Universiteit Leiden
The Netherlands

Computer Science

A Random Forest Approach for Dealing with Missingness: a Case Study in Primary Care Data

Teddy Etoeharnowo

Supervisors:
Dr. M. van Leeuwen (LIACS, Leiden University)
Dr. S. Verberne (LIACS, Leiden University)
Mr. H.J.A. van Os (Department of Neurology, Leiden University Medical Center)

Master Thesis

Leiden Institute of Advanced Computer Science (LIACS)
www.liacs.leidenuniv.nl

28/03/2020

Abstract

In this master's thesis we propose a novel random forests method developed specifically for dealing with missing data, when classical methods of handling with missing data such as imputation are undesirable due to the introduction of bias. We compare this novel method, called *Lost in the Forest* (LITF), with classical random forests methods trained on data in which missingness has been handled in various ways: imputation via either the mean, *k*-nearest neighbors or Multivariate Imputation by Chained Equations, and using a dummy variable that indicates missingness of a value. We first perform this comparison on a large routine primary care data set, predicting five-year risk for cardiovascular events. Imputation of important risk predictors in these data yields biased results when compared with the de facto golden standard, indicating a possible *missing not at random* mechanism for missingness. Although the performance of LITF is comparable to that of other random forests methods, it provides additional insight in the contribution to the prediction of continuous variables without the need for imputation, and thus avoiding possible bias. Validation of our methodology on two clinical validation data sets with varying extent of missingness confirms these results.

Keywords: Lost in the Forest, Random Forest, Missing Data, Missing not at Random, Imputation, Dummy Variable, Primary Care Data

Acknowledgements

I would like to express my deep gratitude to my supervisors, Matthijs van Leeuwen, Suzan Verberne and Hine van Os, for the support and constructive criticism I received during this thesis. I am grateful to have the opportunity to work in cooperation with the Leiden University Medical Center. And I would like to thank everyone at LIACS for teaching me so many things during my time at the university.

I thank all of my friends for their continuous support that has been shown to me. Finally, I would like to thank my family for encouraging me in all of my pursuits and inspiring me to follow my dreams.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

A total of 38 119 deaths due to cardiovascular diseases (CVDs) were recorded in 2017 in the Netherlands, 18 080 of which were men and 20 039 women. This corresponds to an average of 105 per day. On average more than 700 people are hospitalised nationwide every day because of CVDs [1]. Although big steps are being made in the prevention of CVDs, the aforementioned numbers will increase in 2020, due to the aging of the population. [2].

Currently, primary prevention of CVD is carried out by general practitioners (GPs). First, GPs determine the risk of stroke in patients by using the cardiovascular risk management (CRVM) framework [3]. This framework contains instructions for a GP to collect measurements for relevant predictors of stroke, and to estimate the 10-year risk by comparing these measurements to predetermined risk values. The cardiovascular disease risk in patients is determined using a model based on the SCORE-system, which uses risk factors such as sex, age, smoking, systolic blood pressure, and the ratio of total cholesterol / HDL-cholesterol. The SCORE-system is currently decisive in whether to start pharmacological interventions, e.g. statins (lowering of cholesterol) and/or blood pressure medication. However, many more important risk factors for stroke exist, such as a family history of CVDs, migraines, and dementia [4].

With recent innovations in data science and machine learning, it is now possible to account for far more predictors, potentially enabling more accurate predictions of clinical outcomes. In addition, by directly using data from EHRs, automated integration of algorithms in the clinical workflow becomes a possibility, as input for these algorithms are data in EHR format.

Although a larger number of predictors may yield added predictive performance, in routine data many predictor values are only measured in a fraction of the population, leading to a large extent of missingness in the data set. As the majority of (non-Bayesian) prediction models currently in use for prediction cannot handle missing values, one should account for missingness in the analysis pipeline. Not only the extent of

missingness can be problematic, also the mechanism of missingness should be taken into account. Missing data problems can be classified into three categories [5]:

- *missing completely at random* (MCAR) occurs when the missingness of a predictor is unrelated to the values of another predictor or the predictor itself.
- *missing at random* (MAR) occurs when the missingness of a predictor is related to the values of another predictor and not the predictor itself.
- *missing not at random* (MNAR) occurs when the missingness of a predictor is related to the actual values of the predictor itself.

EHR data are almost exclusively collected through routine care practice. This means that, e.g., measurement variables generally only contain values if a medical professional deemed it necessary to perform a measurement, which is a form of reporting bias. Reporting bias in epidemiology is defined as "selective revealing or suppression of information" [6]. This generally results in a varying extent of missingness in different variables, with a mechanism that is most likely MNAR. In contrast to MCAR, the observed data are likely not representative of the population.

In clinical research, dealing with missingness of variables is generally resolved by omitting variables with a large extent of missingness, performing complete case analysis (omitting cases with missing values) or imputation. Imputation is in many cases the favoured option, utilizing as much data as possible, but Fielding [7] showed that imputation is inadequate if MNAR is the likely underlying misingness mechanism, as this results in biased imputed values.

This thesis introduces a novel classification method based on random forests called *Lost in the Forest* (LITF), which for continuous variables can take present values into account without the need to impute missing values. This can be especially valuable in situations where missingness is assumed to be MNAR. We will first apply this methodology in a case study using a large routine primary care data set with a large extent of missingness of important measurement variables, and further we will validate these methods on two open clinical data sets with a varying extent of missingness of important variables. We will then test whether this novel method is competitive with classical random forest (RF) algorithms regarding discriminative performance, and also investigate added value regarding insight in the data by comparing feature importances of LITF and classical RF algorithms. This study will therefore address the following research question: how can we improve random forests to handle missing data in clinical data?

This thesis is written as a master's thesis at the Leiden Institute of Advanced Computer Science (LIACS), the computer science institute of Leiden University with the cooperation of the Leiden University Medical Center (LUMC) for expert knowledge and data used for this thesis.

Chapter 2 presents related works in literature. Chapter 3 describes the data used in this thesis. Chapter 4 will outline how different methods that will be used within this thesis, including the newly developed *Lost in the Forest*. Chapter 5 describes the results. Chapter 6 presents the discussion. Finally, Chapter 7 draws the final conclusions and points out future work.

Chapter 2

Related Work

This thesis will use random forest methods to predict CVDs while handling missing data using different methodologies. This chapter will describe several works in the literature that have addressed similar challenges. In this chapter, we discuss (1) random forest methods, (2) how missing data can be dealt with, and finally (3) how machine learning has been used for primary care data.

2.1 Random Forest

Random forest (RF) [8] is a machine learning method that can be used for both classification and regression. It is able to work with large data sets and uses multiple decision trees to make predictions with.

The first version of the random forests algorithm was proposed by Ho [9] in 1995. Since then, new random forest implementations have been developed, some recent implementations include:

- Adaptive RF (ARF) [10]. This is a variation RF algorithm that can adapt to different types of concept drifts. ARF achieves this without the need for complex techniques.
- Bernoulli RF (BRF) [11]. This novel RF framework uses Bernoulli distributions for the construction of the decision trees. This framework has been proved to have theoretical consistency.
- Probabilistic RF (PRF) [12]. Instead of treating features as deterministic quantities, PRF treats them as probability distributions to account for possible noise in the data.
- Geographical RF (GRF) [13]. This implementation uses an ensemble of decision trees, of which each are constructed using data from only a geographical subspace.
- several random forest implementations that optimize speed and/or memory usage [14–16].

RFs have frequently been used before in the context of medical data, examples include the following works. DuBrava et al. [17] used RF models to find features in electronic health record data that are important for the prediction of peripheral neuropathy caused by diabetes. Alam et al. [18] proposed a model that, using an RF based predictor, can identify important features in clinical data sets for different diseases. Casanova et al. [19] suggested that using an RF can be used for the assessment of retinopathy caused by diabetes. Kumar [20] showed that an RF can be used for the prediction task of chronic kidney disease. He showed that the RF classifier outperforms five other classifiers. Maroco et al. [21] compared ten classifiers to predict dementia. From their analysis, in which they have taken into account multiples classification evaluation metrics, they showed that RF and linear discriminant analysis perform best among all.

2.2 Dealing with Missing Data

Unfortunately, missing data occurs in a lot of medical studies. In some studies the extent of missingness may be higher, e.g. in studies in which certain values are difficult to measure or in studies in which patients may drop out prematurely. Because of this, results that are derived from data from these types of studies may be unreliable [22].

Several methods have been proposed to deal with missing data. This thesis will compare our proposed method to methods using nearest neighbour estimation [23] and multivariate imputation by chained equations (MICE) [24], as these imputation methods have been shown to have very promising performance [25,26].

Waljee et al. [27] compared the performance of different imputation methods for data with missing values. They showed that MICE performed better than the nearest neighbour imputation method.

An alternative to imputing the missing values is to use a placeholder for the missing values. Twalaa and Cartwright [28] proposed a method for decision trees for dealing with missing data. This approach uses a method that is very closely related to the missing indicator method [29], in which "missing" is treated as a category in and of itself. Twalaa and Cartwright have shown that their method performs well on different data types.

