsequent modeling steps are based on a thorough understanding of the data and to preemptively address any concerns that could distort model validity or prediction accuracy.

Phenotyping and Cluster Analysis

Numerical features within the dataset were selected for clustering, with k-means used for unsupervised learning, and PCA applied for dimensionality reduction. The rationale for this choice is based on its widespread use in survival studies for reducing dimensionality while retaining variance, thus preventing overfitting [24]. By clustering the data, I aimed to identify phenotypes or natural groupings that could reveal inherent survival trends. The `ClusteringPhenotyper` from the

`Auton_Survival` library was employed due to its established robustness in handling high-dimensional survival datasets [19].

Cluster Survival Analysis Plots

Once clusters were identified, Kaplan-Meier and Nelson-Aalen estimators were used to assess survival probability and cumulative hazard within each cluster. These non-parametric estimators are foundational in survival analysis due to their capacity to deal with censored data, allowing for more accurate survival function estimates without assumptions about the underlying hazard function [10]. The resulting plots provide clear visual comparisons of survival outcomes across clusters, enabling the identification of patterns that might otherwise be overlooked in raw data. Visualizing survival by cluster offers valuable insights into how specific groupings influence event rates, further validating our clustering approach.

Target Variable Analysis (Survival Time and Event Status)

The survival time (`time`) and event status (`event`) are critical to survival analysis, and thus their distributions must be explored in detail. Summary statistics provide insights into central tendencies and spread, while histograms and boxplots visualize the distribution and any potential skewness or outliers. As suggested by [18], understanding these basic characteristics is essential for making informed decisions about the appropriate survival models and transformations.

Univariate Analysis of Covariates

Univariate analysis was conducted for each covariate to evaluate their distributions and basic summary statistics, such as mean and standard deviation. These steps are necessary to identify whether any features require transformation or imputation, especially when outliers or skewed distributions are present. This aligns with standard EDA practice in survival analysis [1], ensuring that covariates contribute positively to the model's predictive power without introducing bias.

Bivariate Analysis

Bivariate analysis evaluates the relationship between covariates and survival time, enabling the detection of any strong linear or non-linear associations that might warrant further exploration or transformation. For instance, scatter plots were used to observe trends between survival time and numerical covariates. This method was chosen based on [8], where understanding bivariate relationships can guide model selection and reveal interactions that would affect the survival function.

Censoring Analysis

Censoring is a common issue in survival datasets, and understanding its distribution is critical for accurate modeling. I followed best practices outlined by [14] to quantify the proportion of censored observations and to ensure they are evenly distributed across covariates. Any imbalances could distort model estimates, so visualization of censoring distribution was deemed essential.

Correlation Analysis

A correlation matrix was calculated to assess the degree of association between numerical covariates. Visualizing the matrix as a heatmap is a well-established technique for detecting multicollinearity, which could otherwise inflate the variance of coefficient estimates in a regression model. The Pearson correlation coefficient was calculated using the formula:

$$r_{xy} = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \sum(y_i - \bar{y})^2}}$$

where x_i and y_i are individual data points, and $\bar{x}$ and $\bar{y}$ are the means of the variables x and y, respectively [8]. This process aids in identifying collinearity, allowing us to manage covariates that might distort the model through overfitting or redundancy.

3.3.3 Survival Analysis: Model Application and Evaluation

This section outlines the steps for running the models on the preprocessed data. The chosen libraries and methods, as shown in Table 3.1, are widely recognized for survival analysis and machine learning, ensuring both robustness and reproducibility in this study.

Library/Method	Description
Auton-survival [19]	Provides tools for survival analysis including implementations of advanced machine learning models like DeepSurv and Cox-Time.
scikit-survival [25]	Extends scikit-learn [22] to handle survival analysis, enabling use of Cox regression models with extensions such as Lasso.
lifelines [3]	Popular library for survival analysis that includes Kaplan-Meier, Nelson-Aalen, and Cox models, among others.

TABLE 3.1: Libraries to be used during research

Cox Proportional Hazards (CoxPH) Model

Model Fitting: The Cox Proportional Hazards (CoxPH) model is a widely used regression method in survival analysis due to its semi-parametric nature, which allows flexibility in handling the relationship between covariates and survival time without assuming a particular baseline hazard. In this study, the `lifelines` library [3] was used to fit the CoxPH model, which helps identify covariates that significantly affect the hazard function. This aligns with the established methodology in survival analysis research [2].

Lasso Regularization: To address overfitting and improve the interpretability of the CoxPH model, Lasso regularization was be applied during model fitting. This method adds an L1 penalty to the loss function, shrinking the coefficients of less important features to zero, effectively performing feature selection [33]. Lasso has been shown to be effective in high-dimensional survival data, allowing the model to focus on the most influential covariates while ignoring noise [38].

Effect of Lasso on Model Performance: The application of Lasso regularization is expected to enhance model generalizability by reducing complexity. To evaluate the impact of Lasso, performance metrics such as the concordance index (C-index) was computed before and after regularization. This approach ensures that we can assess the predictive power and identify the most critical covariates for survival outcomes, contributing to a more interpretable model [6].

Survival Curve: Post-fitting, survival functions was predicted for different covariate profiles using `lifelines.plot_covariate_groups()`. Survival curves allow visual interpretation of how various covariates influence survival probabilities over time. Plotting these curves for different subgroups is essential for understanding the model's predictions and how different factors impact survival [8].

Assumption Checking: The proportional hazards assumption, a key condition of the CoxPH model, was validated using Schoenfeld residuals. Any detected violations of this assumption necessitate adjustments, such as using stratified Cox models, to ensure the validity of the model's predictions [36]. Checking model assumptions is crucial for avoiding incorrect conclusions in survival analysis.

Random Survival Forest (RSF) Model

Model Fitting: The Random Survival Forest (RSF) model was fitted using the `scikit-survival` library [25], which is tailored for survival data with tree-based methods. RSF offers a non-parametric alternative to the CoxPH model and is particularly useful for datasets with non-linear relationships between covariates and survival outcomes. Tuning hyperparameters, such as the number of trees and maximum depth, was critical for optimizing the model's performance, as suggested by [10].

Survival Curve Prediction: After the RSF model is fitted, survival curves was be generated for each observation. These curves provide a visual understanding of survival probabilities over time for different groups. Comparing RSF-predicted curves to those from CoxPH offer insights into how the models differ, especially in datasets with non-proportional hazards [16].

Variable Importance: The RSF model provides variable importance scores, which help in identifying the most influential covariates for predicting survival outcomes. By analyzing these scores, we can interpret the model's decision-making process and understand which features are driving the predictions. This step is crucial for gaining actionable insights from the RSF model [10].

Model Comparison

The `SurvivalEval` package [26] was utilized for a comprehensive comparison between the RSF and CoxPH models. The package provides metrics for evaluating survival models, including the concordance index, Brier score, and calibration plots. The comparison focus on differences in survival predictions across models, particularly in handling censoring and non-proportional hazards. Understanding these differences is key to selecting the most appropriate model for future survival predictions and accurately interpreting survival probabilities in various contexts [24].

3.4 Limitations

A massive limitation is that the research is tightly coupled, meaning the phases are strictly dependent on each other. This is an antipattern, [13], which should be planned to accommodate any failures during any of the research phases. In cases where results do not converge, or the interpretation is wrong, the tightly coupled nature of the research was also affect the preceding stages. Lastly the computation time, within the modern setting should not be a hindrance but the combination of multiple computational components like the DGM and the model execution and results evaluation, is important to take caution.

3.5 Ethical Considerations

Ethical clearance would not be a component of this study, as the only real data that would be used, was only be selected from open source, or publicly available (public licensing) sources and then further fed into a simulation model which changes the

nature of the data in its entirty and effectivly anonimizes the data. Due to the nature of the topic.

Chapter 4

Results And Discussion

All the Library code used and experiments ran inline with producing the outputs can be found at the github: https://github.com/wjvandermerwe/ResCap/tree/main/research_report/project.

4.1 Simulation: Outputs and Results

In preperation for running the models, as discussed in THE 3, I ran the simulation models for creating synthetic data. This allows two different scenarios, for running a spesific model and predicting outcoms, namely the outputs produced by Survival GAN, the outputs produced by Survival VAE.

To feed the simulation pipeline methods, I use a real dataset to illustrate the expected outputs. The 'flchain' dataset [4] from the SurvSet repository [5] which analyzed data from 15,859 individuals aged 50 years or older, excluding those with known plasma cell disorders, to assess whether the free light chain (FLC) assay could predict overall survival in the general population. The Simualtion methods produced two dataset with 5000 records each. The Columns for the dataset consisted of:

TABLE 4.1: Description of the `flchain` Dataset Variables

Variable	Type	Description
`age`	Numeric	Age of the subject in years.
`sex`	Categorical	Gender of the subject; F = female, M = male.
`sample.yr`	Numeric	The calendar year in which a blood sample was obtained.
`kappa`	Numeric	Serum free light chain, kappa portion (mg/L).
`lambda`	Numeric	Serum free light chain, lambda portion (mg/L).
`flc.grp`	Categorical	FLC group for the subject, as used in the original analysis.
`creatinine`	Numeric	Serum creatinine level (mg/dL).
`mgus`	Binary	Indicator for Monoclonal Gammopathy of Undetermined Significance (MGUS); 1 if diagnosed, 0 otherwise.
`futime`	Numeric	Days from enrollment until death or last contact.
`death`	Binary	Event indicator; 0 = alive at last contact, 1 = dead.
`chapter`	Categorical	Grouping of the primary cause of death by chapter headings of the ICD-9.

4.1.1 Metrics for output data

As discussed the SynthCity [28] library has metrics to evaluate synthetic data, the metrics used and their outputs are summarized in the table below.

