Dynamic updating of survival prediction models in a pandemic setting

Stark, Claudine

Universiteit Leiden
The Netherlands

Leids Universitair Medisch Centrum

Dynamic updating of survival prediction models in a pandemic setting

C.M.V. Stark

Thesis advisor: N. van Geloven, PhD
Second thesis advisor: I. Prosepe, MSc

Defence on August 1st, 2024

MASTER THESIS
STATISTICS AND DATA SCIENCE
UNIVERSITEIT LEIDEN

Foreword

The data used in this thesis comes from patients admitted to four Dutch hospitals with COVID-19. The data contain personal information about each patient and is not publicly available. All figures and results in this thesis do not contain any information that can be directly linked to an individual patient.

Abstract

In fast changing environments prediction models can perform less well over time. With dynamic updating methods one repeatedly updates a prediction model over time when new data becomes available to prevent this loss of performance. Aspects such as the updating frequency and the amount of old data included in each update (also referred to as sliding window) can influence the performance when updating a survival prediction model. In this thesis we investigate the effect of these two aspects on three updating methods for survival outcome; refitting, recalibration of the intercept, and Bayesian dynamic updating. Data was used from hospitalized patients from the first two COVID-19 waves, since this dataset gives a good example of a rapidly changing environment.

The results showed that no single updating method outperformed the others for all the different combinations of updating frequency and sliding windows, and that using one of the dynamic updating methods always outperformed no-updating. Both with refitting and recalibration the performance of the updated models improved when more old data was included (updating less frequent and longer sliding windows) but too much old data can also decrease the performance. The Bayesian method always used old data through the priors, so also adding sliding windows does not have any advantages. The advantage of using a sliding window instead of priors to incorporate old data, is that some events that are lost, through censoring when constructing the updating periods, can be recovered.

Overall, when comparing the results of the three methods, we saw that even though the refitting method requires the most old data, this method gave for most of the updating frequencies the best performance.

Table of content

Chapter 1

Introduction

In the medical setting, prediction models are used to predict diagnosis and prognosis of a patient.[1] A prediction model that targets an outcome that occurs later in time is called a prognostic model, such models are often studied with survival models.[2] Prognostic models are important for counseling of patients and for planning staff at hospitals.[1] Getting a good prediction model is therefore very important.

Prediction models rely on existing data to make accurate predictions of a certain event happening in the future. Due to changes in population characteristics and/or patient mix shifts initially good models can perform less well over time.[1] The COVID-19 pandemic was a fast changing environment with many new treatment strategies and shifts in hospital policy. Smit et al., 2022[3] showed that models developed in the beginning of the pandemic performed worse in later parts of the pandemic possibly due to frequently updated treatment guidelines and changes in the hospitalized population.

1.1 Motivation

Loss of performance of a prediction model can be addressed by retraining the prediction model on new, more recent, data.[4] However, sufficient sample size is required and the collection of new data can take some time. While awaiting the collection of new data, the prediction model loses its accuracy and no good predictions can be made. Also, when refitting the model on new data only, the model can loose potential information about the initial model.[5]

Dynamic updating methods can be a solution to this loss of accuracy problem. The idea behind these methods is to update a prediction model repeatedly over time when new data becomes available.[6] Several updating methods exist, such as recalibration of the intercept, refitting, and Bayesian updating.[6,7] Some studies that investigated different updating methods, using a logistic regression model, suggested that the performance of the methods depend on the sample size of the data used in the update, event rate and other model related characteristics.[7,8] A previous study by Schnellinger et al., 2021[7] investigated the effect of different frequencies of updating (so how often to perform an update) and of the inclusion of old data in each update (also referred to as sliding window), when updating a logistic regression model. Their conclusion was that increasing the amount of old data included and/or the length of the update period lead to improved performance metrics for all updating strategies, but to delay updates for too long or to implement the updates with too much old data resulted in inaccurate model calibration.

Most research until now has focused on looking at the effect of dynamic updating on logistic regression models. The effect certain choices can have on the performance of updating methods, such as how much old data to include while updating, has not been studied broadly for time-to-event data.[6] One study by Tanner et al., 2023 [6] looked into updating proportional hazard survival model. Tanner et al. compared the performance of three different dynamic updating methods for proportional hazard survival models in a simulation study as well as in a real data application on COVID-19 mortality in the general population with data from general practices in the UK during

the COVID-19 pandemic.[6] No single updating method emerged as superior across the different scenarios presented in the simulation study. The results from the real data application were inconclusive, with models that were never updated outperforming dynamic updating strategies in some occasions. Tanner et al. hypothesize that these results may be explained by how the alternation of high and low prevalence periods affects the targeted outcome in the study population. They also hypothesise that making prediction about COVID-19 related mortality in patients with a positive COVID-19 test would be less affected to the waves of the pandemic than when predicting for the whole population. This gives reason to further investigate the different aspects that influence dynamic updating methods when applied to survival models.

1.2 Aim

The aim of this project is to find out what influence the updating frequency and the amount of old data that is used in the updates have on the performance of updating methods, when applied to survival prediction models. We will investigate this using data from hospitalized COVID-19 patients from the first two COVID-19 waves, since this dataset gives a good example of a rapidly changing environment. As mentioned before by Tanner et al.,[6] it can be expected that when using a dataset like this one, with only patients admitted to the hospital with a positive COVID-19 test, the performed updates will be effected to a smaller extent by the underlying incidence of the pandemic. Our dataset is much smaller than the one used by Tanner et al. which will cause some different aspects, such as sample size, to have more influence on the analysis choices.

We will study three different updating methods. For each updating method, we will vary the updating frequency and the amount of old data included in each update. This last factor has not been varied in the study by Tanner et al., 2023.[6] We reason that since we study a setting with a smaller sample size, the inclusion of old data in an update could be needed to create a larger dataset for that update. Using the prediction accuracy over all the updates, for each of the methods, we will investigate the influences of these two aspects. We will assess performance by means of discrimination, calibration, and overall performance of the updated prediction models. For comparison, we will also look at the performance when no-updating is applied, even though we hypothesize that using dynamic updating methods will result in better performance.

The goal is to create insights into the effects the updating frequency and the amount of old data used in updates can have on the different updating methods in different scenarios and when each method would perform best, in the context of updating survival models. This would be useful for future situations of rapidly changing environments.

1.3 Structure of the thesis

We start with giving some descriptives about the dataset, in Chapter 2. After this follows Chapter 3 where some background about survival prediction models is given; theory on survival models will be discussed to give background knowledge about the methods that will be used later on. This chapter also gives an explanation about the different performance measures that will be used. In Chapter 4, the concept and notation of dynamic model updating will be introduced and a general explanation will be given about the different updating methods that will be implemented later on. This chapter also includes an explanation of the two aspects (updating frequency and inclusion of old data) that we will vary in our analyses, and how they might influence the performance of updated models. In the following chapter, Chapter 5, we will discuss the set up of our analysis. And in Chapter 6 the results from the analysis will be presented. Finally, Chapter 7 includes a discussion of the project. Additional information about the data, model and results can be found in the Appendix, as well as a link to the R code repository.

Chapter 2

Data descriptives

In this chapter we give some descriptives of the dataset that will be used in this thesis project. This dataset has not been used before in any project. First some more information is given about the patient selection of the dataset. Then we will have a closer look at the variables presented in the data. We motivate and explain the predictors that we will use to build the prediction model, that we will later dynamically update. Also the chosen outcome of interest will be discussed together with some descriptives of the time-to-event and censoring times. Lastly, we will discuss the missing data in the dataset and how this will be handled.

2.1 Data format

The data that is used in this thesis project comes from electronic health care records from hospitalized COVID-19 patients from different hospitals in the Netherlands (LUMC, HAGA, HMC and Alrijne) between March 2020 and November 2021. This data consist of detailed information of patient characteristics, comorbidities, medication use and lab results from the patients at admission time. The data also includes whether each patient was admitted to the ICU or died, and at which time this happened.

2.2 Patient selection

The initial dataset consisted of 4565 patients of at least age 18 who were admitted to the hospital with COVID-19. All patients were admitted between March 2020 and November 2021. We excluded patients that were transferred from a different hospital, because baseline measurements at first admission are not available for these patients. The final dataset contained 3596 individuals, see Figure 2.1. In Figure 2.2 the amount of patients with COVID-19 admitted to a hospital each month is shown.

2.3 Variables

We used only a part of the variables from the dataset for a prediction model. We selected predictors based on the $4C$ mortality score, a prediction model developed to estimate the risk of mortality in hospitalized COVID-19 patients.[9] We chose a composite outcome of ICU admission or death for our model. An overview and explanation of the variables in our final dataset is given in Table 2.1. In Appendix A.1 more information can be found about how the data was preprocessed.

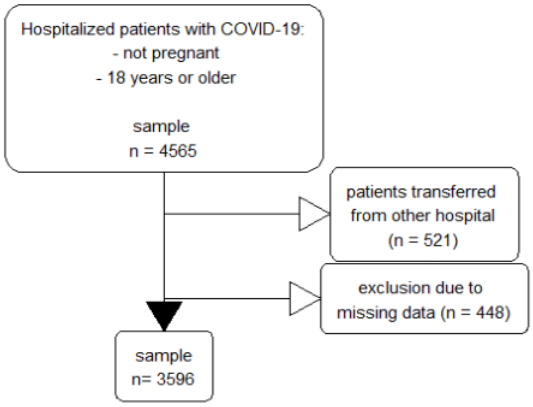

Figure 2.1: Flowchart of patient selection and exclusion together with sample size.

Table 2.1: Variables in the dataset of patients with COVID-19 admitted to the hospital.

Type of variable	Variable name	Description
descriptive	pseudo id	patient id
	admitted date	date of admission
	hospital	hospital of admission
baseline covariate	age	age at admission
	sex	female or male
	numb comorb	number of comorbidities
	RR...min	respiratory rate (per minute)
	CRP..mg.L	c-reactive protein (mg/l)
	SpO2...	peripheral oxygen saturations (percentage)
	Creatinine.SER..µmol.L.	creatinine (µmol/l)
outcome	discharge date	date of discharge
	date ICU	date of ICU admission (if admitted to ICU)
	died date	date of death (if died)
	discharge destination	destination after discharge
	status	indicator of the event (death or ICU admission)
	survtime	time from hospital admission until event or censoring

2.3.1 Predictors

Our chosen predictors are similar to the predictors of the 4C mortality score. The predictors from the 4C model are age, sex, number of comorbidities, respiratory rate, peripheral oxygen saturation, Glasgow coma score, urea and C-reactive protein.[9] Not all these variables are available in our dataset, we ended up using the following 7 predictors: age, sex, number of comorbidities, respiratory rate, peripheral oxygen saturation, creatinine and C-reactive protein. Urea is available in the data, but has more than 30% missing data, so, based on clinical expertise, we decided to use creatinine as predictor instead. The number of comorbidities variable indicates how many comorbidities each patients has. Its values are 0, 1 and 2 or more. The comorbidities that are integrated in this score are chronic cardiac disease, chronic respiratory disease (excluding astma), astma, chronic lever disease, diabetes, malignancy, chronic kidney disease, chronic neurological disease and hypertension, again based on similar procedure from the 4C model and the comorbidities provided in our dataset as well as the ones that were advised to also include by clinical expertise.

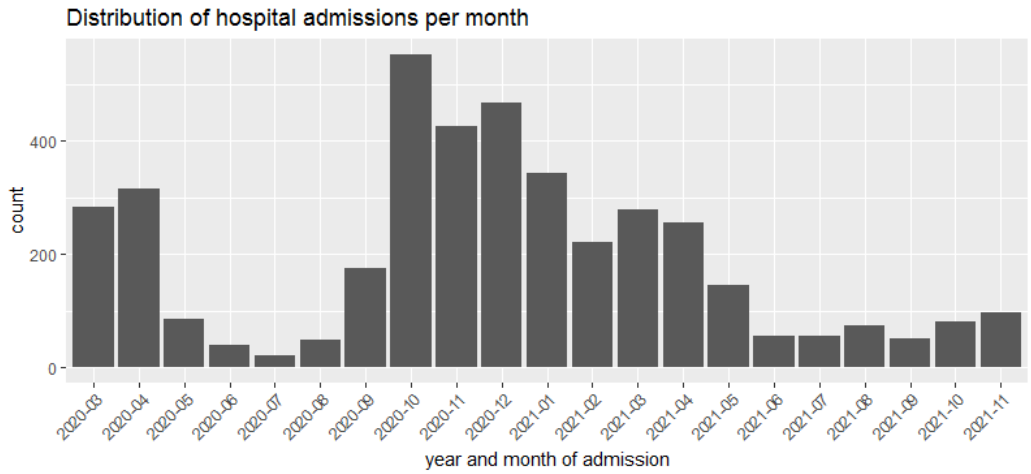

Figure 2.2: Number of admissions of patients with COVID-19 to a hospital, between March 2020 and November 2021.

All the predictors are measured at time of admission of each individual.

Summary descriptives of these covariates can be found in Table 2.2, the overall summary based on the whole dataset is given as well as separate summaries for each of the waves. In Table 2.2 we can see that more males were admitted to the hospital than females, and also the largest part of the patients have 0 comorbidities. In Appendix A.2 additional tables and figures of the data can be found.

2.3.2 Outcome

Our outcome of interest is the 28-day composite outcome of ICU admission or death. We consider patients who are transferred to hospice as death, reasoning that patients are transferred there because they can not be helped anymore in the hospital and will die soon. Patients who are discharged to go home are assumed to have survived past 28 days. Patients who are discharged to a different hospital are censored upon discharge. Since we will be working with a 28-day prediction horizon, we censor at 28 days all patients with survival time longer than 28 days. This, and the fact that we assume that patients who are discharged to go home are assumed to have no event in the 28 days interval, explains the high peak around 28 days in the censoring time distribution in Figure 2.3. In Figure 2.4 we can see that most patients don't experience the event within the 28 days. We can also note that there are indeed two waves visible, one before and one after august 2020, as is also visible in Figure 2.2.

2.3.3 Missing values

The amount of missing data in each predictor is shown in Figure 2.5. All covariates seem to have less than 10% missing data. In the analysis we will only use the complete cases in the data, with respect to the predictors. This leads to the removal of 448 individuals, see Figure 2.1.

Table 2.2: Summary descriptives of covariates (mean value with standard deviation, or count in sample with percentage in total sample) in the whole dataset and for each of the wave separately.

Characteristic	**Overall**, N = 4,064[1]	**wave 1**, N = 723[1]	**wave 2**, N = 3,341[1]
Age	66 (15)	65 (16)	66 (15)
Sex			
Female	1,705 (42%)	285 (39%)	1,420 (43%)
Male	2,359 (58%)	438 (61%)	1,921 (57%)
Comorbidity count			
0	2,496 (62%)	415 (58%)	2,081 (62%)
1	690 (17%)	148 (21%)	542 (16%)
2 or more	861 (21%)	153 (21%)	708 (21%)
Unknown	17	7	10
Respiratory Rate (per min)	23 (8)	23 (7)	23 (8)
Unknown	240	69	171
Peripheral oxygen saturations (%)	93 (7)	94 (5)	93 (7)
Unknown	212	66	146
C-reactive protein (mg/l)	102 (88)	104 (91)	102 (87)
Unknown	299	77	222
Creatinine (µmol/l)	97 (51)	94 (47)	98 (52)
Unknown	273	70	203

[1] Mean (SD); n (%)

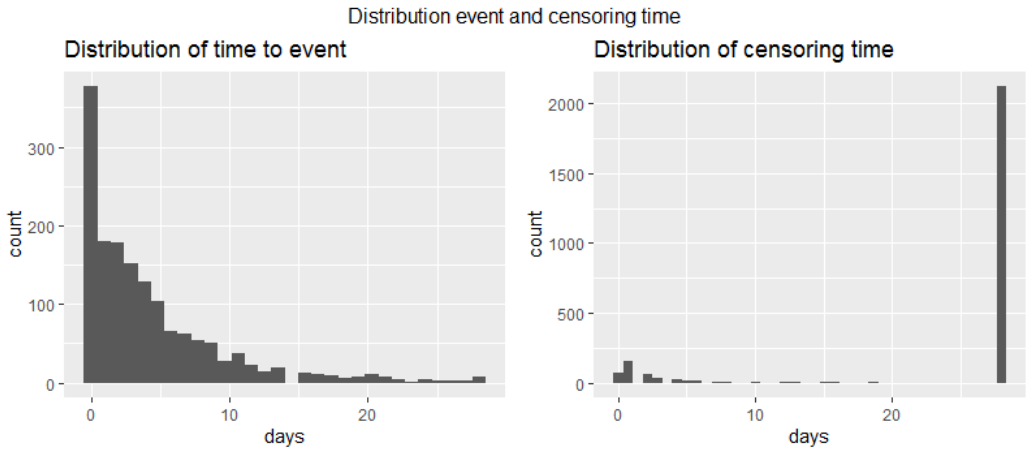

Figure 2.3: Frequencies of time to event and censoring time.

Figure 2.4: Display of total number of events that have taken place and distribution of events per admission month, between March 2020 and November 2021.