The method that we introduce in this thesis involves a different way of constructing the decision trees. Beaulac and Rosenthal [30] also proposed a different decision tree construction approach for handling missing values. Predictors with missing values can be handled as this approach will partition the data according to that predictor only in spaces where it does not contain missing values. In the case of our case study, which has a high ratio of missing values, this comes with problems as there is very little feature space in which there are no missing values for some specific predictors. This means that some predictors are never utilized, although predictors that are mostly unavailable may potentially give the most information.

2.3 Machine Learning for Primary Care Data

Machine learning shows promising results in providing predictions using primary care data [31–33]. Weng et al. [34] showed that using machine learning improves the performance of cardiovascular risk prediction.

Instances exist within primary care data in which machine learning algorithms did not outperform less complex statistical models, like logistic regression [35, 36]. Obermeyer and Emanuel [37] state, however, that machine learning will become an even more important role in the future for clinicians, as the medical field becomes more complex.

Chapter 3

Data

The data used in this thesis have been collected from around 250 general practitioner practices in the Extramural Leiden Academic Network (ELAN). These are primary care practices that are connected to the Leiden University Medical Center. The data are collected and cleaned by Stizon (Foundation for Information Provision for Care and Research).

3.1 Data Exploration

The data set consists of multiple files, each containing data collected for a specific registration category: demographic information, diagnosis, medication prescription, blood pressure, smoking, renal functions, blood laboratory values or referrals. Each of these files has more than ten columns, but not all columns contain useful information for this thesis. The columns that are used are shown in Table 3.1.

File	Original # columns	Used columns
Patient	24	PID, sex, entry date, exit date, birth year, year of death
Journal	13	PID, start date, symptom
Symptom	14	PID, start date, symptom
Blood pressure	15	PID, memo, value, test date
Blood lab values	14	PID, test date, memo, value, flag
Renal function	14	PID, test date, memo, value, flag
Body mass index	15	PID, test date, memo, value, flag
Medication	16	PID, prescription date, drug code
Referral	12	PID, referral date, specialism
Smoking	11	PID, memo, value, test date

Table 3.1: The original files have a lot of columns, only a few of these are used in this thesis. The ones that are used are shown in this table. PID is the patient identification number.

An overview of the amount of entries, unique patients and unique, so-called, "codes" in each of the files can

be found in Table 3.2. Each code represents a specific type of medical event, measurement, referral or drug, classified by clinical coding. Examples of codes include: *K87* for hypertension complicated, *A10* for drugs used in diabetes, *RRSYKA* for systolic blood pressure. The number of unique codes and patients differs considerably between the files.

File	Entries	Unique patients	Unique codes
Patient	206 303	132 121	N/A
Journal	32 388 034	105 757	2 389
Symptoms	4 092 202	105 619	2 421
Blood pressure	1 105 524	43 931	35
Blood lab values	5 763 405	63 837	643
Renal function	224 809	41 260	6
Body mass index	827 467	29 491	6
Medication	28 847 812	96 967	1 627
Referral	2 289 352	88 945	4 687
Smoking	267 199	23 548	3

Table 3.2: Number of entries, unique patients and codes in each of the data files

None of the files covers all patients in the database. For example, the blood pressure file covers a mere 33% of the patients. As a result, many patients in the database seem to have little to no data. To test if the data points are clustered on a small number of patients in our data set, a missingness plot has been made after duplicate patients and patients that are not eligible for the data set are removed and features that are uncommon (presence in patients $\leq 0.5\%$) are dropped, leaving 263 features for further analysis. The results are shown in Fig. 3.1. Shown in the figure is how most data comes from a small portion of the patients and how many missing values the data actually contain.

To further analyse patterns of missingness in the data, the presence of different measurements is studied for different subpopulations. These are subpopulations by certain medical diagnoses or age. Patients younger than 50 are shown to have a lower change of CVDs [1]. To see if this also correlates with missingness in measurement data, this age group is included in the table. The second age interval of 35 to 85 is chosen, because these are the patients that would benefit mostly from a predictive model. When patients are younger, their risk is very low, and when older, they are already under increased medical supervision. Therefore, this subpopulation will be used for the experiments.

Table 3.3 displays the ratios of missingness per subpopulation. As seen from the table, missingness is very high in the total population, ranging between 87% and 95%. Missingness is much lower in the subpopulation having hypertension and cerebral infarction. Patients younger than 50 years have a larger missingness ratio than average, and patients with the age between 35 and 85 have a lower missingness than average. From this data it can be concluded that certain subpopulations are overrepresented in the data in terms of measurement values. The data is clustered on a certain group of patients.

The characteristics of the study patients are summarized in Table 3.4. In the table, the patients are divided

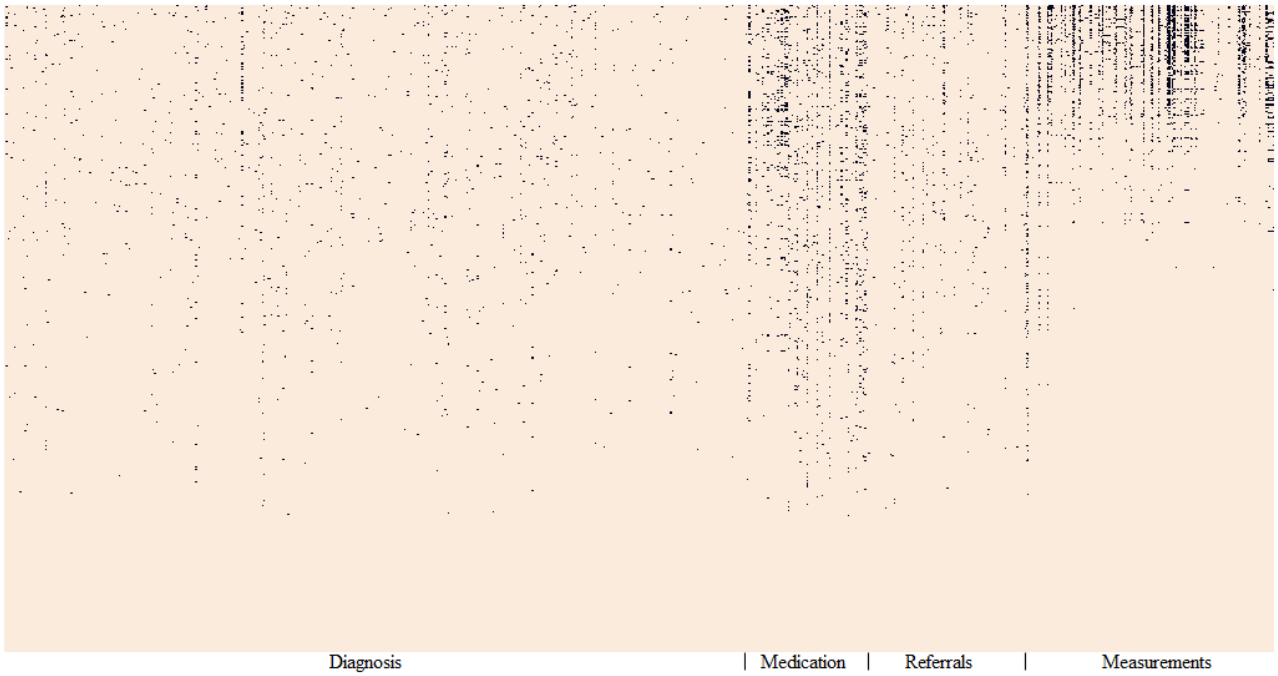


Figure 3.1: This graph shows the presence of events and measurements in the data using black dots. On the y-axis are the different patients sorted to have the number of events and measurements in decreasing order. The x-axis is grouped by diagnosis, medication, referrals and measurements.

	all	Hypertension (uncomplicated and complicated)	Diabetes mellitus type 2	Age ≤ 50	$35 \leq$ age ≤ 85
Number of patients	54989	2737	1278	36852	30180
Systolic blood pressure	0.88	0.20	0.17	0.95	0.80
Cholesterol	0.90	0.43	0.29	0.97	0.83
Fasting glucose	0.92	0.61	0.28	0.97	0.88
Non fasting glucose	0.95	0.83	0.53	0.98	0.92
Kreatinine	0.87	0.37	0.22	0.94	0.80

Table 3.3: Missingness ratio of the measurements are shown in the rows. Each column is a subset of the population having a certain feature.

into two groups: patients with main cardiovascular event (cases) and without main cardiovascular event (controls). Note how the male proportion is larger in the cases group. The two groups differ not only in measurement values but also in the ratio of missingness. This shows that having only the missingness indicator as a feature can already increase classification performance.