Metric	Metric Description	Survival GAN	Survival VAE
Close Values	Measures how closely the generated values match the real data distribution. Higher values indicate better similarity.	0.8738	0.8454
Data Mismatch	Quantifies the mismatch between the generated and real datasets. A value of 0 indicates no mismatch.	0.0	0.0
Proportion	Assesses the proportion of generated data that matches the real data. A value of 0 indicates perfect alignment.	0.0	0.0
NN Distance	Nearest neighbor distance between real and synthetic samples. Lower values indicate greater similarity.	0.1061	0.1164
Distant Values	Measures the proportion of outlier values in the generated data. Lower values are preferred.	0.0005	0.0010
PRDC Score - Precision	Precision of the synthetic data when comparing the real and synthetic distributions. Higher values are better.	0.9968	0.9944
PRDC Score - Recall	Recall score, indicating how much real data is captured by the synthetic model. Higher values show better coverage.	0.9945	0.9933
PRDC Score - Density	Density metric to assess how well the synthetic data replicates the real data density.	0.9782	0.9832
PRDC Score - Coverage	Measures the coverage of the real data by the synthetic data. Higher values indicate better generalization.	0.7366	0.7362
invKL Score - Marginal	Inverse KL divergence between the marginal distributions of real and synthetic data. Lower values are better.	0.9651	0.9601
Chi Score - Marginal	Chi-square statistic comparing the marginal distributions. Lower values indicate better alignment.	0.8555	0.6883
Iden Score	Identification score reflecting the model's ability to distinguish real from synthetic data. Higher values are better.	0.3141	0.3128
Iden Score OC	Outlier-corrected identification score, with higher values reflecting better alignment between real and synthetic data.	0.3050	0.3026
Kanon Score - GT	K-anonymity score for ground truth data, indicating the degree of privacy maintained in real data.	160	160
Kanon Score - SYN	K-anonymity score for synthetic data, assessing privacy in the generated data.	149.00000001	158.00000001
Ldiv Score - GT	L-diversity score for real data, reflecting how well privacy is maintained in sensitive data attributes.	160	160
Ldiv Score - SYN	L-diversity score for synthetic data, evaluating privacy-preserving properties in the synthetic dataset.	149.00000001	158.00000001

TABLE 4.2: Metric Descriptions used from [28]

The models preform quite similar, however in the practical the Survival VAE training is alot more robust, in terms of training time and overall completeness, accross different datasets of the SurvSet Repo, and thus the **results from the Survival VAE is used in the simulation Study.**

4.2 Exploratory Analysis

For all the exploratory analysis plots I do not show the categorical plots, as it wont be as valueable insights as the numerical. The one hot encoded structure will show alot of the same variable during plots which results in two value / bar plots.
This figure illustrates the correlation matrix, which depicts the pairwise correlation coefficients between variables in the dataset. The color intensity in each cell represents the strength of the correlation, with darker colors indicating stronger relationships. This matrix helps to identify multicollinearity among variables, which is crucial for understanding the dependencies within the dataset and for selecting features for further analysis.

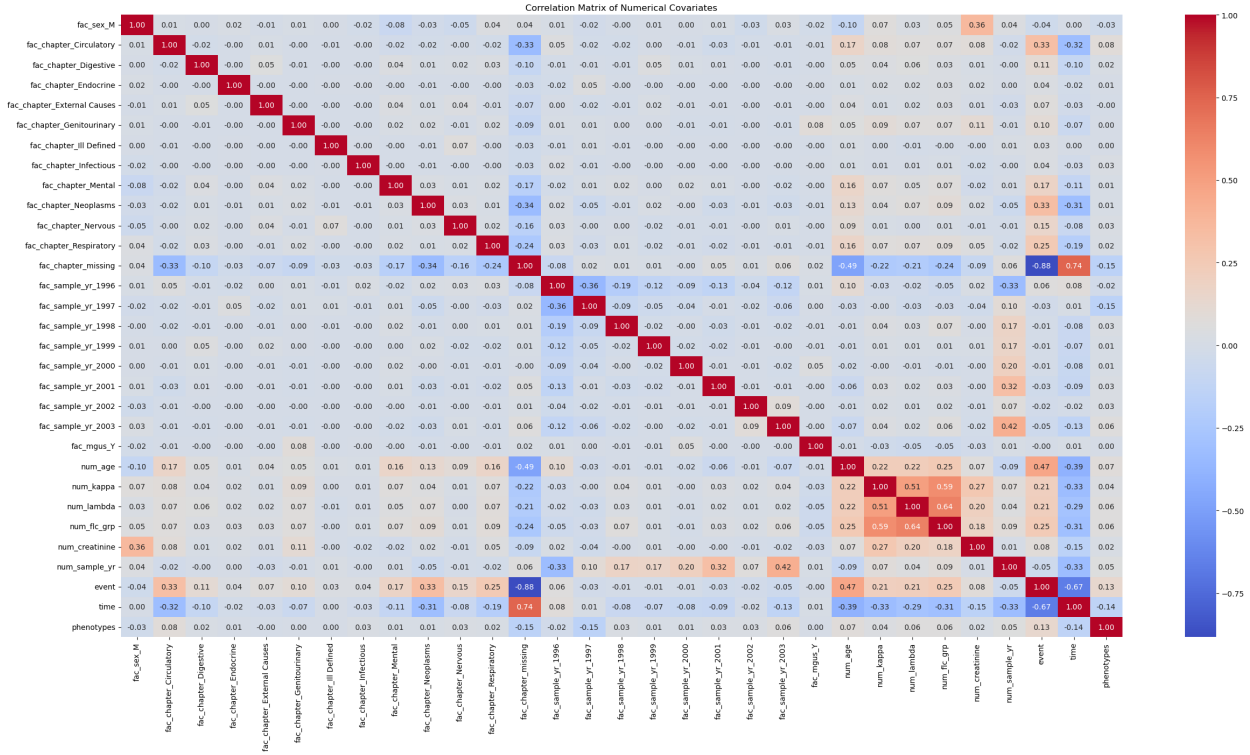


FIGURE 4.1: Correlation Matrix For Variables

The univariate analysis plots presented here offer a comprehensive view of the distribution of individual variables. While four one-hot encoded categorical features are shown for illustration of the non informative nature of these plots, the general tendency observed in the numerical variables is a roughly normal distribution centered around specific values. This suggests that the data is symmetrically distributed with most observations clustered around the mean, with some variation present across different numerical features.

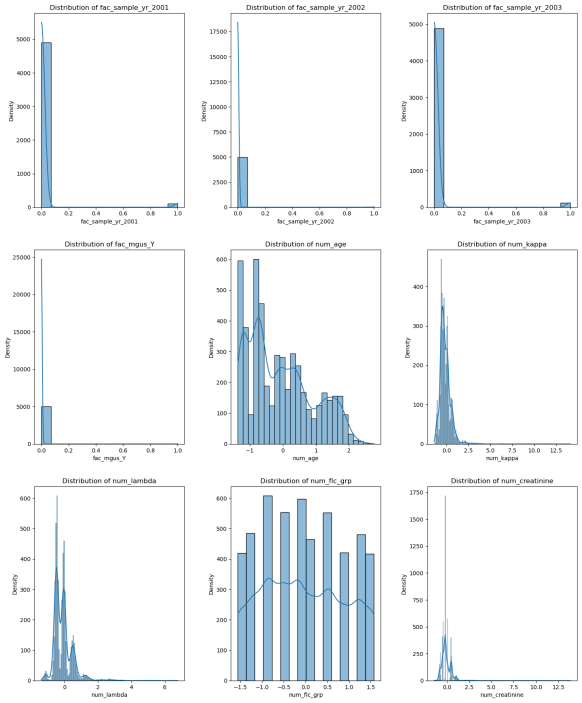


FIGURE 4.2: Univariate Analysis Plots

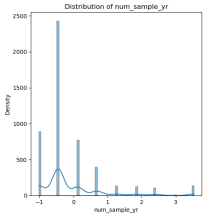


FIGURE 4.3: Univariate Analysis Plots

These plots help visualize interactions, allowing us to explore possible correlations, trends, and dependencies. While most variables appear to follow a normal distribution, the age variable shows a noticeable skew, which is expected given that age naturally increases over time and influences other factors in the dataset. Understanding these patterns is crucial for developing robust predictive models and gaining deeper insights into the data.

