Interpretable Recurrent Neural Networks for Heart Failure Re-hospitalisation Prediction

Marijn Valk - 3830810

June 4, 2018

Abstract

Interpretability and predictive performance are important aspects of a machine learning model. Typically, there is a trade-off between interpretability and predictive performance. This trade-off results in a choice between accurate but opaque models such as multilayer perceptron (MLP) and less accurate but more transparent models such as logistic regression (LR). In the healthcare domain, model interpretability is especially important because the real-life goals (e.g. patient well-being) are hard to model (and thus optimize) formally. Traditional methods such as LR & MLP use aggregate features and are therefore not able to effectively model temporal dimension that is inherent in Electronic Health Records (EHR) data. Recently, Recurrent Neural Network (RNN) approaches have been successful in modelling healthcare data because they are able to effectively take the temporal dimension into account. However, the RNN model is notoriously hard to interpret. We have looked at three recently proposed RNN-based models for medical event prediction that claim to be interpretable (Dipole, GRNN-HA and RETAIN). The interpretability of these models is tied to the implementation of a neural attention mechanism. Having considered how well the models are able to relate the input to the output in understandable terms, we devised an ordering of the interpretability of these models. Then, we compared performance in predicting 30 re-hospitalisation on an EHR dataset with 37,287 medical histories using admission and diagnosis data. The interpretability/performance trade-off within the three 'interpretable' models was partly observed. Although the performance of the RNN-based models was quite similar, the difference in interpretability is more substantial. Therefore we believe that the interpretable RNN-based models are the better overall option to use for predicting events in the healthcare domain.

Acknowledgements

Dear reader,

This document contains the report of my master's thesis, presented in order to obtain the master's degree in Business Informatics from the University of Utrecht. Almost 7 years ago I enrolled in the Information Science bachelor's programme in Utrecht. At that time I was a young man of 18 years old, not too interested in working with data & algorithms. My interests have changed a lot during these 7 years and looking back, I should probably have enrolled in either the Artificial Intelligence or Computer Science programme. All in all, I took quite a detour, following philosophy courses and a cognitive psychology minor. Only to realize during my second year in the Business Informatics master's that I really enjoyed doing the 'nerdy data stuff'. I postponed the writing of my master's thesis, started reading books, following on-line data science courses and enrolling in extra data science related courses.

I could not postpone the writing of my thesis indefinitely though, so I started the project that would result in this document around September 2017. It has been quite a journey in which I learned a lot, got to experience daily life in an academic hospital and had the opportunity to work with the high performance computing cluster at the UMC (a geek's dream come true). Under the supervision of Matthieu Brinkhuis (UU), Folkert Asselbergs (UMC) and Marco Spruit (UU), I was given all the resources, advice, and independence I needed to productively work for 8 months. I would like to thank them for this. I would also like to thank Jochem de Boer for working with me in setting up the project. Finally, I would like to express gratitude to my friends & family for having my back.

Marijn Valk
Utrecht, May 2018

Contents

Part I

Preliminaries

Chapter 1

Introduction

Heart Failure (HF) is one of of the leading causes of hospitalisation in patients over the age of 65 years, and is associated with high morbidity, mortality, and healthcare expenditures (Writing et al., 2016). It is estimated that around 38 million people worldwide suffer from HF (Braunwald, 2015). Also, projections show that the prevalence of HF will increase due to the ageing of the population. At present, in-hospital mortality for acute HF is around 15% and 60-day re-hospitalisation rates in patients who survive to discharge approach 30% (Gheorghiade and Pang, 2009). The 5-year mortality rate after HF diagnosis approximates 50% (Writing et al., 2016).

It has been shown that in outpatients with chronic HF, a hospital admission is one of the strongest prognostic factors for an increased rate of mortality (Koudstaal et al., 2016). This is because severely ill patients are more likely to be hospitalised than patients who are less ill. Therefore, in order to reduce HF mortality it is imperative to be able to detect patients that have a high risk of (re-)hospitalisation and provide them with the care they need. In the majority of patients, gradual signs and symptoms of worsening heart failure emerge in the days or weeks prior to the re-hospitalisation. Several studies have already shown that using ML techniques can be effective in improving the quality of healthcare (Chaudhry et al., 2006; Jha et al., 2009; Black et al., 2011; Shekelle et al., 2006; Jones et al., 2014).