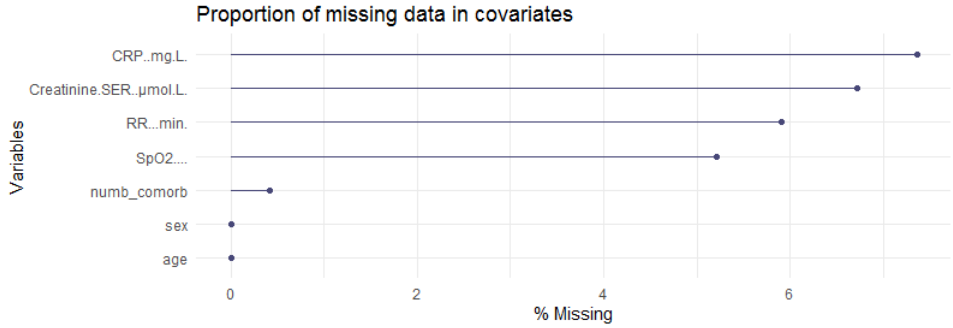

Figure 2.5: Proportion of missing data in each of the covariates respiratory rate per minute (RR...min), peripheral oxygen saturation percentage (SpO2...), c-reactive protein mg/l (CRP.mg.L.), creatinine µmol/l (Creatinine.SER.µmol.L), and age (at admission).

Chapter 3

Prediction in survival models

Prediction models are trained on data to predict an outcome. There are a lot of different models one can use. This thesis project uses data about hospitalized COVID-19 patients for which time until death or ICU admission is recorded. This type of data, known as time-to-event data, is commonly analyzed using survival analysis, which provides tools to account for censoring. A survival prediction model relies on the estimation methods provided by survival analysis. The accuracy of a survival prediction models on new data can be evaluated using several performance measures.

In this chapter we will look at the development and validation of survival models. We will start with an introduction to developing a prediction model with survival analysis. Next we will explain how to make predictions with these models. The second part of this chapter shows which measures can be used to assess the performance of a survival model.

3.1 Survival model development

In this section, we will provide a brief overview of the survival analysis theory that is needed for our analysis. First, we will explain some fundamental concepts. Then, we will build to one of the most used survival models; Cox proportional hazard model. Lastly, we show how to make predictions from this survival model. Theory and notations are based on Kleinbaum and Klein, 2012[10] and Klein and Moeschberger, 1997.[11]

3.1.1 Survival analysis

Survival analysis is used for the analysis of data for which the outcome variable is the time until a certain event occurs. This time is usually referred to as the survival time. An important issue in survival analysis is censoring, which occurs when the exact survival time of an individual or object is unknown. There can be various reasons for censoring, such as loss of follow up or the event not occurring before the end of the study; these are examples of right-censoring. In this project we will work exclusively with right-censored data.

Let T, a random variable, be the survival time. The cumulative distribution of the survival time is then defined as $F_T(t) = P(T \leq t)$ and the probability density function as $p_T(t) = \frac{d}{dt}F_T(t)$. The survival function, which is defined as the probability of an individual surviving to time t, can be written as

$$S(t) = P(T > t) = 1 - F_T(t), \tag{3.1}$$

where this last equality only holds if T is continuous. The hazard (rate) function, which is defined as the instantaneous potential per unit time for the event to occur given that the individual has

survived up to time t, can be written as

$$h(t) = \lim_{\Delta t \to 0} \frac{P(t \leq T \leq t + \Delta t | T \geq t)}{\Delta t} \geq 0. \tag{3.2}$$

This value is not a probability, but a rate since it represents a probability per unit time. From the hazard function, the cumulative hazard function can be obtained in the following way

$$H(t) = \int_0^t h(x)dx. \tag{3.3}$$

If T is continuous, there is a relationship between the survival and the (cumulative) hazard function, namely:

$$h(t) = \frac{1}{S(t)} \lim_{\Delta t \downarrow 0} \frac{S(t + \Delta t) - S(t)}{\Delta t} = -\frac{d \ln S(t)}{dt}, \tag{3.4}$$

which implies

$$H(t) = -\ln(S(t)) \tag{3.5}$$

and so

$$S(t) = \exp(-H(t)). \tag{3.6}$$

One can estimate the survival function using the non-parametric Kaplan-Meier estimator. For each time point t_i the number of events (d_i) and the number of individuals at risk (Y_i) are used to calculate the estimate of the conditional probability that an individual has survived up to time t_i and has the event at time t_i: $\frac{d_i}{Y_i}$. The survival function at time t is then estimated by the product-limit estimator:

$$\hat{S}(t) = \prod_{i:t_i \leq t} (1 - \frac{d_i}{Y_i}), \tag{3.7}$$

where $1 - \frac{d_i}{Y_i}$ denotes the estimate of the conditional probability of an individual to survive up to time t_i and then survive at time t. This estimation is based on the assumption that the censoring mechanism is non-informative, meaning the censoring time of an individual is not related to the survival of that individual if they had continued in the study. Using this estimation, one can also estimate the cumulative hazard function, following (3.6).

An alternative estimator for the cumulative hazard function is the Nelson-Aalen estimator. Using the same notation as for the Kaplen-Meier estimator, the Nelson-Aalen estimator is given by

$$\hat{H}(t) = \sum_{i:t_i \leq t} \frac{d_i}{Y_i}. \tag{3.8}$$

Again, one can also use this estimation to retrieve the estimate of the survival function, using (3.5). This estimator is also based on the assumption of non-informative censoring.

3.1.2 Survival predictions

To predict the survival probabilities of individuals we can use Cox proportional hazard model. This model uses patients characteristics to predict the survival probability, by modelling the hazard semi-parametrically. We will first look closer into this Cox proportional hazards model. Then, we will describe how to predict survival probabilities over time.

Cox proportional hazard model

The Cox proportional hazard model consists of two part. The first one is the baseline hazard, $h_0(t)$, which is non-parametric and an unspecified function by the model. The second part is the exponent of the linear sum over the predictors and their coefficients: $\exp(\beta^\top X)$. This part constitutes the parametric part of the model. The product of these two part forms Cox proportional hazard model:

$$h(t|X) = h_0(t)\exp(\beta^\top X) \tag{3.9}$$

with X a vector of the covariates, and β a vector of the associating coefficients. The combination of the non-parametric and parametric is the reason why this model is referred to as semi-parametric.

This model is build based on the assumptions of proportional hazards, linearity and independent censoring, which are discussed in Appendix B.

Predictions

Using a dataset, we can fit a Cox proportional hazards model and estimate the coefficients (β) and the baseline hazard ($h_0(t)$). The estimates of the coefficients ($\hat{\beta}$) are the maximum likelihood estimates. The estimated coefficients are usually reported in their hazard ratio form, namely $\exp(\hat{\beta}_i)$ for the coefficient β_i of covariate i. A covariate with hazard ratio larger than 1 implies higher probabilities to experience of interest for patients that have a higher value for that covariate.

The baseline hazard is estimated by first using Breslow's estimator to estimate the cumulative baseline hazard ($H_0(t)$), from which then the baseline hazard can be retrieved. The Breslow's estimator is given by

$$\hat{H}_0(t) = \sum_{i:t_i \leq t} \frac{d_i}{\sum_{j \in R(t_i)} \exp(\hat{\beta}^\top X_j)} \tag{3.10}$$

with $R(t_i)$ denoting the group of individuals at risk at time t_i. This estimator is similar to the Nelson-Aalen estimator (3.8), only here each individual j is reweighed based on the exponent of their linear predictor ($\hat{\beta}^\top X_j$).

Together, this results in

$$\hat{h}(t|X) = \hat{h}_0(t)\exp(\hat{\beta}^\top X). \tag{3.11}$$

Using the Cox proportional hazard model it is possible to make predictions on new data of survival probabilities. The estimated coefficients and cumulative baseline hazard from the fitted Cox model (3.11) lead to the predicted survival probabilities on new data in the following way

$$\hat{S}(t|X) = \exp\{-\hat{H}_0(t)\exp(\hat{\beta}^\top X)\}. \tag{3.12}$$

The predicted risk of event by time t can also be obtained as 1-$\hat{S}(t|X)$.

3.2 Survival model validation

The performance of a survival prediction model can be evaluated using measures of discrimination, calibration and overall performance. We will use these measures later on during the analysis. In this section a more in depth explanation will be given on the different measures. All the explanations relate to survival prediction models using time-to-event data with right-censoring.

3.2.1 Discrimination

Predictions made by a model should be able to make a distinction between individuals with and without the events. With time-to-event data this, more specifically, means that a model should be able to assign higher risk estimates to individuals that will experience the event earlier than other

individuals.[12] This is also called the discrimination ability of a model. In other words, how well the model predictions separates high- and low-risk individuals.[13]

Several metrics can be used to sum up the discrimination of a model. We will discuss the concordance index and AUCt.

Concordance index (c-index)

The first discrimination measure we discuss is the concordance index (c-index). The c-index looks at the ordering of the predictions for all pairs of individuals. These pairs are constructed in a way that every pair has at least one individual with the event and the other individual is not censored earlier than that event. The c-index is then calculated as the proportion of the pairs for which the individual with the event observed to have the highest estimated risk.[12] The calculated value can range from 0.5 to 1, where 0.5 corresponds to no discrimination ability (no better than a random ordering of the estimated risks) and 1 corresponds to perfect discrimination.[14] To adjust for the censored individuals, inverse probability of censoring weighting can be applied to estimate the c-index. With this method the case weights from censored individuals are assigned to individuals with longer follow-up time.[13] This reweighed version of the c-index is also called Uno's c-index.[15]

AUCt

Another way to measure the discrimination of a model is the AUCt (area under the time-dependent receiver operating characteristic curve). The AUCt is useful when one is interested in the ability of the model to predict the event occurring by one specific time point and not at the full range of follow-up.[12] Pairs of individuals are compared, which is also done for the calculation for the c-index, but here only the pairs consisting of one individual with the event before time t and the other individual not yet having the event before time t. The AUCt can be calculated at different times, but in our case we use only our time horizon of 28 days. Perfect discrimination corresponds to the value 1, lower values indicate less discrimination ability.[12] Again, inverse probability of censoring weighting is applied to account for censoring.

3.2.2 Calibration

In general we want the estimated outcome predicted by a model to align with the observed outcome. This alignment is referred to as the calibration ability of a model. For a survival prediction model, calibration describes how well the risk estimates agree with the observed outcome proportion.[12] Several metrics can be used to sum up the calibration of a model. We will discuss the calibration plot, calibration slope and intercept, and the observed-expected ratio.

Calibration plot

In a calibration plot the observed and estimated outcomes are compared among patients with the same estimated risk.[12] Individuals can be divided into groups of approximately the same size, based on their estimated outcome.[12] For each group, the proportion of observed outcomes (y-axis) is plotted against the estimated outcome (x-axis) at a specific time point, which for us is the time horizon of 28 days. In the presence of censoring, the proportion of observed outcomes in each group can be estimated by using the Kaplan-Meier estimator.[13] Alternatively, pseudo-observation may be used to get a proxy observed event indicator for all the individuals, including the censored ones. These pseudo-observations can be used to make a smoothed curve of the observed outcome proportions against the estimated risks.[12] Perfect calibration would be reflected in this plot as a diagonal line.

Calibration intercept and slope

It is possible to numerically summarise a calibration plot by calculating the calibration intercept and calibration slope.

The calibration slope captures whether the estimated risks are too extreme (slope smaller than 1) or too moderate (slope larger than 1).[16] A slope equal to 1 means that the predictors match the observed strength in the validation data.[12] The calibration slope can be obtained by applying a Cox proportional hazard model to the data with the linear predictor as only covariate.[17]

The calibration intercept captures whether there is any systematic under- or over-estimation.[13] It measures this by evaluating how close the estimated risk is to the overall observed outcome proportion, which is done by looking at their difference. If they are on average equal, the intercept will be 0. An intercept larger than 0 reflects under-estimation (predicted risks are too low on average), and an intercept smaller than 0 reflects over-estimation (the predicted risks are too high on average).[16] The calibration intercept is a measure of the so-called "calibration in the large" or the mean calibration,[13] as it compares the estimated risks and the observed outcome proportion in the general population.

Together, the calibration slope and intercept give a numeric summary of the calibration ability of a model. Both values need to be near their target value to reflect a good calibration of the model.

Observed-expected ratio

Another way to numerically summarize the "calibration in the large" of a model is by the observed-expected ratio (O/E), which compares the average estimated risk and observed outcome proportion, and assesses whether the predictions are systematically too high or too low.[18] It differs from the calibration intercept as it considers the ratio and not the difference between the overall observed outcome proportion and the average estimated risk at the chosen time point. The chosen time point will be our 28-days prediction horizon. The observed outcome can be estimated using the Kaplan-Meier estimator.[13]

A ratio of 1 reflects perfect calibration, a ratio lower than 1 means that on average the predictions are too high (over-estimation) and a ratio higher than 1 means that on average the predictions are too low (under-estimation).[12]

3.2.3 Overall performance

The overall performance is a combination of the discrimination and calibration of a model; it assesses the overall ability of the model to predict if an individual experiences the event by a particular time point.[12] This is assessed by looking at how far the predicted outcomes are from the observed outcomes.[1] A measure that captures this goodness of fit of the model is the (scaled) Brier score, we will discuss this measure here.

(Scaled) Brier score

The Brier score assesses how far the predicted outcomes are from the observed outcomes by averaging over the the squared distance between the event indicators and the predicted risks, both at time horizon.[12] A scaled version of the Brier score can be used to make the score easier to interpret and can be computed the following way: $1 -$ (model Brier score $\div$ null model Brier score), with null model referring to a model without covariates.[19] The scaled version ranges up to 1, with 1 corresponding to a perfect model.[12] The scaled Brier score can be interpreted as proportion of the variation explained (R-squared).[20] When the scaled Brier score is below 0, the model gives worse predictions than when using the null model.

Chapter 4

Dynamic updating methods

In this chapter, we describe methods for dynamic model updating. We explain and discuss the different updating methods that we investigate in this project, as well as, analysis choices influencing these methods. Specifically, we have a detailed look at use of old data in the updates (data that was already used in the previous update) and the length between updates.

4.1 Dynamic model updating

Dynamically updating a prediction model refers to the mechanism of repeatedly updating a prediction model over time with new data.[6] Using same notation as in Tanner et al.,[6] this updating process starts at period $u = 0$, where an original model M_0 is fitted on a development dataset D_0. This development data set contains data from time t_{-1} to time t_0 (referred to as period $u = 0$). After time t_0, new data becomes available in period $u = 1$ forming dataset D_1, which includes data from $(t_0, t_1]$. The time between the updates, also referred to as the length (or 1/frequency), will be discussed more in detail below (see 4.3.2). Predictions are made on the new dataset D_1 using model M_0 and the performance is measured. This measurement of performance is done using the performance assessment discussed in Chapter 3.2. After this, model M_0 is updated with the data from D_1, and the process repeats itself: for $u \geq 1$ new data is collected forming D_u, predictions are made on D_u using model M_{u-1} and the performance is measured. Subsequently, model M_{u-1} is updated with the data from D_u which then results in model M_u.[6] In Table 4.1, an illustration is given of the dynamical updating procedure.

Table 4.1: Illustration of dynamic updating with the data that will be used in this project.

Dataset	Wave 1	Wave 2		
	D_0	D_1	D_2	D_3
Original model	Fit model	Predict and measure performance using D_1 data		
Update 1		Update model with D_1 data	Predict and measure performance using D_2 data	
Update 2			Update model with D_2 data	Predict and measure performance using D_3 data

For the data that we will use, the first dataset D_0 corresponds to the data from the first wave and will be used to fit the original model. The data from the second wave will be used for the following datasets $D_1, D_2, \ldots$. The dataset D_i used for the i-th update contains both the covariates and

outcome from patients with COVID-19 at admission to the hospital. The precise construction of these datasets will be discussed in the next chapter.

4.2 Methods for model updating

We will now discuss the different updating methods that will be studied in this project. The explanation is specific for the type of model that will be used, namely a Cox proportional hazards model. In Chapter 3.1.2 we already discussed how one can estimate the survival probabilities from a Cox proportional hazards model.

4.2.1 Refitting

With the refitting method both the baseline hazard and the hazard ratios are re-estimated on the newly available data, discarding the previous estimates.[6] When updating model M_{u-1} to model M_u, covariates X and outcomes from data D_u are used to fit a Cox model:

$$h(t|X) = h_{0u}(t) \exp(\beta_u^\top X) \tag{4.1}$$

from which $h_{0u}(t)$ and β_u can be estimated. This updating method also allows to integrate new predictors into the model.[6]

With this updating method, in each updating step one can use either only new data or a combination of new and the most recent old data, for re-estimating the parameters of the model.[7] This last case is also referred to as using a sliding window. In the Section 4.3, we will provide a more in-depth explanation about this concept and its possible impact on the model performance.

Since the refitting method re-estimates all the parameters, a large sample size is required to obtain a reliable updated model. Compared to the next method, this can be seen as a disadvantage. Also, this updating method can be seen as the most extreme in the sense that it re-estimates all parameters.

4.2.2 Recalibration of intercept

The "recalibration of intercept" method only re-estimates the baseline hazard of the Cox proportional hazard model at each update, while the association between the predictors and the outcome is assumed to remain unchanged (estimated coefficients of the original model M_0 are untouched).

Recalibration of the intercept is also referred to as "calibration in the large", which is the concept that the mean observed outcome should be equal to the mean of the predicted outcomes.[1] Recalibrating the baseline hazard aims at aligning the mean observed outcomes and the mean predicted outcomes in the new dataset.[21] In practice, this means that the baseline hazard is re-estimated with the new data of the next interval the following way.[6]

1. First, the linear predictor ($\hat{\beta}_0 X$) is calculated based on the new data (X) and the estimated coefficients ($\hat{\beta}_0$) from the original model (M_0).