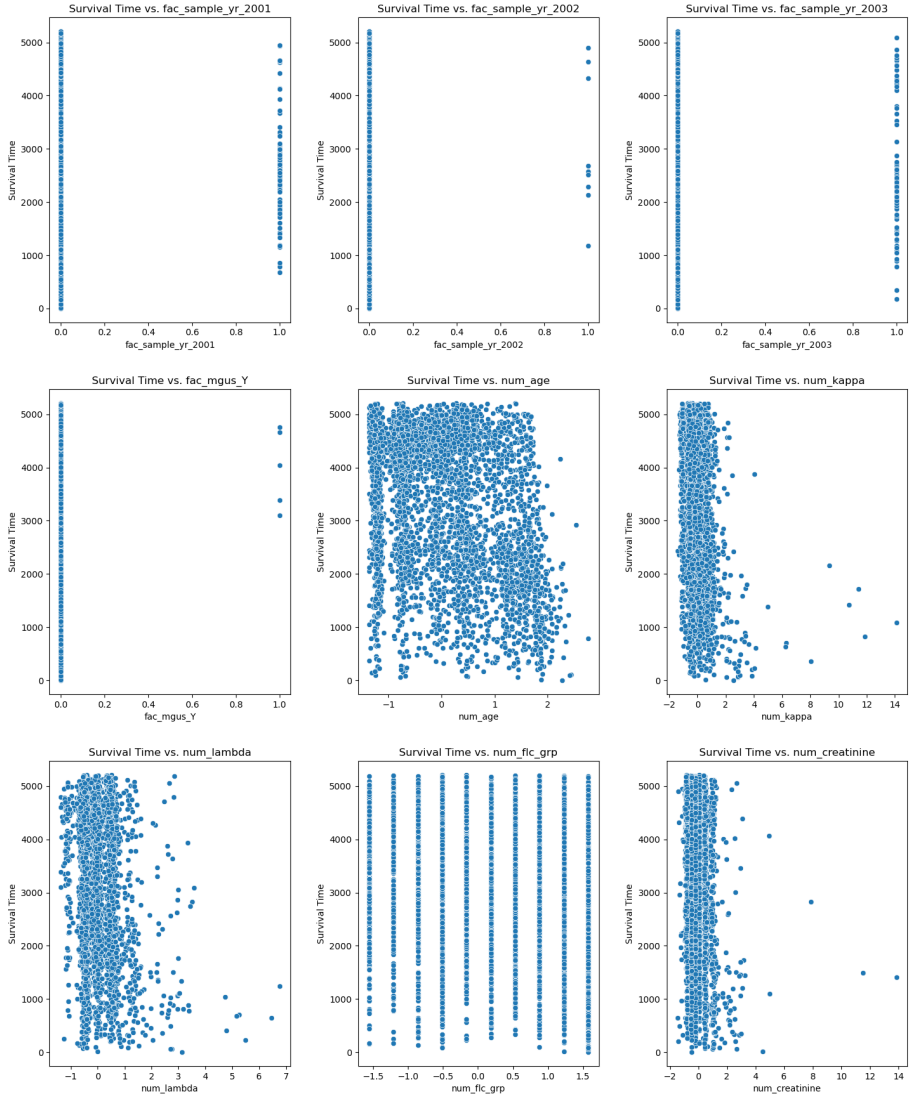


FIGURE 4.4: Bivariate Scatter Plots

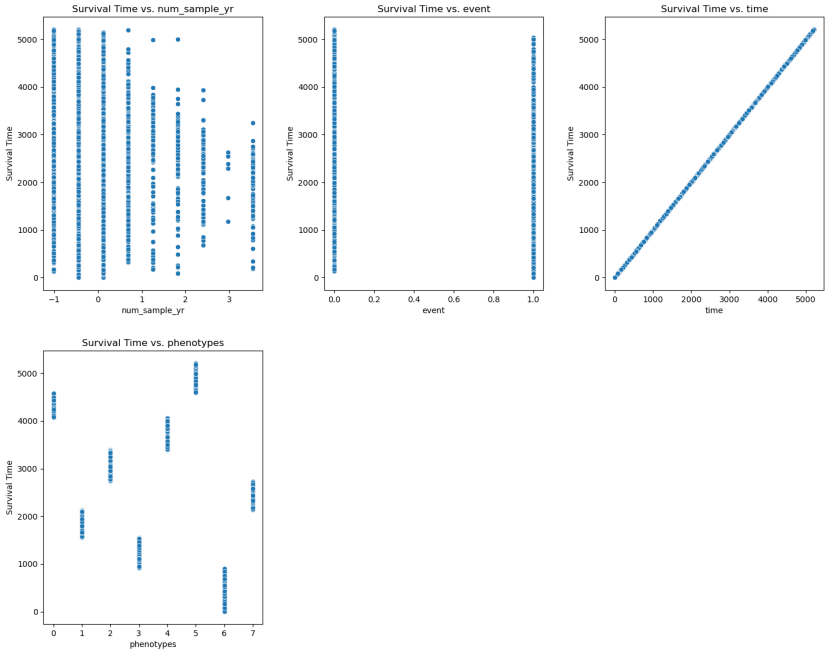


FIGURE 4.5: Bivariate Scatter Plots

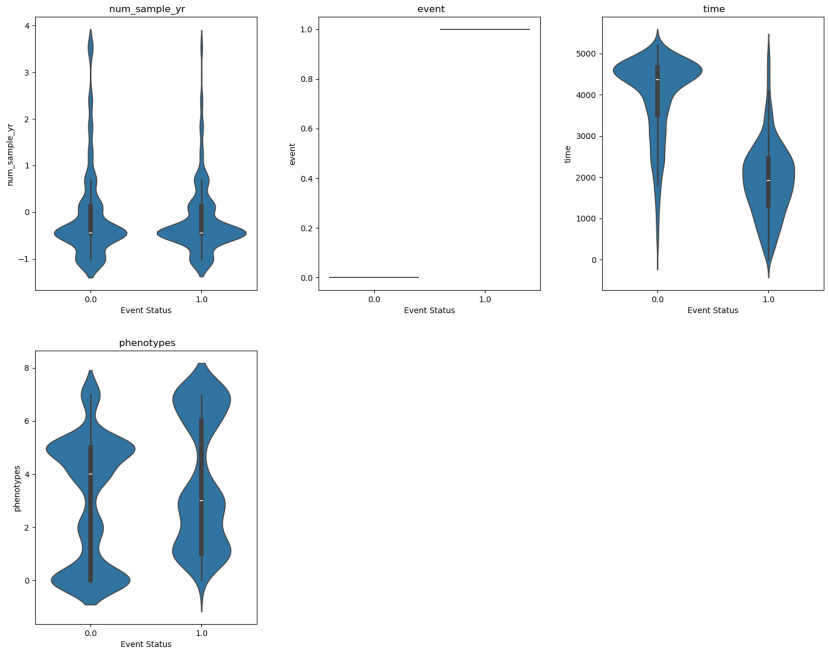


FIGURE 4.6: Bivariate Violin Plots

The violin plots effectively highlight the detailed distribution of value ranges for each event type, revealing clear groupings across various features. A notable example is the num_flc_grp feature, where we observe a distinct pattern: positive values are predominantly associated with the occurrence of the event, while negative values are more common among patients who are still alive. This distinction provides valuable insight into how certain features influence the likelihood of survival or the occurrence of the event.

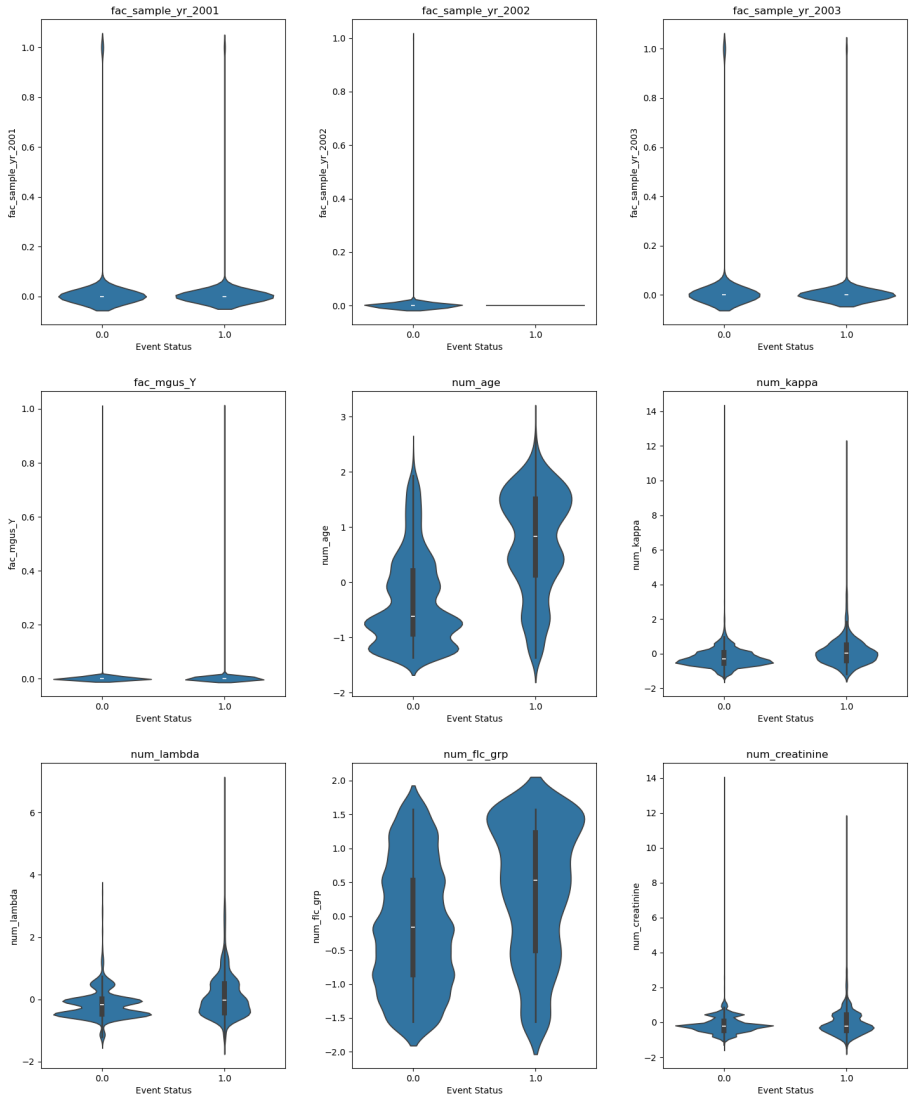


FIGURE 4.7: Bivariate Violin Plots

The per-variable censoring analysis, examines the distribution of censored observations across different variables. Censoring is a common occurrence in survival analysis, and this plot helps in understanding how censoring is distributed within the dataset. We can see the analysis will be preformed on a 75% overall censoring distribution.

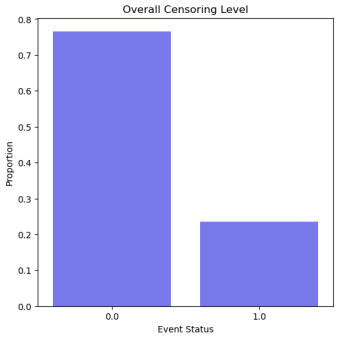


FIGURE 4.8: Per Variable Censoring

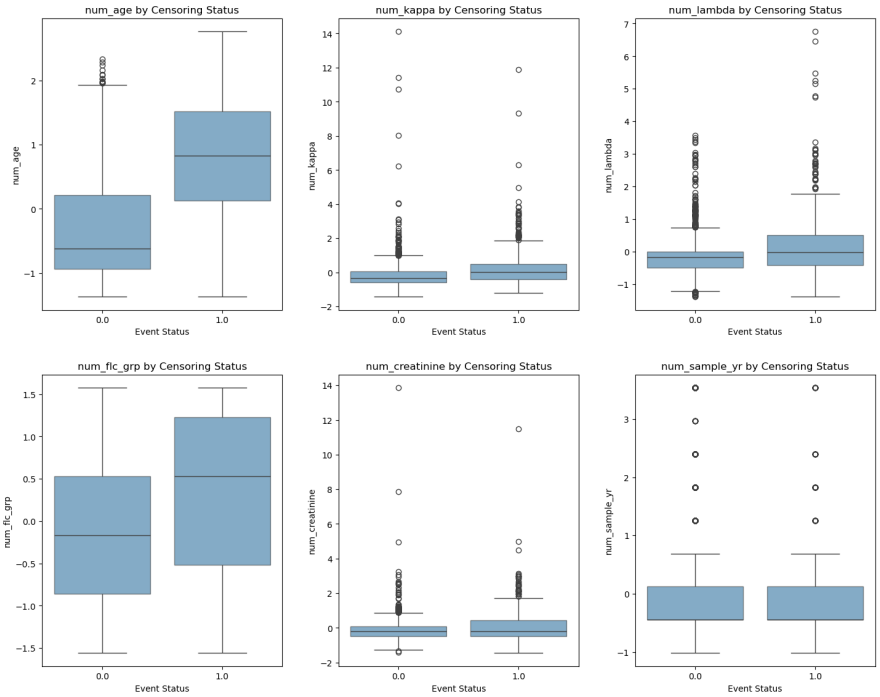


FIGURE 4.9: Numerical Censoring

In this figure, I present the results of the Auton-Survival phenotyping analysis, along with the compositions of the identified clusters. The phenotyping process involves grouping similar observations into clusters based on survival-related features. The figure highlights the characteristics of each cluster, offering insights into the heterogeneity within the population and helping to identify distinct survival patterns.

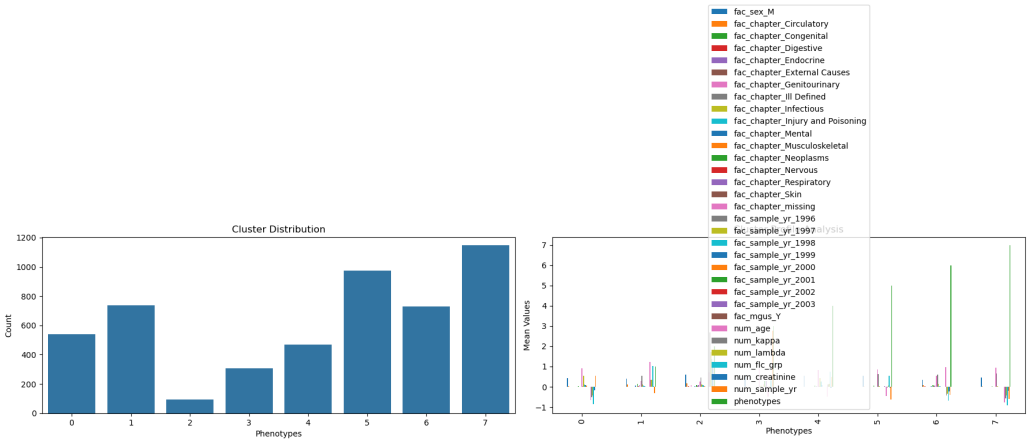


FIGURE 4.10: Auton-Survival Phentotying Along with the cluster compositions

The Kaplan-Meier plot shown in this figure provides a survival analysis of the clusters identified in the phenotyping process. Each curve represents the survival probability over time for a specific cluster. This plot is useful for comparing the survival experiences of different groups within the dataset, revealing significant differences or similarities in survival rates across clusters.