At present, prediction models based on 'traditional' epidemiological approaches such as Logistic Regression (LR) analysis have produced moderate results in the context of HF re-hospitalisation prediction (Ouwerkerk et al., 2014). The subtle dynamics and time-varying nature of real-world clinical predictors are often lost in the oversimplified traditional logistic regression analyses. Clearly, if we could use this information without aggregating over the temporal dimension, significant steps could be taken to improve our ability to recognise and intensify medical care to those who are at high-risk for an impending HF (re-)hospitalisation. The Recurrent Neural Network (RNN) model is able to take into account the temporal dimension and has recently been applied in several healthcare applications (Choi et al., 2016a; Lipton et al., 2015b; Esteban et al., 2016; Lipton et al., 2016). The RNN-based models consistently outperform models that use aggregate features (e.g. LR). However, the gain in performance comes at the cost of reduced interpretability.

The need for interpretability arises when there is a mismatch between the machine learning model optimization objective and the 'real-life' goals of the user of a machine learning model (Lipton, 2016), or when there is a multi-objective trade-off (Doshi-Velez and Kim, 2017). The actuality of mismatched objectives is clear in the healthcare domain where (some of) the real-life goals are to provide the most efficient and effective care while taking ethical considerations and patient subjective well-being into account. The goal of the machine learning model is to minimize prediction errors (as measured by some loss function). However, we are not (yet) able to define the real-life goals in terms of a real-valued function and a machine can therefore not really optimize them (Lipton, 2016). Also, objectives such as healthcare cost and patient well-being can be a trade-off (Doshi-Velez and Kim, 2017). This is where interpretation and human judgement are used. With an interpretable model, a human can make (and justify) decisions with regards to mismatched objectives or objectives that trade-off. In that sense, an interpretable model can serve as a decision support system.

1.1 Problem Statement

The pairing of the four elements that were just introduced are the basis for the problem statement of this study. The problem statement is reached by the following line of reasoning:

1. Heart Failure is a big healthcare problem because of high mortality, morbidity and healthcare costs.
2. In order to improve healthcare for HF patients we need to be able to detect patients that have a high risk of re-hospitalisation. This allows us to keep these patients in the hospital and give them better treatment.
3. Additionally, in predicting re-hospitalisation risk model interpretability is important because it allows a human to make an informed decision with regards to his/her objectives.
4. **Problem Statement**:

   > *The Recurrent Neural Network model has been shown to be successful in modelling EHR data and predicting medical events but suffers from a lack of interpretability.*

1.2 Objective, Scope and Structure

The main objective of this research is to address the problem of providing an interpretable prediction of re-hospitalisation risk for HF patients. Although our scope was a specific condition (i.e. HF) and a specific outcome (i.e. re-hospitalisation), we believe this study addresses the problem of providing accurate and interpretable predictions of medical events in the healthcare domain more generally.

This document is divided into several parts. We complete the preliminaries with a description of our research approach in Chapter 2. After that, this document has the following structure:

Part II: Background

This part consists of the background that is needed for a complete understanding of the problem and the context for a possible solution. We provide an overview of the condition of HF and studies that attempt to predict re-hospitalisation in literature in Chapter 3. In Chapter 4 we describe the structure of Electronic Health Records, challenges for machine learning in the healthcare domain and a way to measure predictive performance of a machine learning model. We continue with the definition of several machine learning approaches that are relevant in our context in Chapter 5.

Part III: Theory

In this part we delve deep into the theory that aims to address our problem. We describe a taxonomy of interpretability that can be used to anchor a discussion about the interpretability of a model in Chapter 6. Following that we discuss several extensions to the RNN model that aim to improve its predictive performance and interpretability in Chapter 7. Finally we outline three RNN-based models that claim to be interpretable and evaluate that claim in Chapter 8.

