Tilburg University

PREDICTING DELIBERATE EMPLOYEE ATTRITION USING SURVIVAL ANALYSIS:

A COMPARISON BETWEEN STATE-OF-THE-ART AND NOVEL SURVIVAL METHODS

LUCA ILARDA

THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE IN DATA SCIENCE & SOCIETY
AT THE SCHOOL OF HUMANITIES AND DIGITAL SCIENCES
OF TILBURG UNIVERSITY

STUDENT NUMBER

294352

COMMITTEE

Dr. Peter Hendrix
Dr. Yash Satsangi

LOCATION

Tilburg University
School of Humanities and Digital Sciences
Department of Cognitive Science &
Artificial Intelligence
Tilburg, The Netherlands

DATE

June 25, 2021

ACKNOWLEDGMENTS

To my supervisor Peter, for all his valuable advice, in-depth knowledge, and for having inspired me to investigate survival methods through his studies.
To professor Satsangi, for having developed in me an interest in machine learning.
To my family and Silvia.

PREDICTING DELIBERATE EMPLOYEE ATTRITION USING SURVIVAL ANALYSIS:

A COMPARISON BETWEEN STATE-OF-THE-ART AND NOVEL SURVIVAL METHODS

LUCA ILARDA

Abstract

Predicting voluntary employee attrition is valuable for a firm as it may proactively act to retain a focal component of the staff. This thesis explores the potential of survival analysis in forecasting deliberate workforce turnover. In particular, the current work compares the predictive performance of state-of-the-art Cox (1972) Proportional Hazard (Cox PH) model with novel survival machine learning methods as the Random Survival Forest (Ishwaran, Kogalur, Blackstone, Lauer, et al., 2008), DeepSurv (Katzman et al., 2018), and DeepHit (C. Lee, Zame, Yoon, & van der Schaar, 2018). The comparison revealed that DeepSurv outperformed the traditional Cox PH in foreseeing deliberate employee attrition on synthetic workforce data (IBM, 2017). The length of an employee's career, the total number of companies in which the employee worked, and extra hours spent at work were observed to be the most relevant features for DeepSurv to predict the churn.

1 INTRODUCTION

Employees may voluntarily or involuntarily leave the company they are working for (Alduayj & Rajpoot, 2018). For example, an employee who decides to change organizations because she/he is unsatisfied with the monthly wage is voluntary attrition. On the contrary, an employer dismissing an employee who was often late at work is involuntary employee attrition (Al Mamun & Hasan, 2017). This thesis will also refer to an employee who voluntarily decides to quit a corporation as a churner employee.

Voluntary employee attrition causes direct and indirect costs for a firm (Madariaga, Oller, & Martori, 2018). Indeed, an outgoing employee directly affects an organization as it will need to invest both time and resources to recruit and train a new member of the staff (Sumathi, Balakrishnan, Naveen, Hariharan, & Rahul Iniyan, 2021). Further, a churner employee indirectly costs the company since it influences the state of open projects. This potentially leads to customers' discontent if deadlines are not met and hence damages the business' image (Căplescu, Ilie, & Strat, 2019). If the leaving employee is a focal or talented component of the working team, a company will face high direct and indirect costs from the attrition (Jin et al., 2020).

Predicting deliberate workforce turnover and understanding its prevalent driving factors through data analysis may benefit the employers and the employees. For instance, if the prevailing cause of churn is overtime, the employer may proactively reduce extra working hours to retain workers at risk and save attrition costs. Consequently, employees may be more satisfied with their current position, hence deciding to remain in the corporation and not stress themselves with the task of finding another job. As such, human resources analytics may positively impact even staff members' lives.

Given the societal benefits described, this research aims at predicting voluntary employee attrition and investigates the predominant causes that motivate the churn. To do so, we will use survival analysis, which is a statistical technique that models the time up to a target event (Kleinbaum & Klein, 2012). As data collection is limited in time and information on whether a subject has experienced an event may be unobserved, survival analysis has developed mechanisms to deal with censored data (P. Wang, Li, & Reddy, 2019). In survival analysis, censoring refers to the incomplete observation of an event being studied (Moore, 2016). In the specific case of voluntary employee attrition, right-censored information is present: the fact that an employee has not left the company at the time of data acquisition does not imply that she/he has not quit the organization later on. Traditional machine learning algorithms do not deal with censoring (P. Wang et al., 2019), and estimating whether an employee will leave a job without taking censored information into account may lead to biased results (McCloy, Purl, & Banjanovic, 2019). Moreover, survival analysis offers the advantage of modeling the probability of surviving an event over time. In contrast, traditional machine learning techniques could be used to binary classify whether or not an employee will quit an organization concerning a specific time point (Madariaga et al., 2018; Willett & Singer, 1991). Since traditional machine learning does not deal with censoring and does not model attrition probabilities over time, we opted for survival analysis methods to conduct our study.

This project encloses a novel methodology to predict deliberate workforce turnover. Indeed, several machine learning algorithms have recently been extended to perform survival analysis and deal with censored data (P. Wang et al., 2019). However, novel techniques have not been utilized for employee attrition prediction yet (for this, see the related work section below). Here, we will compare the performances in predicting voluntary employee churn of the state-of-the-art Cox (1972) Proportional Hazard (Cox PH) model with three newly developed machine learning approaches: the Random Survival Forest (Ishwaran et al., 2008), DeepSurv (Katzman et al., 2018), and DeepHit (C. Lee et al., 2018). Hence, we will use the best-performing survival model to determine why employees leave an organization, thus enabling the employer to act proactively and produce the societal benefits discussed.

To predict voluntary employee churn and investigate its main driving factors, we will use the "IBM HR Analytics Employee Attrition & Performance" dataset (IBM, 2017), composed of synthetic recordings about each employee in the company, and fictitious survey data. The motivation behind utilizing this data frame in the context of survival analysis is that it contains both an event (*Attrition*) and a time (*YearsAtCompany*) variable. Indeed, survival analysis requires both an event and time feature in a dataset to model the time to the event being studied (Kleinbaum & Klein, 2012). Furthermore, we opted for this data frame as the total number of covariates it holds is suited for getting insights into the predominant causes of deliberate employee turnover.

In this context, the current paper will investigate the following general research question, from which two sub-research questions RQ1 and RQ2 are derived:

> *How well can voluntary employee attrition be predicted using survival analysis?*

RQ1 *What is the predictive performance of state-of-the-art and novel survival analysis algorithms for voluntary employee attrition data?*

RQ2 *Which predictors are most relevant for predicting voluntary employee attrition?*

This thesis follows the following structure: Section 2 offers a literature background on employee attrition and survival analysis studies; Section 3 explains the survival analysis methods and the evaluation scores we applied, specifically dedicating subsection 3.1 to algorithms, 3.2 to metrics, and 3.3 to methods to assess the most relevant predictors; Section 4 illustrates the dataset we used and the experiments we conducted; Section 5 reports our findings that will be discussed in detail in Section 6. The last section 7 draws the conclusion of this paper.

2 RELATED WORK

Previous research used traditional machine learning techniques to predict whether an employee will quit a job. For example, Rombaut and Guerry (2018) modeled workforce attrition through the usage of Logistic Regression. By interpreting the output of the fitted Logistic Regression model, they found that gender, seniority, and marital status was among the most significant predictors of employee churn. Liu et al. (2018) compared the performances in forecasting workforce turnover of a Logistic Regression model with other supervised learning algorithms such as Random Forest, AdaBoost, and Support Vectors Classifier. Their results indicate that Random Forest and AdaBoost achieved the best predictive performance and that the factors driving the churn were career experience and high job skills. Alduayj and Rajpoot (2018) compared a collection of models, including Random Forest and K-Nearest-Neighbours, on balanced and unbalanced datasets to analyze their performances in foreseeing workers' attrition. They found that extra working hours and years of career were the most relevant predictors in predicting employee churn through balanced data.

Researchers also applied methods different from traditional machine learning techniques to forecast whether or not an employee will leave a company. For instance, Emadi and Staats (2020) modeled workers turnover through econometrics approaches, observing that managers played a focal role in predicting the employee's decision to quit a job. Fang et al. (2018) fitted a conditional semi-Markov model to forecast workforce attrition over time. By interpreting the probabilities' output by their model, they concluded that the years worked by the employee in the current position was a relevant factor for predicting her/his churn. Further, graph embedding techniques were applied and compared with machine learning methods by Cai et al. (2020) to foresee staff members' attrition. They found that employee's job level and educational background were the prominent causes influencing worker's turnover.

Recently, a number of studies have started exploring the potential of survival analysis in the domain of employee attrition. E. Lee (2019) estimated the survival functions of nurses in South Korea using both the Kaplan-Meier (KM) and the Cox PH method. They reported that gender and job satisfaction were among the principal factors motivating employee churn. Cox PH was also utilized by Assefa, Mariam, Mekonnen, and Derbew (2017) to investigate why medical professors in Ethiopia quit their job. They found that academic level and age significantly affected attrition. Similarly, W. Wang (2019) explored staff member's turnover by analyzing the results of a fitted Cox PH. They observed that job position and gender were two of the main causes driving the churn. Madariaga et al. (2018) fitted both a Cox PH and a Logistic Regression model on employee data to examine the prevalent factors leading to worker's turnover. Through an interpretation of the output of the fitted models, they concluded that either the Cox PH and the Logistic Regression were concordant in indicating that employee's income, gender, age, and marital status were the predominant causes of employee attrition. However, they observed that survival analysis methods were more appropriate in examining employee churn than logistic regression due to their capacity of estimating survival probabilities over time and not only at a specific time point (Madariaga et al., 2018). Silva, Vieira, Pimenta, and Teixeira (2018) modeled employee attrition using a Cox PH, with the particular focus of predicting low-income employee churn. Their results showed that gender, age, level of education, and years spent at the company were among the most relevant factors influencing the churn. Moreover, they pointed out that survival analysis was more suited than other methods in forecasting employee attrition due to its ability to deal with censored data, hence not biasing the results as other techniques would have done (Silva et al., 2018).

As the literature review on employee attrition clarified, different methods have been applied for predicting employee churn. Among them, survival analysis should be preferred when forecasting employee attrition (McCloy et al., 2019). In fact, in contrast with survival analysis, traditional machine learning does not model deliberate employee turnover in a longitudinal time manner and does not take censoring into account (Madariaga et al., 2018; Willett & Singer, 1991). Further, for instance, although a semi-Markov model deals with censoring and could output survival probabilities over time (Abner, Charnigo, & Kryscio, 2013; Zhao & Hu, 2013), it is not easy to implement, and it could output imprecise results in the presence of rare events' instances (Abner et al., 2013). Due to its capacity in dealing with censored information in a dataset with relatively straightforward implementation and output survival probabilities over time, survival analysis offers more advantages than other methods for modeling voluntary employee churn.