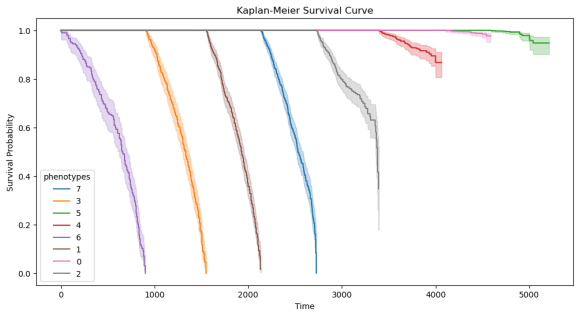


FIGURE 4.11: Kaplan-Meier Plot of the clusters

This figure illustrates the Neelson-Aalen cumulative hazard plot for the clusters identified in the phenotyping analysis. The plot displays the cumulative hazard function for each cluster, offering a different perspective on survival analysis compared to the Kaplan-Meier plot. It helps in understanding the risk accumulation over time and comparing the hazard rates across different clusters.

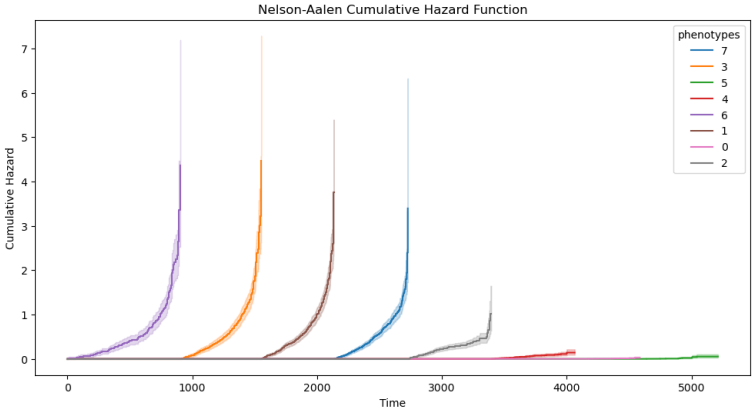


FIGURE 4.12: Neelson-Aalen plot of the clusters

4.3 Survival Analysis Case Study

For the Case Study I present the results of running the models for just the Survival Variational Autoencoder generated data, I do this because the training of this model was faster by a factor of 10 times enabling generation after going back and changing data in the preporcessing stage as Issues arised during runtime of the models.

4.3.1 Cox Proportional Hazards

In the initial phase of the analysis, a Cox proportional hazards model was applied to the dataset. However, the model struggled with convergence, which was indicated by warning messages and issues related to matrix inversion and collinearity. These problems were addressed by following suggestions from the Lifeline documentation I talk about this in 3.3. Specifically, I performed data thinning by dropping columns with low variance and potential complete separation. Additionally, I conducted a variance inflation factor (VIF) analysis to identify and remove highly collinear variables. These steps were crucial in stabilizing the model and resolving convergence issues.

After applying the convergence checks and reprocessing the data, the Cox model successfully fit the dataset. The same preprocessed data was then reused for a Random Survival Forest (RSF) model to ensure consistency across the methods. Both models were evaluated on the test set, providing survival predictions, median survival estimates, and partial hazard calculations. The preprocessing steps proved effective in enabling the Cox model to converge and laid a solid foundation for comparison with the RSF model.

	coef	exp(coef)	se(coef)	coef lower 95%	coef upper 95%	exp(coef) lower 95%	exp(coef) upper 95%	cmp to	z	p	-log2(p)
fac_sex_M	-0.04	0.96	0.06	-0.16	0.08	0.85	1.08	0.00	-0.67	0.51	0.98
fac_chapter_Circulatory	0.85	2.34	0.10	0.66	1.04	1.94	2.82	0.00	8.90	<0.005	60.58
fac_chapter_Digestive	1.07	2.92	0.23	0.63	1.52	1.87	4.56	0.00	4.73	<0.005	18.76
fac_chapter_External Causes	-0.10	0.90	0.34	-0.77	0.57	0.46	1.77	0.00	-0.29	0.77	0.38
fac_chapter_Genitourinary	-0.07	0.93	0.24	-0.54	0.40	0.58	1.49	0.00	-0.30	0.76	0.39
fac_chapter_Ill Defined	-0.23	0.79	0.67	-1.55	1.09	0.21	2.96	0.00	-0.35	0.73	0.46
fac_chapter_Mental	-0.23	0.79	0.16	-0.55	0.08	0.58	1.09	0.00	-1.45	0.15	2.77
fac_chapter_Neoplasms	0.93	2.53	0.09	0.75	1.10	2.12	3.01	0.00	10.33	<0.005	80.67
fac_chapter_Nervous	-0.11	0.89	0.18	-0.46	0.23	0.63	1.26	0.00	-0.64	0.52	0.94
fac_chapter_Respiratory	0.62	1.87	0.12	0.39	0.86	1.47	2.36	0.00	5.20	<0.005	22.24
fac_chapter_missing	-2.63	0.07	0.08	-2.78	-2.47	0.06	0.08	0.00	-32.78	<0.005	780.30
fac_sample_yr_1996	-0.02	0.98	0.06	-0.15	0.10	0.86	1.11	0.00	-0.38	0.70	0.51
fac_sample_yr_1997	0.07	1.07	0.09	-0.11	0.24	0.90	1.28	0.00	0.77	0.44	1.17
fac_sample_yr_1998	0.04	1.04	0.13	-0.22	0.30	0.80	1.36	0.00	0.31	0.75	0.41
fac_sample_yr_1999	0.38	1.47	0.19	0.01	0.76	1.01	2.13	0.00	2.02	0.04	4.51
fac_sample_yr_2000	0.17	1.19	0.28	-0.37	0.71	0.69	2.04	0.00	0.62	0.54	0.90
fac_sample_yr_2001	0.27	1.31	0.26	-0.23	0.77	0.79	2.17	0.00	1.05	0.30	1.76
fac_sample_yr_2003	0.15	1.16	0.25	-0.34	0.63	0.71	1.89	0.00	0.60	0.55	0.87
num_age	0.25	1.28	0.04	0.17	0.32	1.19	1.38	0.00	6.54	<0.005	33.92
num_kappa	0.18	1.20	0.04	0.10	0.26	1.10	1.30	0.00	4.21	<0.005	15.24
num_lambda	0.23	1.25	0.05	0.12	0.33	1.13	1.39	0.00	4.23	<0.005	15.41
num_flc_grp	0.07	1.08	0.04	0.00	0.14	1.00	1.15	0.00	2.09	0.04	4.77
num_creatinine	0.16	1.17	0.05	0.07	0.25	1.07	1.29	0.00	3.31	<0.005	10.08
num_sample_yr	0.16	1.17	0.04	0.08	0.24	1.09	1.27	0.00	4.04	<0.005	14.18