	Controls $N = 3055$	Cases $N = 611$
Proportion Male	48.61%	62.03%
Averages		
Age	53.25	62.82
Systolic blood pressure	83.77	82.15
Diastolic blood pressure	142.63	146.14
Cholesterol total	5.21	5.21
Glucose fasting	5.77	6.36
# Events or measurements present (percentage)		
Ischaemic heart disease w/ angina	17(0.6%)	9(1.5%)
Atrial fibrillation/flutter	16(0.5%)	8(1.3%)
Hypertension uncomplicated	251(8.2%)	70(11.5%)
Hypertension complicated	11(0.4%)	28(4.6%)
Diabetes non-insulin dependent	130(4.3%)	71(11.6%)
Lipid disorder	38(1.2%)	10(1.6%)
Drugs used in diabetes	124(4.1%)	76(12.4%)
Diuretic drugs	225(7.4%)	90(14.7%)
Beta blocking agents	264(8.6%)	119(19.5%)
Calcium channel blockers	113(3.7%)	64(10.5%)
RAS-acting agents	326(10.7%)	161(26.4%)
Lipid modifying agents	261(8.5%)	143(23.4%)
Smoking	227(7.4%)	82(13.4%)
Systolic blood pressure	565(18.5%)	197(32.2%)
Diastolic blood pressure	566(18.5%)	197(32.2%)
Cholesterol total	488(16.0%)	176(28.8%)
Glucose fasting	359(11.8%)	119(19.5%)

Table 3.4: Baseline characteristics of the patients in the data

3.2 Classification Task

Only the data between 2 January 2007 and 1 January 2017 are considered, since all measurements concerning blood pressure and medication are only available after 2 January 2007. The goal is to predict main vascular diseases, namely acute myocardial infarction, ischaemic heart disease without angina, and stroke/cerebrovascular accident.

Data of a patient will be collected for a period of time ("run in period"), and then a prediction will be made whether there will be a cardiovascular event in the next period ("prediction period"). Our data is only reliable for ten years, therefore a run in-period of one year and a prediction period of five years has been decided on.

Figure 3.2: This figure shows how a patient's EHR is divided into its run in and prediction period. The red and green marked areas show the run-in and prediction period, respectively. Years are put on the x-axis, and on the y-axis are examples of different patients. Patient 1, 2, 4 and 5 are eligible for the data, since they all are within the time frame for at least six years. Patient 3, 6 and 7 are not long enough in the time frame.

To have the machine learning algorithm learn from the given data a training set needs to be from one year and the classification should be from the next five years. A total of six years follow up. Figure 3.2 shows how the data of patient are divided into their corresponding run-in and prediction periods.

3.3 Sampling

A sample of the data set is taken for the experiments. Sampling is useful, because the whole data set is too large to analyze within reasonable time. Analyzing a representative sample is more computationally efficient and cost-effective than using the entirety of the data. Using the sampled data set the experiments for this thesis already takes multiple hours, more details about runtimes within this thesis are provided in Appendix A. The number of cardiovascular events is very low, only 611. All of these events are included in the sampled data set. 3055 uniformly sampled control patients are also included in the sampled data set, creating a case:control ratio of 1:5.

Only a certain set of important features is included, based on expert knowledge. The following measurements are used (with the respective percentages of missing values): systolic blood pressure (79.21%), diastolic blood pressure (79.19%), cholesterol (81.89%), and fasting glucose (86.96%). The full list of features is shown in Table 5.3.

3.4 Additional Data Sets for Validation

To validate our approach, two additional clinical data sets from the KEEL-data set repository [38], with varying amounts of missing values, are used.

3.4.1 Hepatitis Data Set

This data set contains information about patients affected by Hepatitis. The task is to predict if these patients will die or survive. This data set has 19 features, 155 instances, and 48.39% of the instances have missing values. The data set contains the following continuous measurements (with the percentage of missing values): Bilirubin (3.87%), AlkPhosphate (18.71%), Sgot (2.58%), AlbuMin (10.32%), and Prothrombin time (43.23%).

3.4.2 Mammographic Mass Data Set

This data set contains information related to the risk of developing breast cancer of patients. The task is to predict the severity of a mammographic mass lesion. This data set has 5 features, 961 instances, and 13.63% of the instances have missing values. The data set contains the following continuous measurements (with the percentage of missing values): Shape (3.23%), Margin (4.99%) and Density (7.91%).

3.5 Data Preprocessing

This section will describe how the data needs to be processed before it can be used by the prediction models. This section divided into two parts: (1) how the different files are merged and (2) how the data is prepared for our specific models. The various preprocessing steps are illustrated in Figure 3.3.

Merging the Files

The raw data is divided into eleven files. The features that are extracted from the files are divided into two groups: Events and Measurements. Symptom codes, medication and referrals are considered to be events. Measurements are values that describe the blood pressure, lab results, renal function and body mass index. Each file contains entries which corresponds to a specific measurement or event on a specific time for a patient. For the data set to be usable for the prediction model we need every entry to represent the medical history of a patient instead. Initially, the raw files are merged into one big file and patients are filtered out that are unsuitable for our training set. This stage is named "Merger" in Figure 3.3. "Merger" starts with reading in the patients file. For the classification analysis that will be done in this thesis all patients must have six years of followup. One year is for the run-in period and the last five years are for the prediction period.

When a patient exits the data, the date will be denoted by *exit date*. But a patient may pass away before the patient exits the data according to *exit date*. This is why the end of a patient's history is not determined

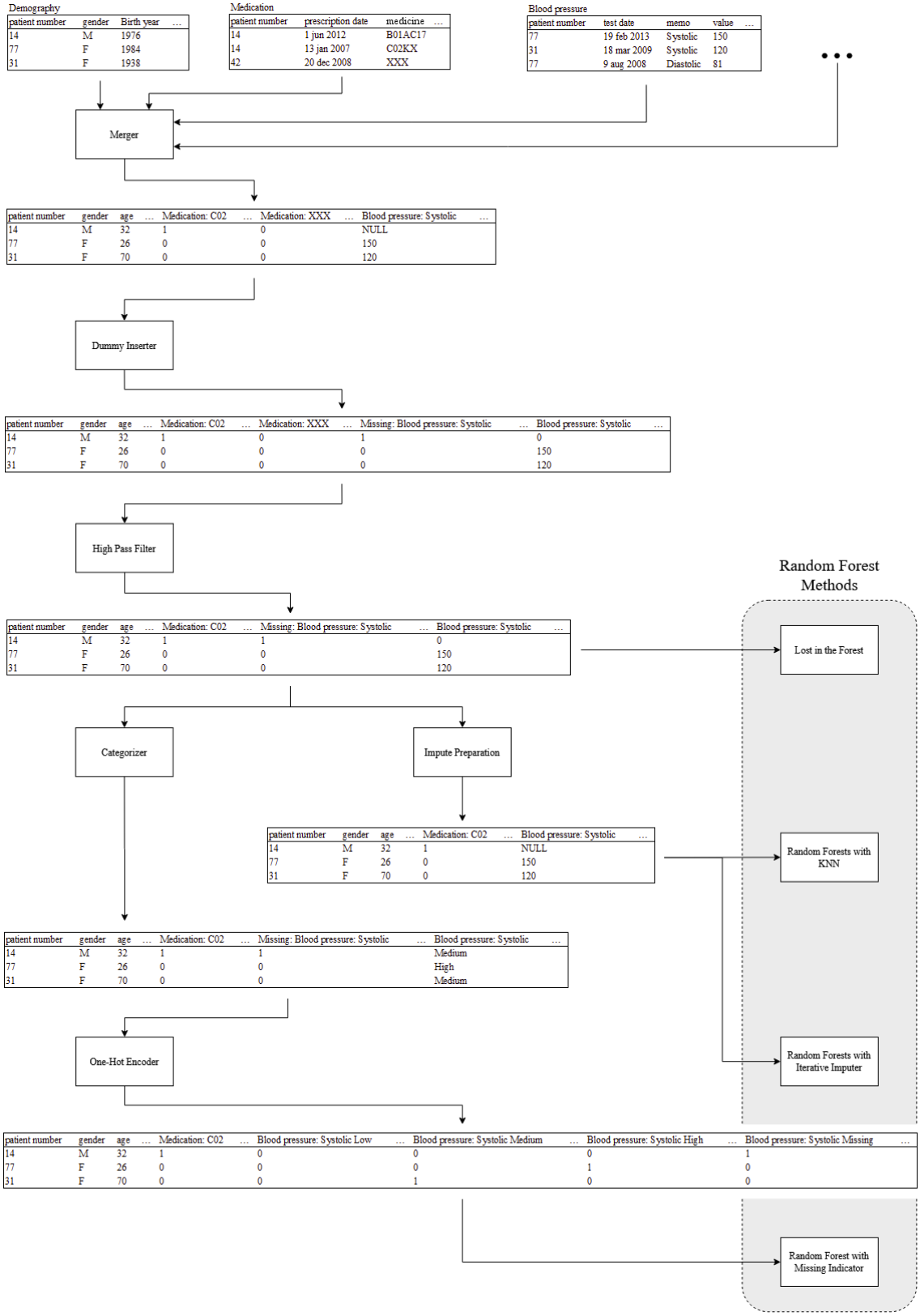


Figure 3.3: This is the pipeline for the preprocessing of the various primary care files to data sets that are usable for the different random forest methods used in this thesis. Not all random forest methods need to have the data to traverse all preprocessing steps. Only a fraction of the columns, rows and files are shown.

by only the *exit date*, but also the patient's *year of death*. Duplicates of patients are dropped, it is unknown why these entries exist, but for the sake of simplicity only the last entry is kept. The age of each patient is calculated at the end of the run-in period. This age is used to filter out patients younger than 35 and older than 85. For simplicity, only two sexes are used, sexes other than male of female are discarded.