Even though prior studies (Madariaga et al., 2018; Silva et al., 2018; Willett & Singer, 1991) pointed out the convenience of using survival analysis in the domain of employee attrition, the current literature on workforce turnover has only explored traditional survival analysis methods such as KM and Cox PH thus far. Moreover, previous papers on employee attrition based on survival analysis have mainly focused on investigating the causes that drive the churn without comparing the predictive capabilities of survival models. In particular, this contrasts with precedent studies on workforce turnover based on machine learning, where different algorithms are compared and the main factors influencing the churn explored.

This thesis aims at contributing to the literature by comparing the predictive performances of novel survival analysis algorithms with state-of-the-art Cox PH while still examining the predominant causes of employee churn. Indeed, novel survival machine learning techniques have recently been proposed and are predominantly utilized in the medical field (P. Wang et al., 2019). For example, Hathaway, Yanamala, Budoff, Sengupta, and Zeb (2021) compared the performances of a Cox PH model with other survival methods, including a Random Survival Forest (RSF) and a survival deep neural network (DeepSurv), in predicting atherosclerosis. They found that both RSF and DeepSurv outperformed the traditional Cox PH. Further, they interpreted the most relevant factors influencing atherosclerosis according to RSF and DeepSurv through permutation importance. Nakagawa et al. (2020) performed a comparison between Cox PH, a survival deep neural network (DeepHit), and Weibull survival deep neural network in forecasting Alzheimer's in patients using extracted features from brain images. They concluded that either DeepHit and the Weibull deep neural network achieved a better performance than Cox PH in predicting Alzheimer's in patients. Kantidakis et al. (2020) compared a RSF model and a Partial Logistic Artificial Neural Network (PLANN) method with Cox PH to forecast liver transplantation. They observed that Cox PH was outperformed by novel survival methods. In addition, they explored the predominant predictors in predicting liver transplantation retrieving features' importance through the RSF.

To address the lack in the usage of novel survival machine learning algorithms in the workforce attrition literature, we decided to compare a Cox PH model used as a baseline with a RSF, DeepSurv, and DeepHit model in forecasting employee churn. In particular, given that precedent studies in the survival analysis background reported an increase in predictive performance while utilizing novel survival procedures, we will investigate how well we can predict voluntary employee attrition using survival analysis approaches.

Furthermore, the performance comparison offered by this study will not affect the ability to get insights into the most relevant features for predicting employee attrition. Indeed, as the literature review conducted on survival analysis specified, previous studies in the medical field explored the focal factors utilized by novel survival methods for making predictions. In the domain of employee attrition, precedent papers shared the task of examining the fitted models by retrieving the importance of the variables they used. Overall, studies on employee turnover observed career length, gender, job level, overtime, and age as the prominent causes of employee churn. Since retrieving the main turnover predictors is common in the employee attrition literature and novel methods will not affect the ability to do so, we will use our best-performing survival model to investigate the predominant factors for predicting employee churn.

3 METHODS

In this study, we decided to use survival analysis to model voluntary employee attrition. In particular, we chose to use survival analysis methods as they offer the opportunity to deal with censored data, whereas other techniques such as traditional machine learning don't. In a dataset, censoring refers to the incomplete observation of an event being studied (Moore, 2016). In the specific circumstances of deliberate workforce turnover, *right-censoring* is present. Indeed, although the event of employee attrition is not observed in the data for a given individual does not signify that the same individual has not left the company after the data collection. To include censored information in our estimates, we opted to use survival analysis over other approaches. In this section, we present the survival algorithms that will be implemented for predicting deliberate workforce turnover.

3.1 *Survival algorithms*

Before specifying the algorithms that will be utilized in this thesis, we determine here some survival analysis' terms beneficial for their description. One of them is the survival function. Precisely, it conveys the probability of a subject not encountering an event until T, which exceeds a chosen time t (In & Lee, 2018). It is denoted as:

$$S(t) = Pr(T > t) \tag{1}$$

Another one is the hazard function. Specifically, conditioning on the fact that so far a particular event has not happened, it indicates the ratio of the examined event manifesting when the next time interval (δt) tends towards zero (Kleinbaum & Klein, 2012). Formally, it is characterized as:

$$\lambda(t) = \lim_{\delta t \to 0} \frac{Pr(t \leq T \leq t + \delta t | T \geq t)}{\delta t} \tag{2}$$

By calculating the integral from 0 to t over the hazard function, the cumulative hazard function $\Lambda(t)$ is derived (Kleinbaum & Klein, 2012):

$$\Lambda(t) = \int_0^t \lambda(t) \delta t \tag{3}$$

3.1.1 *Cox PH*

It is a semi-parametric survival model proposed by Cox (1972), which assumes the hazards to be proportional (PH) (Kleinbaum & Klein, 2012). Given a set of n covariates $x = (x_1, x_2, ...x_n)$, Cox (1972) denoted the hazard function at time t by multiplying the hazard at time t where each covariate is equal to 0 - called baseline hazard, $\lambda_0(t)$ - with the exponential of the sum from $i = 1$ to n between the product of the covariates with their coefficients - denoted as risk term, $e^{\sum_{i=1}^{n} \beta_i x_i}$. In this way, Cox (1972) formulated the hazard function $\lambda(t|x)$ as:

$$\lambda(t|x) = \lambda_0(t)e^{\sum_{i=1}^{n} \beta_i x_i}. \tag{4}$$

By dividing the hazards for an observation with covariates values x^* in a given dataset by the hazards for another observation with covariates values x, Cox (1972) obtained:

$$\frac{\lambda(t|x^*)}{\lambda(t|x)} = \frac{\lambda_0(t)e^{\sum_{i=1}^{n} \beta_i x_i^*}}{\lambda_0(t)e^{\sum_{i=1}^{n} \beta_i x_i}}. \tag{5}$$

As a result of Eq. 5, the ratio between the hazards for observation with covariates values x^* and another with covariates values x is not time-dependent. Consequently, it does not change with the changing of time, which is the meaning of the PH assumption (Kleinbaum & Klein, 2012). Following Cox (1972), the values of the coefficients β in Eq. 4 are computed by maximizing of the given partial log-likelihood, $L(\beta)$ (Kleinbaum & Klein, 2012). Specifically, $L(\beta)$ is obtained by taking the product at every d out of k times of the likelihood of an observation p experiencing the event of interest among a set of j observations being at risk (Katzman et al., 2018):

$$L(\beta) = \prod_{d=1}^{k} \frac{e^{\sum_{i=1}^{n} \beta_i x_{p,i}}}{\sum_{j \in R_{(d)}}^{n} e^{\sum_{i=1}^{n} \beta_i x_{j,i}}} \tag{6}$$

with $R_{(d)}$ being a set containing j observations at risk of experiencing the event at time d out of k times, p the observation chosen, n the total amount of covariates in the dataset and i their current number.
Since the baseline hazard $\lambda(t)$ in Eq. 4 does not assume a determined shape compared to parametric survival models, Cox PH is referred to as being semi-parametric (Kleinbaum & Klein, 2012).

3.1.2 *DeepSurv*

It is a feed-forward neural network extension of the Cox PH, and it has been advanced by Katzman et al. (2018). Indeed, similarly to the Cox PH, DeepSurv relies on the assumption of the proportionality of the hazards

(Katzman et al., 2018). However, if Cox PH presumes the hazards to be linearly proportional, DeepSurv allows the proportionality to occur in a non-linear manner thanks to its neural architecture (Katzman et al., 2018). Hence, DeepSurv accounts for more variability than a Cox PH while modeling the data (Katzman et al., 2018).

DeepSurv is composed of a series of fully connected layers, each of them being followed by drop-out (Katzman et al., 2018). The last layer is linearly activated to output the sum of the product between the input covariates and the network weights θ: $\hat{h}_\theta(x) = \sum_{i=1}^{n} \theta_i x_i$ (Katzman et al., 2018). As such, DeepSurv output $\hat{h}_\theta(x)$ is equivalent to the natural logarithm of the risk term in Eq. 4 with the difference being in the coefficients used (Katzman et al., 2018). While training DeepSurv, the loss function to optimize is derived from the log-likelihood formula used in Eq 6 by taking its mean and changing its sign, and by summing it to a penalty term ($\lambda||\theta||_2^2$) (Katzman et al., 2018):

$$l(\theta) = -\frac{1}{N_{d=1}} \sum_{d=1}^{k} (\sum_{i=1}^{n} \theta x_{p,i} - \log \sum_{j \in R_{(d)}}^{n} e^{\sum_{i=1}^{n} \theta x_{j,i}}) + \lambda||\theta||_2^2 \tag{7}$$

where $N_{d=1}$ is the total number of observations exposed to the event being studied at time d out of k times, θ the network weights, n the total number of covariates and i their current number, p the individual being considered, j the person chosen among those who are at risk in the set $R_{(d)}$, and x the covariates in the dataset.

3.1.3 *Random Survival Forest*

Unlike Cox PH and DeepSurv, it does not assume the ratio between the ratio of the hazards to be time-invariant. Hence, RSF allows for more flexibility while modeling the data as compared to Cox PH and DeepSurv. The RSF proposed by Ishwaran et al. (2008) is obtained by bootstrapping B data points from the dataset and growing B number of trees. Each tree is built using the seventy percent of data available, and the other thirty percent remains outside the bag (Ishwaran et al., 2008). Given x randomly chosen covariates, each parent node in a tree is divided into child nodes by the covariate that produces the highest difference in surviving among the children nodes (Ishwaran et al., 2008). Once a stopping criteria has been met, each of the trees returns a cumulative hazard function $\Lambda_b(t|x)$. By taking the sum of every cumulative hazard function obtained by each of the trees and dividing it by the total number of trees B, the RSF returns the cumulative hazard function of the forest as:

$$\Lambda_e(t|x_i) = \frac{1}{B} \sum_{b=1}^{B} \Lambda_b(t|x_i). \tag{8}$$

Since the exponential of the cumulative hazard function with a negative sign is equal to the survival function, given a set of covariates x the RSF estimates the survival function $\hat{S}(t|x)$ as:

$$\hat{S}(t|x) = \exp\left(-\hat{\Lambda}_e(t|x)\right). \tag{9}$$

3.1.4 *DeepHit*

DeepHit is a feed-forward survival neural network that has been proposed by C. Lee et al. (2018). Differently from both Cox PH and DeepSurv, DeepHit does not assume the hazards' ratio to be time-invariant (C. Lee et al., 2018). Moreover, in contrast with the RSF introduced by Ishwaran et al. (2008), DeepHit also offers the opportunity to be applied to tasks in which individuals are at risk of experiencing not only a single event, but also non-independent events (C. Lee et al., 2018). Nonetheless, as our project only deals with one event of interest (employee attrition), we considered DeepHit into its single-event framework.