2. Secondly, the baseline hazard is re-estimated through the Breslow estimator using the outcomes from the new data (see Chapter 3.1.2 for more explanation about this estimator).

Updating model M_{u-1} to model M_u hence results in:

$$\hat{h}(t|X) = \hat{h}_{0u}(t) \exp(\hat{\beta}_0^\top X). \tag{4.2}$$

This updating method only influences the calibration of the updated model, not the discrimination, since the coefficients are kept constant throughout the updates.[6] The ordering of the individuals risk are not affected by this method, meaning that the distribution of the risks stay

the same. This results in the fact that the calibration slope is also not affected by this updating method.

An advantage of this method is that at each update only the baseline hazard is re-estimated, which makes that this method requires a smaller sample size than the previous method.[7] With this method there is again the option to use a combination of new and old data in each update (sliding window). A disadvantage of this method is that no new predictors can be added at the updates.[6]

4.2.3 Bayesian dynamic updating

With the previous two updating methods there was always the choice to incorporate old data into each update. Bayesian updating inherently includes previous knowledge, updating the model by combining information from the previous model with new data.[6] In each update, the estimated coefficients from the previous model are used as priors, which are then used during updating.[22] The posterior distribution is used to construct the updated survival estimates.[22]

As was done by Tanner et al.,[6] we apply the Bayesian updating technique of McCormick et al.,[22] on a proportional hazard model assuming exponentially distributed survival times. Updating model M_{u-1} to model M_u, we estimate the new parameters β_u and updated baseline hazard λ_u assuming the following:

$$\begin{aligned} T_i &\sim Exp(\omega_i) \\ \omega_i &= \lambda_u + \beta_u^\top X_i \\ \beta_u &\sim \mathcal{N}(\hat{\beta}_{u-1}, \hat{\Sigma}_{u-1}/\xi) \\ \lambda_u &\sim \mathcal{N}(0, \sigma_\lambda) \end{aligned} \tag{4.3}$$

with T_i representing the survival time of individual i and λ_u the log baseline hazard in period u ($\lambda_u = \log h_0$). Further, $\hat{\beta}_{u-1}$ is a vector of the estimated coefficients from the previous period $u-1$, with X_i the vector of covariates values of individual i and $\hat{\Sigma}_{u-1}$ the covariance matrix. And ξ is the "forgetting factor", assumed here smaller or equal to 1, which will be explained in more detail below. Equations (4.3) form the Bayesian updating model.

The posterior distribution of the updated estimate β_u is written as

$$p(\beta_u|T_u) \propto p(T_u|\beta_u)p(\beta_u|T_{u-1}) \tag{4.4}$$

with T_u the observed survival time in period u.[22] Equation (4.4) is the product of the likelihood at period u, denoted by $p(T_u|\beta_u)$, and the so-called prediction equation from (4.3), namely $\beta_u \sim \mathcal{N}(\hat{\beta}_{u-1}, \hat{\Sigma}_{u-1}/\xi)$. The updated model is estimated using Markov Chain Monte Carlo with the R package "rstan".[23] Convergence can be evaluated using the Rhat statistic (smaller than 1.1 for convergence).[24]

A first thing to notice about this updating method is that the baseline hazard is modeled parametrically.[6] This has not been an assumption so far with the other updating methods. Our assumption that the survival times are exponentially distributed ($T \sim Exp(\omega_i)$) implies that the baseline hazard is constant ($h_0(t) = \omega_i$).

Another important aspect of this updating method is the "forgetting factor" ξ. This forgetting factor reflects the uncertainty in the prior of β_u, meaning that for a smaller ξ the prior becomes less informative (so "wider").[6] This causes the used old data, in each update, to be down-weighted.[25] This describes the idea that the older data reflects the new data less accurately.

With this updating method old data is always included in each update. This is a large difference compared to the previous two updating method. The only choice we can make is how often to update, and thus how often the old data is down-weighted. Together with the forgetting factor this can have an influence on the performance of the updating method.

A disadvantage of this updating method is that the mechanism behind the estimation of the updated coefficients is more complicated, and could be less transparent to non-statisticians than the previous discussed methods.

4.3 Sliding window & length of update periods

Before evaluating the updating methods, there are two choices to be made, namely how much old data to include (if any) and how frequent to perform the updates. The amount of old data included in each update is referred to as a sliding window, and how frequent the updates are performed is expressed in terms of the length of the update periods.

With dynamic updating of a model we want to incorporate new data into this model. It is possible to update a model with only new data, but it might be useful to also include some old data. Including some old data in an update will make the updated model more stable since it is based on a larger sample size, but this is only when the new data does not reflect any sudden changes that are expected to continue into the next period.[7]

A previous study by Schnellinger et al., 2021[7] compared dynamic updating strategies on post-lung transplant data. They also investigated the influence of the amount of old data included in each update and the length of the update period on the performance. Their conclusion was that increasing the sliding window and/or the length of the update period led to improved performance metrics for all updating strategies, but to delay updates for too long or to implement the updates with too much old data resulted in inaccurate model calibration.[7] This implies that there probably exists a trade-off between the size of the sliding window and/or length of the update periods and the performance of the updated models.[7] Since they used a logistic regression model, there might be different effects from the use of sliding windows and/or the length of the periods when using a Cox proportional hazard model, which is what we will use.

In our case more choices, concerning the prediction horizon and the censoring mechanism, can have an influence on this trade-off. By cutting the data into periods, we suspect that more censoring is added to the data. Reasoning that this partitioning of the data leads to the loss of complete follow-up for several individuals. This makes the use of a survival model necessary, since it is capable of handling censored data, unlike a logistic regression model. In Chapter 5.1 we will explain in more detail how this additional censoring occurs.

We will discuss here in more detail what choices we need to make regarding the sliding window and the length of update periods, and the influences we expect these choices to have on the performance of the updated models.

4.3.1 Sliding windows

Updating on only new data may lead to an updated model that reflects the new environment more rapidly, but the updated model will probably be made on a small sample size.[6] It could also be the case that using only new data may cause the updated model to overfit, and will lead to poor performance if the new data reflects only part of the population or a momentary change.[6]

The choice of the amount of old data to include in each update also depends on the amount of available new data: if there is few new data available it can be useful to include more old data in each update, since this would lead to a more accurate updated model. But again this also depends on whether the new data reflects moderate or sudden changes. Also using too much old data may result in an inaccurate updated model.[7]

Thus, the choice of the sliding window seems to depend on the amount of new data available, the intensity of any changes that the new data reflect, and whether those changes are temporary or expected to stay.

4.3.2 Length update periods

The length between the updates can be defined by calendar time (eg. quarters of a year) or by number of events (by counting the observed events from the start until the wanted amount). When defining the length of the update periods one should take into account several aspects. Namely, each period should contain enough data to be able to perform an update.[7] Furthermore, the number of events during the period should be reasonably high enough, otherwise it will not be possible to update the model.[7] Also the time frame of the analysis plays a role in choosing the update length, reasoning that with a longer time frame one has more possibilities for the update length. The choice on the length of the periods can also alter for different clinical contexts. In clinical setting it might be inconvenient to have too many updates when the models needs to be approved or when resources are low.[7]

Using a short period, the updated model will faster reflect the changing environment, but less new data is used for each update. Using a long period means that more new data is included in each update, but when the new data reflects sudden changes it will take more time to transmit this onto the updated model, meaning that meanwhile inaccurate predictions might be made.

Thus when making the choice of the length of the periods one needs to take into account the amount of data available, the number of events and the intensity of the changes that the new data reflects.

Chapter 5

Analysis plan

In this chapter we describe the design of our numerical experiment. First we describe how the updating periods are defined. After that, an in depth description is given on how we apply the different methods and compare them. Finally, motivation is given about the different choices that are made regarding the analysis.

5.1 Updating periods

In this section we will discuss how we defined the periods that are used for the updating methods.

Each period has a start date and an end date, with the start date one day after the end date of the previous period. All individuals admitted to the hospital between the start date and the end date are included in that period. We based the length of each period on a chosen fixed number of events: for a period of k events, the end date of the period is when the k-th event is observed. Up to and including that date, individuals are included in the period. This means that each period only contains events of the individuals that were admitted to the hospital in that period.

We made the decision to base the length of the periods on a number of events reasoning that when updating a proportional hazard model a sufficient number of events should be present to create an accurate model. Knowing that the number of events are not uniformly distributed throughout the months (see Figure 2.4 in Chapter 2), defining the length of the periods by events gives more insurance on a stable model than basing the length on calendar time.

We included patients until the date of the k-th event and not until the admission date of individual with the k-th event, reasoning that otherwise individuals that were admitted after the individual with the k-th event and with their event before the date of the k-th event would have been excluded. This would lead to having less than the wanted k events in the periods, because the events take place in a different ordering than the admission of the individuals.

For a period of k events, it is possible that e.g. event $k+1$ and $k+2$ happen on the same date as event k (this can happen when the patients with the $k+1$-th an $k+2$-th event were admitted later than the patient with the k-th event). As all the individuals that are admitted to the hospital after the start of the period and before the date of the k-th event are included in the period, a period can include a few more events than the wanted k events.

Individuals included in the period that have their event later than the end date are censored at the end date, meaning their survival time is put to the date of the k-th event. This means that also individuals who are admitted at the date of the k-th event are added to the period but then also directly censored (their survival time is put to 0). This additional censoring to the data does not affect the independent censoring assumption of the Cox proportional hazard model.[26] This choice of inclusion criteria and censoring for each period leads to not using all the information provided in the data. Indeed, if a patient is included in a period, but has the event after the end date of that period, the event of the patient will not be recorded in any updating step, leading to a loss of events. It would be possible to reduce the loss of events by using sliding windows and incorporating

in the period $u+1$ some old information from period u. This would still potentially lead to some loss of individuals (whenever the sliding window is shorter than time horizon), but less. Using sliding windows in this context would focus more on preserving the lost events in between two periods than on incorporating old data in the updates.

Using sliding windows may cause some events to be included in adjoining periods, generating some overlap in information. This reflects the idea of the sliding window to include old data (previous events) in each period, which is our main focus in this experiment.

In Figure 5.1 an illustration is given of how the first period of 50 event is constructed.

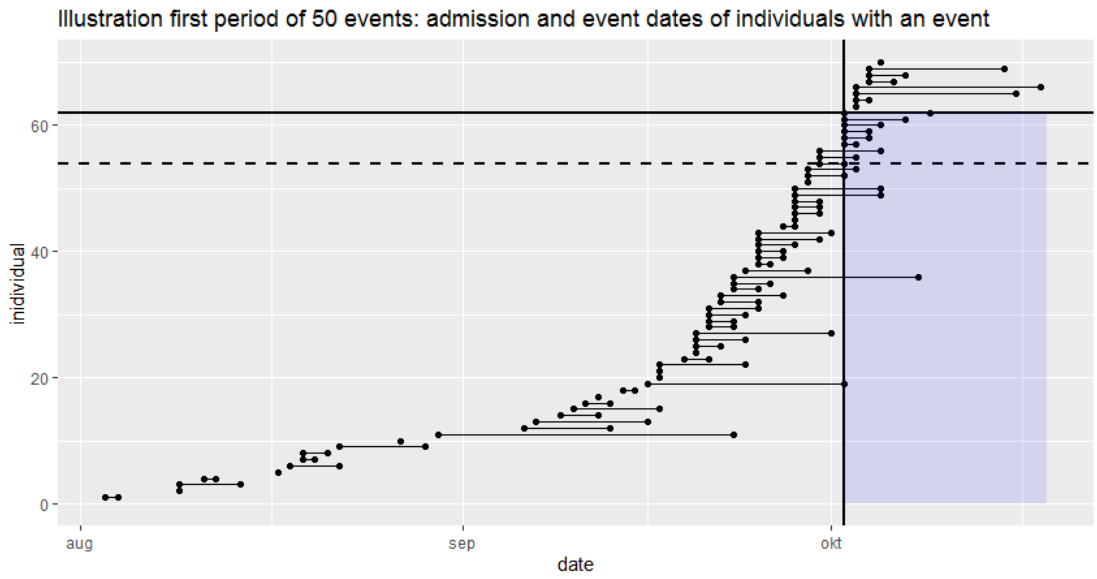

Figure 5.1: Illustration of the first period of 50 events. Each individual with an event is represented by a horizontal line which indicates the time from admission until the event. The individuals (y-axis) are ordered by their admission date (x-axis). The individual with the 50-th event is indicated by the black dashed horizontal line, the black vertical line indicates the date of the 50-th event. The first period consists then of all individuals below the black horizontal line; all individuals admitted before the date of the 50-th event. Individuals with their event in the blue area have their event time censored.

5.2 Methods comparison

First of all we will develop a model on the data from the first wave (D_0), this model is referred to as the original model (M_0).

The main goal of this analysis is to investigate the effect the sliding window and updating frequency have on the performance of updating methods. We will first vary the sliding windows and the length of the update periods for all methods separately to have a closer look at how each method is influenced by these two aspects. To investigate this, we will look at the performance of the updated models. At each update we will evaluate the updated model on the next period of 100 events (independent on the length of the period used for the update) using several performance measures (see Chapter 3.2 for more detailed explanation about these measures). We chose to fix the length of the period used for evaluation to be able to compare the performance between the methods with different updating periods.

To get a better idea of the performances across all the updating periods, we will provide summary measures of the predictive performances. For the Scaled Brier score, c-index and AUCt, we choose

as summary measure the average of each performance over all updates, meaning that higher values reflect better performance. For the OE-ratio and the calibration slope, we take the average over the absolute difference of each performance value with 1, over all updates. And for the calibration intercept, we average over the absolute value of each performance value, over all updates. So, for these last three measures, values closer to 0 reflect better performance. A good thing to note is that the calibration intercept and slope need to be interpreted together, meaning that both values need to be near there target value to reflect good performance.

We then want to compare the results between the updating methods. We will also compare the results to the performance of applying no-updating to the original model M_0, so without updating M_0 with new data.

5.3 Implementation choices

We will now discuss the other choices we made regarding the implementation of the analysis.

For each updating method we considered the lengths for the update periods and sliding windows shown in Table 5.1. With the chosen length of periods we aim to reflect different frequencies of updating, namely more often (every 50 events) or less often (every 150 events), or something in between (every 100 events). The sliding windows are chosen this way such that they represent the different scenarios that we want to take a closer look at. Namely, including only new data and no old data in the update (using no sliding window; 0 weeks), or using a combination of old and new data ($1, 2, 4$ weeks of old data included as sliding window). A sliding window of 4 weeks would also recover all the lost events in between two periods, as we mentioned in Section 5.1.

Table 5.1: Overview of lengths of updating periods and sliding windows that will be used in the method comparison.

Length of periods	Sliding windows
50 events	0 weeks
100 events	1 week
150 events	2 weeks
	4 weeks

The sliding windows are measured in calendar time (weeks) and not in events as the length of periods are, reasoning that the data from the updating periods are based on the ordering of the admissions of the individuals and not by the time of the events. The events have a different ordering than the admissions, this makes it complicated to select the old data based on a number of events. In Table 5.2 an overview is given of the average amount of events in each length of updating period when different lengths sliding windows are used. As mentioned before, we can see that with no sliding window the updating periods on average contain a few more events than the want amount. Additionally, in Appendix C more information can be found about the calendar dates of all the periods.

For the Bayesian updating method, some extra choices need to be made regarding the "forgetting factor" ξ and the prior of the log baseline hazard λ_u. Following Tanner et al.,[6] we set $\xi = 0.9$ and $\lambda_u \sim N(0, 2.5)$. Furthermore, we made the decision to also work with sliding windows, even though Bayesian updating already relies on the data of the previous period, the old data, through the prior. We made this choice reasoning that sliding windows also help with losing less patients with events in between the defined updating periods, we discussed this idea already in section 5.1. And we also reason that this way we can make a general comparison with the other two updating methods that do use sliding windows.

Table 5.2: Average number of events for each updating period when different sliding windows are used.

updating periods of 50 events	
sliding window	mean number of events
0 weeks	51.7
1 week	78.2
2 weeks	105
4 weeks	156.8
updating periods of 100 events	
sliding window	mean number of events
0 weeks	103.2
1 week	128.1
2 weeks	153.4
4 weeks	206.7
updating periods of 150 events	
sliding window	mean number of events
0 weeks	152.3
1 week	178.3
2 weeks	205.5
4 weeks	254.8

Chapter 6

Results

In this chapter results from the analyses are shown. First of all we will briefly discuss the model we fitted on the first wave, also referred to as M_0. From there, the results of each of the updating methods will be shown and explained. After this, the results of the different methods will be compared. We also compare the results to the no-updating method results.

6.1 Model of the first wave

In Chapter 2.3.1 we discussed the predictors we chose to develop a proportional hazard model using data from the first wave. This model makes predictions about the risk of ICU admission or death within 28 days after admission. In Table 6.1 the estimated coefficients and hazard ratios (HR) of this first model (M_0) are shown. And in Table 6.2 the performance of this model on the training data (data from the first wave; D_0) is shown for the different performance metrics. Performance is as expected with assessing performance on the training data.