Entries will have a feature for each event except for the main cardiovascular events that the methods will predict. Patients with a occurrence of a main cardiovascular event before the end of the run-in period will be dropped from the data set. This is because patients with a registered history of the event will receive special care and so the model that will be developed in this thesis will predict first instances of the main cardiovascular event.

Measurements consist of values and a patient can have multiple values for the same measurement, e.g. a patient can have his/her blood pressure measured multiple times within a time frame. When there are multiple values for the same measurement within the run-in period, the average is taken.

The main cardiovascular events are used to generate a target column. If such an event occurs within the prediction period, the target value will be 1, otherwise 0.

Data Transformation

Some data transformation needs to happen to make the data ready for the developed RF methods. The data transformation is done in several stages, these stages are called: "Dummy Inserter", "High Pass Filter", "Categorizer", "One-Hot Encoder" and "Impute Preparation". Here, we will shortly describe the transformations that take place. Not all methods need to have the data to traverse all preprocessing steps, as can be seen in Figure 3.3.

A dummy variable is added for every measurement column. The dummy variable is stored in the *missingness column*. This column denotes if the measurement is missing or not. The value of 1 is used for missing and 0 for not missing. In Figure 3.3, this is done by "Dummy Inserter". Events in the primary use case are considered to be never missing, because an event is either registered or it did not occur and is therefore not registered. So, events do not require dummy variables. Age and gender are always present due to the previous preprocessing steps. These features are not considered to be measurements in the scope of this project, because they do not contain missing values and do not need a dummy variable.

Some events and measurements occur in very few patients only. The "High Pass Filter" filters out features that are rarely or not used in the patients that are still in the data. Predictors that occur in less than 0.5% of patients are removed.

"Categorizer" divides the values of the measurements into nominal categories: high, medium and low. The

cutoff value of each category is either given by experts from the LUMC or generated by using the 25^{th} percentile for the low category and the 75^{th} percentile for the high category. The measurement values need to be discretized into nominal categories, because the different categories can not be ordered, since the missing value does have a numerical value attached to it. In the primary care use case, discretization is only needed for measurements. Other features do not need discretization because they are never missing.

Each measurement column will be split into multiple columns by "One-Hot Encoder". Each measurement will have four columns describing it at the end of this file: missing, low, medium and high. Each feature is binary.

"Impute Preparation" prepares the data for methods using imputation techniques. It inserts $NULL$ in the measurement column where a value is missing, based on the missingness column. Afterwards, the missingness columns are dropped.

Chapter 4

Methods

The methods considered in this thesis are all variations of the random forests method. This chapter explains what a random forest is, describes conventional strategies for dealing with missing data, introduces our new method *Lost in the Forest*, describes how feature importance is calculated, and, lastly, describes how the models are validated.

4.1 Random Forest

Random forest (RF) [8] is a machine learning method that can be used for both classification and regression. It uses multiple generated decision trees and makes a prediction based on the outputs of the individual trees. Using multiple decision trees, random forest is able to overfit less than a single decision tree alone.

The RF algorithm works in two stages: (1) random forest creation and (2) prediction based on the forest. The first stage uses a stochastic element to make the method more robust. In each node of a tree only n features are taken into consideration from all F features, with $n \ll F$. From these n features the best feature is chosen and this is used to split the node. The best feature is chosen based on, for example, Gini impurity. These steps are repeated until a tree is made with a desired depth. Moreover, these tree manufacturing steps are repeated to create multiple trees. The second stage makes a prediction based on the forest. This is done by taking the test features and then using the rules of each created tree to predict the outcome and storing the prediction as a "vote". For classification, is considered the prediction with the highest number of "votes". For regression, an average of the individual predictions can be used.

4.2 Conventional Strategies for Dealing with Missing Data

RFs have a built-in way of working with missing data, albeit potentially not a very effective one, that can be included in RF without any problem. In this technique the missing data is simply set to a certain value. This has the disadvantage of not being able to distinguish between an actual missing value and a value that is close to the value of missing data. There exist approaches that are better able to deal with missing data. In this section, two different approaches are examined that work with data with missing values: random forest with dummy for missingness, and random forest using imputation methods.

4.2.1 Random Forest with Dummy for Missingness (RFDM)

One way to deal with missing values is to represent a continuous variable containing missing values with a new binary dummy variable denoting whether a value is missing or not, 1 for missing and 0 for not missing. This method is developed as a baseline. Besides one dummy variable for missingness, three other binary dummy variables are introduced. That is, the observed values are binned into three nominal categories of high, middle, and low values, defined by expert-based cut-offs, and these categories are then converted to one-hot encoded features. When the cut-offs are unknown the categories will be based on the 25^{th} and 75^{th} percentiles. Although this will make the data less granular, it will not introduce biased data, like imputation would do. For example, when a measurement is considered to be high, the features are in states: $high = 1$, $medium = 0$, $low = 0$, $missing = 0$. And when a measurement is missing, the features are in states: $high = 0$, $medium = 0$, $low = 0$, $missing = 1$. To have the data in this format, the data has to go through "One-Hot Encoder" during the preprocessing steps, as shown in Figure 3.3.

In the experiments another RF (RFonlyDM) is included in the comparison. RFonlyDM only uses the dummy variable for missingness for all measurements and will not consider any measurement value. This is to show how important the missingness feature is and to show how much knowledge can be lost when values are not taken into consideration. So, this RF variation simply disregards every feature for a measurement value, except for the dummy variable.

Examples of possible decision trees using RFDM and RFonlyDM within a random forest are shown in Figure 4.1.

4.2.2 Random Forests with Imputation Methods

Two imputation strategies are used: *k*-nearest neighbors (*k*-NN) and Multivariate Imputation by Chained Equations (MICE), which are used by RFkNN and RFM, respectively.

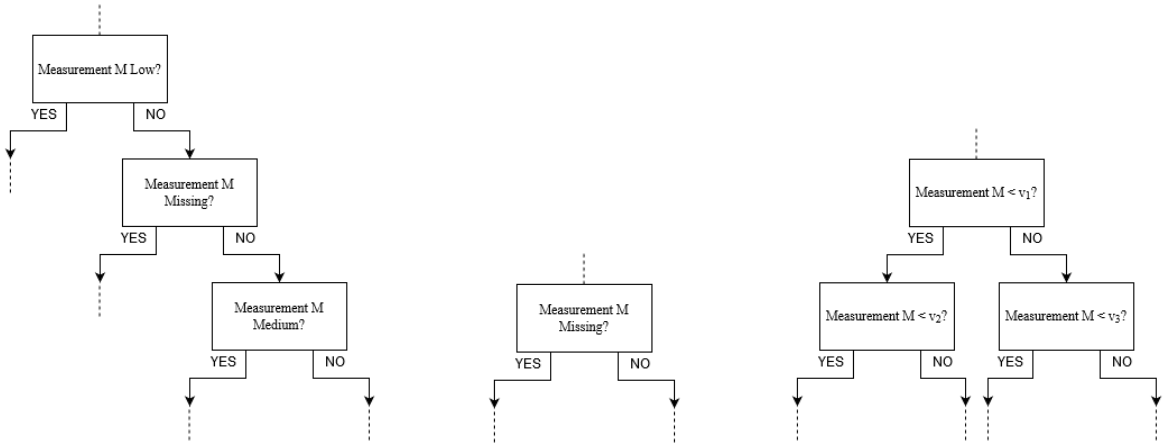
The *k*-NN algorithm [23] imputes missing values by based on the *k* closest neighbors which have the values present. The weighted mean of the nearest *k* neighbors are taken, where the distances (mean squared difference) to neighbors are used as weights, so the closer neighbor is, the more weight it is given.

MICE [24] is a strategy that imputes missing values in an iterated round-robin manner. The standard MICE strategy uses multiple imputations for each missing value to incorporate possible uncertainty in the imputations [39]. This will create multiple data sets to work with. This is not desirable in the context of this thesis, due to the time constraint. For this reason, this thesis will use a modified implementation based on the R MICE package, and differs from it by returning a single imputation instead of multiple imputations.

In this thesis the hyperparameters of MICE are set to their default values, most notable are: *maximum number of imputation rounds* = 10, *tolerance of the stopping condition* = 10^{-3}.

For this method the missing values in the data should be left empty so the imputation algorithms can fill in the empty spaces using the data that is available. No dummy variable is needed for this method. Accordingly, during the preprocessing steps the data has to go through "Impute Preparation", as shown in Figure 3.3.

An example of a possible decision tree using an imputation method within a random forest is shown in Figure 4.1.