DeepHit architecture is composed of a number of fully connected layers (C. Lee et al., 2018). Before entering the network, the time variable T need to be divided m into equal-distant times $\tau_0, \tau_1, ...\tau_n$ as DeepHit treats time in a discrete manner (Kvamme, Borgan, & Scheel, 2019). Given a set of covariates x as input, the last layer of a single-event DeepHit is activated by a *softmax* function to output a vector $y(x)$ of estimated probabilities at times $0, 1...n$ (Kvamme et al., 2019):

$$y(x) = [y_0(x), y_1(x), ...y_n(x)]^T \tag{10}$$

Following Kvamme et al. (2019), given a set of covariates x, the estimate of the survival function at τ_j discrete times for a one-event DeepHit is computed by subtracting to one the sum from k to j discrete times of the estimated probabilities $y(x)$ output by the model at each time k:

$$\hat{S}(\tau_j|x) = 1 - \sum_{k=1}^{j} y_k(x). \tag{11}$$

The objective function $\mathcal{L}_{tot}$ to minimize while training DeepHit is obtained by calculating the sum of two different losses:

$$\mathcal{L}_{tot} = \mathcal{L}_1 + \mathcal{L}_2. \tag{12}$$

$\mathcal{L}_1$ handles right-censored information by extending the log-likelihood of the joint distributed event e and times, considering an individual i, with D_i being an uncensoring indicator (Kvamme et al., 2019; C. Lee et al., 2018). In particular, $\mathcal{L}_1$ is composed of a term $D_i \log(y_{e_i}(x_i))$ that brings information about non-censored observations i experiencing the event e being studied, and another term $(1 - D_i)\log(\hat{S}[T_i|x_i])$ that provides information about censored persons i in the dataset (C. Lee et al., 2018; Roblin, Cournede, & Michiels, 2020). By taking the negative sum from one to the total number of individuals N of both censored and uncensored observations in the data frame, $\mathcal{L}_1$ is derived as:

$$\mathcal{L}_1 = -\sum_{i=1}^{N}\big[D_i \log(y_{e_i}(x_i)) + (1 - D_i)\log(\hat{S}[T_i|x_i])\big]. \tag{13}$$

On the other hand, $\mathcal{L}_2$ tries to avoid the discordance of observations' pairs (Kvamme et al., 2019; C. Lee et al., 2018). Indeed, considering every observation i that experienced the event at time T_i, and each observation j that did not experience the event at time T_j (with T_i being less or equal to T_j), $\mathcal{L}_2$ penalizes the pairs (i, j) incorrectly predicted by DeepHit dividing the difference of the estimated survival functions for i and j $(\hat{S}[T_i|x_i] - \hat{S}[T_i|x_j])$ by a penalizer term σ, and taking the exponential of this division (Pawley, 2020). Hence, $\mathcal{L}_2$ is multiplied by an hyperparameter α, which determines how much the loss $\mathcal{L}_2$ is taken into account with respect to the total DeepHit loss ($\mathcal{L}_{tot} = \mathcal{L}_1 + \mathcal{L}_2$) (Pawley, 2020). As such, $\mathcal{L}_2$ is formally defined as:

$$\mathcal{L}_2 = \alpha \sum_{i,j}[D_i \mathbb{1}\{T_i \le T_j\} \exp(\frac{\hat{S}[T_i|x_i] - \hat{S}[T_i|x_j]}{\sigma})] \tag{14}$$

where both the relative importance hyperparameter α and the penalizer σ are to be optimized.

3.2 *Survival Metrics*

To evaluate the algorithms mentioned above, we will use the metrics described in this subsection.

3.2.1 *Concordance Index*

It was proposed by Harrell, Califf, Pryor, Lee, and Rosati (1982). To evaluate the model performance, it considers all the *comparable* pairs of observations (x, y) in a dataset and their survival period as recorded in the data frame (Longato, Vettoretti, & Di Camillo, 2020). In particular, a pair (x, y) is *comparable* if the event occurred for at least x or y, and if only x or y has experienced the event the other observation in the pair survived for a longer recorded time (Harrell Jr, Lee, & Mark, 1996; Longato et al., 2020). To clarify this statement, consider the examples reported in Table 1. Looking at the first row, the pair (a, b) is not comparable as neither the individual a nor the individual b experienced the event being studied.

Table 1: Harrell's concordance index: examples of *comparable* pairs.

Pair	Event	Observed Survival Time	*Comparable*
a	0	4	no
b	0	5	
c	1	5	yes
d	0	8	
e	0	5	no
f	1	8	
g	1	6	no
h	1	6	
i	1	6	yes
j	1	9	

Indeed, it is true that the dataset reports a survival time of four years for observation a, and of five years for observation b. However, since both a and b are censored, the model's predictions for the pair (a, b) can not be compared as it is unknown for how long they did not experience the event *after* the data collection (Longato et al., 2020). As it can not be determined whether a or b survived the event for longer, the pair (a, b) is excluded from Harrell's concordance index computation. In contrast, the pair (c, d) in the second row is *comparable* as the individual c who experienced the event survived for a shorter observed time than the censored subject d. In this way, it is possible to compare the model's prediction for the pair as it is observed that the censored subject d survived for a longer time than c. On the contrary, the pair (e, f) in the third row, with the event occurring for the individual f and e being censored at a shorter observed survival time than f, is not comparable. In fact, since the subject e is censored at five years and f experienced the event at eight years, it is unknown whether e experienced the event before or after f. Furthermore, as the fourth row of Table 1 shows, a pair (g, h) experiencing the event at the same time is not comparable as it can not be determined who survived for longer (Harrell Jr et al., 1996). Conversely, if the event occurred for both the individuals in the pair but at different survival times, the pair is *comparable* (Harrell Jr et al., 1996). As indicated in the last row of Table 1, in fact, it is established that in the pair (i, j), the observation i experienced the event before the individual j.

A *comparable* pair is also *concordant* if the model *predicts* the survival time (rank score) for the observation in the pair with the highest *observed* survival time to be higher than the predicted survival time for the other observation (Harrell Jr et al., 1996). To better illustrate this concept, consider the

Table 2: Harrell's concordance index: examples of *concordant* pairs.

Comparable Pair	Observed Survival Time	Rank Score	*Concordant*
c	5	0.6	yes
d	8	0.75	
i	6	0.90	no
j	9	0.5	

examples in Table 2. The *comparable* pair (c, d) is also *concordant* since the individual with the highest observed survival time d was predicted by the model to have a rank score higher than c. On the other hand, the comparable pair (i, j) in the second row of Table 2 is not concordant as the model predicted the observation with the highest observed survival time j to have a lower rank score than i.

Once the *comparable* and *concordant* pairs are determined, Harrell's concordance index is obtained by dividing the number of *concordant* pairs by the quantity of all the pairs in a dataset that are *comparable*:

$$C\text{-}index = \frac{concordant\ pairs}{comparable\ pairs} \tag{15}$$

with a concordant pair counting 1 (or 0.5 if the fitted model predicts the same survival time for both the observations in the pair) (Harrell Jr et al., 1996).

In this project, we used Harrell's concordance index to evaluate PH models. For RSF and DeepHit, however, we utilized the concordance index as modified by Antolini, Boracchi, and Biganzoli (2005). Indeed, as C. Lee et al. (2018) noted, for a non-proportional hazards model it is relevant to compute how the model captures modifications in risks with the changing of time (Kvamme et al., 2019). Further, Antolini's concordance index offers the advantage to be still comparable to the method proposed by Harrell, given that Harrell's concordance index is computed to evaluate a PH model in the framework of a single event of interest (C. Lee et al., 2018).

Antolini et al. (2005) modified the notion of *concordant* pairs with respect to Harrell's definition of the concordance index while maintaining the notion of *comparable*, leaving Equation 15 unaltered. To present Antolini's modifications to the concordance index, consider a comparable pair (x_i, x_j) where at least x_i experienced the event with observed survival times $T_i < T_j$. According to Antolini et al. (2005), the comparable pair (x_i, x_j) is also *concordant* if the model predicts the *survival function* for the observation in the pair with the highest observed survival time x_j to be higher than the predicted *survival function* for the other observation x_i. Specifically, each survival function is estimated at the lowest observed survival time in the pair T_i. Hence, for the comparable pair (x_i, x_j) to be concordant it should result that $\hat{S}(T_i|x_j) > \hat{S}(T_i|x_i)$. Since both the *survival functions* predicted by the model depend on T_i, where T_i varies per each comparable pair (x_i, x_j) being considered, Antolini's concordance index diverges from Harrell's definition of concordance as only the former takes different times into account (Antolini et al., 2005). Consequently, Antolini's concordance index is referred to as being time-dependent.

For both versions of the concordance index, a model predicting naïvely would result in a concordance score of 0.5, while an optimal model in a concordance score of 1. The concordance index has the advantage of dealing with censored information in the dataset, of being easily interpretable and of being suited for discriminating between correctly and incorrectly predicted risk of experiencing an event in observations' pairs (Vickers & Cronin, 2010). However, it has the disadvantage of not evaluating the model's calibration (Alba et al., 2017). That is, the concordance index does not capture whether the model overestimated or underestimated the observed risk of an event occurring (Alba et al., 2017).

3.2.2 *Integrated Brier Score*

It is a metric that extends the Brier score calculated at one time t to a range of time $\delta t = (t_1 - 0)$ (Kantidakis et al., 2020; Kvamme et al., 2019). In fact, the integrated Brier score (IBS) is obtained by integrating from 0 to t_1 and multiplying the integral by one over a range of time δt:

$$IBS = \frac{1}{\delta t} \int_0^{t_1} BS(t) dt \tag{16}$$

where the Brier score (BS) computed at a specified time t, following Kvamme et al. (2019) and Graf, Schmoor, Sauerbrei, and Schumacher (1999) is:

$$BS(t) = \frac{1}{n} \sum_{i=1}^{n} [\frac{\mathbb{1}\{T_i \leq t, D_i = 1\} + \hat{S}(t|x_i)^2}{\hat{G}(T_i)} + \frac{\mathbb{1}\{T_i > t\}(1 - S(t|x_i))^2}{\hat{G}(t)}] \tag{17}$$

with $D_i = 1$ indicating the occurrence of the event and $\hat{G}(t)$ being an estimate of the $\hat{S}(t|x)$ obtained by using the Kaplan-Meier method (Kvamme et al., 2019).