Part IV: Results

In the fourth and final part we outline our results. The experimental setting and evaluation of the results is given in Chapter 9. We conclude with a return to our research questions, the limitations of our study and considerations for future research in Chapter 10.

Chapter 2

Research approach

2.1 Research Questions

Following from the problem statement that was formulated in the opening of this document, we devised the main question of this research as follows:

> *Can the Recurrent Neural Network model be improved such that it can provide an accurate and interpretable prediction of (re-)hospitalisation risk for Heart Failure patients?*

In order to answer the main research question there are several sub-questions that need to be answered. In order to make assertions regarding the interpretability of a model we need to be precise in a definition of the concept. This is reflected in the first sub-question:

> **SQ1.** *What constitutes an interpretable model?*

After fixing on a definition of the concept of interpretability, we are looking for ways that aim to improve the RNN model in terms of interpretability and predictive performance:

> **SQ2.** *How can the regular RNN model be adapted in order to allow for better predictive performance and interpretation?*

Finally, we are interested in the comparison between the 'interpretable' RNN-based models and their non-interpretable counterparts:

> **SQ3.** *How do the 'interpretable' RNN-based models compare against each other, the traditional approaches and the regular RNN model with regards to predictive performance and interpretability?*

2.2 Research Design

The Cross-Industry Standard Process for Data Mining (CRISP-DM) is a widely adopted standard for machine learning projects. It recognizes that machine learning is an iterative process but provides structure to a project by dividing activities into specific phases. A useful feature is that it guides users to gain an understanding of the application (business) context and the characteristics of the data first before moving on the actual machine learning activities (data preparation, modelling and evaluation). The CRISP-DM method was used as guidance in planning and executing the research project. For this research project, the 'Deployment' phase was out of scope and is therefore not discussed. The research methods used in this machine learning project consist of a literature review and an experimental study. See Figure 2.1 for our application of the CRISP-DM to this research project.

The **literature review** was a part of the 'Business- and Data Understanding' phases of the CRISP-DM. It was performed in order to answer **SQ.1** & **SQ.2**. The results of this literature review are laid out in Parts II & III of this document. Chapter 6 specifically aims to answer **SQ.1** while Chapter 7 & Chapter 8 aim to answer **SQ.2**.