(a) RFDM (b) RFonlyDM (c) After imputation

Figure 4.1: A comparison of conventional strategies for dealing with missing data using an example using only one measurement. (a) RFDM is able to use both the missingness and the bins as features to split on. (b) RFonlyDM is only able to use the missingness of a measurement as a feature to split on. (c) After imputation, measurements can be used as if there are no missing values.

4.3 Our Novel Approach: *Lost in the Forest* (LITF)

The results that are derived from the previously mentioned approaches are hard to justify. They either add possibly biased data, by using a imputation method that does not consider reporting bias, or make the data less granular. The new approach that we propose is a variation on random forests that does not use any imputation methods and does not have the need to divide the values into bins beforehand.

The main idea of *Lost in the Forest* is that it uses a dummy variable for the missingness of the measurement values, and it enforces to split on this missingness column *before* a split on the values of the measurements themselves are allowed.

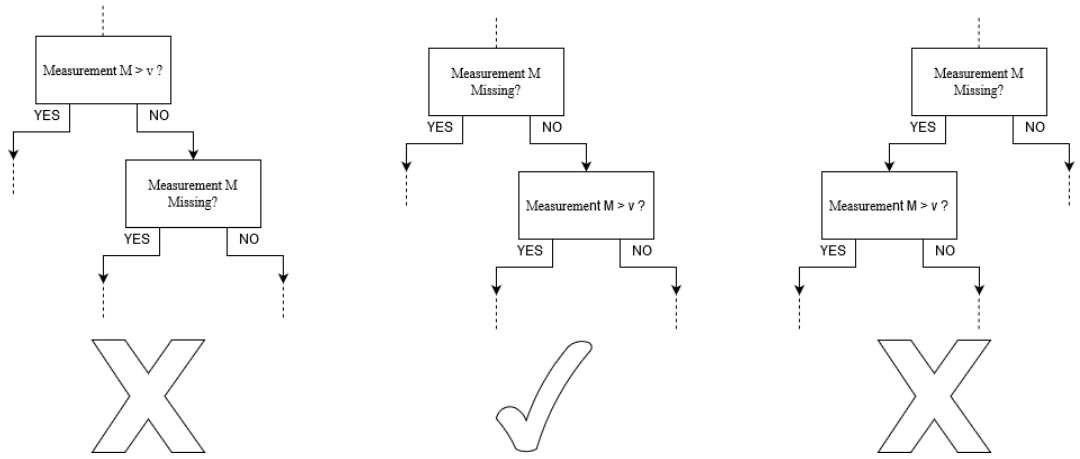
This allows this variation of RF to still make a distinction between when a value is missing or not, while still being able to make use of the available actual values.

Initially only the set of features that describe the missingness of measurements are considered by the RF for splitting. After the missingness feature of a measurement is chosen, the feature containing the measurement values are included in the set of features that the RF can split on. So, only once the "missingness" feature of a measurement has been "resolved" then the choice of the corresponding numerical feature is allowed.

After a split on a dummy variable, there are two branches: one in which the values are present and one in which the values are missing. To make sure there is no split after the branch in which the values are missing, every missing value should be set to one specific value (or $NULL$) as described by Groenwold et al. [29]. This ensures that a split on the value under this branch will give no information gain. The algorithm will then choose another feature to split on.

Unlike the approach that uses the dummy variable, LITF will only split on the actual value of a variable *after* the split of the dummy variable and *only* in the branch for which the dummy variable states that the value is not missing.

An example of a possible LITF tree, along with some examples of what is not possible in LITF, is shown in Figure 4.2.



(a) (b) (c)

Figure 4.2: (a) The left shows an impossible LITF tree; the value of a measurement can only be used after the "missing" feature of that measurement. (b) The middle tree is a possible LITF, because the the split on the measurement value is after the split on the "missing" feature. (c) The tree on the right shows a technically possible tree, but this tree will never occur since all missing measurement values are set to one value.

4.4 Feature Importance

To look further into how the random forests are structured, we will take a look at how the decision trees, within the forest, are structured. A feature importance measure will give insight into how the predictive features impact the variable of interest [40].

Feature importance is calculated by dividing the number of leaves under a feature by the total number of leaves in the decision tree. When the feature is used multiple times in a tree, only the highest value is considered.

In random forests the importance of a feature is the average of its importance in all individual trees in the forest. In this thesis *normalized* feature importances are used, which is done by unity-based normalization to bring all values into the range $[0, 1]$.

4.5 Model Validation

Nested cross-validation (CV) is needed when hyperparameters are tuned and classifier performance are measured simultaneously. Tuning hyperparameters using a non-nested CV will give unrealistic model performance [41]. Varma and Simon [42] showed that to be able to give a realistic estimate of a model a nested CV should be used.

We use the variant of CV as described by Wang et al. [43]. This method uses two loops of CV: one inner loop and one outer loop. The outer loop has the same function as a normal CV, it is used to evaluate the model. The inner loop is used to determine the best hyperparameter setting. The outer loop will determine the performance of the model using the hyperparameter setting chosen by the inner CV loop. In our implementation the number of folds for the outer and inner CV loop are denoted by k and l, respectively. When imputation is used, missing values are imputed between folds so each training set does not gain information of any data outside of the set. The validation set is imputed with knowledge about the training set. Figure 4.3 shows a schematic of our nested CV implementation.

The best hyperparameters are determined by choosing the setting with the highest AUC. AUC stands for the area under the curve [44], which here implies the area under the receiver operating characteristic curve. This curve shows the true positive rate against the false positive rate of a classification model at different classification thresholds. Accuracy only measures the percentage of points correctly classified for only a given classification threshold. The AUC considers all possible thresholds. AUC is preferred in this project as it provides a broader view of the performance of the model. AUC varies between 0 and 1, and a higher AUC means that the model is better at predicting.

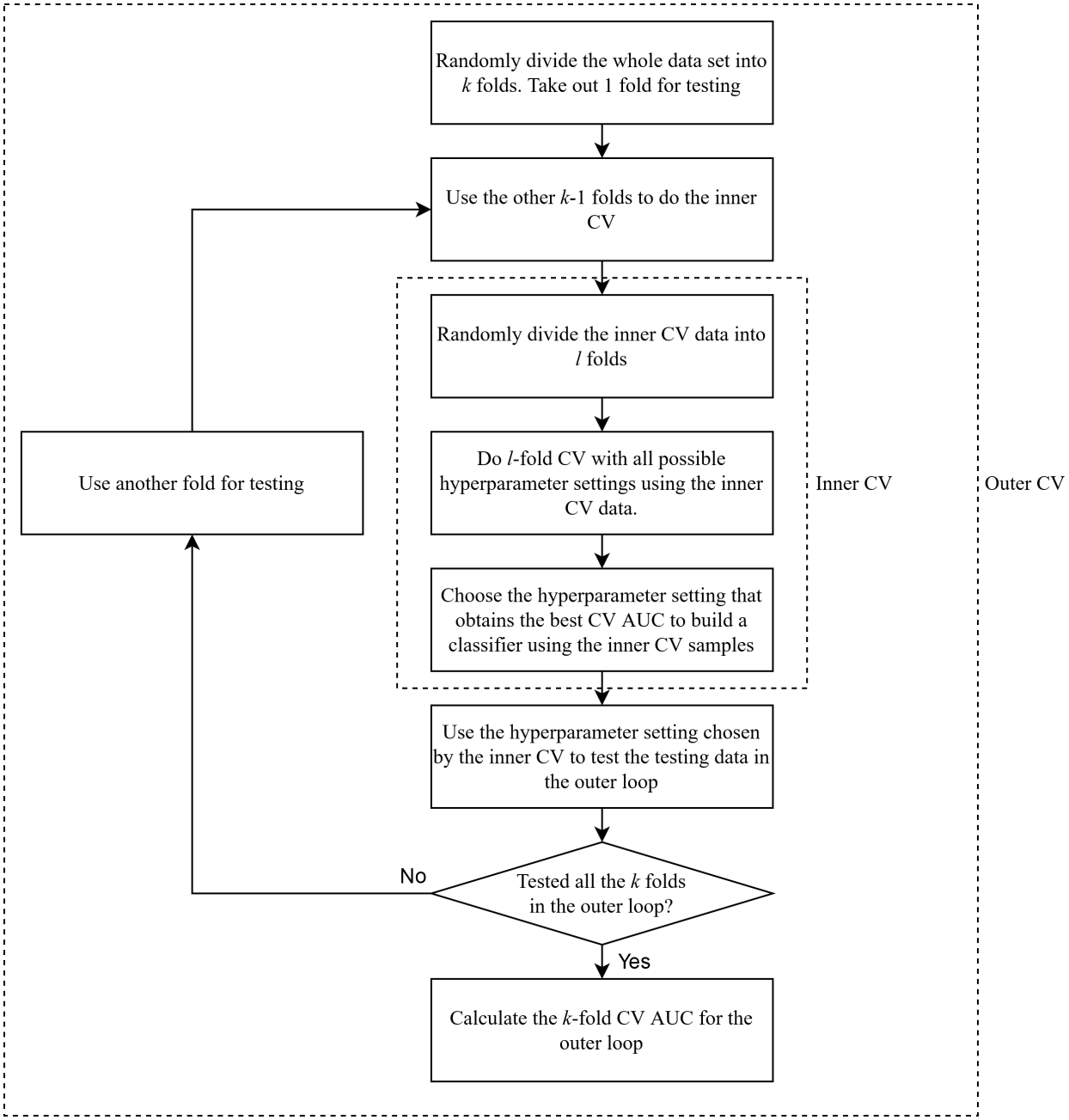


Figure 4.3: Schematic representation of nested cross-validation methodology.