An ideal model with perfect predictions would return an IBS = 0, while a model predicting everything wrong would result in an IBS of 1. The integrated Brier score has the advantage of dealing with censored information in the dataset while assessing the fitted survival model's performance and of capturing both the model's calibration and discrimination (Kattan & Gerds, 2018). Nevertheless, it is sensitive to the choice of the time range over which the Brier score is integrated. In fact, for example, the integrated Brier score computed using a range of time equal to ten years could be different from the integrated Brier score calculated over a time range of eleven years. Thus, the range of time δt in Eq. 16 should be determined carefully. To do so, prior studies in the survival analysis' literature (Haider, Hoehn, Davis, & Greiner, 2020; Kantidakis et al., 2020) suggested utilizing a time range from zero to the maximum observed survival time in the dataset. Further, another drawback of the integrated Brier score is that it is not an easily interpretable metric (Kattan & Gerds, 2018).

3.3 *Methods to assess the most important predictors*

The method that we will use to assess the most relevant factors for predicting voluntary employee attrition will depend on which model will have the best predictive performance.

If the Cox PH results in the highest C-index and the lowest IBS, we will utilize the absolute value of Wald's test statistic z-score output by the model. Indeed, in the case of a considerable absolute value of a z-score, there is not enough evidence to retain the null hypothesis that the coefficient β is equal to 0 (Moore, 2016), hence indicating that the covariate has relevance in the model.

If the RSF has the best results, we will utilize the variable importance score. In particular, the RSF computes the importance of a feature by first introducing into the fitted trees the samples that were not used to grow that tree (Ishwaran et al., 2008). Then, the feature's values are randomly allocated at every child node division for the variable they meet (Ishwaran et al., 2008). Every tree output is thus evaluated, and the variable importance score for that tree is obtained by subtracting the error computed without randomization to the score obtained while randomizing the feature (Ishwaran et al., 2008). Hence, the importance score is the average of the feature importance scores across the trees (Ishwaran et al., 2008).

If DeepSurv or DeepHit results in being the best performing survival models, we will use the permutation importance score as previously done by Hathaway et al. (2021) for investigating DeepSurv most relevant predictors. To achieve this, we will adopt the permutation importance score provided by Python's library `eli5` (Korobov & Lopuhin, 2021). Specifically, permutation importance is retrieved by monitoring the decrease in the model predictive performance while shuffling the values of each variable at times in the test set: the more the reduction in the model performance while noising a feature, the more that variable is considered influential (Korobov & Lopuhin, 2021).

4 EXPERIMENTAL SETUP

4.1 *Data*

To predict voluntary employee attrition with survival analysis methods and investigate its prevalent causes, we used the "IBM HR Analytics Employee Attrition & Performance" dataset (IBM, 2017). We accessed it through Kaggle, where it was published in 2017.

The IBM data frame consists of a csv file of synthetic data, which holds 1470 rows and 35 columns. Specifically, the 35 features characterizing the workforce in the dataset provide information regarding employee's demographics (such as *MaritalStatus*, *Gender*, *DistanceFromHome*, or *Education*), employee's job position (as *MonthlyIncome*, *JobRole*, *PerformanceRating*, or *JobInvolvement*), and employee's gratification for her/his current workplace (for example, *Environmental*, *Job*, or *Relationship* satisfaction data). Moreover, the IBM dataset indicates whether the event of an employee leaving the company occurred (*Attrition*) and how many years passed since the employee joined the corporation (*Years At Company*), hence being suitable for time-to-event analysis.

Exploring the IBM dataset, we found that 237 employees out of 1470 left the company. Since 1233 employees did not experience attrition, the presence of right-censoring in the data was in the order of 1233 individuals. Further, we observed that, on average, employees were in the company for 7.01 years ($SD = 6.13$).

4.2 *Cleaning Process*

The IBM dataset contained neither missing values nor duplicates into its observations. However, we noticed that the covariates *Over18*, *StandardHours*, and *EmployeeCount* reported the same value per employee in the data frame. Indeed, each person in the company was over eighteen years old, every individual was working for forty hours a week, and each counted as one. Further, we observed that every employee had a unique associate id number through the covariate *EmployeeNumber*. Since staff members' ids and constant features did not provide insights towards the attrition event, we decided to remove the covariates *Over18*, *EmployeeCount*, *StandardHours*, and *EmployeeNumber* from the dataset.

When a variable in a data frame carries a valuable quantity of information about another independent feature, they are said to be multicollinear (Vatcheva, Lee, McCormick, & Rahbar, 2016). Multicollinearity needs to be addressed to allow the regression model to estimate coefficients in a trustable manner (Vatcheva et al., 2016). To do so, we measured the correla-

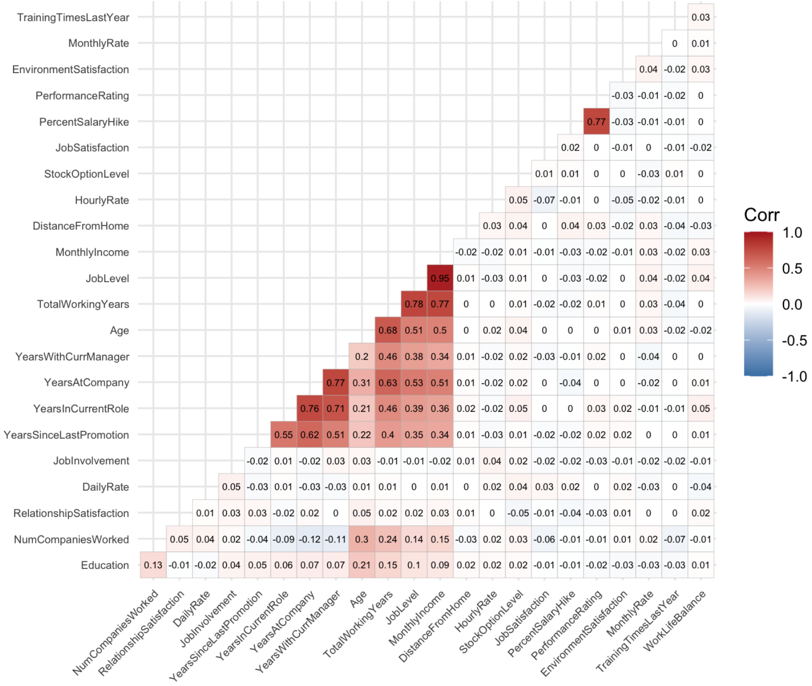

Figure 1: Pearson's correlations between numerical covariates

tions between the numerical features using the Pearson's pairwise method and plotting them through the the R `ggcorrplot` package (Kassambara, 2019) as reported in figure 1. Examining Pearson's correlations, we noticed that collinearity in our data was present with severe strength mainly among duration variables and *JobLevel* covariate. For *JobLevel*, we observed almost a perfect correlation with *MonthlyIncome* (r = 0.95) and a very strong collinearity with *TotalWorkingYears* (r = 0.78). Among duration features, we found that our time covariate *YearsAtCompany* was severe correlated both with *YearsWithCurrManager* (r = 0.77) and with *YearsInCurrentRole* (r = 0.76). Further, *YearsWithCurrManager* and *YearsInCurrentRole* suffered of a very strong collinearity (r = 0.71). Overall, other variables in the data frame such as *JobSatisfaction* or *JobInvolvement* did not show very strong correlations between them excepting for *PercentSalaryHike* and *PerformanceRating* (r = 0.77). To address multicollinearity, we did not include in our analysis the features *JobLevel*, *YearsInCurrentRole*, *YearsWithCurrManager*, and *PercentSalaryHike*. A new correlation plot between the numerical variables utilized in this project can be found in appendix A (page 34).

In our cleaning process, we also decided to check for multicollinearity in categorical data. For this purpose, we plotted with `ggcorrplot` a correlation matrix obtained using the Cramer's V method through the R's `creditmodel` package (Dongping, 2021), where perfect collinearity is indi-

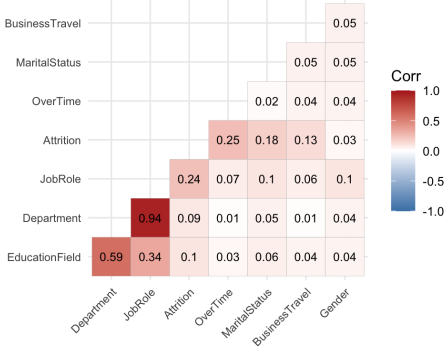

Figure 2: Cramer's V correlations between categorical covariates

cated with the value of 1 and its absence with a value of 0. In analyzing the plot reported in figure 2, we observed that the covariate *Department* was almost perfectly correlated with *JobRole* ($V = 0.94$). Indeed, they carry similar information as a *JobRole* depends on the department an employee is assigned to. Moreover, a strong Cramer's V correlation was noticed between *Department* and *EducationField* ($V = 0.59$). Other features did not present severe correlations between them. To also address collinearity in categorical data, we removed the covariate *Department* from our data frame. At the end of the data cleaning procedure, our dataset was reduced to 1470 rows and 26 columns.

4.3 *Dataset Visualization*

To better understand our dataset, we conducted an exploratory data analysis. The main findings of our visualizations are shown in figure 3, in particular:

- The majority of the employees who quit the company left before working ten years at the firm [a].
- Churner employees were more densely distributed at the beginning of their career [b].
- Employee attrition occurs with higher percentages among employees that have worked for more than four companies so far [c].
- One out of three employees who worked extra hours left the organization [d].

We obtained each graph using the R package `ggplot2` (Wickham, 2016).

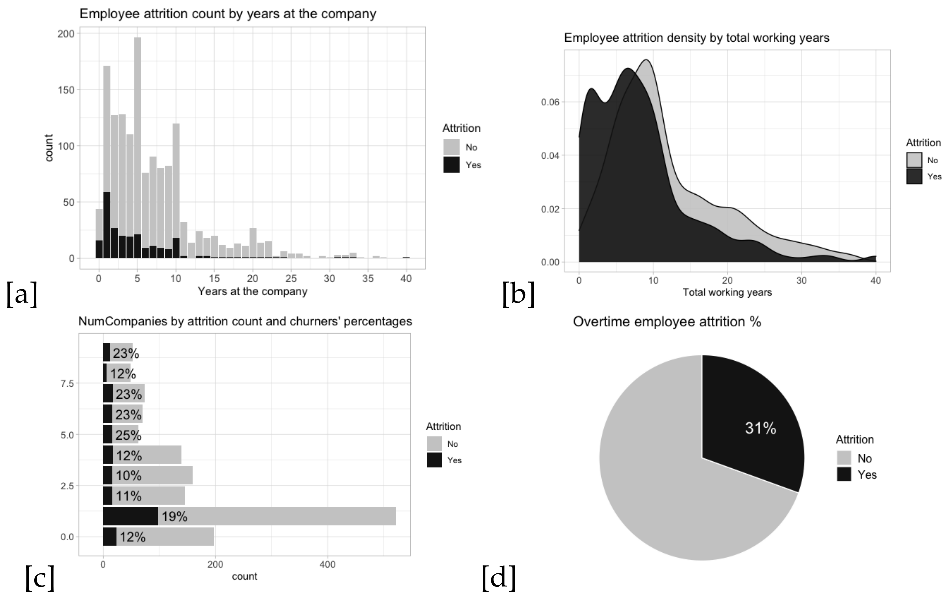

Figure 3: Key findings of the exploratory data analysis conducted

4.4 *Model building*

Before building any model, we one-hot encoded all the categorical variables present in our dataset. To do so, we utilized the *dummycols* function available through the R package `fastDummies` (Kaplan, 2020). We dropped the reference column in each dummy covariate to avoid multicollinearity. After this process, the dimensions of our data frame were in the order of 1470 rows and 39 columns. All the 39 covariates were used as input features to train our models.