FIGURE 4.13: coefficient values

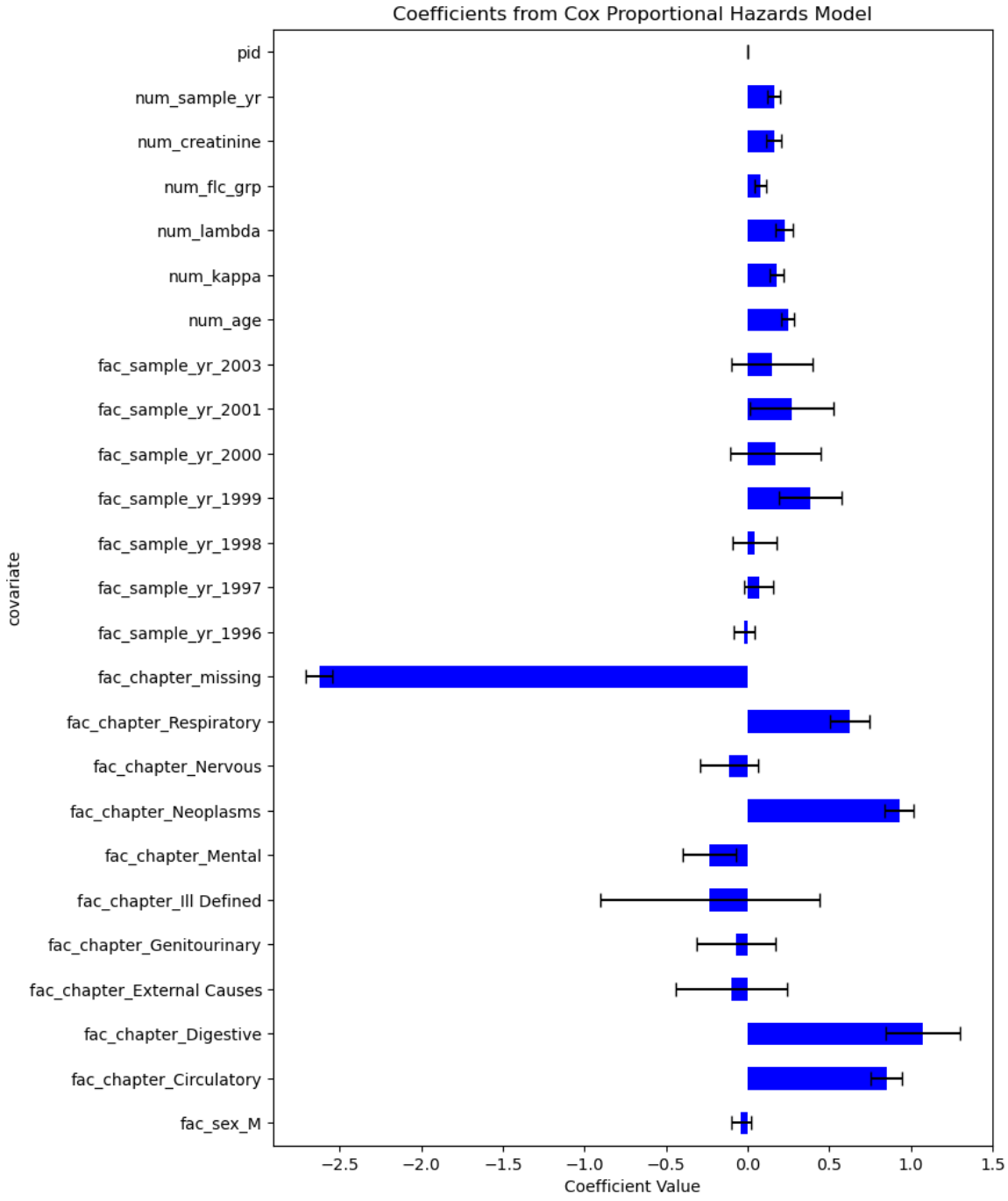


FIGURE 4.14: Boxplots of coefficients

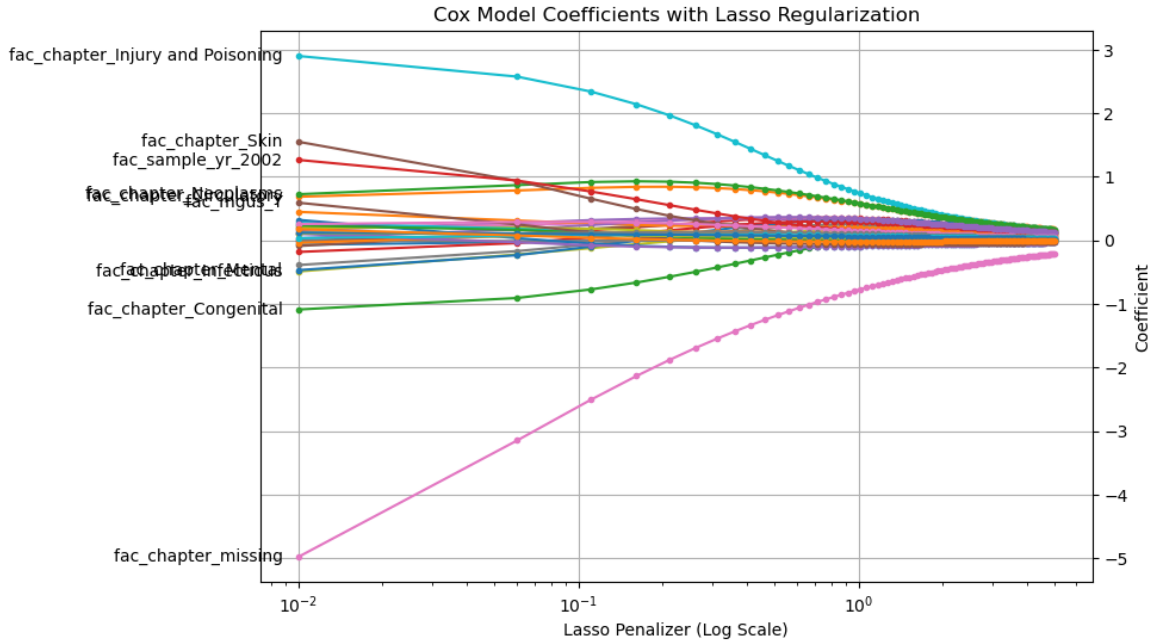


FIGURE 4.15: Lasso regularized coefficient panning closer to zero

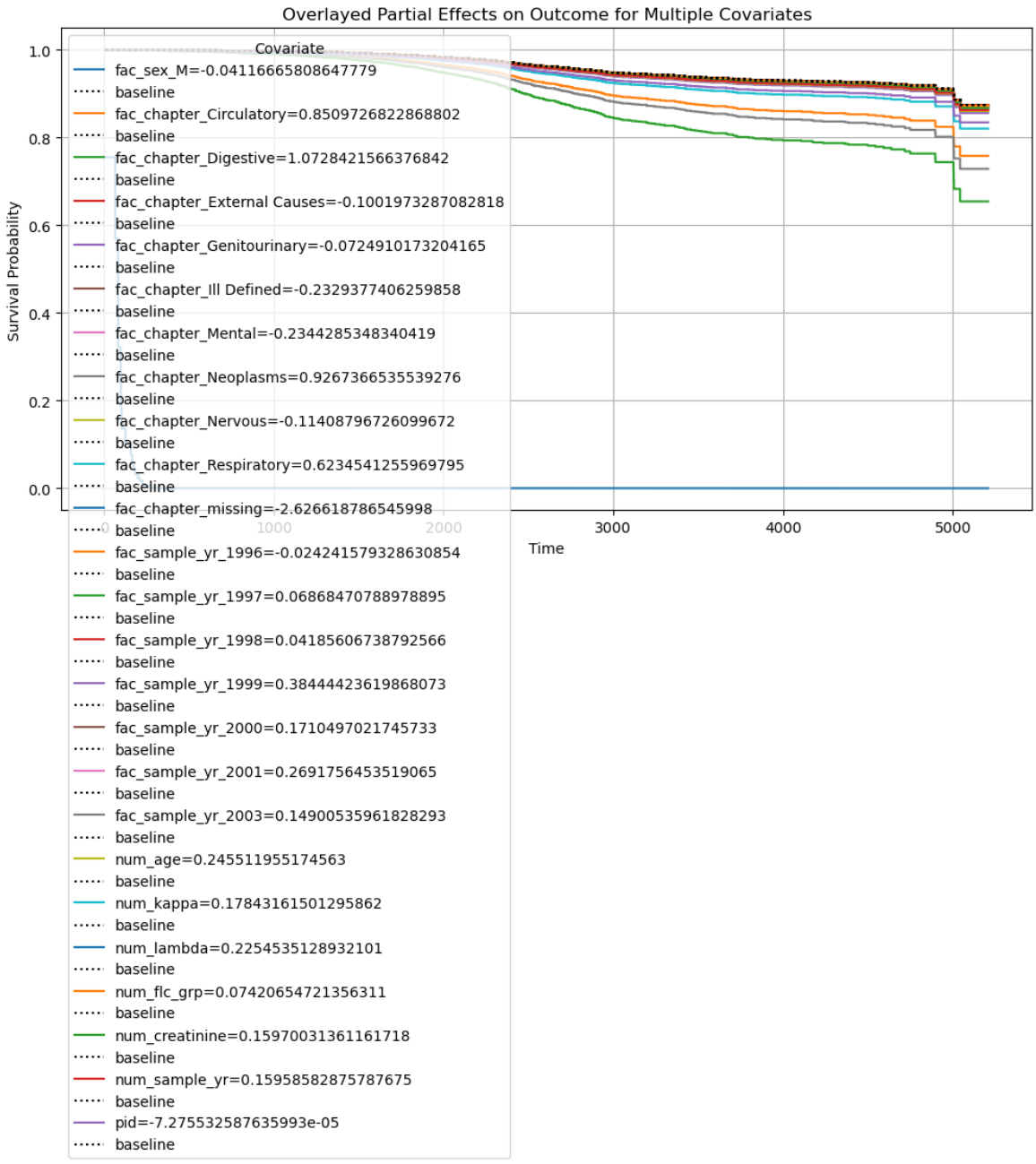


FIGURE 4.16: Survival curves for covariates

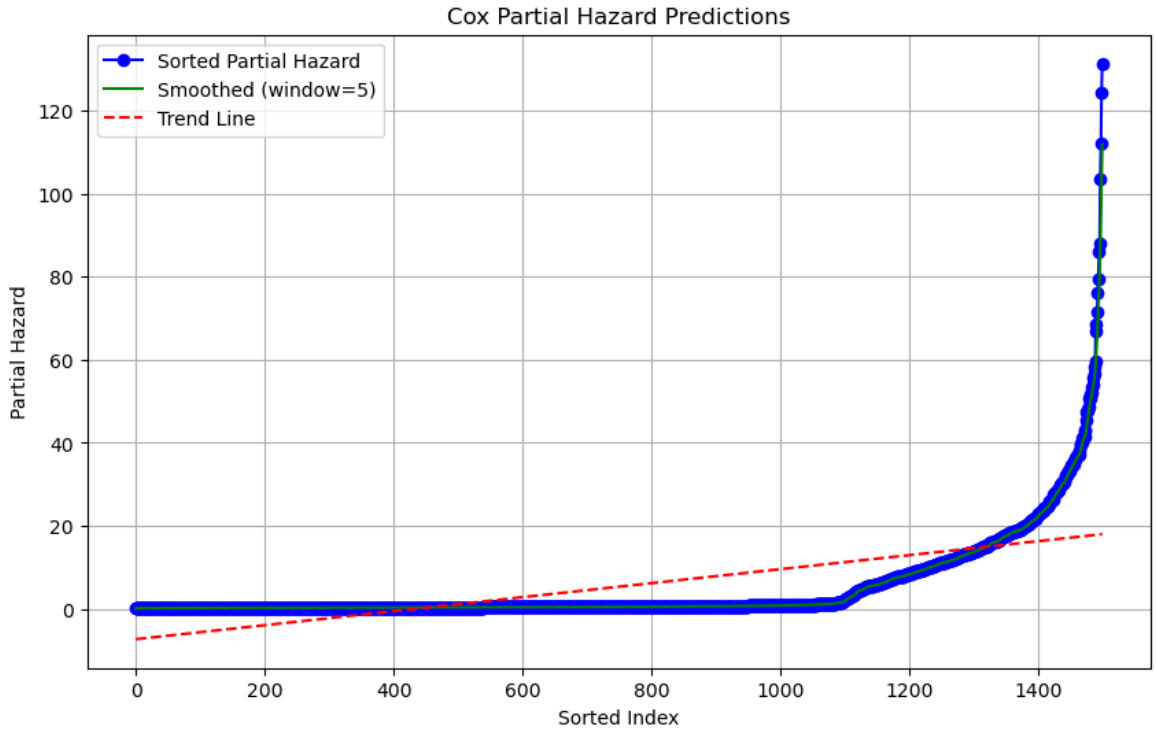


FIGURE 4.17: Mean Hazard Visualisation

Assumption Check

TABLE 4.3: Test Statistics, p-values, and -log2(p) for Different Variables

Variable	Test Type	Test Statistic	p-value	-log2(p)
fac_chapter_Circulatory	km	16.29	<0.005	14.16
	rank	17.65	<0.005	15.20
fac_chapter_Digestive	km	0.97	0.32	1.63
	rank	0.95	0.33	1.60
fac_chapter_External Causes	km	2.86	0.09	3.46
	rank	2.24	0.13	2.90
fac_chapter_Genitourinary	km	0.03	0.87	0.20
	rank	0.03	0.86	0.22
fac_chapter_Ill Defined	km	0.00	0.95	0.07
	rank	0.00	0.95	0.07
fac_chapter_Mental	km	0.41	0.52	0.93
	rank	0.32	0.57	0.80
fac_chapter_Neoplasms	km	5.65	0.02	5.84
	rank	5.82	0.02	5.98
fac_chapter_Nervous	km	1.00	0.32	1.65
	rank	0.90	0.34	1.54
fac_chapter_Respiratory	km	0.72	0.40	1.34
	rank	0.90	0.34	1.54
fac_chapter_missing	km	57.74	<0.005	44.93
	rank	58.70	<0.005	45.63
fac_sample_yr_1996	km	1.04	0.31	1.70
	rank	1.00	0.32	1.65
fac_sample_yr_1997	km	0.34	0.56	0.84
	rank	0.37	0.54	0.88
fac_sample_yr_1998	km	0.04	0.85	0.23
	rank	0.01	0.91	0.13
fac_sample_yr_1999	km	0.89	0.34	1.54
	rank	0.96	0.33	1.61
fac_sample_yr_2000	km	0.49	0.48	1.05
	rank	0.58	0.45	1.16
fac_sample_yr_2001	km	0.11	0.73	0.44
	rank	0.12	0.73	0.45
fac_sample_yr_2003	km	0.23	0.63	0.67
	rank	0.25	0.62	0.70
fac_sex_M	km	0.45	0.50	0.99
	rank	0.36	0.55	0.86
num_age	km	1.34	0.25	2.01
	rank	1.32	0.25	2.00
num_creatinine	km	5.45	0.02	5.68
	rank	5.59	0.02	5.79
num_flc_grp	km	0.06	0.80	0.32
	rank	0.06	0.81	0.31
num_kappa	km	0.13	0.72	0.47
	rank	0.12	0.73	0.46
num_lambda	km	0.13	0.72	0.48
	rank	0.08	0.77	0.37
num_sample_yr	km	1.29	0.26	1.96
	rank	1.39	0.24	2.07