4.5.1 Hyperparameter Settings

Hyperparameter settings are chosen using a grid search. This is a fairly simple method of tuning hyperparameters. The following set of hyperparameters are being adjusted:

- the number of trees in the forest of the model. A larger number of trees usually means that the random forest learns the data better.
- the maximum depth of each tree. When the tree is deeper it has more splits, and it will be able to extract more information from the data. Having deeper trees makes it easier to overfit on the data.
- the minimum number of samples needed for a leaf node. A split can only occur when there is more than the minimum number of samples in each branch. Increasing this value can cause underfitting.
- the number of features to consider when choosing for the best split. Increasing this too much nullifies the benefit of the randomness of the random forest, which helps in decreasing overfitting.

The hyperparameters were optimized using a grid with following values: *number of trees* $\in \{16, 32, 64\}$; *maximum depth of trees* $\in \{10, 20, 30\}$; *minimum number of samples* $\in \{1, 3, 10\}$; *number of features for the best split* $\in \{0.2 \times$ *total # features* , $0.5 \times$ *total # features*$\}$.

Chapter 5

Results

The experiments are divided into four sections: (1) imputation, (2, 3) performance and feature selection on the data of the primary care case study, and (4) performance on the other data sets. In our experiments, all methods are run using the nested $k \times l$-fold cross-validation scheme, with both k and l set to 5. This is chosen for a good balance between computational cost and solution accuracy. As mentioned, we will assess the performance by the AUC of the receiver operating characteristics curve.

5.1 Imputation Results

The following experiment evaluates the bias of the imputation methods. In this experiment the missing values of the primary care data are imputed and compared to the population and data averages. The average of the imputed values should, ideally, approach the population average. But this is a difficult task, because MNAR data are not likely representative of the population.

Table 5.1 shows the average imputed values by different imputation methods. This experiment only considers the measurements of the sampled data set, as described in Subsection 3.3. If the data would not be MNAR, the data would be representative of the whole population. The imputed values would then approach the population average. As seen from this table, the imputed values approaches the data average instead of the population average. This shows that the imputation methods are unlikely to be able to impute the values to the values that the patients actually have, due to the bias that MNAR data introduces.

	Population		Data		3NN		10NN		MICE	
Measurement	Mean	σ	Mean	σ	Mean	σ	Mean	σ	Mean	σ
Sys. blood pressure	128.70 [45]	15.63	143.94	17.53	137.01	12.26	136.78	9.90	141.05	8.24
Dias. blood pressure	77.88 [45]	9.88	83.40	9.47	82.34	6.29	82.52	4.35	84.05	1.48
Cholesterol	5.42 [46]	n/a	5.24	1.10	5.43	0.68	5.40	0.51	5.23	0.10
Fasting glucose	5.15 [47]	n/a	5.84	1.49	5.43	0.66	5.46	0.47	5.84	0.05

Table 5.1: This table shows the mean and standard deviation (σ) of the imputed values that were initially missing. The imputation methods used are Multivariate Imputation by Chained Equations (MICE) and k-nearest neighbor (3/10NN). These are then arranged next to the averages and the standard deviations of the population and the data. Data that are not available for the Dutch population are denoted with n/a

5.2 Predictive Performance

This experiment compares the performance of the different methods on the primary care case study data set. Table 5.2 shows the different AUCs of the different methods run on the the sampled set. As seen from the table, the methods that use a dummy variable for missingness (LITF, RFDM and RFonlyDM) have the best performance. This experiment indicates that using an imputation method for this data set will perform worse than using other methods to deal with the missing data.

Method	Mean AUC
LITF	0.70
RFDM	0.70
RFonlyDM	0.70
RF3NN	0.66
RF10NN	0.65
RFM	0.65

Table 5.2: The mean of the AUC on the CV test sets are displayed. The methods used are: *Lost in the Forest* (LITF), RF with dummy for missingness (RFDM), RF without any measurement value only using the dummy for missingness (RFonlyDM), and RFs with imputation methods: k-NN (RF3NN, RF10NN), and MICE (RFM).

5.3 Feature Importance Analysis

Table 5.3 shows the normalized feature importances of all features of all methods on the primary care data set. As the missingness increases, the importance of the value of the measurement decreases: diastolic blood pressure has the least values missing (79.19%) and it has the highest importance of the measurement values (0.02), next is systolic blood pressure (79.21%, 0.01, resp.), cholesterol (81.89%, 0.01, resp.) and fasting glucose (86.96%, 0.00, resp.).

With RFDM the dummy variable is just a little bit more important that the value itself. One special case is the Low Diastolic Blood Pressure, which has a higher importance than the dummy variable. The RFs using imputation have high feature importances for the measurement values. Note that around 80% of the measurement values were initially missing.

Feature	LITF	RFDM	RFonlyDM	RF3NN	RF10NN	RFM
Age	1.00	1.00	1.00	1.00	0.46	1.00
Gender	0.43	0.54	0.48	0.25	0.08	0.19
Hypertension complicated	0.43	0.25	0.20	0.21	0.12	0.20
Lipid modifying agents	0.43	0.55	0.55	0.28	0.12	0.28
RAS-acting agents	0.34	0.49	0.46	0.24	0.10	0.19
Drugs used in diabetes	0.21	0.19	0.15	0.07	0.03	0.08
Beta blocking agents	0.19	0.25	0.18	0.10	0.07	0.12
Smoking	0.16	0.18	0.10	0.07	0.07	0.10
Calcium channel blockers	0.13	0.25	0.23	0.11	0.04	0.10
Diabetes non-insulin dependent	0.09	0.17	0.12	0.05	0.05	0.07
Hypertension uncomplicated	0.08	0.12	0.04	0.03	0.04	0.05
Diuretic drugs	0.07	0.12	0.03	0.04	0.04	0.03
Lipid disorder	0.06	0.08	0.01	0.02	0.00	0.02
Ischaemic heart disease w/ angina	0.05	0.08	0.00	0.01	0.00	0.03
Atrial fibrillation/flutter	0.04	0.07	0.00	0.00	0.00	0.00
Cholesterol Total *Missing?*	0.14	0.15	0.10			
Systolic blood pressure *Missing?*	0.12	0.11	0.11			
Diastolic blood pressure *Missing?*	0.10	0.13	0.09			
Glucose fasting *Missing?*	0.07	0.10	0.02			
Cholesterol total	0.01			0.51	0.41	0.40
Systolic blood pressure	0.01			0.47	0.66	0.41
Diastolic blood pressure	0.02			0.90	0.79	0.86
Glucose fasting	0.00			0.94	1.00	0.93
High cholesterol total		0.12				
Low cholesterol total		0.07				
Medium cholesterol total		0.09				
High systolic blood pressure		0.03				
Low systolic blood pressure		0.10				
Medium systolic blood pressure		0.10				
High diastolic blood pressure		0.10				
Low diastolic blood pressure		0.21				
Medium diastolic blood pressure		0.11				
High glucose fasting		0.07				
Low glucose fasting		0.00				
Medium glucose fasting		0.05				

Table 5.3: Normalized feature importance of all features using the different methods. 1 is most important, 0 is least important. The table is roughly sorted on the LITF column for readability. The dummy variable for missingness is denoted with *Missing?*. The methods used are: *Lost in the Forest* (LITF), RF with dummy for missingness (RFDM), RF without any measurement value only using the dummy for missingness (RFonlyDM), and RFs with imputation methods: *k*-NN (RF3NN, RF10NN), and MICE (RFM). Note that the first groups of features (Age, ... , Atrial fibrillation/flutter) are not considered to be measurements.

5.4 Validation on other Clinical Data Sets

Table 5.4 shows how all methods perform with the KEEL data sets. With the KEEL data sets the RFs using imputations seem to perform better than in the EHR use case. RF10NN slightly outperforms other methods in the Hepatitis data set. LITF, RFDM, RF3NN and RFM show similar performance. RFonlyDM performs poorly. In the Mammographic Mass data set LITF and the RFs using imputation show similar AUCs. RFonlyDM performs poorly again.

Data set	Method	Mean AUC
Hepatitis	LITF	0.85
	RFDM	0.86
	RFonlyDM	0.56
	RF3NN	0.87
	RF10NN	0.88
	RFM	0.87
Mammographic Mass	LITF	0.90
	RFDM	0.80
	RFonlyDM	0.54
	RF3NN	0.88
	RF10NN	0.88
	RFM	0.88

Table 5.4: Different performance metrics generated by the runs on different methods on the KEEL data sets. The mean of the AUC on the CV test sets are displayed. The methods used are: *Lost in the Forest* (LITF), RF with dummy for missingness (RFDM), RF without any measurement value only using the dummy for missingness (RFonlyDM), and RFs with imputation methods: *k*-NN (RF3NN, RF10NN), and MICE (RFM).