Table 6.1: Coefficients and hazard ratio (HR) of fitted proportional hazard model on wave 1 (n=621, 284 events) together with 95% confidence intervals and p-value. Note that the variables have been centered and scaled, see Table A.1 for more infromation.

Variable	Coefficient (95% CI)	HR (95% CI)	p-value
Age (at admission)	$0.37\ (0.23, 0.51)$	$1.45\ (1.26, 1.67)$	< 0.001
Sex (Male)	$0.02\ (-0.23, 0.27)$	$1.02\ (0.80, 1.31)$	0.86
Number of comorbidities	$-0.15\ (-0.30, 0.00)$	$0.86\ (0.74, 1.00)$	0.05
Respiratory rate (per minute)	$0.20\ (0.07, 0.33)$	$1.22\ (1.07, 1.39)$	0.003
Pheripheral oxygen saturation (percentage)	$-0.23\ (-0.38, -0.09)$	$0.79\ (0.69, 0.91)$	0.001
C-reactive protein (mg/l)	$0.22\ (0.11, 0.33)$	$1.24\ (1.11, 1.39)$	< 0.001
Creatinine (µmol/l)	$0.22\ (0.11, 0.33)$	$1.24\ (1.10, 1.39)$	< 0.001

6.2 Results per method

In this section the results from the refitting, recalibration of the intercept and Bayesian dynamical updating method are shown. Figures with the performance of the updated models in each period for each updating method can be found in Appendix D.1. The summary measures of the predictive performances, for each of the different lengths of updating periods and sliding windows, are for each method thoroughly discussed below and can be seen in Tables 6.3, 6.4 and 6.5. The highlighted values indicate the best performance for each performance metric. The performance of the no-updating method is also shown.

Table 6.2: Performance of the original model (M_0) on the training data (D_0). For the scaled Brier, c-index and AUCt higher values reflect better performance. For the OE-ratio and the calibration slope best performance is achieved at 1, and for the calibration intercept at 0.

Performance metric	Value (95% CI)
Scaled Brier	$0.160\ (0.202, 0.119)$
c-index	$0.687\ (0.654, 0.778)$
AUCt	$0.739\ (0.700, 0.778)$
OE-ratio	$1.024\ (0.911, 1.150)$
Calibration intercept	$0.061\ (-0.062, 0.184)$
Calibration slope	$1.097\ (0.857, 1.337)$

In Appendix D.2, a visual overview can be found of how the coefficients of the updated models change throughout the updates, for both the refitting and Bayesian dynamic updating method. In Appendix D.3 we have a closer look whether the updating methods also influence the individual predictions.

6.2.1 Refitting

From the summary measures in Table 6.3 we see the following. For shorter updating periods (updating more frequently), such as periods of 50 events, the use of a longer sliding window (4 weeks) led to better performance for all performance measures of the updated models. The longer the updating periods (updating less frequently) the less the length of the sliding window seemed to influence the performance, meaning that there was no specific sliding window that resulted in the best performance for all the measures.

For the overall and discrimination performance measures the performance increased when comparing the updating periods length of 50 to 100 events, but decreased when comparing the performance between periods of 100 to 150 events. Refitting is an updating method that requires enough data and events to perform an update. The results that we saw here are thus not unexpected; the updates where more data was included, so longer sliding windows and/or longer updating periods, performed best overall. But too much data (events) did not guarantee best performance; updating every 100 events with the use of a 4 week sliding window gave better performance than when updating every 150 events with a 4 week sliding window, which was 206.7 events on average compared to 254.8 events (see Table 5.2). Reasoning that the data we used reflects a rapidly changing environment, waiting too long for new data caused this new data to reflect a longer period and will thus also include information that already got older over time, which then resulted in an unstable model. This was reflected by the decrease in performance for the overall performance and discrimination between updating periods of 100 and 150 events. For the same reason, including a sliding window when already using longer updating periods did not give much improvement in the performance of this updating method. Updating with periods of 100 events and a sliding window of 4 weeks resulted in best performance compared to the other two lengths of updating periods and all lengths of sliding windows, based on the overall performance.

The no-updating method performed better for the discrimination measures when updating every 50 events. In all the other cases, the refitting methods outperformed the no-updating method. To conclude, the refitting method performed overall best when updates were made not too frequently and a long sliding window was used. Including more data improved the performance of this method, but when this data became too old it reduced the performance.

Table 6.3: Summary measures of the performance measures for the different lengths of updating periods for the varying sliding windows over all updates. Also showing the average performance of the no-updating method for comparison. Best values for each measure are highlighted. The summary measure of the Scaled Brier score, c-index and AUCt is the average of each performance over all updates and higher values reflect better performance. For the OE-ratio and the calibration slope, it is the average over the absolute difference of each performance value with 1, over all updates. And for the calibration intercept, it is the average over the absolute value of each performance value, over all updates. For these last three measures, values closer to 0 reflect better performance.

Refitting

updating periods of 50 events

sliding window	Scaled Brier	Unos c-index	AUCt	OE-ratio	Cal.intercept	Cal.slope
0 weeks	0.112	0.686	0.723	0.195	0.267	0.270
1 week	0.130	0.686	0.727	0.125	0.189	0.198
2 weeks	0.144	0.688	0.729	0.093	0.132	0.168
4 weeks	0.154	0.691	0.736	0.083	0.120	0.184
no updating	0.125	0.697	0.744	0.220	0.358	0.147

updating periods of 100 events

sliding window	Scaled Brier	Unos c-index	AUCt	OE-ratio	Cal.intercept	Cal.slope
0 weeks	0.146	0.690	0.737	0.103	0.180	0.168
1 week	0.150	0.691	0.738	0.079	0.140	0.192
2 weeks	0.153	0.691	0.738	0.076	0.132	0.160
4 weeks	0.164	0.695	0.743	0.078	0.128	0.177
no updating	0.126	0.692	0.740	0.202	0.330	0.130

updating periods of 150 events

sliding window	Scaled Brier	Unos c-index	AUCt	OE-ratio	Cal.intercept	Cal.slope
0 weeks	0.149	0.689	0.739	0.057	0.085	0.201
1 week	0.153	0.691	0.740	0.066	0.082	0.209
2 weeks	0.155	0.692	0.741	0.082	0.102	0.201
4 weeks	0.157	0.690	0.741	0.082	0.096	0.188
no updating	0.102	0.687	0.733	0.232	0.386	0.133

6.2.2 Recalibration of intercept

Similar to the previous paragraph on the results of the refitting method, here we present the results for the recalibration method. Looking at the summary measures in Table 6.4 we see the following.

First of all, the recalibration of the intercept method did not effect the discrimination ability of the model and the calibration slope as expected.

For shorter updating periods, so updating more frequently, the use of a longer sliding window led to better performance for both the calibration measures and the overall performance of the updated models. The longer the updating periods became, so updating less frequently, the less the length of the sliding window seemed to influence the performance. Meaning that a sliding window did not give much improvement when already using longer updating periods.

We can also note that only the calibration improved when longer updating periods were used, the overall performance and discrimination got worse. Reasoning that this updating method focuses on aligning the mean observed outcome and the mean predicted outcome in the updated models, also called "calibration in the large", only improvement in calibration was expected. Also, because

the discrimination ability of the updated model decreased when longer updating periods were used, the overall performance also decreased even though the calibration did improve. Looking at Table 5.2, updating every 100 events with a sliding window of 4 weeks (corresponding to on average 206.7 events) performed overall worse than when updating every 50 events with a sliding window of 4 weeks (corresponding to on average 156.8 events). Meaning that overall the performance did not improve when more old data, so longer updating periods and sliding windows, was used. Best performance for this method was when updating every 50 events with a sliding window of 4 weeks.

We knew that recalibration of the intercept only re-estimates the baseline hazard, so we expected that this method required less data. Only the calibration of the updated models did improve when more data (events) was used (longer sliding window and/or updating period) but not much. So including more data in the updates did indeed lead to only small improvements in performance for this updating method. As with the previous method, using new data that reflects a longer period and thus also includes information that already got older over time, led to a decrease in the performance.

When looking at the overall performance measure and the calibration measures, the recalibration of the intercept method seemed to perform better than the no-updating method. We came to the conclusion that with this method including more data in each update only increased the performance slightly, a shorter updating period with a long sliding window already resulted in the best overall performance for this method.

Table 6.4: Summary measures of the performance measures for the different lengths of updating periods for the varying sliding windows over all updates. Also showing the average performance of the no-updating method for comparison. Best values for each measure are highlighted. The summary measure of the Scaled Brier score, c-index and AUCt is the average of each performance over all updates and higher values reflect better performance. For the OE-ratio and the calibration slope, it is the average over the absolute difference of each performance value with 1, over all updates. And for the calibration intercept, it is the average over the absolute value of each performance value, over all updates. For these last three measures, values closer to 0 reflect better performance.

Recalibration of intercept

updating periods of 50 events

sliding window	Scaled Brier	Unos c-index	AUCt	OE-ratio	Cal.intercept	Cal.slope
0 weeks	0.122	0.697	0.744	0.288	0.361	0.147
1 week	0.141	0.697	0.744	0.191	0.258	0.147
2 weeks	0.150	0.697	0.744	0.150	0.212	0.147
4 weeks	0.158	0.697	0.744	0.093	0.149	0.147
no updating	0.125	0.697	0.744	0.220	0.358	0.147

updating periods of 100 events

sliding window	Scaled Brier	Unos c-index	AUCt	OE-ratio	Cal.intercept	Cal.slope
0 weeks	0.144	0.692	0.740	0.143	0.203	0.130
1 week	0.148	0.692	0.740	0.134	0.196	0.130
2 weeks	0.151	0.692	0.740	0.111	0.168	0.130
4 weeks	0.154	0.692	0.740	0.096	0.149	0.130
no updating	0.126	0.692	0.740	0.202	0.330	0.130

updating periods of 150 events

sliding window	Scaled Brier	Unos c-index	AUCt	OE-ratio	Cal.intercept	Cal.slope
0 weeks	0.144	0.687	0.733	0.083	0.124	0.133
1 week	0.145	0.687	0.733	0.068	0.106	0.133
2 weeks	0.145	0.687	0.733	0.071	0.104	0.133
4 weeks	0.145	0.687	0.733	0.076	0.115	0.133
no updating	0.102	0.687	0.733	0.232	0.386	0.133

6.2.3 Bayesian updating

Again, when looking at the summary measures in Table 6.5 we see the following.

First of all, for each of the different lengths of updating periods, the sliding window of 0 weeks, so no old data included in the dataset used for the updates, resulted in the best performance. Only for the discrimination, the use of a sliding window of 1 week led to better performance in some cases, but the performance with a 0 week sliding window was still quite close. We also noted that updating less frequently, so longer updating periods, resulted in worse performance for all metrics. For this method, best performance was achieved when using periods of 50 events to update with no sliding window, and this is no surprise. We know that for the Bayesian updating method, the length of the updating periods not only indicates how frequent the updates are performed, but also indicates how often the forgetting factor down-weights the previously used data through the priors. The same data is down-weighted more often with shorter updating periods (higher updating frequency) than when using longer updating periods (lower updating frequency). Updating more frequently makes thus more distinction between very old data and more recent old data.

Using a sliding window was also a disadvantage for this updating method regarding the performance. Each update already incorporated old data (in the form of coefficients) throughout the priors. The priors in this method could be seen as a type of sliding window; using updating periods of 50 events means that the prior is based on the previous period of 50 events, so each update is then based on 2x50 events. Looking back to Table 5.2, updating every 50 events with no sliding window in the Bayesian method corresponded to using the same amount of information as when updating every 50 events with a sliding window of 2 weeks or updating every 100 events with no sliding window, for the other two updating methods. We can reason the same way for updating every 100 events with the Bayesian method; a sliding window of 4 weeks when updating every 100 events incorporates the same amount of information for the two other methods. For an updating period of 150 events, a sliding window longer than 4 weeks when updating every 150 events would represent the same amount of information. When making this comparison we have to keep in mind that the prior actually does not only incorporate information from the previous period but also the information from all the earlier periods, this information is however down-weighted over time.

For the Bayesian method using old data in the form of a sliding window, in each update, causes the same information to be used twice. For the previous two methods a sliding window was not only valuable to incorporate old data into each update, but also to recover some of the events that were lost in between the updating periods, as we explained in 5.1. This is the main difference with how the Bayesian method includes old data.

Lastly, we also noted that the Bayesian updating led to better overall performance than the no-updating method, for all the lengths of updating periods. Our conclusion is that updating more frequently results in the best performance for this method. And using a sliding window gives no improvement in performance since (part of) the same data is already incorporated through the priors.

Table 6.5: Summary measures of the performance measures for the different lengths of updating periods for the varying sliding windows over all updates. Also showing the average performance of the no-updating method for comparison. Best values for each measure are highlighted. The summary measure of the Scaled Brier score, c-index and AUCt is the average of each performance over all updates and higher values reflect better performance. For the OE-ratio and the calibration slope, it is the average over the absolute difference of each performance value with 1, over all updates. And for the calibration intercept, it is the average over the absolute value of each performance value, over all updates. For these last three measures, values closer to 0 reflect better performance.

Bayesian updating

updating periods of 50 events

sliding window	Scaled Brier	Unos c-index	AUCt	OE-ratio	Cal.intercept	Cal.slope
0 weeks	0.146	0.701	0.745	0.149	0.263	0.260
1 week	0.143	0.699	0.746	0.160	0.276	0.227
2 weeks	0.138	0.699	0.745	0.166	0.292	0.243
4 weeks	0.137	0.698	0.745	0.179	0.317	0.248
no updating	0.125	0.697	0.744	0.220	0.358	0.147

updating periods of 100 events

sliding window	Scaled Brier	Unos c-index	AUCt	OE-ratio	Cal.intercept	Cal.slope
0 weeks	0.144	0.697	0.747	0.175	0.300	0.210
1 week	0.140	0.697	0.746	0.181	0.316	0.239
2 weeks	0.135	0.696	0.745	0.188	0.334	0.237
4 weeks	0.138	0.695	0.746	0.178	0.320	0.223
no updating	0.126	0.692	0.740	0.202	0.330	0.130

updating periods of 150 events

sliding window	Scaled Brier	Unos c-index	AUCt	OE-ratio	Cal.intercept	Cal.slope
0 weeks	0.127	0.691	0.741	0.198	0.348	0.268
1 week	0.122	0.691	0.740	0.200	0.360	0.284
2 weeks	0.118	0.690	0.740	0.205	0.374	0.284
4 weeks	0.119	0.688	0.738	0.205	0.371	0.275
no updating	0.102	0.687	0.733	0.232	0.386	0.133

6.3 Comparison of the three updating methods

We will now compare the results of the three methods. We will give detailed comparison where we focus on the effect of the length of updating periods and sliding windows.

For both refitting and recalibration of the intercept, longer sliding windows resulted in better performance when updates were made more frequently (shorter update periods). When updates were made less frequently (longer updating periods), the length of the sliding windows had less impact on the performances for both methods. Reasoning that both methods needed enough data to obtain a reliable updated model, using a longer updating period means that more data is included in each update and thus the updated model gets more reliable. But the proportion of data included by a sliding window in each update becomes smaller when the updating periods get longer, resulting in the size of sliding windows having less influence on the performance of the updated models. However, we also saw that including too much old data led to a decrease in performance for both methods. This is because longer updating periods contain more data that over time already has become old, and so using sliding window would add even older information to the update. We

also saw that the recalibration method needed less data to achieve best performance compared to the amount of data needed for the best performance of the refitting method; updating every 50 events with a 4 weeks sliding window (corresponding to an average of 156.8 events per update) and updating every 100 events with a 4 week sliding window (corresponding to an average of 206.7 events per update), respectively.

For the Bayesian method, best performance was achieved when no old data was included in each update and updates were made most frequently; updating every 50 events and using no sliding window. All performance measures got worse for this method when updating less frequently (longer updating periods) and when using a sliding window. This can be explained by the forgetting factor; when updating less frequently, the previously used data is down-weighted less often and causes there to be less distinction between older and newer old data. The old data in a sliding window is also already incorporated through the priors (in the form of coefficients), meaning that a sliding window has no added value for this updating method. The prior incorporates the information from all the previous periods down-weighted over time, so this method makes good use of all the previous information in every update. However, when comparing the overall performance of three methods we can see that the Bayesian method only outperformed the other two method when updating every 50 events with a sliding window of 0 and 1 week. This means that including old information through a sliding window for the refitting and recalibration method resulted for all the other cases in better overall performance than the Bayesian method that uses a prior. The main advantage of the usage of the sliding windows with the recalibration and refitting method is that the lost events in between the periods are saved, which does not happen with the priors of the Bayesian method.

When comparing the results of the methods to the performance of no-updating, so using the original model for the predictions, we see the following. No-updating only outperformed the discrimination of the refitting method when updating most frequently (updating every 50 events). For all other cases, using the dynamic updating methods outperformed the no-updating method. We did expect that dynamic updating of the original model would result in better performance than not updating. Most results for each of the methods were as expected; for refitting and recalibration using more data in the updates gave improvement for the performance but including too much old data also turned out to decrease the performance, and for the Bayesian method sliding windows are not adding value because this method already uses priors. When looking at the highest overall performance value each method reached, the refitting method reached the highest scaled Briers score of 0.164 (when updating every 100 events and a sliding window of 4 weeks).

6.4 Conclusion

The results showed that using one of the dynamic updating methods always outperformed the original model (no-updating).