The variable `num_creatinine` failed the non-proportional hazards test, as indicated by the p-value of 0.0181, is likely due to a violation of the proportional hazards assumption. This assumption requires that the effect of the covariate on the hazard function remains constant over time [14]. When a variable fails this test, it suggests

that its relationship with the hazard may change over time, which could be due to several factors:

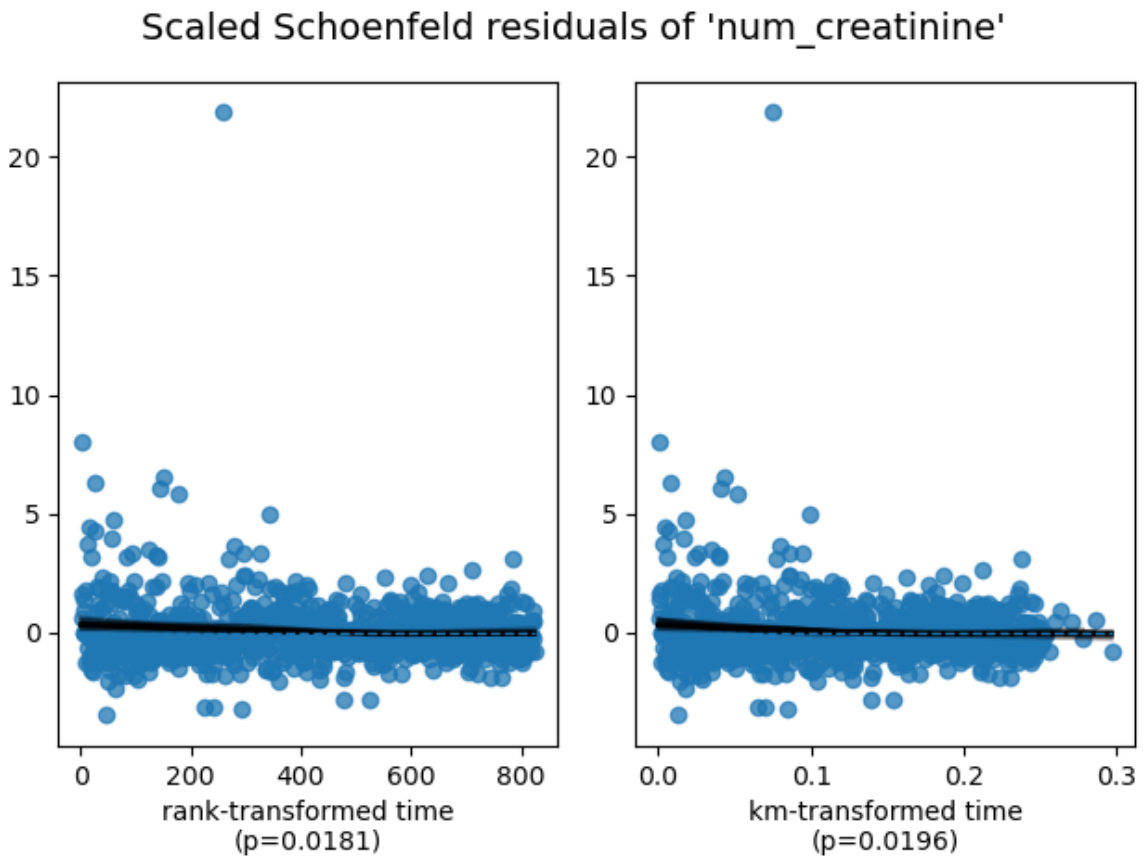


FIGURE 4.18: Schoenfeld Residuals for num_creatinine

- **Incorrect Functional Form:** The relationship between `num_creatinine` and the outcome might not be linear, and missing non-linear terms could be causing the violation [8]. The proportional hazards test is highly sensitive to such misspecifications.
- **Non-Linearity:** The variable `num_creatinine` may have different effects at different levels, which could be addressed by transforming the variable or using binning (e.g., with `pd.cut`) to categorize it. This approach helps account for non-proportional effects by stratifying the data.

- **Time-Varying Effects:** The effect of `num_creatinine` on the hazard may change over time, suggesting that adding an interaction term with time could better capture this dynamic relationship.

`num_creatinine` may not meet the proportional hazards assumption due to non-linearity or time-varying effects, but can be addressed by modifying the functional form, using stratification, or introducing interaction terms with time.

4.3.2 Random Survival Forest

In the initial Random Survival Forest (RSF) model, a static set of parameters was used, producing expected results. However, to improve performance, I introduced dynamic parameter tuning, similar to the approach used in Lasso models with varying alpha parameters. This involved using GridSearchCV with a search grid of 12 parameters, allowing the model to explore various configurations to find the optimal combination.

While this dynamic tuning enhanced the model's accuracy, it significantly increased training time, with each run taking an average of one hour. Despite the longer processing time, the parameter-tuned RSF model provided better results, with improved survival function predictions and more accurate cumulative hazard estimates. The grid search also helped to identify important features through permutation importance analysis, offering deeper insights into the model's performance.