Hyperparameter tuning results of the experiments in Section 5.2 and Section 5.4 are presented in Appendix B.

Chapter 6

Discussion

This thesis compares the performance of the newly developed *Lost in the Forest* (LITF) RF implementation with RF models trained on data with either a dummy variable for missingness or data in which missing values have been imputed in various ways. Regarding the routine primary care use case, all imputation methods likely resulted in biased imputed values when compared with the population average as the golden standard.

Discriminative predictive performance on the routine primary care data set was similar for LITF, RFDM and RFonlyDM, the algorithms that incorporate the dummy variable for missing values. Furthermore, in the context of the routine primary care data set, this study shows that the dummy for missingness method performs better than simply using imputation methods to fill in the missing data, when used on EHR data. This shows that a lot of predictive performance can be extracted from the missingness of a feature. More importantly, eventual performance of a model is depending on more than the predictive performance during internal validation alone. When using imputation methods for missing values that are MNAR, it may result in biased imputation which will finally lead to worse model results in external validation. Thus, in data with high probability of MNAR missingness, imputation is likely a poor strategy to begin with.

The performance of the various RF methods using the KEEL data sets shows that imputation using RFs still is a viable option when missingness is not as prevalent as in the primary care data set. In the KEEL data sets the performance of MICE is similar to the performance of the k-nearest neighbor imputation method. The RF that exclusively uses the dummy for missingness for measurements (RFonlyDM) shows poor performance in these data sets, showing that using only the dummy for missingness on its own is not enough to perform comparable to other methods. This and the fact that the RFs using imputation perform relatively well show that LITF and RFDM lose their advantage when the data has few missing values and possibly a different missingness type. However, using LITF has no risk of bias due to imputation of possible MNAR data. The

high AUCs for LITF and the RFs using imputation methods, when used on the mammographic mass data, show how important it can be to take advantage of continuous values, since the other methods are unable to use the continuous values.

Our study has several limitations. First regarding classical RF methods, since only a limited number of hyperparameters have been tested using limited ranges, it is possible that the optimal set of hyperparameters has not been identified, leading to suboptimal model performance. Second, the missingness mechanism in the routine primary care data is assumed to be largely MNAR, based on expert knowledge of the data collection process in primary care. However, there is no way of exactly determining the type of missingness. The missingness mechanism of the KEEL data sets is largely unknown, as no detailed information is known about these data sets. This suggests that using synthetic data in future experiments will resolve this problem.

Strong points of this study include use of detailed, large routinely available primary health data set and a strong method of model validation using both a nested cross-validation technique and other clinical data sets.

Chapter 7

Conclusions

In our routine primary care data set imputing missing values using different methods appears to lead to biased imputation results for all tested continuous measurement variables, since missing data are likely MNAR. We developed a novel RF implementation that uses all continuous data, without the need to impute missing values and thus avoiding the possibility of bias introduced by imputing. Although LITF performance never exceeded the performance of RF algorithms using imputation or using dummy variable to deal with missing data, it performs well on both types of data sets without needing to impute these variables. This makes LITF a versatile tool to use on clinical data.

This thesis uses its own implementation and the focus of this thesis was not on time or computational efficiency. Incorporating LITF into a more optimized and better tested implementation from, for example, scikit-learn [48] may have a positive impact on ease of use. Alternatively, a random forest implementation that optimizes speed and/or memory usage [14–16] could be included.

Different test and experiments have been left for the future due to lack of time, i.e. experiments with the sampled data required almost a day to finish the run for all algorithms. An option in the future would be to use the entirety (or a more significant portion) of the data to make predictions with. Not only would it be used to classify potential CVD patients, but it can be used to determine a broader range of risk patients, e.g. for cancers, or respiratory diseases. Having a system that could predict multiple facets of health would decrease overhead in medicine, by allowing experts to focus on high-risk patients.

Ultimately, LITF could be expanded to other fields where missing data, expecially MNAR occurs.

Bibliography

[1] Hartstichting, "Hart- en vaatziekten in nederland: cijfers over leefstijl, risicofactoren, ziekte en sterfte, 2018.." https://www.hartstichting.nl/getmedia/a6e15c10-2710-41b9-bcf8-8185feaf54b2/cijferboek-hartstichting-hart-vaatziekten-nederland-2018.pdf. Accessed: 2010-09-30.

[2] J. N. Struijs, M. L. van Genugten, S. M. Evers, A. J. Ament, C. A. Baan, and G. A. van den Bos, "Modeling the future burden of stroke in the Netherlands," *Stroke*, vol. 36, no. 8, pp. 1648–1655, 2005.

[3] Nederlands Huisartsen Genootschap, "Cardiovasculair risicomanagement (CVRM)." https://www.nhg.org/standaarden/volledig/cardiovasculair-risicomanagement.

[4] P. Appelros, I. Nydevik, Åke Seiger, and A. Ternt, "Predictors of severe stroke," *Stroke*, vol. 33, no. 10, pp. 2357–2362, 2002.

[5] D. B. Rubin, "Inference and missing data," *Biometrika*, vol. 63, no. 3, pp. 581–592, 1976.

[6] M. S. Porta, S. Greenland, M. Hernn, I. d. S. Silva, and J. M. Last, *A dictionary of epidemiology*. Oxford University Press, 2014.

[7] S. Fielding, P. M. Fayers, A. McDonald, G. McPherson, M. K. Campbell, and the RECORD study group, "Simple imputation methods were inadequate for missing not at random (MNAR) quality of life data," *Health and Quality of Life Outcomes*, vol. 6, p. 57, Aug 2008.

[8] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[9] T. K. Ho, "Random decision forests," in *Proceedings of the Third International Conference on Document Analysis and Recognition (Volume 1) - Volume 1*, ICDAR '95, (Washington, DC, USA), pp. 278–, IEEE Computer Society, 1995.

[10] H. M. Gomes, A. Bifet, J. Read, J. P. Barddal, F. Enembreck, B. Pfharinger, G. Holmes, and T. Abdessalem, "Adaptive random forests for evolving data stream classification," *Machine Learning*, vol. 106, pp. 1469–1495, June 2017.

[11] Y. Wang, S. Xia, Q. Tang, J. Wu, and X. Zhu, "A novel consistent random forest framework: Bernoulli random forests," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, pp. 3510–3523, Aug 2018.

[12] I. Reis, D. Baron, and S. Shahaf, "Probabilistic random forest: A machine learning algorithm for noisy data sets," *The Astronomical Journal*, vol. 157, p. 16, Dec. 2018.

[13] S. Georganos, T. Grippa, A. N. Gadiaga, C. Linard, M. Lennert, S. Vanhuysse, N. Mboga, E. Wolff, and S. Kalogirou, "Geographical random forests: a spatial extension of the random forest algorithm to address spatial heterogeneity in remote sensing and population modelling," *Geocarto International*, pp. 1–16, June 2019.

[14] M. N. Wright and A. Ziegler, "ranger: A fast implementation of random forests for high dimensional data in C++ and R," *Journal of Statistical Software*, vol. 77, no. 1, 2017.

[15] Y. Mishina, R. Murata, Y. Yamauchi, T. Yamashita, and H. Fujiyoshi, "Boosted random forest," *IEICE Transactions on Information and Systems*, vol. E98.D, no. 9, pp. 1630–1636, 2015.

[16] A. Bayat, P. Szul, A. R. O'Brien, R. Dunne, O. J. Luo, Y. Jain, B. Hosking, and D. C. Bauer, "VariantSpark, a random forest machine learning implementation for ultra high dimensional data," *bioRxiv*, 2019.

[17] S. DuBrava, J. Mardekian, A. Sadosky, E. J. Bienen, B. Parsons, M. Hopps, and J. Markman, "Using Random Forest Models to Identify Correlates of a Diabetic Peripheral Neuropathy Diagnosis from Electronic Health Record Data," *Pain Medicine*, vol. 18, pp. 107–115, 01 2017.

[18] M. Z. Alam, M. S. Rahman, and M. S. Rahman, "A random forest based predictor for medical data classification using feature ranking," *Informatics in Medicine Unlocked*, vol. 15, p. 100180, 2019.

[19] R. Casanova, S. Saldana, E. Y. Chew, R. P. Danis, C. M. Greven, and W. T. Ambrosius, "Application of random forests methods to diabetic retinopathy classification analyses," *PLoS ONE*, vol. 9, p. e98587, June 2014.

[20] M. Kumar, "Prediction of chronic kidney disease using random forest machine learning algorithm," *International Journal of Computer Science and Mobile Computing*, vol. 5, no. 2, pp. 24–33, 2016.