When one has data with a lot of events (data can be both new and old) and no need to update frequently, the refitting updating method results in best performance. When in need of a faster updating method, so in occasions where the environment is expected to change more rapidly, the Bayesian updating method is preferred. Another option is to use recalibration; this method also gives good performance when updating more frequently. The main difference between the Bayesian and recalibration method is the way in which they incorporate old information. The Bayesian method uses priors based on all the data from the previous periods down-weighted over time, the recalibration method uses a sliding window. The advantage of using a sliding window is that the events that are lost, due to censoring, in between periods are recovered. Even though the refitting method required the most old data, this method gave for most of the updating frequencies the best overall performance measures, when comparing the results of the three methods.

Chapter 7

Discussion

In this project we studied the effect of the updating frequency and the use of sliding windows (the inclusion of old data in an update) on the performance of a dynamically updated Cox proportional hazard model for three different updating methods: refitting, recalibration of the intercept, and Bayesian updating. For each method, we varied the updating frequency and the length of sliding window and evaluated the performance using measures of discrimination, calibration, and overall performance. Each performance measure was summarised over all the updates to give an overall summary of the results. We used data from patients admitted to the hospital with COVID-19 in the first two waves of the pandemic in The Netherlands.

No single updating method outperformed the others for all the different combinations of updating frequency and sliding windows. In every combination, there was always a dynamic updating method that outperformed the original model (no-updating).

For refitting, updating less frequently and using longer sliding windows improved the performance of the updated models. However, we saw that for this method the performance can also decrease when too much old data is included in each update. This occurred when the update frequency was lowest with longest sliding window. For this method a large amount of data is required to achieve best performance, which can be an issue when one requires to update more frequently.

For the recalibration of the intercept method, updating more frequently and using a longer sliding window improved the performance of the updated models. As well as for the refitting method, including too much old data in each update decreased the performance. But compared to the refitting method, relatively less data was needed to achieve the best performance for this method.

With the Bayesian dynamic updating method, the best performance was achieved by updating most frequently and using no sliding window. We saw that the prior of this method incorporated the data from all the previous periods down-weighted over time and resulted in best performance when updating most frequently without using a sliding window. However, the refitting and recalibration method both outperform the Bayesian method when using longer sliding windows, based on overall performance. We also noted that when using a prior instead of a sliding window to incorporate old data into each update, the lost events in between the updating periods would not be recovered.

Compared to the other two methods, the Bayesian method was the most computationally intensive and required additional assumptions on the distribution of the survival times and the rate at which data became old (forgetting factor). Following Tanner et al., 2023[6] we assumed the survival times to follow an exponential distribution and fixed the forgetting factor at 0.9. For our analysis, assuming an exponential distribution of the survival times seemed reasonable due to the short horizon and focus on 28-day outcomes only.[6] Future research could look into the use of other parametric survival distributions. We also saw in the results that for the Bayesian dynamic updating the incorporation of a sliding window had no added value. Since this updating method already incorporates the old data in the form of priors based on the previous estimated coefficient, we reason that it could be more interesting to investigate what the influence of the forgetting factor

would be on the performance of the updated models.

Comparing the results of the three methods, we can note that even though the refitting method required the most old data, this method gave for most of the updating frequencies the best performance measures.

An important fact to note is that we evaluated each updated model on the next period of 100 events, for all lengths of updating periods. This implies that for the models that were updated every 150 events we do not know their performance on the 50 events that lay in between the validation period of 100 events and the next updating period of 150 events. To prevent this we advise for future applications to evaluate the updated models on a larger dataset. However, we also have to note that when evaluating an updated model on a period that is larger than the periods used for the updates, the updated model would already have been updated again at some point (e.g. when updating a model every 50 events and evaluating on periods of 100 events, after the first 50 events their would already have been an update).

Another aspect that was not looked into in our analysis, is the assumptions check of the updated models at each update. Every time a Cox proportional hazards model is updated one should check if the model assumptions are still valid. In this project, we focused on illustrating dynamic updating. Nevertheless, when one would apply dynamic updating the assumptions of the updated models should not be forgotten.

A limitation of this analysis is that it is restricted to only one scenario, namely the one provided by our real data application. A simulation study may be needed to draw more general conclusions. Another limitation is that patients were artificially censored at the end of each updating period and, as a consequence, some events were lost due to how the updating periods were constructed. It could be useful to look into other ways to construct these updating periods.

The choice regarding the updating frequency and the sliding window will always depend on the amount of new data that is available and on the intensity of the underlying changes in the data. This implies that when making these choices one has to take into account the data that will be used, since we concluded that there is not one method that works best in every scenario. The results of this project can be useful for future application of dynamic survival model updating now that we know the influence of the updating frequency and use of sliding windows on the performance of the three updating methods.

Acknowledgment

I would like to express my deepest appreciation to my supervisors Dr.ir. N. van Geloven and I. Prosepe of Leiden Univeristy Medical Center for their guidance and support. Our weekly meetings have been extremely useful for the progress of my thesis and the feedback on various thesis drafts were really helpful to me during the project's completion. I would like to express my special thanks to Ilaria for her constant support and motivation.

Additionally, I want to thank all the PhD students from the Medical Statistics department for making me feel at home.

Finally, I would like to thank my parents and friends for their unconditionally support throughout my studies.

Bibliography

1. Steyerberg EW. *Clinical Prediction Models A Practical Approach to Development, Validation, and Updating*. Springer International Publishing 2019.

2. Smeden M, Reitsma JB, Riley RD, Collins GS, Moons KG. Clinical prediction models: diagnosis versus prognosis. *Journal of clinical epidemiology.* 2021;132:142-145.

3. Smit JM, Krijthe JH, Tintu AN, *et al.* Development and validation of an early warning model for hospitalized COVID-19 patients: a multi-center retrospective cohort study. *Intensive care medicine experimental.* 2022;10(1):38-38.

4. Janssen K, Moons K, Kalkman C, Grobbee D, Vergouwe Y. Updating methods improved the performance of a clinical prediction model in new patients. *Journal of clinical epidemiology.* 2008;61(1):76-86.

5. Steyerberg EW, Borsboom GJJM, Houwelingen HC, Eijkemans MJC, Habbema JDF. Validation and updating of predictive logistic regression models: a study on sample size and shrinkage. *Statistics in medicine.* 2004;23(16):2567-2586.

6. Tanner KT, Keogh RH, Coupland CAC, Hippisley-Cox J, Diaz-Ordaz K. Dynamic updating of clinical survival prediction models in a changing environment. *Diagnostic and prognostic research.* 2023;7(1):1-14.

7. Schnellinger EM, Yang W, Kimmel SE. Comparison of dynamic updating strategies for clinical prediction models. *Diagnostic and prognostic research.* 2021;5(1):20-20.

8. Davis SE, Greevy J, Fonnesbeck C, Lasko TA, Walsh CG, Matheny ME. A nonparametric updating method to correct clinical prediction model drift. *Journal of the American Medical Informatics Association.* 2019;26(12):1448-1457.

9. Knight SR, Ho A, Pius R, *et al.* Risk stratification of patients admitted to hospital with covid-19 using the ISARIC WHO Clinical Characterisation Protocol: development and validation of the 4C Mortality Score. *BMJ.* 2020;370.

10. Kleinbaum DG, Klein M. *Survival analysis.* Statistics for biology and health. Springer 2012.

11. Klein JP, Moeschberger ML. *Survival analysis: techniques for censored and truncated data.* Statistics for biology and health. Springer 1997.

12. Geloven N, Giardiello D, Bonneville EF, *et al.* Validation of prediction models in the presence of competing risks: a guide through modern methods. *BMJ (Online).* 2022;377.

13. McLernon DJ, Giardiello D, Van Calster B, *et al.* Assessing Performance and Clinical Usefulness in Prediction Models With Survival Outcomes: Practical Guidance for Cox Proportional Hazards Models. *Annals of internal medicine.* 2023;176(1):105-114.

14. Hartman N, Kim S, He K, Kalbfleisch JD. Concordance indices with left-truncated and right-censored data. *Biometrics.* 2023;79(3):1624-1634.

15. Uno H, Cai T, Pencina MJ, D'Agostino RB, Wei LJ. On the C-statistics for evaluating overall adequacy of risk prediction procedures with censored survival data. *Statistics in medicine.* 2011;30(10):1105-1117.

16. Van Calster B, McLernon DJ, Smeden M, Wynants L, Steyerberg EW. Calibration: the Achilles heel of predictive analytics. *BMC medicine.* 2019;17(1):230-230.

17. Houwelingen HC. Validation, calibration, revision and combination of prognostic survival models. *Statistics in medicine.* 2000;19(24):3401-3415.

18. Binuya MAE, Engelhardt EG, Schats W, Schmidt MK, Steyerberg EW. Methodological guidance for the evaluation and updating of clinical prediction models: a systematic review. *BMC medical research methodology.* 2022;22(1):316-316.

19. Graf E, Schmoor C, Sauerbrei W, Schumacher M. Assessment and comparison of prognostic classification schemes for survival data. *Statistics in medicine.* 1999;18(17-18):2529-2545.

20. Steyerberg EW, Vickers AJ, Cook NR, *et al.* Assessing the Performance of Prediction Models: A Framework for Traditional and Novel Measures. *Epidemiology (Cambridge, Mass.).* 2010;21(1):128-138.

21. Su TL, Jaki T, Hickey GL, Buchan I, Sperrin M. A review of statistical updating methods for clinical prediction models. *Statistical methods in medical research.* 2018;27(1):185-197.

22. McCormick TH, Raftery AE, Madigan D, Burd RS. Dynamic logistic regression and dynamic model averaging for binary classification. *Biometrics.* 2012;68(1):23—30.

23. Stan Development Team . RStan: the R interface to Stan. 2023. R package version 2.26.3.

24. Gelman A. *Bayesian data analysis.* Texts in statistical science. Chapman & Hall/CRC 2004.

25. Jenkins D, Martin G, Sperrin M, *et al.* Comparing Predictive Performance of Time Invariant and Time Variant Clinical Prediction Models in Cardiac Surgery. *Studies in health technology and informatics.* 2024;310:1026-1030.

26. Prentice RL, Kalbfleisch JD, Peterson AV, Flournoy N, Farewell VT, Breslow NE. The Analysis of Failure Times in the Presence of Competing Risks. *Biometrics.* 1978;34(4):541–554.

Appendix A

Dataset information

A.1 Adaptations on dataset

The following adaptions were applied to the data before the implementation of the analysis:

- All dates as year-month-day format;
- Number of comorbidities defined as count of comorbidities;
  - Same chosen as in $4C$ when available in dataset;
  - Instead of ureum we used creatinine (both values of kidney, ureum had a lot of missing values $> 30\%$);
- Exclusion criteria (see Chapter 2 for more details);
- End first wave: 2020-06, begin second wave: 2020-08;
  - In July 2020 only 20 events were recorded; did not use the data from this month;
- Continuous predictors were centered by their mean and scaled by their standard deviation (see Table A.1);
- Extreme values were cut off (based on clinical expertise);
  - RR larger than 60 cut to 60;
  - SpO2 smaller than 60 cut to 60;
  - creatinine larger than 300 cut to 300.

Table A.1: Overview of mean and standard deviation of continuous predictors before they were centered and scaled.

variable	mean	sd
age	65.93	15.43
respiratory rate per minute (RR...min)	23.11	7.67
peripheral oxygen saturation percentage (SpO2...)	93.18	6.66
c-reactive protein mg/l (CRP.mg.L.)	102.11	88.03
creatinine μmol/l (Creatinine.SER.µmol.L)	97.10	50.87

A.2 Data descriptives

In this section some additonal tables and figures of the dataset are provided.

In Table A.2 an overview of the hospitals and their total number of patients is provided. In Figure A.1 we can see the distribution of the continuous predictors. The proportion of patients admitted that were older than 50 is much larger than the proportion of patients under the age of 50. We can also note that for the predictors respiratory rate and peripheral oxygen some outliers are present even after the adaptions (from the previous section) are made; for the respiratory rate two values are much larger compared to the other values, and for peripheral oxygen a few values are much lower (and close to 0) compared to the other values.

Table A.2: Table of hospitals and number of patients with COVID-19 admitted there.

Hospital	Number of admitted patients
Alrijne	1320
Haga	911
HMC	1175
LUMC	658

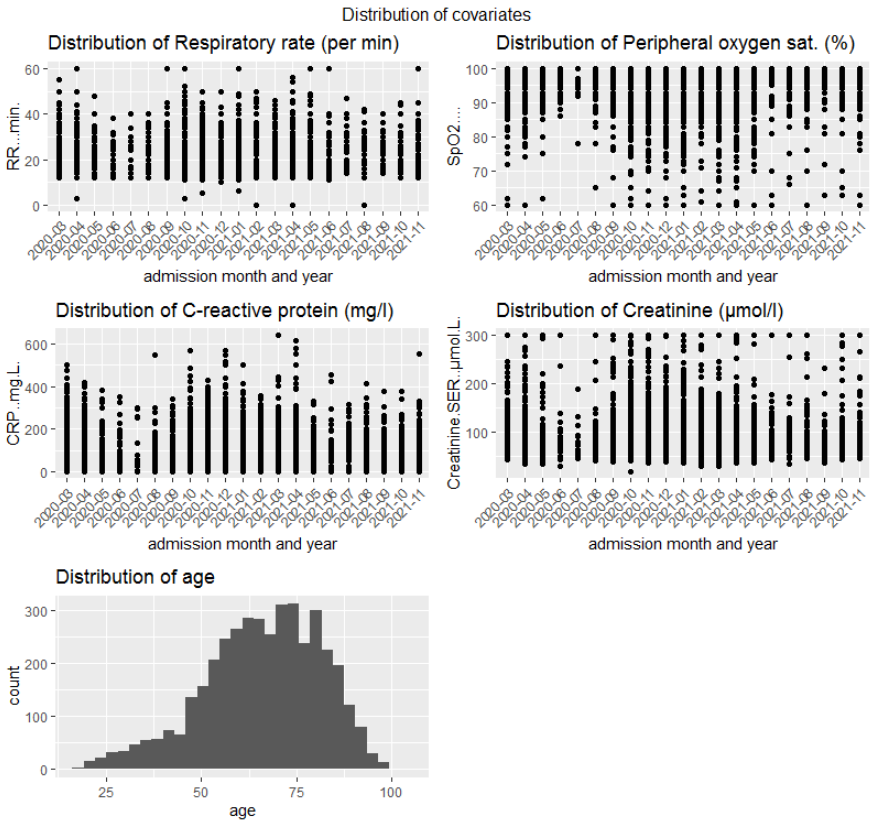

Figure A.1: Distribution of the continuous covariates respiratory rate per minute (RR...min), peripheral oxygen saturation percentage (SpO2...), c-reactive protein mg/l (CRP.mg.L.), creatinine μmol/l (Creatinine.SER.µmol.L), and age (at admission).

Appendix B

Model assumptions

In this appendix the assumptions of a Cox proportional hazard are discussed. The assumptions of the model we developed on the data from the first wave of the COVID-19 pandemic (M_0) are also checked.

B.1 Assumptions Cox proportional hazards model

When using a Cox proportional hazards model one needs to check the assumptions of this model. We will discuss here the assumptions of the model and how they can be checked. Explanation of the assumtpions is based on Kleinbaum and Klein, 2012[10] and Klein and Moeschberger, 1997.[11]

B.1.1 Proportional hazard assumption

The first assumption that we will discuss here is the proportional hazard assumption. This assumption states that the hazards must be proportional. With this is meant that the ratio between two hazard of any two individuals must be time independent. In general this would mean that

$$HR(t) = \frac{h_2(t)}{h_1(t)} =: HR$$

is assumed to be independent of time for any two hazard functions $h_1(t)$ and $h_2(t)$. In the Cox model we have that the hazard ratio for two sets of covariate values X_i and X_j equals

$$\frac{h(t|X_i)}{h(t|X_j)} = \frac{h_0(t)\exp(\sum_{k=1}^{p}\beta_k X_{ik})}{h_0(t)\exp(\sum_{k=1}^{p}\beta_k X_{jk})} = \frac{\exp(\sum_{k=1}^{p}\beta_k X_{ik})}{\exp(\sum_{k=1}^{p}\beta_k X_{jk})} = \exp(\sum_{k=1}^{p}\beta_k(X_{ik} - X_{jk}))$$

from which we can directly conclude that the ratio is constant through time, and thus that the hazards are proportional.

One can validate this assumption by visually inspecting through plots. A first possibility to check the assumption is through cumulative hazard plots, although this is only possible for the categorical covariates. If the cumulative hazards don't cross the assumption holds. Another possibility is to us a log-minus-log plot, here the survival is plotted through a log-minus-log transformation: $\log(-\log S(t|X=1)) = \log H(t|X=1) = \log H_0(t) + \beta_1$, for categorical covariate X, so again only possible for categorical covariates. The curves for the different levels of X should not cross for the assumption to hold.

It is also possible to test this assumption with *cox.zph()* in R for each covariate and also global.

B.1.2 Linearity assumption for continuous variables

The second assumption is the linearity assumption. Namely, the model assumes that there is a linear relation between the log hazard and each of the continuous variables.

The formulation of the Cox model can be rewritten as

$$\log h(t|X) = \log h_0(t) + \beta^\top X$$

such that this linear relation between the continuous covariates and the log hazard can be viewed more clearly.