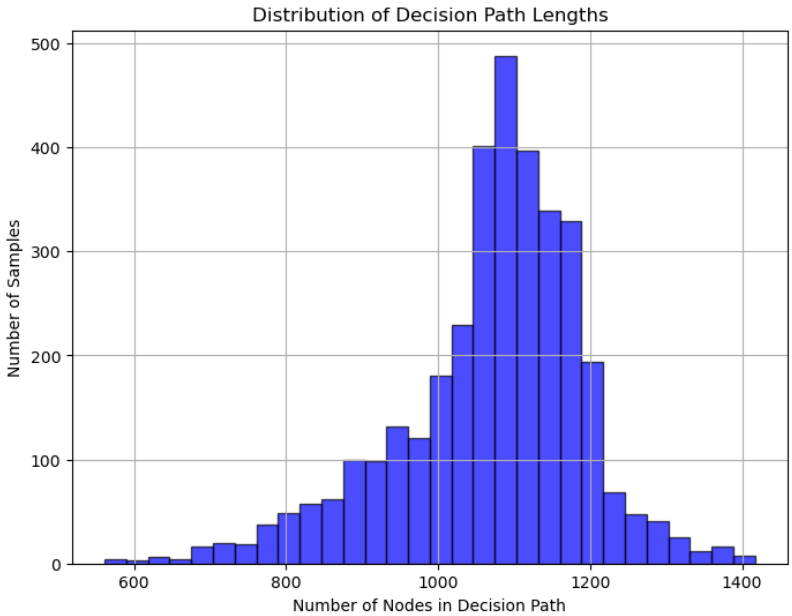


FIGURE 4.19: Descision Tree Matrix Visualisation

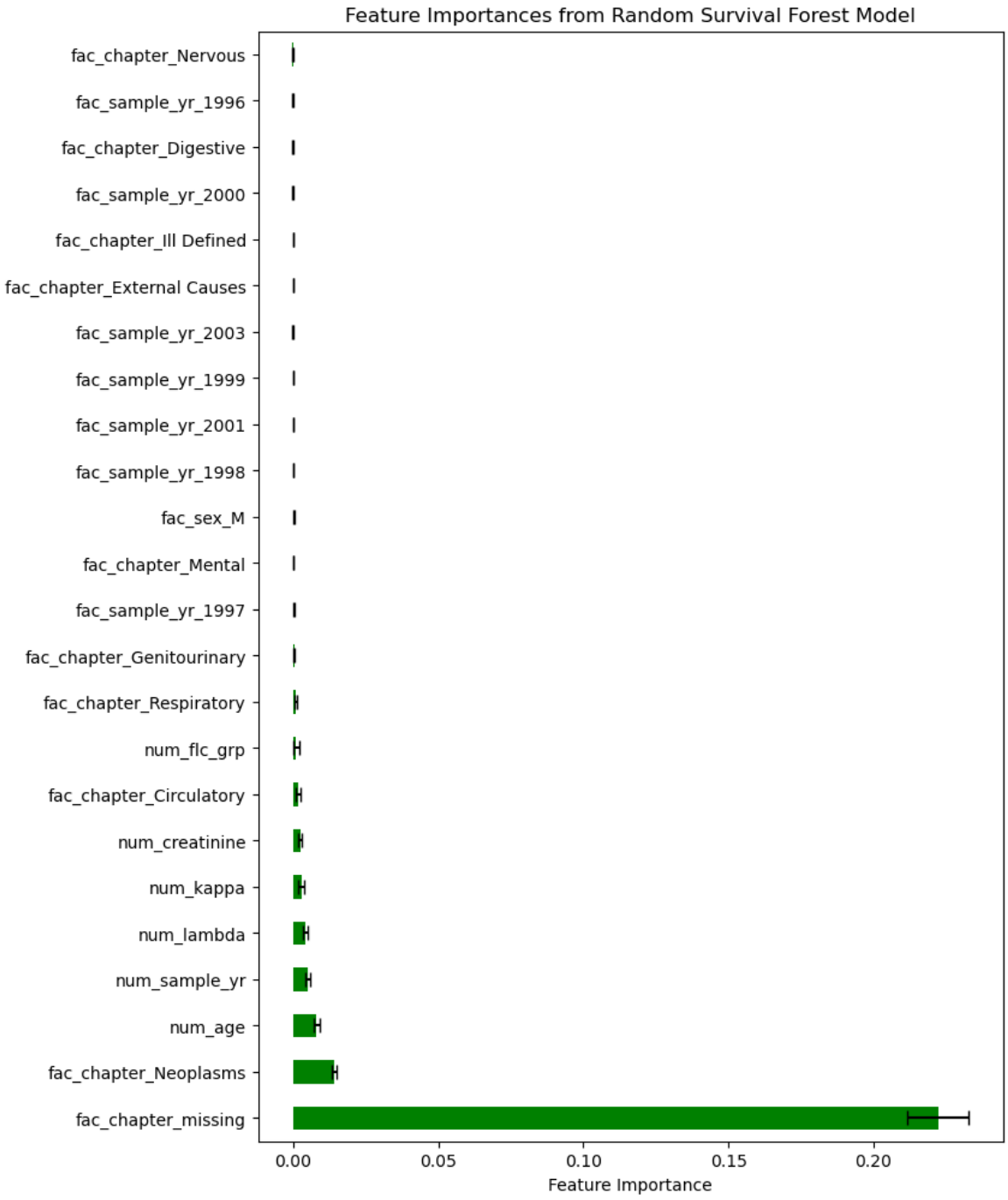


FIGURE 4.20: Variable Importance Boxplots

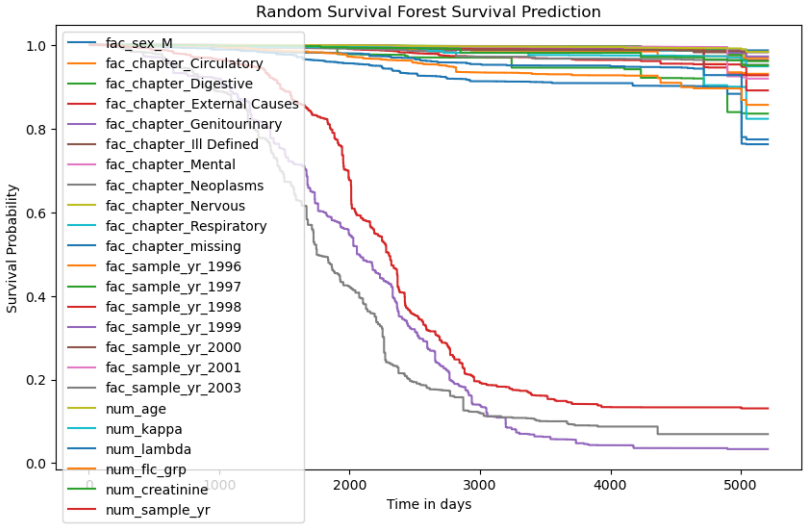


FIGURE 4.21: RSF Surival Curves

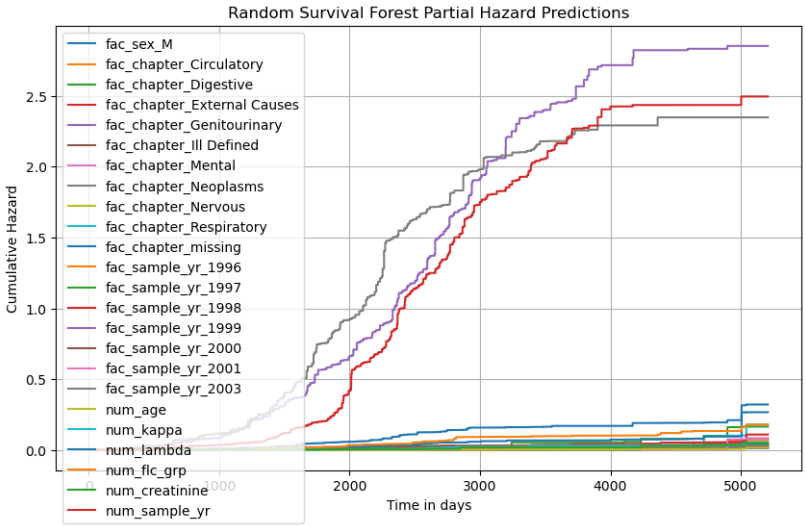


FIGURE 4.22: RSF Hazard Curves

4.3.3 Models comparison

The final metrics as described in 3.1 is shown here:

Metric	Cox Model	RSF Model
Concordance	0.9447	0.9545
Brier Score	0.0207	0.0129
Integrated Brier Score (IBS)	0.0295	0.0244
MAE	267.0475	277.8896
RMSE	2518.4543	2997.5985
One Calibration Error (One-Cal)	3.55e-15	0.0011
D-Calibration Error (D-Cal)	1.10e-06	0.1065

TABLE 4.4: Comparison of Cox and RSF Models for Survival Analysis

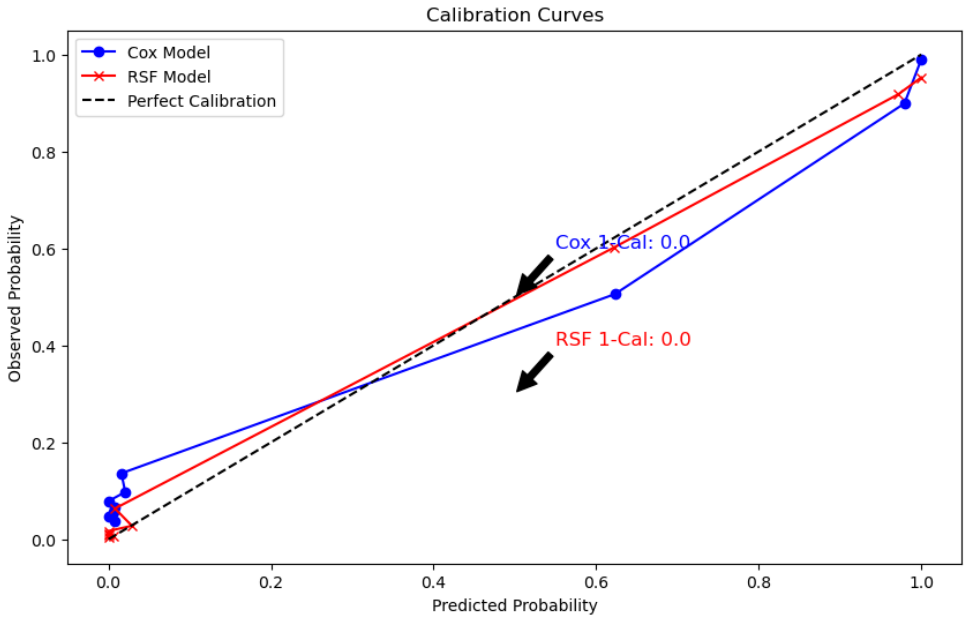


FIGURE 4.23: One Calibration Errors

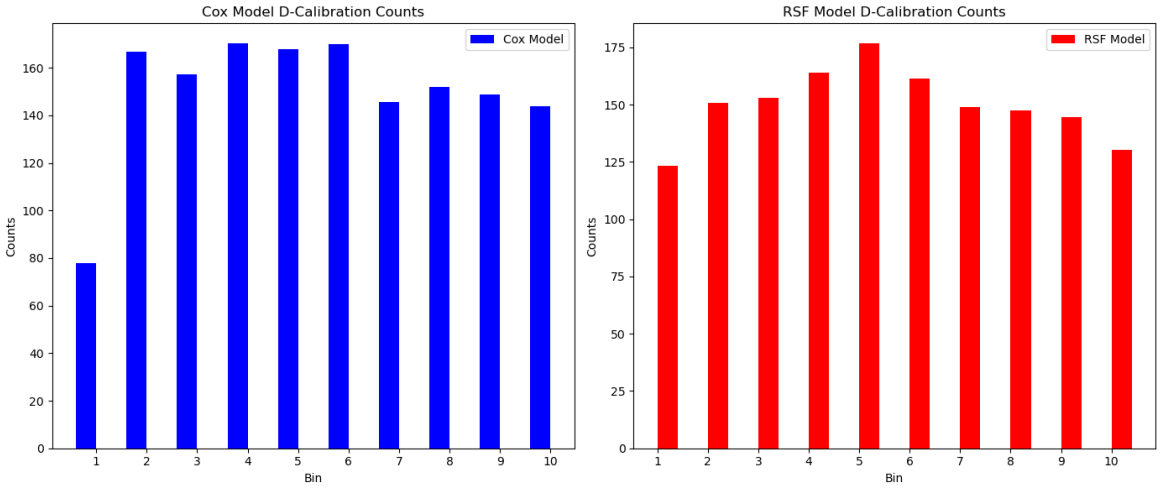


FIGURE 4.24: Binned D-calibration Errors

Chapter 5

Conclusion

In this study, I explored the application of Random Survival Forests (RSF) and Cox Proportional Hazards models in survival analysis, focusing on evaluating their performance through dynamic parameter tuning and the use of synthetic data generated by Survival GAN and Survival VAE models. The simulation results demonstrated that both models provide valuable insights into the underlying patterns of the dataset, with the RSF model showing slightly improved performance in survival prediction and cumulative hazard estimates. However, the Cox model, once convergence issues were addressed through preprocessing and variable selection techniques, proved to be a robust alternative for simpler survival data structures.

The comparison between the models revealed that while the Cox model performed well in terms of metrics like concordance and calibration, the RSF model's flexibility in handling complex interactions and high-dimensional data provided it with an edge, particularly in Brier score and integrated Brier score (IBS). Despite the computational cost associated with tuning parameters in the RSF model, the results highlight its effectiveness in survival analysis, especially when dealing with more intricate datasets.

In addition, the exploratory data analysis (EDA) provided crucial insights into the structure of the dataset, with the correlation matrix and bivariate analysis identifying key relationships between variables, such as the skewed distribution of age and its effect on survival. The visualizations generated through the RSF and Cox models helped validate these findings, further emphasizing the importance of selecting appropriate models based on the complexity of the data.

Ultimately, this simulationstudy underscores the value of using a robust framework like ADMEP [18] to easily apply advanced machine learning techniques like RSF in survival analysis, particularly in scenarios where high-dimensional data and complex variable interactions come into play. The results also suggest that model selection should be guided by both performance metrics and the computational feasibility of tuning parameters, especially in resource-constrained environments.

Bibliography

[1] Tomasz Burzykowski. "Survival analysis: Methods for analyzing data with censored observations". en. In: *Seminars in Orthodontics* 30.1 (Feb. 2024), pp. 29–36. ISSN: 10738746. DOI: 10.1053/j.sodo.2024.01.008. URL: https://linkinghub.elsevier.com/retrieve/pii/S1073874624000094 (visited on 04/16/2024).

[2] D. R. Cox. "Regression Models and Life-Tables". In: *Journal of the Royal Statistical Society. Series B (Methodological)* 34.2 (1972). Publisher: [Royal Statistical Society, Wiley], pp. 187–220. ISSN: 00359246. URL: http://www.jstor.org/stable/2985181 (visited on 04/16/2024).

[3] Cameron Davidson-Pilon. *lifelines, survival analysis in Python*. Jan. 2024. DOI: 10.5281/ZENODO.805993. URL: https://zenodo.org/doi/10.5281/zenodo.805993 (visited on 04/21/2024).

[4] Angela Dispenzieri et al. "Use of Nonclonal Serum Immunoglobulin Free Light Chains to Predict Overall Survival in the General Population". In: *Mayo Clinic Proceedings* 87.6 (June 2012), pp. 517–523. ISSN: 00256196. DOI: 10.1016/j.mayocp.2012.03.009. URL: https://linkinghub.elsevier.com/retrieve/pii/S0025619612003886 (visited on 09/01/2024).

[5] Erik Drysdale. *SurvSet: An open-source time-to-event dataset repository*. Mar. 6, 2022. arXiv: 2203.03094[cs,stat]. URL: http://arxiv.org/abs/2203.03094 (visited on 08/22/2024).

[6] Laura Freijeiro-González, Manuel Febrero-Bande, and Wenceslao González-Manteiga. "A Critical Review of LASSO and Its Derivatives for Variable Selection Under Dependence Among Covariates". en. In: *International Statistical Review* 90.1 (Apr. 2022), pp. 118–145. ISSN: 0306-7734, 1751-5823. DOI: 10.1111/insr.12469. URL: https://onlinelibrary.wiley.com/doi/10.1111/insr.12469 (visited on 04/21/2024).

[7] Humza Haider et al. *Effective Ways to Build and Evaluate Individual Survival Distributions*. arXiv:1811.11347 [cs, stat]. Nov. 2018. URL: http://arxiv.org/abs/1811.11347 (visited on 04/28/2024).