The IBM dataset was then split into a train and test set, reserving 70% of the data for training and 30% for testing. We performed this process applying the function *createDataPartition* offered by the R package `caret` (Kuhn, 2021) to not alter the distribution of censoring across the data. Hence, we saved the train and test datasets into two separate csv files. They were imported in Python using `pandas` (The pandas development team, 2020).

The first model we built was Cox PH. Since our baseline did not necessitate hyperparameter tuning, we directly fitted it on the training data through the `lifelines` library (Davidson-Pilon, 2019). To interpret the model, we retrieved its summary statistics. Moreover, we checked if the assumption of the proportionality of the hazard held for the covariates used. We observed that it was violated by 4 out of 37 features as reported in Table 3. As a consequence of the proportionality hazards assumption violation, we expected a less accurate model fit as the hazard ratio of four covariates will change over time. However, since only a limited number

Table 3: Covariates violating the PH assumption with a p-value treshold of 0.05. The null-hypothesis is that there are no violations.

covariate	p-value
BusinessTravel_Travel_Rarely	0.04
JobSatisfaction	0.04
TotalWorkingYears	0.02
YearsSinceLastPromotion	<0.005

Table 4: RSF hyperparameter space

min_node_size	{8, 16, 24, 32}
max_features	{2, 4, 6, 8}
trees	{500, 1000}

of covariates violated the PH assumption, we might expect Cox PH to perform reasonably well for the current IBM employee dataset. Therefore, after checking the proportionality of the hazards assumption, Cox PH was tested on unseen data by computing both the Harrel's C-index and the integrated Brier score.

RSF, DeepSurv, and DeepHit required hyperparameter optimization. To test the best-performing combination of hyperparameters, we performed 5-folds cross-validation. Specifically, the event variable *Attrition* was stratified across all the folds to maintain censoring distribution in the dataset unaltered. For each model, we conducted a stratified 5-folds cross-validation for the same hyperparameter space twice. Indeed, the optimal combination of hyperparameters was selected monitoring their performance in terms of concordance index during the first and integrated Brier score during the second. Thus, we concluded the tuning process by selecting two different combinations of hyperparameters per model.

For RSF, hyperparameter optimization was performed using grid search due to the limited number of parameters to adjust. As reported in Table 4, we tuned the minimum amount of data to collect at the end of each leaf node (`min_node_size`), the maximum number of covariates to randomly consider at every node (`max_features`), and the trees' quantity that composes the forest (`trees`). The log-rank rule proposed by Segal (1988) was used as a criterion to split the nodes. We chose the values reported in Table 4 based on prior studies (Kantidakis et al., 2020; Kvamme et al., 2019), thus exploring 32 possible combinations of parameters. Once the best performing hyperparameters in terms of Antolini's concordance index were found using a stratified 5-folds cross-validation, we fitted a RSF on the complete training dataset (RSF$_c$). Further, another RSF was trained on the whole training set with the tuned hyperparameters found through a stratified

Table 5: DeepSurv hyperparameter space

`num_nodes`	{32}, {32, 32}, {38}, {32, 38}, {38, 38}, {38, 32}
`dropout`	$[0.1, 0.2, \ldots, 0.8]$
`batch_size`	$[32, 96, \ldots, 576]$
`epochs`	$[10, 15, \ldots, 120]$
`learning_rate`	LogSpace [0.0001, 0.4]
`optimizer`	{Adam, SGD}
`activation`	{ReLU, SELU}

5-folds cross-validation while monitoring the integrated Brier score (RSF$_i$). Both RSF$_c$ and RSF$_i$ were built using the Python library `pysurvival` (Fotso et al., 2019), and tested on unseen data.

For DeepSurv and DeepHit, we standardized the numerical covariates in the IBM dataset before tuning. Due to the large number of hyperparameters to optimize, a random search of 160 iterations was implemented. Both DeepSurv and DeepHit were built in Python using the `pycox` framework (Kvamme, 2018). Early-stopping and dropout were implemented for DeepSurv and DeepHit to avoid overfitting.

The values of the hyperparameters we utilized for tuning DeepSurv are reported in Table 5. To set up DeepSurv's hyperparameter space, we referred to the experiments conducted by Katzman et al. (2018). In particular, we followed Katzman et al. (2018) in testing a moderate number of layers (`num_nodes`), optimizers, and activation functions. To determine the nodes' values, we tried the default input suggested in `pycox` (32) and the total number of covariates we had in the dataset minus the event variable (38). Further, we researched the best performing hyperparameters using values of `batch_size`, `dropout`, `epochs`, and `learning_rate` that are commonly used for training neural networks. Similar to what we did for the RSF, DeepSurv's hyperparameters were tuned monitoring Harrel's C-index with a stratified 5-folds cross-validation and then used to fit a DeepSurv on the training set (DeepSurv$_c$). Moreover, another DeepSurv was trained with the hyperparameters selected through a stratified 5-folds cross-validation monitoring the integrated Brier score (DeepSurv$_i$). Either DeepSurv$_c$ and DeepSurv$_i$ were hence tested on testing data.

As mentioned above in 3.1.4., DeepHit requires discrete times as input. To treat our time variable *YearsAtCompany* discretely, we divided it into 34 equal distant intervals. We split *YearsAtCompany* into 34-time points as CoxPH and DeepSurv made predictions for 34 survival times. The hyperparameter space for DeepHit is reported in Table 6. To select the hyperparameters' values, we referred to prior studies (Kvamme et al., 2019; C. Lee et al., 2018; Nagpal, Li, & Dubrawski, 2021). For example, we implemented considerable `dropout` as suggested by C. Lee et al. (2018),

Table 6: DeepHit hyperparameter space

`num_nodes`	{2, 3, 156}, {32, 32}, {32}, {38, 38}
`dropout`	{0.6, 0.7}
`batch_size`	$[32, 96, \dots, 576]$
`epochs`	$[10, 30, \dots, 120]$
`learning_rate`	LOGSPACE [0.0001, 0.2]
α	{0, 1}
σ	{0.1, 0.25, 0.5, 1, 2.5}
`activation`	{ReLU, ELU}

ELU `activation` as done by Nagpal et al. (2021), α and σ as indicated by Kvamme et al. (2019). We chose values of `batch_size`, `epochs`, and `learning_rate` among those that are frequently used for tuning neural networks, and *Adam* as optimizer. Once the best performing hyperparameters in terms of Antolini C-index were selected through a stratified 5-folds cross-validation, we fitted DeepHit on the training set (DeepHit$_c$). Evaluating the same hyperparameter space in terms of IBS with a stratified 5-folds cross-validation, we fitted another DeepHit model on the training set (DeepHit$_i$). Both DeepHit$_c$ and DeepHit$_i$ were then tested on testing data.

The best combinations of hyperparameters selected per model are reported below in appendix B (page 34), where even the training loss for DeepSurv and DeepHit are displayed.

4.5 *Software implementation*

In order to conduct the experiments described above, `R` software (R Core Team, 2020) was utilized for cleaning, exploring, and splitting the IBM dataset. On the contrary, all the models and the visualizations of their results were implemented in `Python` (Van Rossum & Drake Jr, 1995).

5 RESULTS

This section will report the performances in predicting voluntary employee attrition obtained with the survival analysis methods described above in subsection 3.1. Further, it will present the most relevant predictors in forecasting whether an employee will leave an organization according to our best-performing model.

Table 7: Performances of the tuned models in terms of C-index and IBS. Subscript c indicates a model tuned with a stratified 5-folds cross-validation monitoring the C-index, whereas subscript i monitoring the IBS.

Models	C-index	IBS
Cox PH	0.904	0.060
RSF$_c$	0.908	0.162
RSF$_i$	0.909	0.148
DeepSurv$_c$	0.917	0.064
DeepSurv$_i$	0.917	**0.055**
DeepHit$_c$	0.918	0.108
DeepHit$_i$	**0.919**	0.112

5.1 *Predicting employee attrition*

The results of the implemented algorithms in predicting employee churn on IBM testing data are provided in Table 7. Overall, every survival machine learning algorithm we fitted outperformed the state-of-the-art Cox PH in relation to the concordance index. Indeed, either RSF, DeepSurv, and DeepHit showed a higher concordance score than our baseline (Cox PH = 0.904), regardless of whether their hyperparameters were tuned monitoring the C-index or the IBS. In particular, among the survival machine learning models applied, DeepHit resulted in the highest concordance score with a C-index equal to 0.919. DeepSurv occurred to have a concordance index equal to 0.917 and RSF to 0.909. The fact that DeepHit was the best performing survival algorithm in terms of concordance index on unseen data fits appropriately with the findings of previous studies (Kvamme et al., 2019; C. Lee et al., 2018).

Substantial improvements in concordance index were not observed by tuning the model's hyperparameters optimizing the C-index or the IBS. In fact, for example, the concordance score of a DeepHit trained with the cross-validated hyperparameters that maximized the C-index (DeepHit$_c$ = 0.919) was close to the concordance score of a DeepHit tuned while minimizing the IBS (DeepHit$_i$ = 0.918). Similarly, the concordance index of a RSF$_c$ (C-index = 0.908) was almost the same as the concordance score of a RSF$_i$ (C-index = 0.909). No difference was found between the concordance index of a DeepSurv$_c$ (0.917), and a DeepSurv$_i$ (0.917).