This assumption can be validated by inspecting a plot of the Martingale residuals together with each of the continuous covariates. The assumption then holds if a linear trend is visual. When this is not the case, one can try to improve the model by adding a spline to the covariate in question.

B.1.3 Independent censoring

The last assumption, regarding the censoring of the data, is the independent censoring assumption conditional on the covariates. For this assumption to hold, it is requires that the individuals that are censored are representative of the individuals that remain at risk, conditional on the covariates.

B.2 Assumptions check of original model M_0

For the Cox model that we developed using the data from the first wave, also referred to as the original model M_0, we looked at the assumptions that this model should satisfy.

The first two assumptions, proportional hazards and linearity assumption, we checked using visual aids in R and *cox.zph()*.

Figures B.1 and B.2 show the proportional hazard assumption check for the categorical covariates sex and number of comorbidities, and Figure B.3 shows linearity assumption check for the continuous covariates. In Table B.1 the results from the *cox.zph()* test are displayed.

For the independent censoring assumption we take a closer look at the censoring mechanism. The censoring in the data is due to the patients that were discharged to a different hospital. We believe that independent censoring conditional on the covariates is safe to assume for these patients, as the discharge can mostly be explained by the combination of values of the covariates.

We can see that not all assumptions seem to hold; in Figure B.3 two of the three lines cross, and Table B.1 not all variables pass the proportional hazards test. Our goal is not to develop a new prediction model for ICU or mortality in COVID-19 admission to the hospital, but to illustrate dynamic model updating, so this is not an issue. However we did look into these assumption because it is always relevant to check the assumptions of the model one uses to get a better understanding of the model.

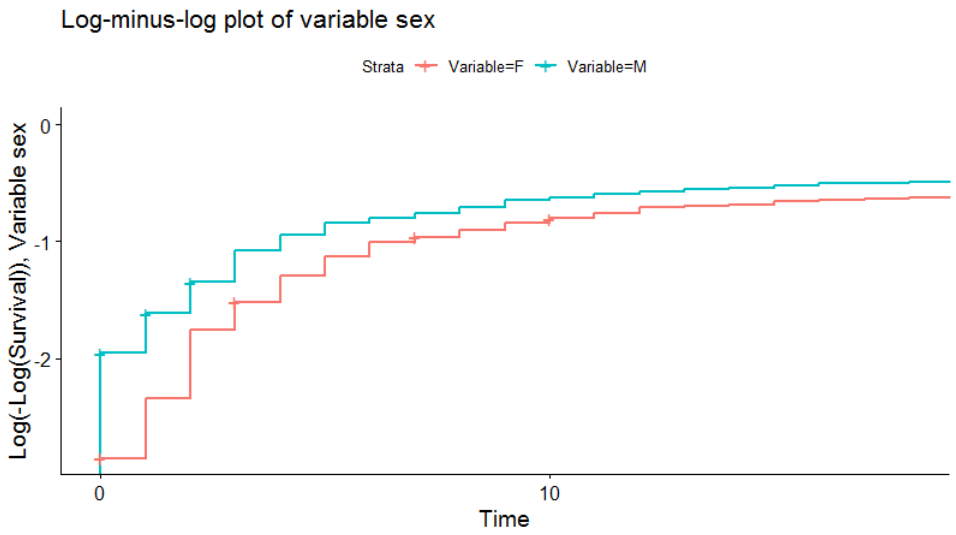

Figure B.1: Log-minus-log plot of the variable sex to check the proportional hazard assumption.

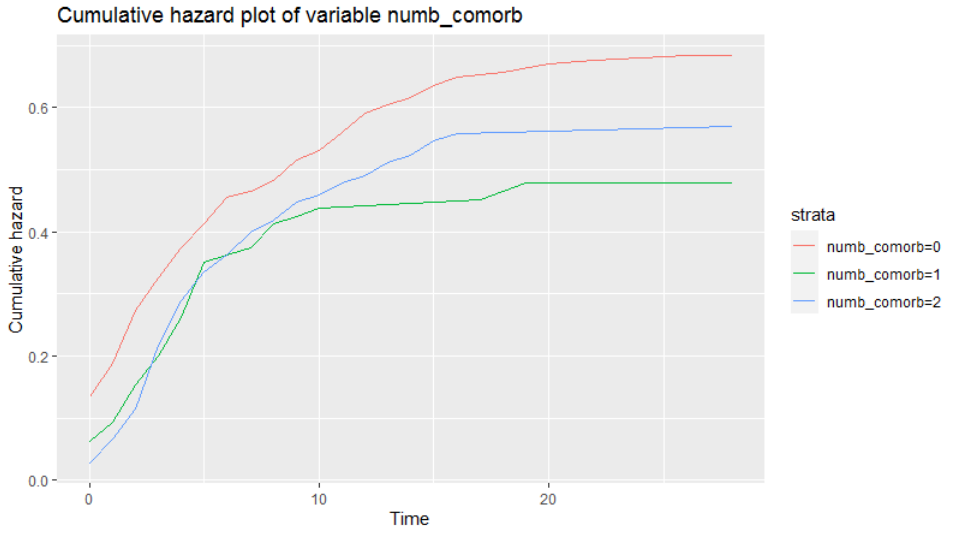

Figure B.2: Cumulative hazard plot of the variable number of comorbidities to check the proportional hazard assumption.

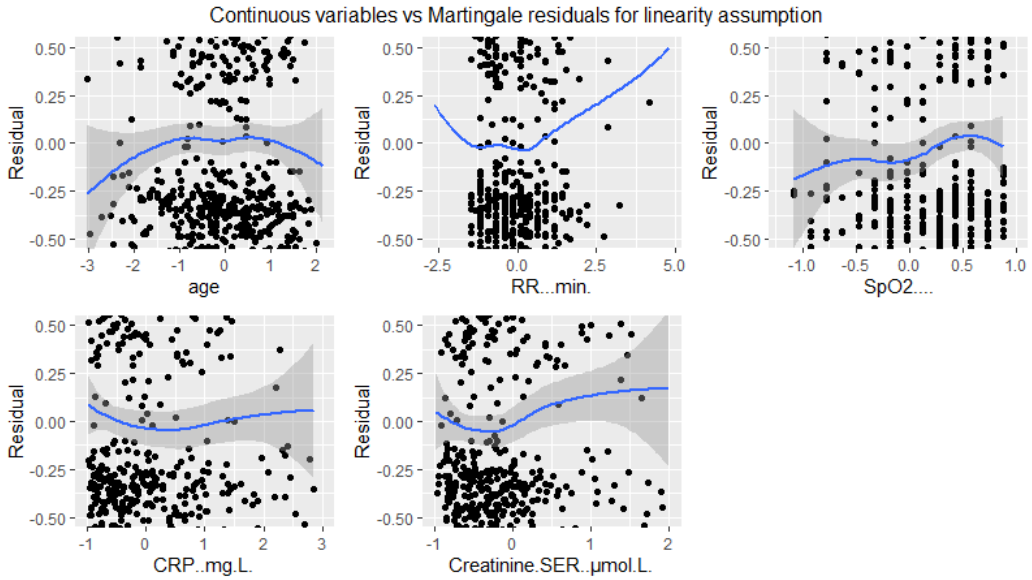

Figure B.3: Martingale residuals of the continuous variables to check the linearity assumption. The continuous variables are age (at admission), respiratory rate per minute (RR...min), peripheral oxygen saturation percentage (SpO2...), c-reactive protein mg/l (CRP.mg.L.), and creatinine μmol/l (Creatinine.SER.µmol.L).

Table B.1: Output of *cox.zph()* test for proportional hazard for each predictor seperate and global. The predictors are age (at admission), sex, number of comorbidities (numb comorb), respiratory rate per minute (RR...min), peripheral oxygen saturation percentage (SpO2...), c-reactive protein mg/l (CRP.mg.L.), and creatinine μmol/l (Creatinine.SER.µmol.L).

variable	chisq	df	p-value
age	61.84	1	3.7e-15
sex	6.38	1	0.012
numb comorb	6.39	1	0.012
RR...min	16.81	1	4.1e-05
SpO2	2.55	1	0.110
CRP..mg.L.	33.78	1	6.2e-09
Creatinine.SER..μmol.L.	3.31	1	0.069
GLOBAL	101.76	1	<2e-16

Appendix C

Updating periods

A table with additional information about the updating periods is given; Table C.1 shows the start (and end) dates of the updating periods.

Table C.1: Start dates of updating periods for the different lengths of periods. The end date of each period is a day before the start date of the next period (for the last periods the end date is explicitly mentioned).

periods of 50 events	start date	periods of 100 events	start date	periods of 150 events	start date
1	2020-08-01	1	2020-08-01	1	2020-08-01
2	2020-10-03	2	2020-10-15	2	2020-10-24
3	2020-10-17	3	2020-11-05	3	2020-11-25
4	2020-10-29	4	2020-12-02	4	2021-01-01
5	2020-11-11	5	2020-12-27	5	2021-02-12
6	2020-11-26	6	2021-01-19	6	2021-04-08 (end date 2021-07-25)
7	2020-12-13	7	2021-02-25		
8	2020-12-27	8	2021-04-04		
9	2021-01-10	9	2021-05-07 (end date 2021-09-07)		
10	2021-01-27				
11	2021-02-19				
12	2021-03-13				
13	2021-04-04				
14	2021-04-20				
15	2021-05-17 (end date 2021-08-05)				

Appendix D

Additional figures of the results

D.1 Performance throughout updating periods

In each figure, update i refers to updating the previous model M_{i-1} using the i-the period (D_i) of 50/100/150 events and evaluating the updated model on the next period of 100 events.

The sliding windows vary between 0, 1, 2 and 4 weeks of old data included in each update. For reference, the performance of the no-updating method is also displayed.

For the Scaled Brier score, c-index and AUCt, higher values reflect better performance. For the other three measures, OE-ratio, calibration intercept and slope, preferred values are close to the horizontal red line (so close to 0 or 1).

D.1.1 Refitting

We present three figures, one for each of the three different lengths of updating periods (Figure D.1, D.2 and D.3). In each plot, the performance metrics are shown at each update for the different sliding windows.

First thing to note in Figure D.1 is for the discrimination measures the performance looks similar to that of the no-updating method, but for the calibration measures the performance of the refitting method differs more from the performance of the no-updating method. In Figures D.2 and D.3 we can see the same. More specifically, for the calibration intercept, the performance of the refitting method for all sliding window seems to be better than the performance of the no-updating method, for all the lengths of updating periods. For the OE-ratio, the measured values seem to vary in a similar way as for the calibration intercept around the target value.

Secondly, when updating more frequently, so using shorter updating periods, the measured performance measures fluctuate more throughout the updates compared to using longer updating periods. This variation is also reflected in the performance of the no-updating method. Specifically, the calibration intercept and slope seem to fluctuate wider around their target values 0 and 1, respectively, than when using longer updating periods. We can also see that in some periods the performance between the sliding windows can vary a lot for some performance metrics. For example, when updating every 50 events at update 4 for the OE-ratio and calibration intercept, and at update 3 for the calibration slope (see Figure D.1). When updating less frequently, so using longer updating periods, this seems to happen less. Using a sliding window has less impact; it does not improve the performance of the updated models.

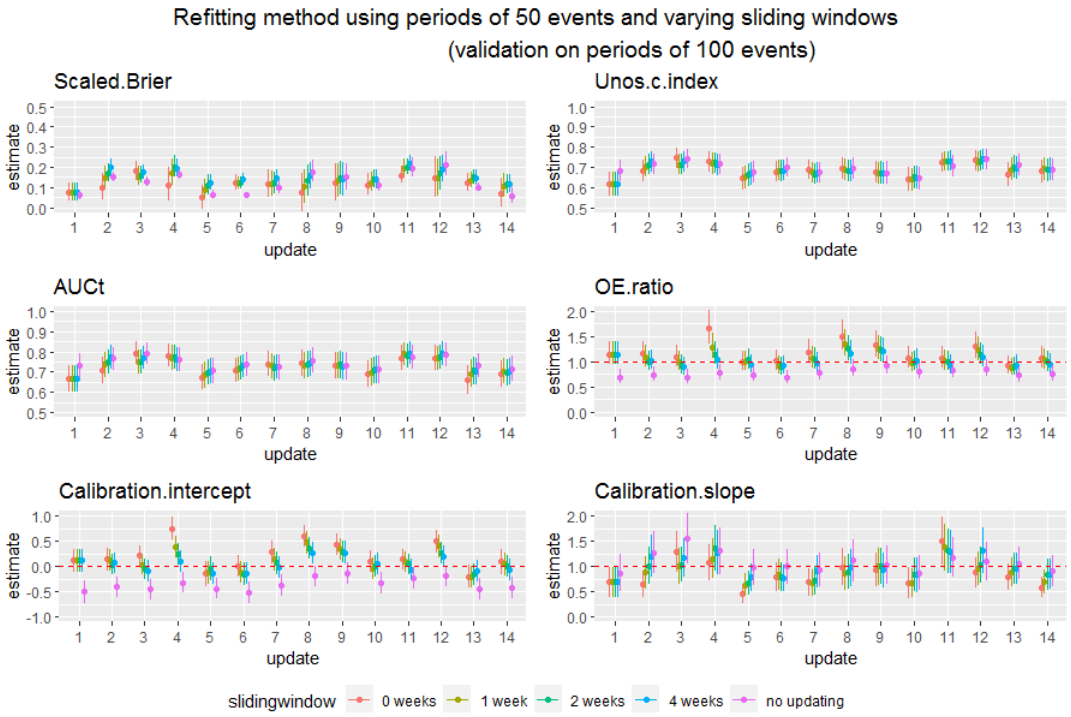

Figure D.1: Refitting every 50 events. Using varying sliding windows and also showing the performance of the no-updating method. At each update the model is validated on the next period of 100 events. For the Scaled Brier score, c-index and AUCt, higher values reflect better performance. For the OE-ratio, calibration intercept and slope, preferred values are close to the horizontal red line (so close to 0 or 1).

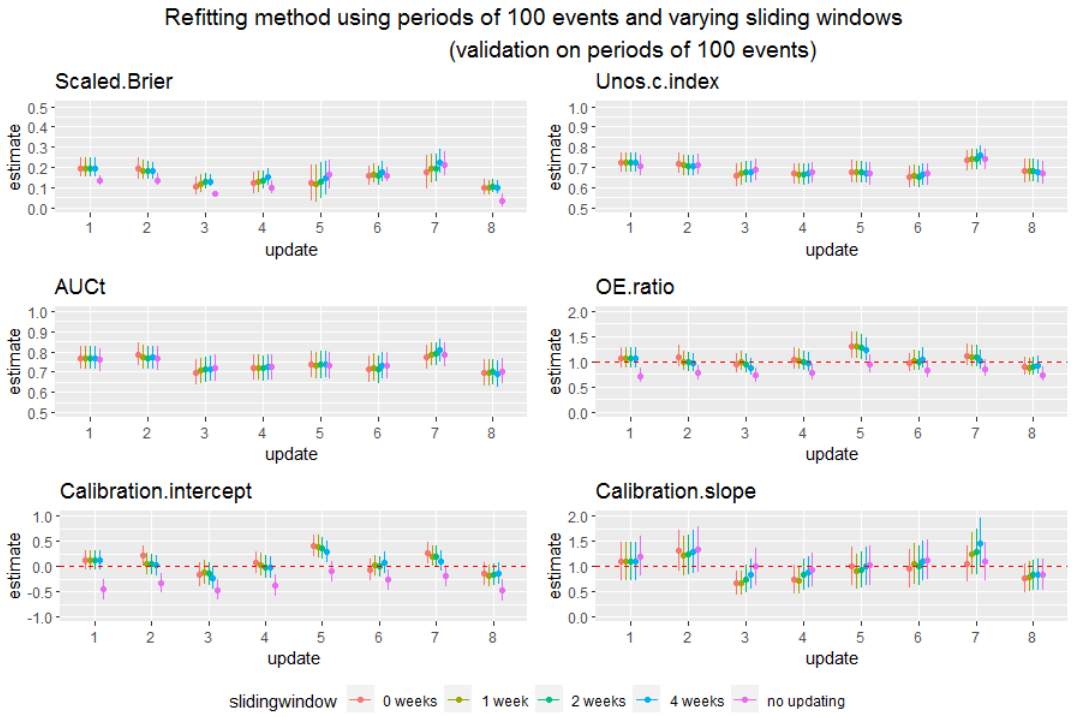

Figure D.2: Refitting every 100 events. Using varying sliding windows and also showing the performance of the no-updating method. At each update the model is validated on the next period of 100 events. For the Scaled Brier score, c-index and AUCt, higher values reflect better performance. For the OE-ratio, calibration intercept and slope, preferred values are close to the horizontal red line (so close to 0 or 1).