[8] Frank E. Harrell. *Regression Modeling Strategies: With Applications to Linear Models, Logistic and Ordinal Regression, and Survival Analysis*. Springer Series in Statistics. Cham: Springer International Publishing, 2015. ISBN: 978-3-319-19424-0 978-3-319-19425-7. DOI: 10.1007/978-3-319-19425-7. URL: https://link.springer.com/10.1007/978-3-319-19425-7 (visited on 08/22/2024).

[9] J. M. Ichida et al. "Evaluation of protocol change in burn-care management using the Cox proportional hazards model with time-dependent covariates". In: *Statistics in Medicine* 12.3 (Feb. 1993), pp. 301–310. ISSN: 0277-6715, 1097-0258. DOI: 10.1002/sim.4780120313. URL: https://onlinelibrary.wiley.com/doi/10.1002/sim.4780120313 (visited on 08/30/2024).

[10] Hemant Ishwaran et al. "Random survival forests". In: *The Annals of Applied Statistics* 2.3 (Sept. 2008). arXiv:0811.1645 [stat]. ISSN: 1932-6157. DOI: 10.1214/08-AOAS169. URL: http://arxiv.org/abs/0811.1645 (visited on 04/16/2024).

[11] Byron C. Jaeger et al. *Accelerated and interpretable oblique random survival forests*. arXiv:2208.01129 [stat]. Aug. 2022. URL: http://arxiv.org/abs/2208.01129 (visited on 04/16/2024).

[12] Man Jin. "Imputation methods for informative censoring in survival analysis with time dependent covariates". en. In: *Contemporary Clinical Trials* 136 (Jan. 2024), p. 107401. ISSN: 15517144. DOI: 10.1016/j.cct.2023.107401. URL: https://linkinghub.elsevier.com/retrieve/pii/S1551714423003245 (visited on 04/16/2024).

[13] Bipin Joshi. *Beginning SOLID Principles and Design Patterns for ASP.NET Developers*. en. Berkeley, CA: Apress, 2016. ISBN: 978-1-4842-1847-1 978-1-4842-1848-8. DOI: 10.1007/978-1-4842-1848-8. URL: http://link.springer.com/10.1007/978-1-4842-1848-8 (visited on 04/28/2024).

[14] John D. Kalbfleisch and Douglas E. Schaubel. "Fifty Years of the Cox Model". en. In: *Annual Review of Statistics and Its Application* 10.1 (Mar. 2023), pp. 1–23. ISSN: 2326-8298, 2326-831X. DOI: 10.1146/annurev-statistics-033021-

014043. URL: https://www.annualreviews.org/doi/10.1146/annurev-statistics-033021-014043 (visited on 04/21/2024).

[15] Georgios Kantidakis et al. "A Simulation Study to Compare the Predictive Performance of Survival Neural Networks with Cox Models for Clinical Trial Data". en. In: *Computational and Mathematical Methods in Medicine* 2021 (Nov. 2021). Ed. by Zoran Bursac, pp. 1–15. ISSN: 1748-6718, 1748-670X. DOI: 10.1155/2021/2160322. URL: https://www.hindawi.com/journals/cmmm/2021/2160322/ (visited on 04/16/2024).

[16] Imran Kurt Omurlu, Mevlut Ture, and Füsun Tokatli. "The comparisons of random survival forests and Cox regression analysis with simulation and an application related to breast cancer". en. In: *Expert Systems with Applications* 36.4 (May 2009), pp. 8582–8588. ISSN: 09574174. DOI: 10.1016/j.eswa.2008.10.023. URL: https://linkinghub.elsevier.com/retrieve/pii/S0957417408007343 (visited on 04/16/2024).

[17] Can Meng et al. "Simulating time-to-event data subject to competing risks and clustering: A review and synthesis". en. In: *Statistical Methods in Medical Research* 32.2 (Feb. 2023), pp. 305–333. ISSN: 0962-2802, 1477-0334. DOI: 10.1177/09622802221136067. URL: http://journals.sagepub.com/doi/10.1177/09622802221136067 (visited on 04/16/2024).

[18] Tim P. Morris, Ian R. White, and Michael J. Crowther. "Using simulation studies to evaluate statistical methods". In: *Statistics in Medicine* 38.11 (May 2019). arXiv:1712.03198 [stat], pp. 2074–2102. ISSN: 0277-6715, 1097-0258. DOI: 10.1002/sim.8086. URL: http://arxiv.org/abs/1712.03198 (visited on 04/16/2024).

[19] Chirag Nagpal, Willa Potosnak, and Artur Dubrawski. *auton-survival: an Open-Source Package for Regression, Counterfactual Estimation, Evaluation and Phenotyping with Censored Time-to-Event Data*. arXiv:2204.07276 [cs, stat]. Aug. 2022. URL: http://arxiv.org/abs/2204.07276 (visited on 04/16/2024).

[20] Alexander Norcliffe et al. *SurvivalGAN: Generating Time-to-Event Data for Survival Analysis*. arXiv:2302.12749 [cs]. Feb. 2023. URL: http://arxiv.org/abs/2302.12749 (visited on 04/16/2024).

[21] Samuel Pawel, Lucas Kook, and Kelly Reeve. "Pitfalls and potentials in simulation studies: Questionable research practices in comparative simulation studies allow for spurious claims of superiority of any method". In: *Biometrical Journal* 66.1 (Jan. 2024). arXiv:2203.13076 [stat], p. 2200091. ISSN: 0323-3847, 1521-4036. DOI: 10.1002/bimj.202200091. URL: http://arxiv.org/abs/2203.13076 (visited on 04/16/2024).

[22] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[23] Hoang Pham, ed. *Springer Handbook of Engineering Statistics*. en. Springer Handbooks. London: Springer London, 2023. ISBN: 978-1-4471-7502-5 978-1-4471-7503-2. DOI: 10.1007/978-1-4471-7503-2. URL: https://link.springer.com/10.1007/978-1-4471-7503-2 (visited on 04/16/2024).

[24] Evan M. Polce and Kyle N. Kunze. "A Guide for the Application of Statistics in Biomedical Studies Concerning Machine Learning and Artificial Intelligence". en. In: *Arthroscopy: The Journal of Arthroscopic & Related Surgery* 39.2 (Feb. 2023), pp. 151–158. ISSN: 07498063. DOI: 10.1016/j.arthro.2022.04.016. URL: https://linkinghub.elsevier.com/retrieve/pii/S0749806322002821 (visited on 04/16/2024).

[25] Sebastian Pölsterl. *scikit-survival*. Dec. 2023. DOI: 10.5281/ZENODO.3352342. URL: https://zenodo.org/doi/10.5281/zenodo.3352342 (visited on 04/21/2024).

[26] Shi-ang Qi, Weijie Sun, and Russell Greiner. "SurvivalEVAL: A Comprehensive Open-Source Python Package for Evaluating Individual Survival Distributions". In: *Proceedings of the AAAI Symposium Series* 2.1 (Jan. 22, 2024), pp. 453–457. ISSN: 2994-4317. DOI: 10.1609/aaaiss.v2i1.27713. URL: https://ojs.aaai.org/index.php/AAAI-SS/article/view/27713 (visited on 09/04/2024).

[27] Shi-ang Qi et al. *An Effective Meaningful Way to Evaluate Survival Models*. arXiv:2306.01196 [cs, stat]. June 2023. URL: http://arxiv.org/abs/2306.01196 (visited on 04/16/2024).

[28] Zhaozhi Qian, Bogdan-Constantin Cebere, and Mihaela van der Schaar. *Synthcity: facilitating innovative use cases of synthetic data in different data modalities*. Jan. 18, 2023. arXiv: 2301.07573[cs]. URL: http://arxiv.org/abs/2301.07573 (visited on 08/22/2024).

[29] Susan M. Shortreed and Ashkan Ertefaie. "Outcome-Adaptive Lasso: Variable Selection for Causal Inference". en. In: *Biometrics* 73.4 (Dec. 2017), pp. 1111–1122. ISSN: 0006-341X, 1541-0420. DOI: 10.1111/biom.12679. URL: https://academic.oup.com/biometrics/article/73/4/1111-1122/7537777 (visited on 04/16/2024).

[30] Hayley Smith et al. "A scoping methodological review of simulation studies comparing statistical and machine learning approaches to risk prediction for time-to-event data". en. In: *Diagnostic and Prognostic Research* 6.1 (June 2022), p. 10. ISSN: 2397-7523. DOI: 10.1186/s41512-022-00124-y. URL: https://diagnprognres.biomedcentral.com/articles/10.1186/s41512-022-00124-y (visited on 04/16/2024).

[31] Raphael Sonabend et al. *Flexible Group Fairness Metrics for Survival Analysis*. arXiv:2206.03256 [cs, stat]. July 2022. URL: http://arxiv.org/abs/2206.03256 (visited on 04/16/2024).

[32] Maria Thurow et al. *How to Simulate Realistic Survival Data? A Simulation Study to Compare Realistic Simulation Models*. arXiv:2308.07842 [stat]. Aug. 2023. URL: http://arxiv.org/abs/2308.07842 (visited on 04/16/2024).

[33] Robert Tibshirani. "Regression Shrinkage and Selection via the Lasso". In: *Journal of the Royal Statistical Society. Series B (Methodological)* 58.1 (1996). Publisher: [Royal Statistical Society, Wiley], pp. 267–288. ISSN: 00359246. URL: http://www.jstor.org/stable/2346178 (visited on 04/26/2024).

[34] Hong Wang and Gang Li. "A Selective Review on Random Survival Forests for High Dimensional Data". In: *Quantitative Bio-Science* 36.2 (Nov. 2017), pp. 85–96. DOI: 10.22283/QBS.2017.36.2.85. URL: https://doi.org/10.22283/QBS.2017.36.2.85 (visited on 08/24/2024).

[35] Mark D. Wilkinson et al. "The FAIR Guiding Principles for scientific data management and stewardship". en. In: *Scientific Data* 3.1 (Mar. 2016), p. 160018.

ISSN: 2052-4463. DOI: 10.1038/sdata.2016.18. URL: https://www.nature.com/articles/sdata201618 (visited on 04/20/2024).

[36] Hyun-Soo Woo, Jisun Kim, and Albert A. Cannella. "Time Dependence in the Cox Proportional Hazard Model as a Theory Development Opportunity: A Step-by-Step Guide". en. In: *Organizational Research Methods* (Oct. 2023), p. 10944281231205027. ISSN: 1094-4281, 1552-7425. DOI: 10.1177/10944281231205027. URL: http://journals.sagepub.com/doi/10.1177/10944281231205027 (visited on 04/21/2024).

[37] Hiroki Yanagisawa. *Proper Scoring Rules for Survival Analysis*. arXiv:2305.00621 [cs, stat]. June 2023. URL: http://arxiv.org/abs/2305.00621 (visited on 04/16/2024).

[38] Hao Helen Zhang and Wenbin Lu. "Adaptive Lasso for Cox's Proportional Hazards Model". In: *Biometrika* 94.3 (2007). Publisher: [Oxford University Press, Biometrika Trust], pp. 691–703. ISSN: 00063444, 14643510. URL: http://www.jstor.org/stable/20441405 (visited on 04/16/2024).