In terms of integrated Brier score, we observed that only DeepSurv outperformed our baseline Cox PH. Indeed, while Cox PH resulted in an integrated Brier score equal to 0.060, DeepSurv achieved a performance of 0.055. On the other hand, non-proportional hazards models performed

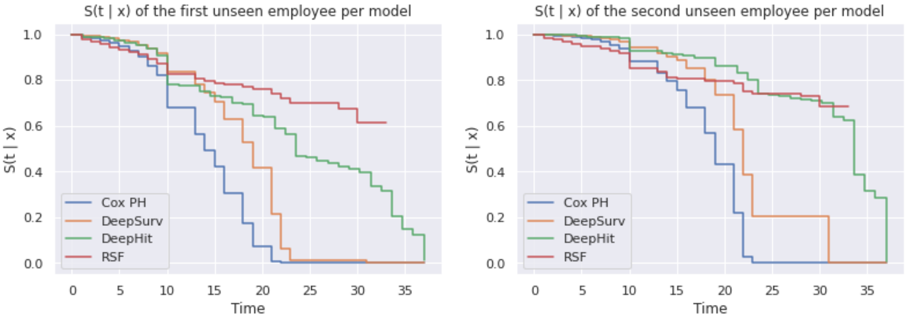

Figure 4: Survival curves of first and second unseen employee per model.

considerably worse than the Cox PH: the integrated Brier score for our best-tuned RSF was equal to 0.148, and for DeepHit to 0.108. A limited number of experiments in previous studies obtained similar results (Kvamme et al., 2019; Vale-Silva & Rohr, 2020) as frequently non-proportional hazards models showed lower integrated Brier score's values than proportional hazards algorithms (Kantidakis et al., 2020; Kvamme & Borgan, 2019; Kvamme et al., 2019; Vale-Silva & Rohr, 2020).

As the results presented in Table 7 show, improvements in terms of integrated Brier score were observed by tuning the model's hyperparameters monitoring the IBS or the concordance index. In fact, the integrated Brier score of a DeepHit optimized by maximizing the C-index trough cross-validation (DeepHit$_c$ = 0.108) was lower than a DeepHit tuned by minimizing the integrated Brier score (DeepHit$_i$ = 0.112). On the other hand, a DeepSurv tuned by minimizing the integrated Brier score (DeepSurv$_i$ = 0.055) resulted in a better integrated Brier score than a DeepSurv optimized by maximizing the C-index (DeepSurv$_c$ = 0.064) as well as RSF$_i$ (IBS = 0.148) had a better performance than RSF$_c$ (IBS = 0.162).

To further compare the fitted models, we plotted the predicted survival curves of the first two employees in the test set, as shown in the left and right panels of figure 4. Specifically, Cox PH and the best-tuned version of every model (RSF$_i$, DeepSurv$_i$, and DeepHit$_c$) were used to forecast the survival probabilities of each employee over time. For the first unseen employee plotted on the left of figure 4, we observed that every model predicted her/his survival curve concordantly until a ten-year window. After ten working years, the proportional hazards models diverged from the non-proportional hazards models in predicting employee churn. A similar pattern was found for the second unseen employee on the right plot of figure 4, where proportional hazards models were concordant with non-proportional hazards models until approximately a twelve-year window and diverged afterward.

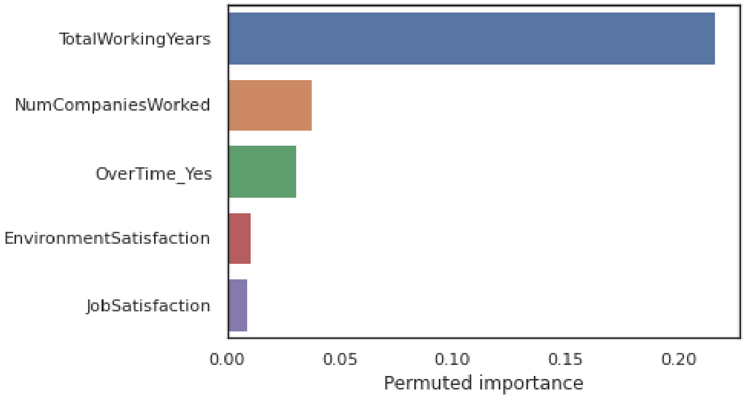

Figure 5: Top five factors influencing voluntary employee attrition according to the best performing model slected DeepSurv$_i$. They were retrieved using permutation importance.

5.2 *Most relevant predictors for predicting employee churn*

As previously discussed, DeepHit achieved the best score on employee data regarding the concordance index, while DeepSurv was the best-performing model in terms of integrated Brier score. Since we observed that DeepSurv resulted in a concordance index close to the best value obtained by DeepHit, whereas DeepHit was far from the best integrated Brier score, we chose DeepSurv$_i$ to investigate the main causes driving employee attrition. Figure 5 presents the most relevant factors influencing the workforce's deliberate turnover retrieved through the permutation importance of the features on the IBM test set. According to DeepSurv$_i$, we observed that the focal predictors of employee churn in the company were *TotalWorkingYears* (permutation importance: 0.216), *NumCompaniesWorked* (permutation importance: 0.038), and *OverTime* (permutation importance: 0.031). Further, results visualized in figure 5 indicate that *EnvironmentSatisfaction* (permutation importance: 0.010) and *JobSatisfaction* (permutation importance: 0.009) played a moderate role in predicting voluntary employee attrition.

6 DISCUSSION

Several machine learning algorithms have recently been extended to perform survival analysis (E. Lee, 2019; P. Wang et al., 2019). Nonetheless, prior studies on employee attrition only used traditional survival analysis methods to investigate why an employee will leave a job (Assefa et al., 2017; Li, Ge, Zhu, Xiong, & Zhao, 2017; P. Wang et al., 2019). Here, we compared the performances of state-of-the-art Cox PH with the newly developed survival algorithms RSF, DeepSurv, and DeepHit in forecasting deliberate workforce turnover. Furthermore, we utilized our best-performing model to investigate the most relevant factors influencing employee churn. As such, this thesis contributes to the deliberate turnover literature by exploring the potential of novel survival methods in predicting whether an employee will quit an organization.

The concordance index and the integrated Brier score were applied to analyze the predictive capabilities of Cox PH and novel survival methods on employee data. Our findings indicate that all the survival models implemented resulted in a better concordance score than state-of-the-art Cox PH, consistently with prior studies (Kvamme & Borgan, 2019; Kvamme et al., 2019; C. Lee et al., 2018). However, it is interesting to notice that only DeepSurv achieved a better result concerning the integrated Brier score than Cox PH. In particular, this is not in line with the majority of the experiments reported by previous papers (Kantidakis et al., 2020; Kvamme & Borgan, 2019; Kvamme et al., 2019; Vale-Silva & Rohr, 2020), where novel survival analysis methods resulted in a better integrated Brier score than Cox PH. Only a limited number of experiments in previous studies (Kvamme & Borgan, 2019; Vale-Silva & Rohr, 2020) reported that DeepSurv resulted in a better integrated Brier score than other methods, where Kvamme and Borgan (2019) argued that the proportionality of the hazards played a role in DeepSurv's performance. We also suspect that our results diverge from the majority of the literature due to the proportionality of the hazards assumption. Indeed, our findings show that non-proportional hazards models performed considerably worse in terms of integrated Brier score than proportional-hazards models. We argue that this could be the case because only four covariates out of thirty-seven violated the hazards assumption of proportionality in the synthetic IBM dataset used (see table 1). Moreover, the capability of DeepSurv in modeling the proportionality of the hazards in a non-linear manner could motivate its better performance in terms of integrated Brier score than Cox PH.

This thesis also aimed at selecting the best performing survival model to investigate the most relevant predictors in predicting voluntary employee attrition. Once we identified DeepSurv as the best performing model, the most influential factors for employee churn were retrieved using permutation importance. Our results show that the number of years since an employee has begun to work, the number of companies for which she/he had worked, and overtime are the focal factors that influence attrition. These findings are in line with previous papers (Alduayj & Rajpoot, 2018; Jin et al., 2020), where they also resulted in being among the most relevant predictors of deliberate workforce turnover. However, our results are in contrast with other studies conducted on the IBM data frame (Fallucchi, Coladangelo, Giuliano, & William De Luca, 2020; Yang, Ravikumar, & Shi, 2020), where other variables as *MonthlyIncome* and *Age* were indicated as having the most substantial influence on employee churn. We suspect that the difference in results relies on the methodology used to retrieve the most influential predictors. For example, Yang et al. (2020) obtained discordant predictors while interpreting the coefficients of logistic regression or the variable importance score given by a random forest. Future studies may implement different methodologies to retrieve the most influential predictors of employee churn with more reliable estimates.

From the societal perspective, our findings may positively impact both the employer and the employees. Indeed, suppose it is observed that an employee is at a high risk of churn in the next time window by interpreting the survival probabilities output by our best-performing model DeepSurv. In that case, the employer could take proactive measures to avoid employee attrition using the most relevant predictors of employee churn we found. For instance, if the employee at risk is working overtime, the employer may proactively reduce her/his extra working hours spent at work. Consequently, the employee may be more satisfied with her/his job, hence deciding not to quit the company. As such, data-driven proactive measures based on our results may save attrition costs, thus producing a positive impact on a firm as pointed out by previous studies (Alduayj & Rajpoot, 2018; Căplescu et al., 2019; Madariaga et al., 2018). Moreover, our findings may have a positive effect even on staff members' lives. In fact, an employee working overtime may be satisfied with spending fewer extra hours at work, hence choosing not to stress herself/himself with the research of another job. In this way, the results presented in this project may have positive societal implications.

We argue that a limitation of this study concerns the strategy implemented for handling multicollinearity between variables. Indeed, prior studies (Belsley, Kuh, & Welsch, 2005; Tomaschek, Hendrix, & Baayen, 2018) indicated that correlation matrices could not guarantee that there isn't any collinearity issue even though a high correlation between variables is not observed in the matrix (Tomaschek et al., 2018). Since we used correlation matrices not only for diagnostics, but also for decision-making, the Cox PH implemented could have suffered from our choices. Further researches on voluntary employee attrition could implement more advanced techniques to avoid multicollinearity, such as the principal component analysis (**?**) strategy for numerical variables.

Another limitation of this thesis regards the usage of permutation importance as the only method used for exploring the most relevant predictors of employee attrition. Indeed, previous studies (Molnar, 2020) indicated that permutation importance could produce inconsistent results and suffer from correlations between variables. Further studies could implement different techniques such as Olden's algorithm (Olden, Joy, & Death, 2004) to estimate the relevance of the features through the network's weights and compare its findings with the results given by permutation importance. Through this comparison, more reliable estimates of the most relevant factors influencing employee attrition could be obtained.

7 CONCLUSION

The objective of this study was to investigate how well voluntary employee attrition can be predicted using survival analysis. The reported results indicate that survival analysis methods can effectively predict deliberate workforce turnover. In particular, our findings reveal that novel machine learning models extended to survival analysis outperformed state-of-the-art Cox PH in terms of concordance index in forecasting employee attrition. Among them, DeepSurv had a better performance than Cox PH with respect to the integrated Brier score. Moreover, we observed that the number of years since the employee has begun her/his career, the number of companies in which an employee worked, and extra working hours are the most relevant predictors in predicting the churn.