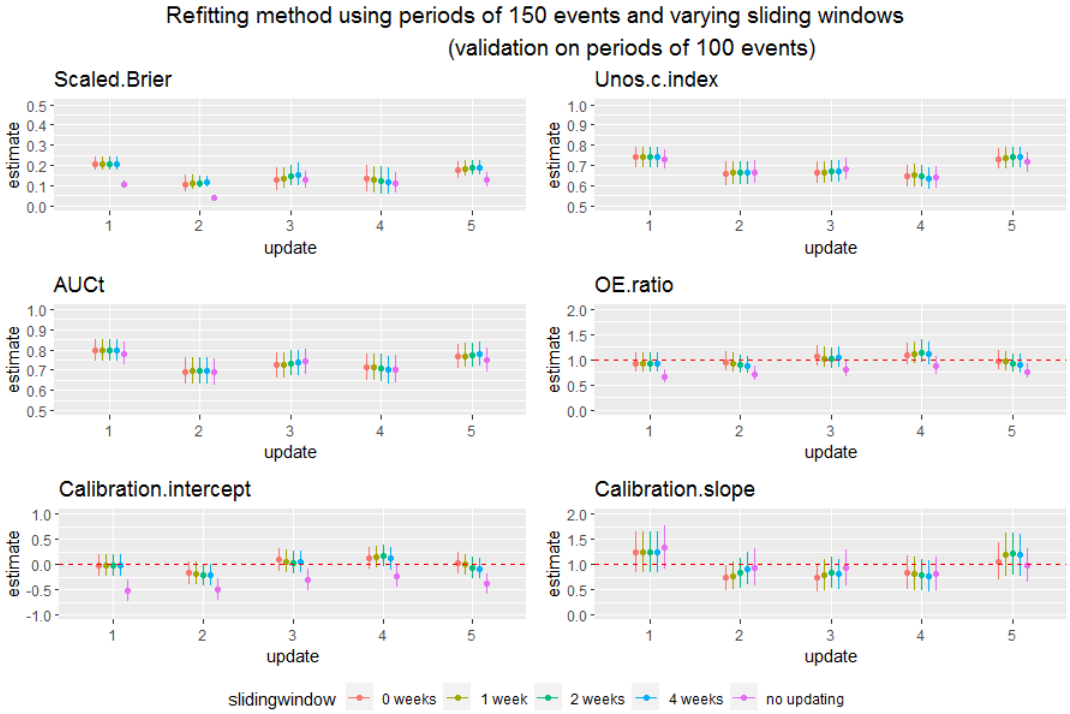

Figure D.3: Refitting every 150 events. Using varying sliding windows and also showing the performance of the no-updating method. At each update the model is validated on the next period of 100 events. For the Scaled Brier score, c-index and AUCt, higher values reflect better performance. For the OE-ratio, calibration intercept and slope, preferred values are close to the horizontal red line (so close to 0 or 1).

D.1.2 Recalibration of the intercept

Again, we present three figures, one for each of the three different lengths of updating periods (Figure D.4, D.5 and D.6). In each plot, the performance metrics are shown at each update for the different sliding windows.

Similar to what we saw in the results of the refitting method, also here we observe that the calibration measures differ to that of the no-updating method, shorter updating periods result in fluctuating performance measures compared to longer periods, and when updating less frequently using a sliding window has less impact on the performance measures.

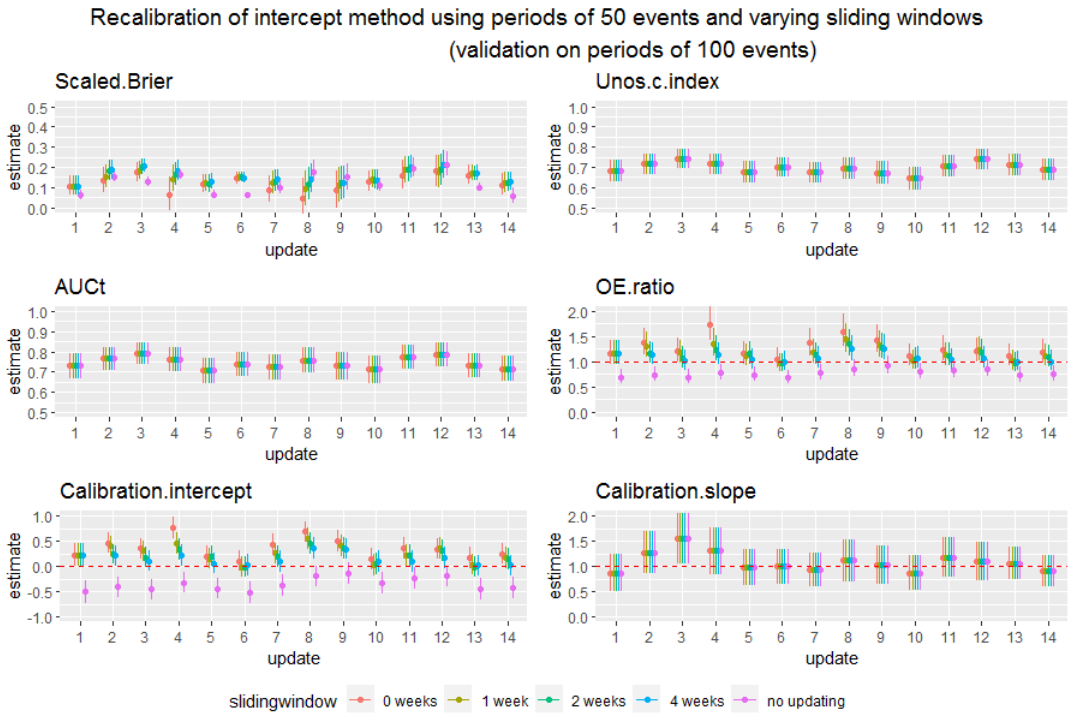

Figure D.4: Recalibrating the intercept every 50 events. Using varying sliding windows and also showing the performance of the no-updating method. At each update the model is validated on the next period of 100 events. For the Scaled Brier score, c-index and AUCt, higher values reflect better performance. For the OE-ratio, calibration intercept and slope, preferred values are close to the horizontal red line (so close to 0 or 1).

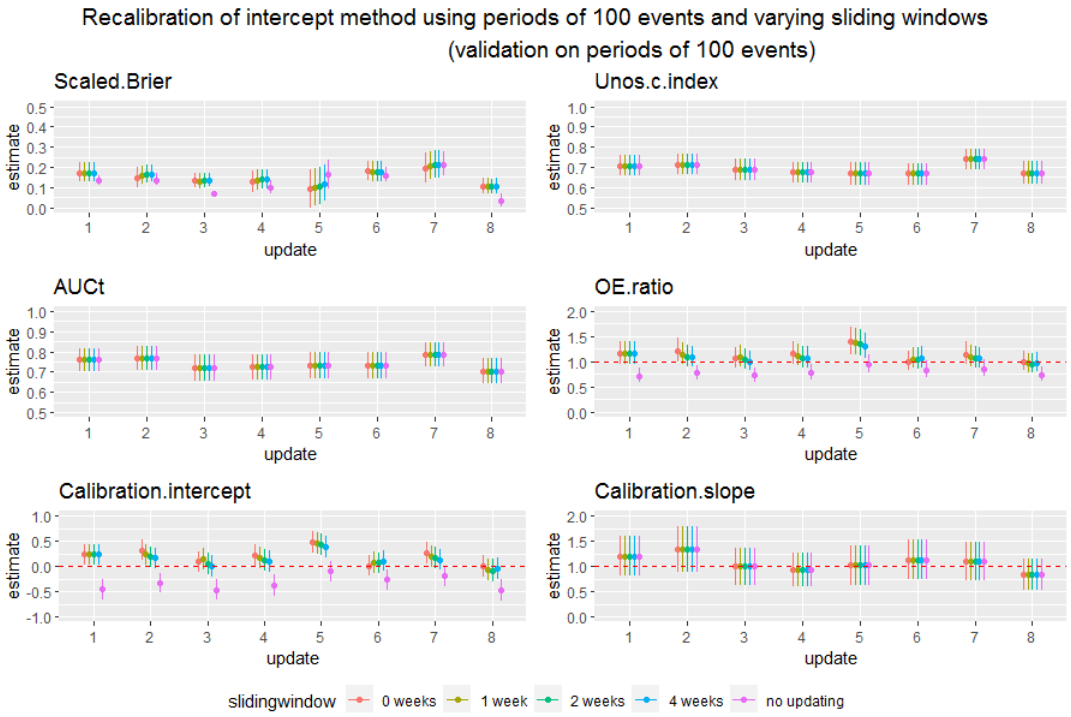

Figure D.5: Recalibrating the intercept every 100 events. Using varying sliding windows and also showing the performance of the no-updating method. At each update the model is validated on the next period of 100 events. For the Scaled Brier score, c-index and AUCt, higher values reflect better performance. For the OE-ratio, calibration intercept and slope, preferred values are close to the horizontal red line (so close to 0 or 1).

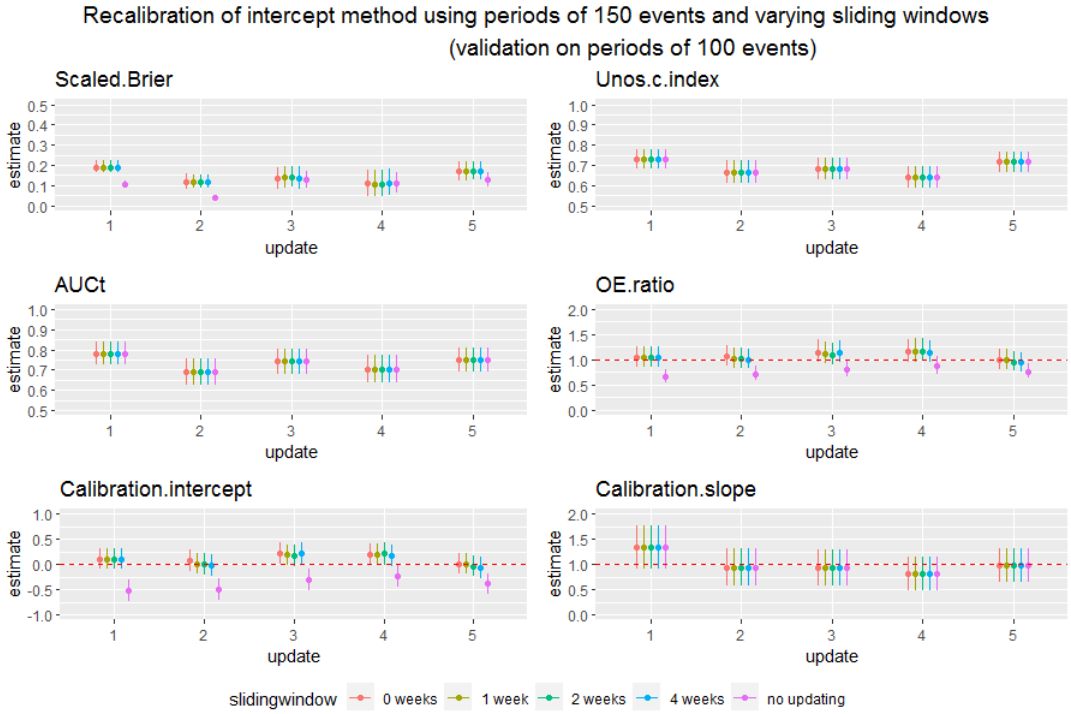

Figure D.6: Recalibrating the intercept every 150 events. Using varying sliding windows and also showing the performance of the no-updating method. At each update the model is validated on the next period of 100 events. For the Scaled Brier score, c-index and AUCt, higher values reflect better performance. For the OE-ratio, calibration intercept and slope, preferred values are close to the horizontal red line (so close to 0 or 1).

D.1.3 Bayesian updating

Again, we present three plots, one for each of the three different lengths of updating periods (Figure D.7, D.8 and D.9). In each figure, the performance metrics are shown at each update for the different sliding windows.

Similar to the two previous methods, the discrimination performance when using the Bayesian method stay similar to the performance of the no-updating method. This seems to be also the case for the calibration measures, this is different from what we saw before in the previous two methods.

We can also note that when updating more frequently, so using shorter updating periods, the performances seem to fluctuate more throughout the updates, for all the performance metrics. This stays visible for the overall performance (scaled Brier) when using longer updating periods.

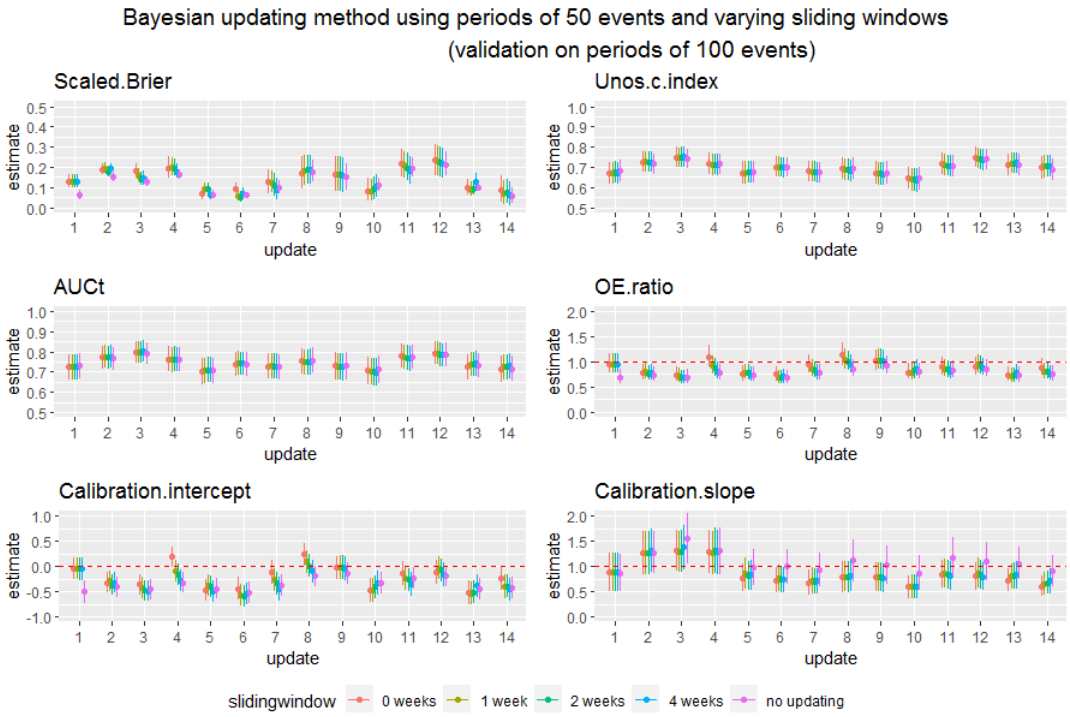

Figure D.7: Bayesian updating every 50 events. Using varying sliding windows and also showing the performance of the no-updating method. At each update the model is validated on the next period of 100 events. For the Scaled Brier score, c-index and AUCt, higher values reflect better performance. For the OE-ratio, calibration intercept and slope, preferred values are close to the horizontal red line (so close to 0 or 1).

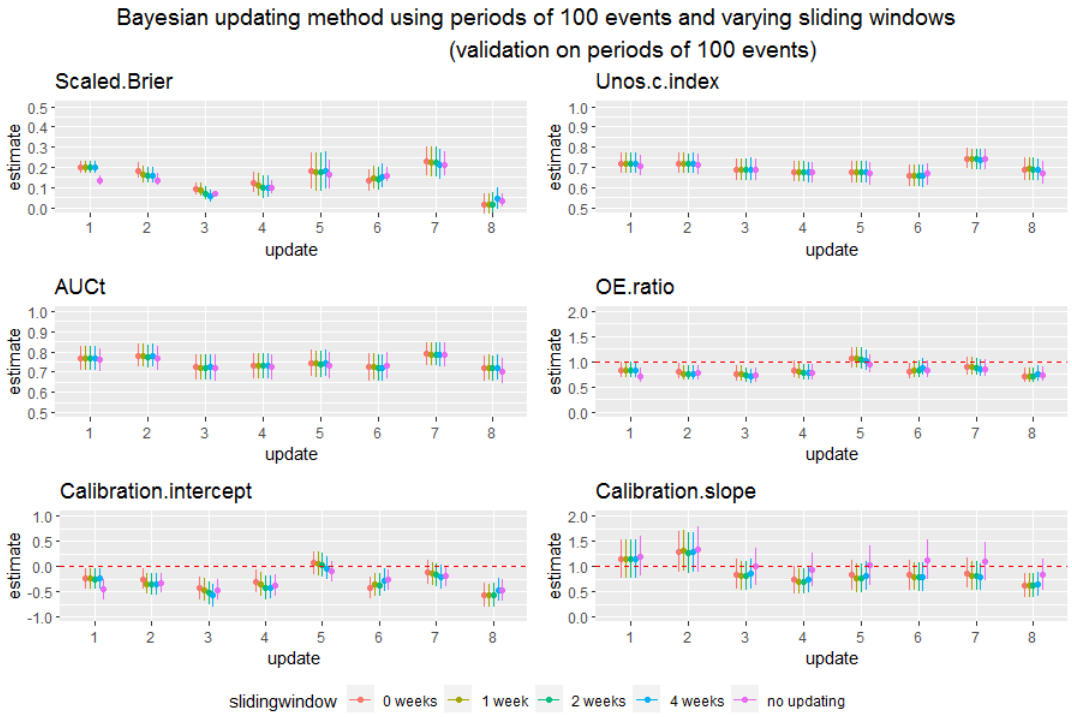

Figure D.8: Bayesian updating every 100 events. Using varying sliding windows and also showing the performance of the no-updating method. At each update the model is validated on the next period of 100 events. For the Scaled Brier score, c-index and AUCt, higher values reflect better performance. For the OE-ratio, calibration intercept and slope, preferred values are close to the horizontal red line (so close to 0 or 1).