[21] J. Maroco, D. Silva, A. Rodrigues, M. Guerreiro, I. Santana, and A. de Mendonça, "Data mining methods in the prediction of dementia: A real-data comparison of the accuracy, sensitivity and specificity of linear discriminant analysis, logistic regression, neural networks, support vector machines, classification trees and random forests," *BMC Research Notes*, vol. 4, Aug. 2011.

[22] J. Ibrahim, H. Chu, and M. Chen, "Missing data in clinical studies: Issues and methods," *Journal of Clinical Oncology*, vol. 30, pp. 3297–3303, 9 2012.

[23] J. Chen and J. Shao, "Nearest neighbor imputation for survey data," *Journal of official statistics*, vol. 16, no. 2, p. 113, 2000.

[24] S. Buuren and C. Groothuis-Oudshoorn, "MICE: Multivariate imputation by chained equations in R," *Journal of Statistical Software*, vol. 45, 12 2011.

[25] G. Batista and M.-C. Monard, "A study of k-nearest neighbour as an imputation method.," *Hybrid Intelligent Systems, ser Front Artificial Intelligence Applications*, vol. 30, pp. 251–260, 01 2002.

[26] R. Giorgi, A. Belot, J. Gaudart, and G. Launoy, "The performance of multiple imputation for missing covariate data within the context of regression relative survival analysis," *Statistics in Medicine*, vol. 27, no. 30, pp. 6310–6331, 2008.

[27] A. K. Waljee, A. Mukherjee, A. G. Singal, Y. Zhang, J. Warren, U. Balis, J. Marrero, J. Zhu, and P. D. Higgins, "Comparison of imputation methods for missing laboratory data in medicine," *BMJ Open*, vol. 3, no. 8, 2013.

[28] B. Twala, M. Jones, and D. Hand, "Good methods for coping with missing data in decision trees," *Pattern Recognition Letters*, vol. 29, no. 7, pp. 950 – 956, 2008.

[29] R. Groenwold, I. White, R. Donders, J. Carpenter, D. Altman, and K. Moons, "Missing covariate data in clinical research: When and when not to use the missing-indicator method for analysis," *CMAJ : Canadian Medical Association journal = journal de l'Association medicale canadienne*, vol. 184, pp. 1265–9, 02 2012.

[30] C. Beaulac and J. S. Rosenthal, "Handling missing values using decision trees with branch-exclusive splits," *CoRR*, vol. abs/1804.10168, 2018.

[31] D. R. Morales, R. Flynn, J. Zhang, E. Trucco, J. K. Quint, and K. Zutis, "External validation of ADO, DOSE, COTE and CODEX at predicting death in primary care patients with COPD using standard and machine learning approaches," *Respiratory Medicine*, vol. 138, pp. 150 – 155, 2018.

[32] P. Lambin, R. G. P. M. van Stiphout, M. H. W. Starmans, E. Rios-Velazquez, G. Nalbantov, H. J. W. L. Aerts, E. Roelofs, W. van Elmpt, P. C. Boutros, P. Granone, V. Valentini, A. C. Begg, D. D. Ruysscher, and A. Dekker, "Predicting outcomes in radiation oncology—multifactorial decision support systems," *Nature Reviews Clinical Oncology*, vol. 10, pp. 27–40, Nov. 2012.

[33] A. G. Singal, A. Mukherjee, J. B. Elmunzer, P. D. R. Higgins, A. S. Lok, J. Zhu, J. A. Marrero, and A. K. Waljee, "Machine learning algorithms outperform conventional regression models in predicting development of hepatocellular carcinoma," *American Journal of Gastroenterology*, vol. 108, pp. 1723–1730, Nov. 2013.

[34] S. F. Weng, J. Reps, J. Kai, J. M. Garibaldi, and N. Qureshi, "Can machine-learning improve cardiovascular risk prediction using routine clinical data?," *PLOS ONE*, vol. 12, p. e0174944, Apr. 2017.

[35] T. van der Ploeg, D. Nieboer, and E. W. Steyerberg, "Modern modeling techniques had limited external validity in predicting mortality from traumatic brain injury," *Journal of Clinical Epidemiology*, vol. 78, pp. 83–89, Oct. 2016.

[36] H. J. A. van Os, L. A. Ramos, A. Hilbert, M. van Leeuwen, M. A. A. van Walderveen, N. D. Kruyt, D. W. J. Dippel, E. W. Steyerberg, I. C. van der Schaaf, H. F. Lingsma, W. J. Schonewille, C. B. L. M. Majoie, S. D. Olabarriaga, K. H. Zwinderman, E. Venema, H. A. Marquering, M. J. H. Wermer, and the MR CLEAN Registry Investigators, "Predicting outcome of endovascular treatment for acute ischemic stroke: Potential value of machine learning algorithms," *Frontiers in Neurology*, vol. 9, p. 784, 2018.

[37] Z. Obermeyer and E. J. Emanuel, "Predicting the future — big data, machine learning, and clinical medicine," *New England Journal of Medicine*, vol. 375, pp. 1216–1219, Sept. 2016.

[38] J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, and S. García, "Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework," *Multiple-Valued Logic and Soft Computing*, vol. 17, pp. 255–287, 2011.

[39] M. Azur, E. Stuart, C. Frangakis, and P. Leaf, "Multiple imputation by chained equations: What is it and how does it work?," *International Journal of Methods in Psychiatric Research*, vol. 20, pp. 40–49, 3 2011.

[40] A. Altmann, L. Toloi, O. Sander, and T. Lengauer, "Permutation importance: a corrected feature importance measure," *Bioinformatics*, vol. 26, pp. 1340–1347, 04 2010.

[41] G. C. Cawley and N. L. C. Talbot, "On over-fitting in model selection and subsequent selection bias in performance evaluation," *J. Mach. Learn. Res.*, vol. 11, pp. 2079–2107, Aug. 2010.

[42] S. Varma and R. Simon, "Bias in error estimation when using cross-validation for model selection," *BMC Bioinformatics*, vol. 7, pp. 91 – 91, 2005.

[43] L. Wang, F. Chu, and W. Xie, "Accurate cancer classification using expressions of very few genes," *IEEE/ACM transactions on computational biology and bioinformatics / IEEE, ACM*, vol. 4, pp. 40–53, 01 2007.

[44] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861 – 874, 2006. ROC Analysis in Pattern Recognition.

[45] Rijksinstituut voor Volksgezondheid en Milieu, "Bloeddruk naar leeftijd en geslacht." `https://www.rivm.nl/bloeddruk-en-hypertensie-naar-leeftijd-en-geslacht`.

[46] Rijksinstituut voor Volksgezondheid en Milieu, "Prevalentie van totaal- en HDL-cholesterol, 2009-2010." `https://www.volksgezondheidenzorg.info/onderwerp/cholesterol/cijfers-context/huidige-situatie`.

[47] Bronovo Ziekenhuis, "Onderzoek naar referentiewaarden van laboratoriumonderzoek in een algemeen ziekenhuis: resultaten en bevindingen," *Ned Tijdschr Klin Chem Labgeneesk*, vol. 34, no. 1, 2008.

[48] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

Appendix A

Runtime

Data set	Method	Runtime
Mammographic Mass	LITF	15 minutes
	RFDM	7 minutes
	RFonlyDM	2 minutes
	RF3NN	6 minutes
	RF10NN	6 minutes
	RFN	7 minutes
Hepatitis	LITF	67 minutes
	RFDM	22 minutes
	RFonlyDM	5 minutes
	RF3NN	20 minutes
	RF10NN	21 minutes
	RFN	24 minutes
Primary Care Case Study	LITF	4 hours
	RFDM	4 hours
	RFonlyDM	3 hours
	RF3NN	9 hours
	RF10NN	11 hours
	RFN	11 hours

Table A.1: Runtimes of the experiments in Section 5.2 and Section 5.4

All experiments in this thesis were run on the following system:

- Operating system: Ubuntu 16.04 LTS 64-bit
- Processor: Intel® Core™i7-3610QM CPU @ 2.30GHz × 8
- Memory: 7,7 GiB DDR3 @ 1600 MHz
- Graphics: NVIDIA® GeForce® 610M with 2GB DDR3 VRAM

Appendix B

Hyperparameter Settings

Data set	Method	Number of trees	Maximum depth	Minimum number of samples	Ratio of features for a split
Mammographic Mass	LITF	64	30	10	0.2
	RFDM	64	10	3	0.5
	RFonlyDM	64	30	10	0.2
	RF3NN	64	20	3	0.5
	RF10NN	64	20	1	0.5
	RFN	64	10	10	0.5
Hepatitis	LITF	64	20	3	0.2
	RFDM	64	20	3	0.2
	RFonlyDM	64	20	3	0.5
	RF3NN	64	30	3	0.5
	RF10NN	64	20	10	0.5
	RFN	64	10	1	0.5
Primary Care Case Study	LITF	64	20	3	0.5
	RFDM	64	10	10	0.5
	RFonlyDM	64	30	10	0.5
	RF3NN	64	30	1	0.5
	RF10NN	64	20	3	0.5
	RFN	64	30	3	0.5

Table B.1: Median hyperparameters chosen by the nested CV for experiments in Section 5.2 and Section 5.4