By comparing recently proposed survival methods with state-of-the-art Cox PH, this project enclosed a novel methodology to predict deliberate attrition in a company. Since satisfactory results were obtained through this comparison, future studies could further explore the potential of survival analysis in forecasting employee attrition by comparing other novel machine learning techniques extended to survival tasks. In addition, it would be interesting for future studies to investigate models' performance differences with respect to the integrated Brier score on employee data where most of the covariates violate the proportional hazards assumption.

REFERENCES

Abner, E. L., Charnigo, R. J., & Kryscio, R. J. (2013). Markov chains and semi-markov models in time-to-event analysis. *Journal of biometrics & biostatistics*(e001), 19522. https://doi.org/10.4172/2155-6180.S1-e001

Alba, A. C., Agoritsas, T., Walsh, M., Hanna, S., Iorio, A., Devereaux, P., ... Guyatt, G. (2017). Discrimination and calibration of clinical prediction models: users' guides to the medical literature. *Jama*, *318*(14), 1377–1384. https://doi.org/10.1001/jama.2017.12126

Alduayj, S. S., & Rajpoot, K. (2018). Predicting employee attrition using machine learning. In *2018 international conference on innovations in information technology (iit)* (p. 93-98). IEEE.

https://doi.org/10.1109/INNOVATIONS.2018.8605976

Al Mamun, C. A., & Hasan, M. N. (2017). Factors affecting employee turnover and sound retention strategies in business organization: A conceptual view. *Problems and Perspectives in Management*, *15*(1), 63–71. https://doi.org/10.21511/ppm.15(1).2017.06

Antolini, L., Boracchi, P., & Biganzoli, E. (2005). A time-dependent discrimination index for survival data. *Statistics in medicine*, *24*(24), 3927–3944. https://doi.org/10.1002/sim.2427

Assefa, T., Mariam, D. H., Mekonnen, W., & Derbew, M. (2017). Survival analysis to measure turnover of the medical education workforce in ethiopia. *Human resources for health*, *15*(1), 1–11. https://doi.org/10.1186/s12960-017-0197-0

Belsley, D. A., Kuh, E., & Welsch, R. E. (2005). *Regression diagnostics: Identifying influential data and sources of collinearity* (Vol. 571). Hoboken, New Jersey: John Wiley & Sons.

Cai, X., Shang, J., Jin, Z., Liu, F., Qiang, B., Xie, W., & Zhao, L. (2020). Dbge: employee turnover prediction based on dynamic bipartite graph embedding. *IEEE Access*, *8*, 10390–10402. https://doi.org/10.1109/ACCESS.2020.2965544

Căplescu, R.-D., Ilie, M., & Strat, V. A. (2019). Voluntary employee attrition. descriptive and predictive analysis. *Proceedings of the International Conference on Applied Statistics*, *1*(1), 145–161. https://doi.org/10.2478/icas-2019-0013

Cox, D. R. (1972). Regression models and life-tables. *Journal of the Royal Statistical Society: Series B (Methodological)*, *34*(2), 187–220. Retrieved from `http://www.jstor.org/stable/2985181`.

Davidson-Pilon, C. e. (2019). *lifelines.* Retrieved from `https://doi.org/10.5281/ZENODO.2638135` (version 0.21.0)

Dongping, F. (2021). *creditmodel: Toolkit for credit modeling, analysis and visualization.* Retrieved from `https://CRAN.R-project.org/package=creditmodel` (version 1.3.0)

Emadi, S. M., & Staats, B. R. (2020). A structural estimation approach to study agent attrition. *Management Science*, *66*(9), 4071–4095. https://doi.org/10.1287/mnsc.2019.3401

Fallucchi, F., Coladangelo, M., Giuliano, R., & William De Luca, E. (2020). Predicting employee attrition using machine learning techniques. *Computers*, *9*(4), 86. https://doi.org/10.3390/computers9040086

Fang, M., Su, J., Liu, J., Long, Y., He, R., & Wang, T. (2018). A model to predict employee turnover rate: observing a case study of chinese enterprises. *IEEE Systems, Man, and Cybernetics Magazine*, *4*(4), 38–48. https://doi.org/10.1109/MSMC.2018.2834829

Fotso, S., et al. (2019). *PySurvival: Open source package for survival analysis* *modeling.* Retrieved from https://www.pysurvival.io/ (version 0.1.2)

Graf, E., Schmoor, C., Sauerbrei, W., & Schumacher, M. (1999). Assessment and comparison of prognostic classification schemes for survival data. *Statistics in medicine*, *18*(17-18), 2529–2545. https://doi.org/10.1002/(sici)1097-0258(19990915/30)18:17/18<2529::aid-sim274>3.0.co;2-5

Haider, H., Hoehn, B., Davis, S., & Greiner, R. (2020). Effective ways to build and evaluate individual survival distributions. *Journal of Machine Learning Research*, *21*(85), 1–63. Retrieved from https://www.jmlr.org/papers/volume21/18-772/18-772.pdf

Harrell, F. E., Califf, R. M., Pryor, D. B., Lee, K. L., & Rosati, R. A. (1982). Evaluating the yield of medical tests. *Jama*, *247*(18), 2543–2546. https://doi.org/10.1001/jama.1982.03320430047030

Harrell Jr, F. E., Lee, K. L., & Mark, D. B. (1996). Multivariable prognostic models: issues in developing models, evaluating assumptions and adequacy, and measuring and reducing errors. *Statistics in medicine*, *15*(4), 361–387. https://doi.org/10.1002/(SICI)1097-0258(19960229)15:4<361::AID-SIM168>3.0.CO;2-4

Hathaway, Q. A., Yanamala, N., Budoff, M. J., Sengupta, P. P., & Zeb, I. (2021). Deep neural survival networks for cardiovascular risk prediction: The multi-ethnic study of atherosclerosis (mesa). *medRxiv*. https://doi.org/10.1101/2021.04.12.21255286

IBM. (2017). *Ibm hr analytics employee attrition & performance.* Retrieved from https://www.kaggle.com/pavansubhasht/ibm-hr-analytics-attrition-dataset/home

In, J., & Lee, D. K. (2018). Survival analysis: Part i – analysis of time-to-event. *Korean journal of anesthesiology*, *71*(3), 182. https://doi.org/10.4097/kja.d.18.00067

Ishwaran, H., Kogalur, U. B., Blackstone, E. H., Lauer, M. S., et al. (2008). Random survival forests. *Annals of Applied Statistics*, *2*(3), 841–860. https://doi.org/10.1214/08-AOAS169

Jin, Z., Shang, J., Zhu, Q., Ling, C., Xie, W., & Qiang, B. (2020). Rfrsf: Employee turnover prediction based on random forests and survival analysis. In *International conference on web information systems engineering – wise 2020* (pp. 503–515). Springer. https://doi.org/10.1007/978-3-030-62008-0_35

Kantidakis, G., Putter, H., Lancia, C., de Boer, J., Braat, A. E., & Fiocco, M. (2020). Survival prediction models since liver transplantation-comparisons between cox models and machine learning techniques. *BMC medical research methodology*, *20*(1), 1–14. https://doi.org/10.1186/s12874-020-01153-1

Kaplan, J. (2020). *fastdummies: Fast creation of dummy (binary) columns and rows from categorical variables.* Retrieved from https://CRAN.R-project.org/package=fastDummies (version 1.6.3)

Kassambara, A. (2019). *ggcorrplot: Visualization of a correlation matrix using 'ggplot2'.* Retrieved from https://CRAN.R-project.org/package=ggcorrplot (version 0.1.3)

Kattan, M. W., & Gerds, T. A. (2018). The index of prediction accuracy: an intuitive measure useful for evaluating risk prediction models. *Diagnostic and prognostic research*, *2*(1), 1–7. https://doi.org/10.1186/s41512-018-0029-2

Katzman, J. L., Shaham, U., Cloninger, A., Bates, J., Jiang, T., & Kluger, Y. (2018). Deepsurv: personalized treatment recommender system using a cox proportional hazards deep neural network. *BMC medical research methodology*, *18*(1), 1–12. https://doi.org/10.1186/s12874-018-0482-1

Kleinbaum, D. G., & Klein, M. (2012). *Survival analysis* (3rd ed.). New York, NY: Springer.

Korobov, M., & Lopuhin, K. (2021). *Eli5.* Retrieved from https://pypi.org/project/eli5/ (version 0.11.0)

Kuhn, M. (2021). *caret: Classification and regression training.* Retrieved from https://CRAN.R-project.org/package=caret (version 6.0-88)

Kvamme, H. (2018). *Pycox: Time-to-event prediction with pytorch.* Retrieved from https://pypi.org/project/pycox/ (version 0.2.2)

Kvamme, H., & Borgan, Ø. (2019). Continuous and discrete-time survival prediction with neural networks. *arXiv preprint arXiv:1910.06724*. Retrieved from https://arxiv.org/abs/1910.06724

Kvamme, H., Borgan, Ø., & Scheel, I. (2019). Time-to-event prediction with neural networks and cox regression. *arXiv preprint arXiv:1907.00825*. Retrieved from https://arxiv.org/abs/1907.00825

Lee, C., Zame, W., Yoon, J., & van der Schaar, M. (2018). Deephit: A deep learning approach to survival analysis with competing risks. *Proceedings of the AAAI Conference on Artificial Intelligence*, *32*(1). Retrieved from https://ojs.aaai.org/index.php/AAAI/article/view/11842

Lee, E. (2019). Why newly graduated nurses in south korea leave their first job in a short time? a survival analysis. *Human resources for health*, *17*(1), 1–9. https://doi.org/10.1186/s12960-019-0397-x

Li, H., Ge, Y., Zhu, H., Xiong, H., & Zhao, H. (2017). Prospecting the career development of talents: A survival analysis perspective. In *Proceedings of the 23rd acm sigkdd international conference on knowledge discovery and data mining* (pp. 917–925). https://doi.org/10.1145/3097983.3098107

Liu, J., Long, Y., Fang, M., He, R., Wang, T., & Chen, G. (2018). Analyzing employee turnover based on job skills. In *Pro-*

ceedings of the international conference on data processing and applications (pp. 16–21). Association for Computing Machinery. https://doi.org/10.1145/3224207.3224209

Longato, E., Vettoretti, M., & Di Camillo, B. (2020). A practical perspective on the concordance index for the evaluation and selection of prognostic time-to-event models. *Journal of Biomedical Informatics*, *108*, 103496. https://doi.org/10.1016/j.jbi.2020.103496

Madariaga, R., Oller, R., & Martori, J. C. (2018). Discrete choice and survival models in employee turnover analysis. *Employee Relations*, *40*(2), 381–395. https://doi.org/10.1108/ER-03-2017-0058