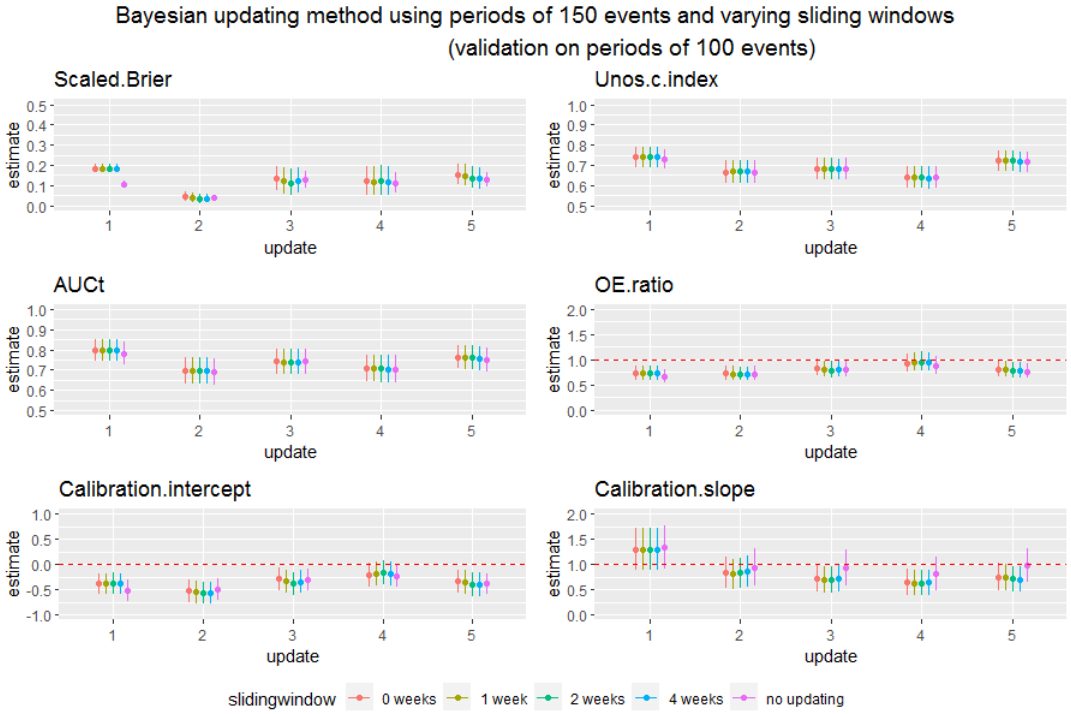

Figure D.9: Bayesian updating every 150 events. Using varying sliding windows and also showing the performance of the no-updating method. At each update the model is validated on the next period of 100 events. For the Scaled Brier score, c-index and AUCt, higher values reflect better performance. For the OE-ratio, calibration intercept and slope, preferred values are close to the horizontal red line (so close to 0 or 1).

D.2 Updated model coefficients

In this section figures are given that show the behaviour of the updated coefficients for the refitting and Bayesian dynamic updating method. The predictors are age (at admission), sex, number of comorbidities (numb comorb), respiratory rate per minute (RR...min), peripheral oxygen saturation percentage (SpO2...), c-reactive protein mg/l (CRP.mg.L.), and creatinine μmol/l (Creatinine.SER.µmol.L). The behaviour of the updated baseline hazard is also shown for the Bayesian method, the Bayesian method was the only method that modeled the baseline hazard parametrically. For the recalibration of the intercept method, the coefficient are untouched throughout the updates and thus equal to the coefficients given by no-updating.

D.2.1 Refitting method

In Figure D.10, D.11 and D.12 the behaviour of the updated coefficients for the refitting method are shown.

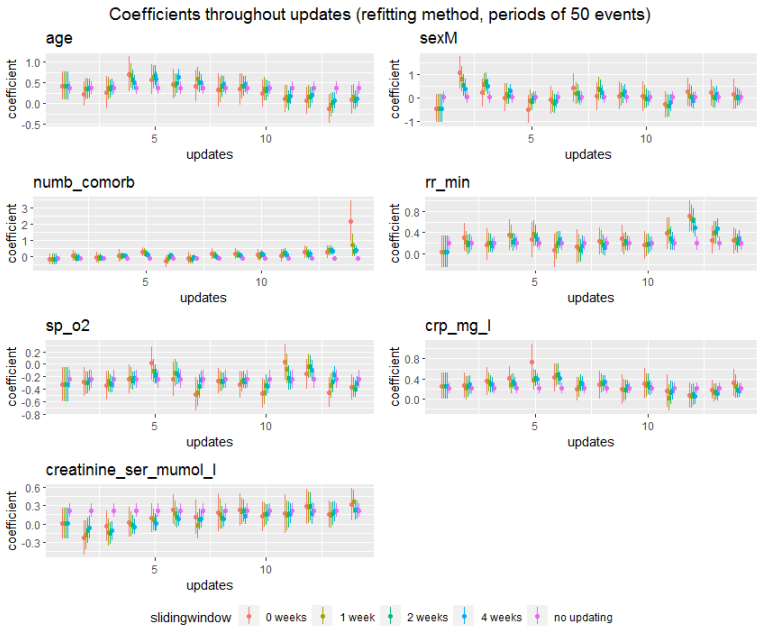

Figure D.10: Visualisation of how the estimated coefficients of each of the predictors change throughout the updates using the refitting method every 50 events with different sliding windows. For reference the estimated coefficients of the original model (no-updating) are also shown at each update.

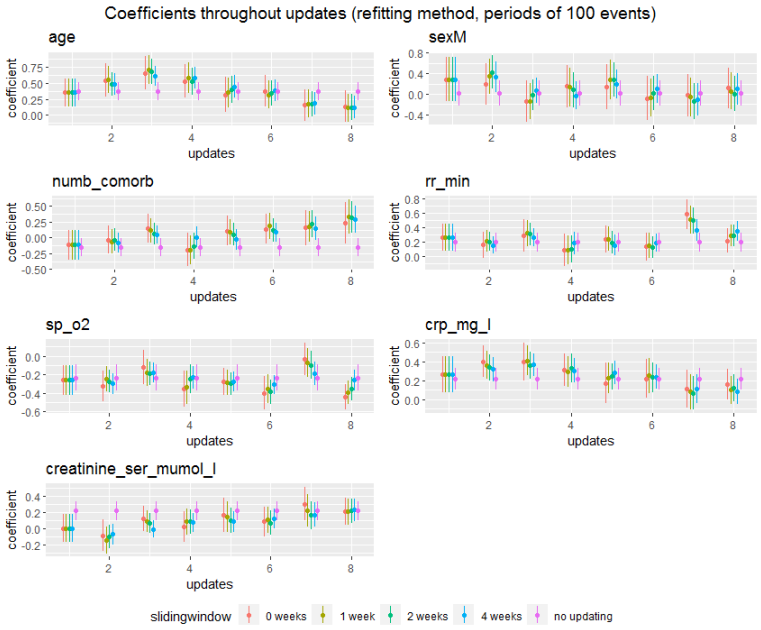

Figure D.11: Visualisation of how the estimated coefficients of each of the predictors change throughout the updates using the refitting method every 100 events with different sliding windows. For reference the estimated coefficients of the original model (no-updating) are also shown at each update.

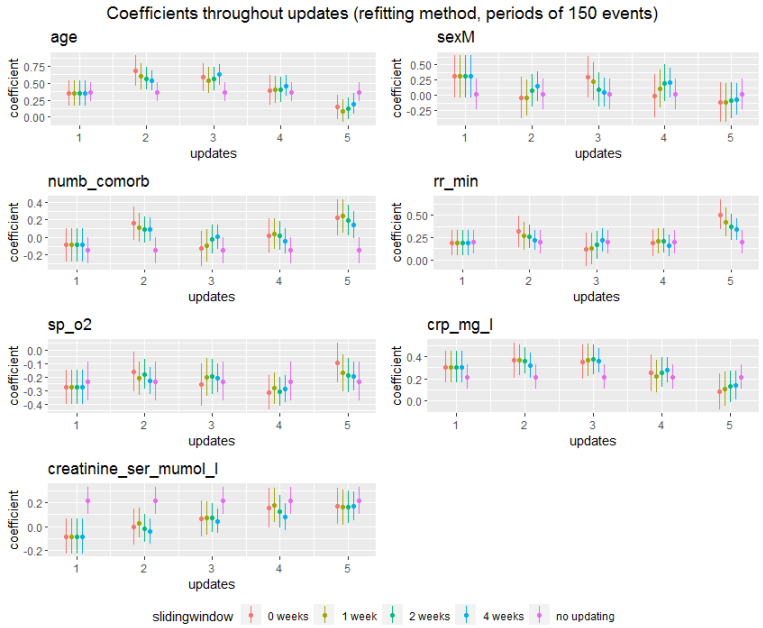

Figure D.12: Visualisation of how the estimated coefficients of each of the predictors change throughout the updates using the refitting method every 150 events with different sliding windows. For reference the estimated coefficients of the original model (no-updating) are also shown at each update.

D.2.2 Bayesian updating method

In Figure D.13, D.14 and D.15 the behaviour of the updated coefficients for the Bayesian updating method are shown.

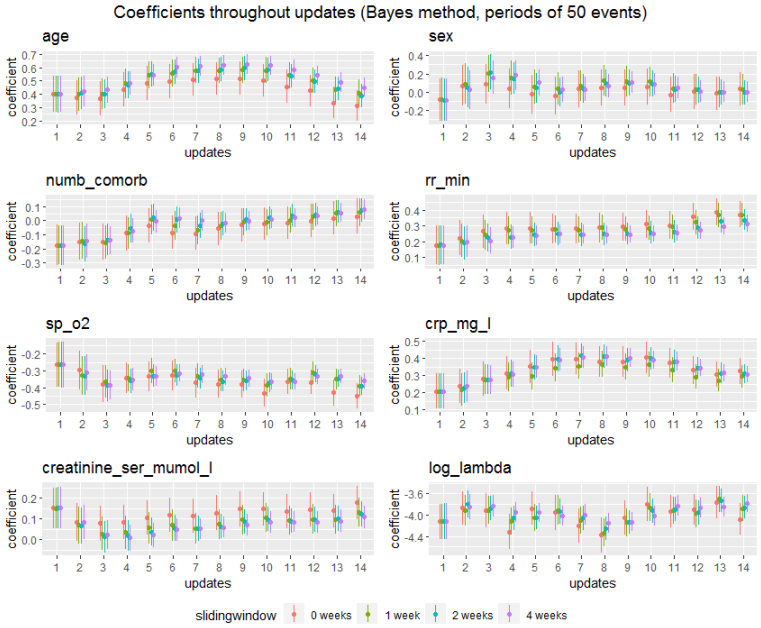

Figure D.13: Visualisation of how the estimated coefficients of each of the predictors and the baseline hazard (log lambda) change throughout the updates using the Bayesian updating method every 50 events with different sliding windows. For reference the estimated coefficients of the original model (no-updating) are also shown at each update.

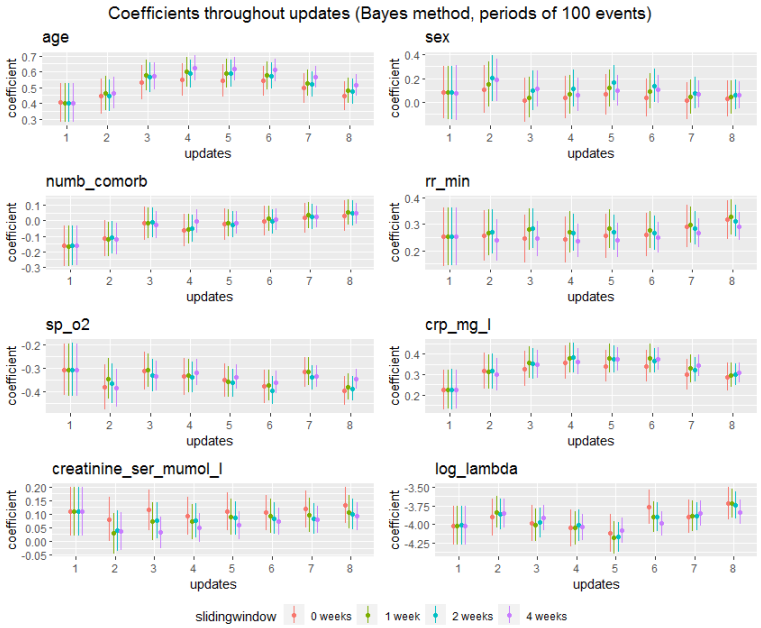

Figure D.14: Visualisation of how the estimated coefficients of each of the predictors and the baseline hazard (log lambda) change throughout the updates using the Bayesian updating method every 100 events with different sliding windows. For reference the estimated coefficients of the original model (no-updating) are also shown at each update.

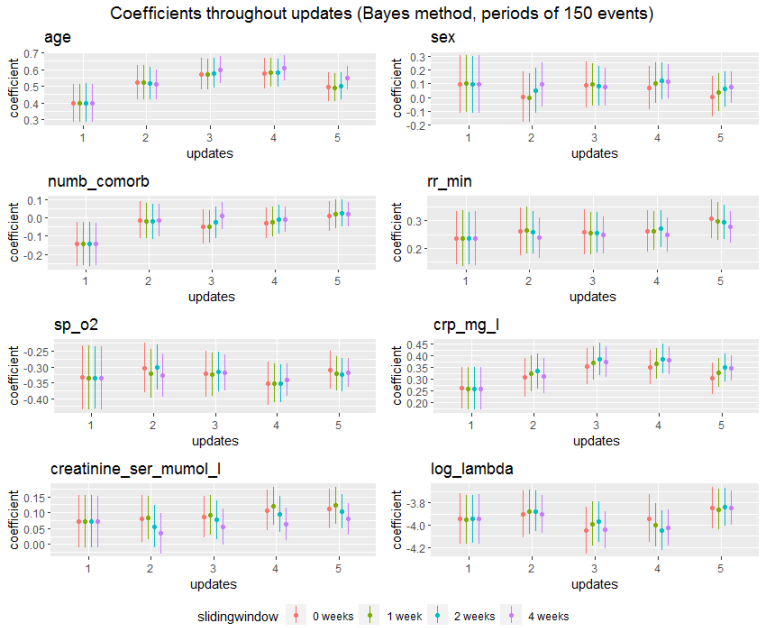

Figure D.15: Visualisation of how the estimated coefficients of each of the predictors and the baseline hazard (log lambda) change throughout the updates using the Bayesian updating method every 150 events with different sliding windows. For reference the estimated coefficients of the original model (no-updating) are also shown at each update.

D.3 Individual predictions

In our analysis we focus on analysing how the updated models perform in a new dataset. One could also be interested to see if for individual predictions the updating method, the length of updating periods and the size of sliding window also have an influence.

In Figure D.16 the difference is shown between the predicted risk from the refitting method and the recalibration of the intercept method, after updating the original model once using a period of 50 events with no sliding window, for the individuals of the next period of 100 events. When the difference is larger than 0, the predicted risk from the refitted model is higher than the predicted risk from the recalibrated model. When the difference is smaller than 0, the predicted risk from the refitted model is lower than the predicted risk from the recalibrated model. If there would be no difference in the predicted risk between the two updating methods, all points would be at 0. We see that this is not the case, meaning that the individual predictions also vary for the different updating methods. This demonstrates that for individual predictions the updating method is also of influence.

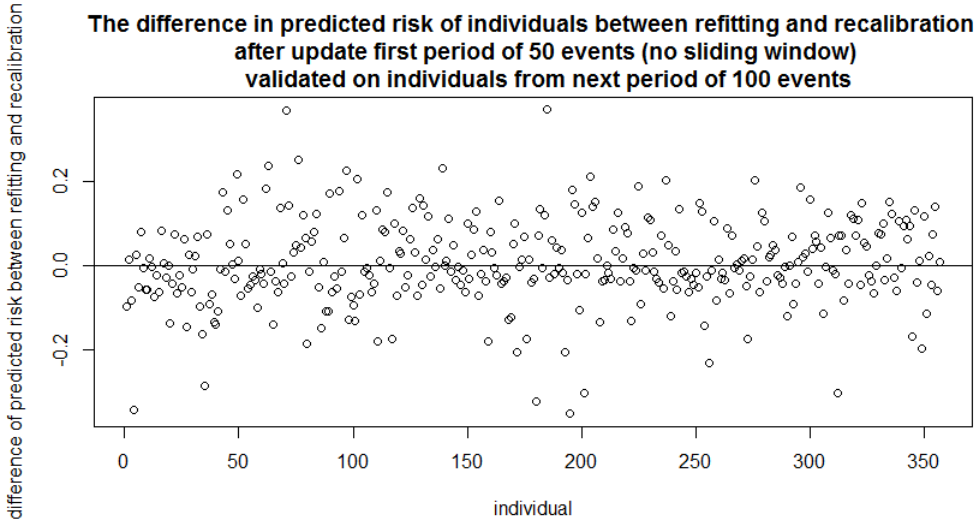

Figure D.16: The difference in the predicted risk for individuals between the refitting method and recalibration method is shown. Based on the first model update (using updating periods of 50 events and no sliding window) predictions are made for the individuals of the next period of 100 events.

Appendix E

R code

The R-code can be found on https://github.com/ClaudineStark/Master-Thesis-Dynamic-Updating-of-Survival-Prediction-Models.