McCloy, R. A., Purl, J. D., & Banjanovic, E. S. (2019). Turnover modeling and event history analysis. *Industrial and Organizational Psychology*, *12*(3), 320–325. https://doi.org/10.1017/iop.2019.56

Molnar, C. (2020). *Interpretable machine learning*. Retrieved from `https://christophm.github.io/interpretable-ml-book/`

Moore, D. F. (2016). *Applied survival analysis using r*. Springer. https://doi.org/10.1007/978-3-319-31245-3

Nagpal, C., Li, X. R., & Dubrawski, A. (2021). Deep survival machines: Fully parametric survival regression and representation learning for censored data with competing risks. *IEEE Journal of Biomedical and Health Informatics*. https://doi.org/10.1109/jbhi.2021.3052441

Nakagawa, T., Ishida, M., Naito, J., Nagai, A., Yamaguchi, S., Onoda, K., & Initiative, A. D. N. (2020). Prediction of conversion to alzheimer's disease using deep survival analysis of mri images. *Brain communications*, *2*(1), fcaa057. https://doi.org/10.1093/braincomms/fcaa057

Olden, J. D., Joy, M. K., & Death, R. G. (2004). An accurate comparison of methods for quantifying variable importance in artificial neural networks using simulated data. *Ecological modelling*, *178*(3-4), 389–397. https://doi.org/10.1016/j.ecolmodel.2004.03.013

Pawley, M. (2020). *Deepweibull: a deep learning approach to parametric survival analysis* (Master's Thesis, Imperial College London, London, United Kingdom). Retrieved from `https://people.bath.ac.uk/mtp34/Projects/Imperial/MSciThesis.pdf`

R Core Team. (2020). *R: A language and environment for statistical computing*. Vienna, Austria. Retrieved from `https://www.R-project.org/`

Roblin, E., Cournede, P.-H., & Michiels, S. (2020). On the use of neural networks with censored time-to-event data. In *International symposium on mathematical and computational oncology* (pp. 56–67). Springer. https://doi.org/10.1007/978-3-030-64511-3_6

Rombaut, E., & Guerry, M.-A. (2018). Predicting voluntary turnover through human resources database analysis. *Management Research Review*, *41*(1), 96–112. https://doi.org/10.1108/MRR-04-2017-0098

Segal, M. R. (1988). Regression trees for censored data. *Biometrics*, *44*(1), 35–47. https://doi.org/10.2307/2531894

Silva, F., Vieira, J., Pimenta, A., & Teixeira, J. (2018). Duration of low-wage employment: a study based on a survival model. *International Journal of Social Economics*, *45*(2), 286–299. https://doi.org/10.1108/IJSE-11-2016-0332

Sumathi, K., Balakrishnan, D., Naveen, V., Hariharan, P., & Rahul Iniyan, M. (2021). Talent flow employee analysis based turnover prediction on survival analysis. *Annals of the Romanian Society for Cell Biology*, *25*(4), 3844–3857. Retrieved from https://annalsofrscb.ro/index.php/journal/article/view/2934

The pandas development team. (2020). *pandas-dev/pandas: Pandas*. Zenodo. (version 1.2.5) https://doi.org/10.5281/zenodo.3509134

Tomaschek, F., Hendrix, P., & Baayen, R. H. (2018). Strategies for addressing collinearity in multivariate linguistic data. *Journal of Phonetics*, *71*, 249–267. https://doi.org/10.1016/j.wocn.2018.09.004

Vale-Silva, L. A., & Rohr, K. (2020). Multisurv: Long-term cancer survival prediction using multimodal deep learning. *medRxiv*. https://doi.org/10.1101/2020.08.06.20169698

Van Rossum, G., & Drake Jr, F. L. (1995). *Python tutorial*. Centrum voor Wiskunde en Informatica Amsterdam, The Netherlands.

Vatcheva, K. P., Lee, M., McCormick, J. B., & Rahbar, M. H. (2016). Multicollinearity in regression analyses conducted in epidemiologic studies. *Epidemiology*, *6*(2). https://doi.org/10.4172/2161-1165.1000227

Vickers, A. J., & Cronin, A. M. (2010). Everything you always wanted to know about evaluating prediction models (but were too afraid to ask). *Urology*, *76*(6), 1298–1301. https://doi.org/10.1016/j.urology.2010.06.019

Wang, P., Li, Y., & Reddy, C. K. (2019). Machine learning for survival analysis: A survey. *ACM Computing Surveys (CSUR)*, *51*(6), 1–36. https://doi.org/10.1145/3214306

Wang, W. (2019). Using survival analysis in human resource management research: Staff retention. *SAGE Publications Ltd*. https://doi.org/10.4135/9781526495570

Wickham, H. (2016). *ggplot2: Elegant graphics for data analysis*. Springer-Verlag New York. Retrieved from https://ggplot2.tidyverse.org

Willett, J. B., & Singer, J. D. (1991). From whether to when: New methods for studying student dropout and teacher attrition. *Review of educational research*, *61*(4), 407–450. https://doi.org/10.2307/1170572

Yang, S., Ravikumar, P., & Shi, T. (2020). Ibm employee attrition analysis. *arXiv preprint arXiv:2012.01286*. Retrieved from https://arxiv.org/abs/2012.01286v2

Zhao, L., & Hu, X. J. (2013). Estimation with right-censored observations under a semi-markov model. *Canadian Journal of Statistics*, *41*(2), 237–256. https://doi.org/10.1002/cjs.11176

8 APPENDIX A

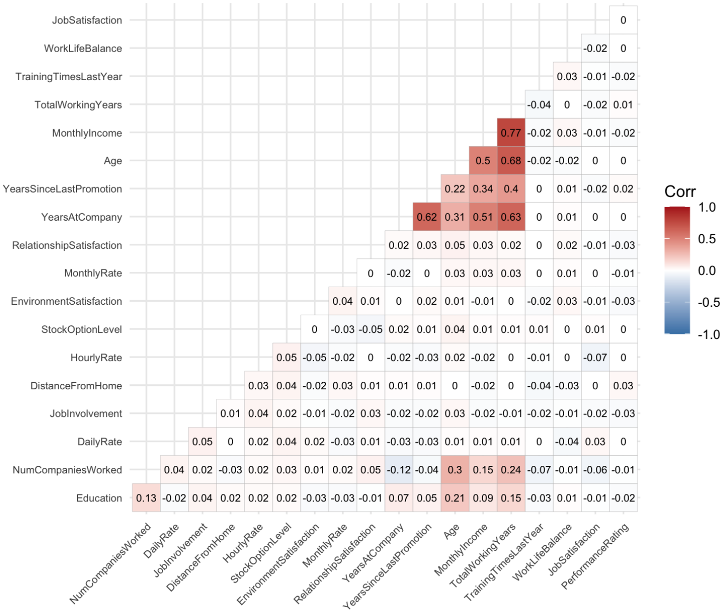

Figure 6: Correlations of the numeric variables used

9 APPENDIX B

Table 8: RSF$_c$ tuned

`min_node_size`	8
`max_features`	2
`trees`	1000

Table 9: RSF$_i$ tuned

`min_node_size`	32
`max_features`	6
`trees`	500

Table 10: DeepSurv$_c$ tuned

num_nodes	[38, 38]
dropout	0.6
batch_size	224
epochs	25
learning_rate	0.018076592746497238
activation	SELU
optimizer	Adam

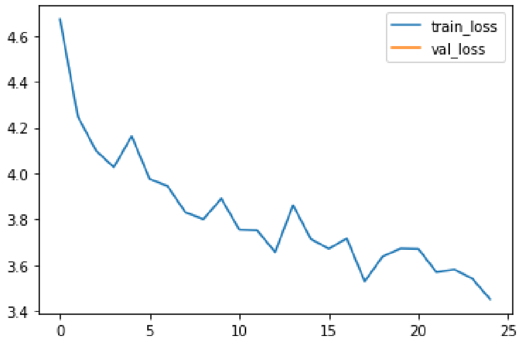

Figure 7: DeepSurv$_c$ training loss

Table 11: DeepSurv$_i$ tuned

num_nodes	[32]
dropout	0.3
batch_size	224
epochs	105
learning_rate	0.045597250222444564
activation	SELU
optimizer	SGD

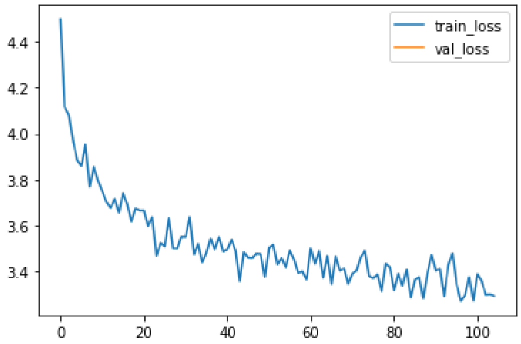

Figure 8: DeepSurv$_i$ training loss

Table 12: DeepHit$_c$ tuned

`num_nodes`	[38, 38]
`dropout`	0.7
`batch_size`	96
`epochs`	30
`learning_rate`	0.023579057190571906
α	1
σ	0.25
`activation`	ReLU

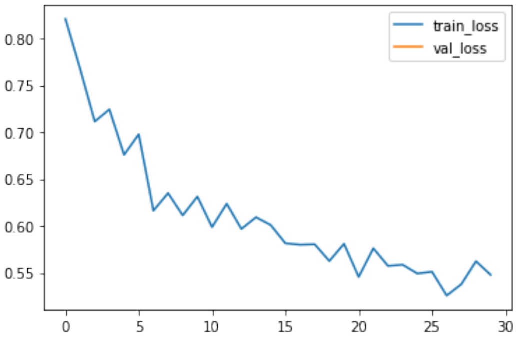

Figure 9: DeepHit$_c$ training loss

Table 13: DeepHit$_i$ tuned

`num_nodes`	[32, 32]
`dropout`	0.6
`batch_size`	32
`epochs`	110
`learning_rate`	0.0035446036055156544
α	1
σ	0.5
`activation`	ELU

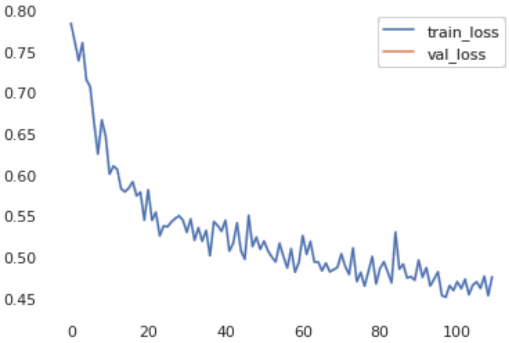

Figure 10: DeepHit$_i$ training loss