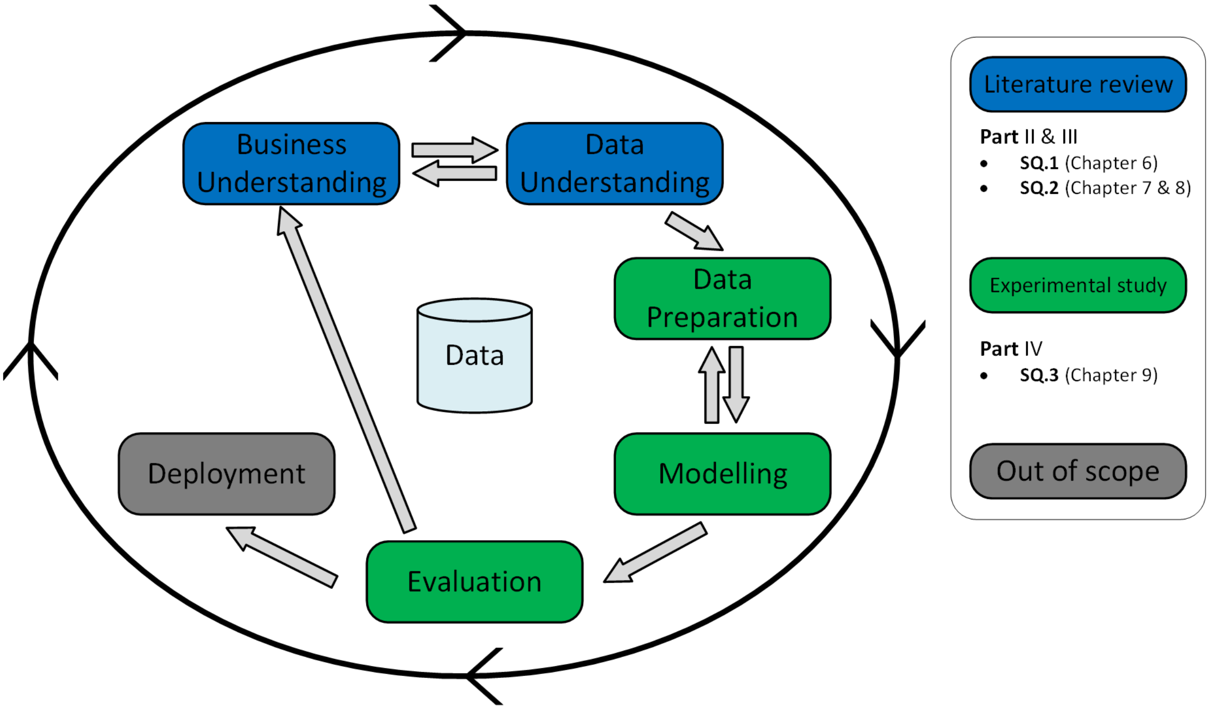


Figure 2.1: The CRISP-DM. The color coding illustrates the link between phases in the CRISP-DM and the parts, chapters and sub-questions in this document.

The **experimental study** was part of the ‘Data Preparation’, ‘Modelling’ and ‘Evaluation’ phases. It was performed in order to answer **SQ.3** in Chapter 9. With the answers to all the sub-questions we answer the main research question in Chapter 10. The design and results of the experimental study are laid out in Chapter 9.

2.2.1 Contributions

The proposed research project will contribute to the scientific body of knowledge in the following two ways:

1. An operationalisation of the concept of interpretability in the context of predicting re-hospitalisation for heart failure patients. This provides a basis for assertions regarding the interpretability of a machine learning model in this context.

2. A comparison of ‘interpretable’ Recurrent Neural Network models with each other (and with baseline approaches) in an experimental study on a cohort of HF patients from the University Medical Centre Utrecht. The models are compared with regards to their:

    - Predictive performance
    - Interpretability

Part II

Background

Chapter 3

Heart Failure and Re-hospitalisation Prediction

In this chapter we give a more complete description of the condition of Heart Failure. In addition, we provide an overview of studies from literature that aim to predict Heart Failure re-hospitalisation.

3.1 Definition of Heart Failure

Heart Failure (HF) is a clinical syndrome caused by cardiac abnormality. The result is reduced cardiac output (i.e. a reduced amount of blood that the heart is able to pump through the body). HF may also result in elevated levels of pressure within the heart. Classical symptoms are breathlessness, fatigue and swollen ankles. The cardiac abnormality that is usually the cause of HF has to do with the muscular tissue of the heart (myocardial abnormality) which in turn results in ventricular dysfunction. However, there are abnormalities related to other parts of the heart that can also cause HF.

HF patients are usually differentiated with regards to the left ventricular ejection fraction (LVEF). This measure is the fraction of the blood that is ejected from the left ventricle with a heartbeat. It is measured with an echo cardiogram (a sonogram of the heart). The European Society of Cardiology defines three groups of heart failure patients according to their LVEF (see table 3.1) (Ponikowski et al., 2016). The differentiation is done because treatment options, morbidity and mortality are different for these groups. HF with reduced ejection fraction is also known as Systolic HF. In this case the heart is not able to pump with enough force to circulate the blood through the body. In contrast, HF with preserved ejection fraction is also known as diastolic HF. In this type of HF the heart is not able to relax normally in between each heartbeat and allow its ventricles to be filled with blood.

Type of HF	**LVEF**
preserved ejection fraction (HFpEF) / Diastolic HF	$\geq 50\%$
mid-range ejection fraction (HFmrEF)	40–49%
reduced ejection fraction (HFrEF) / Systolic HF	<40%

Table 3.1: Differentiation of HF. The type of HF is established by measuring the Left Ventricular Ejection Fraction (LVEF). In other words, the fraction of blood that is ejected by the left ventricle after each contraction of the heart muscle.

Another way to characterize the type of HF is by looking at the time of onset. The ESC guidelines make a distinction between patients who have developed HF slowly over time (Chronic HF) and those that are presented suddenly (Acute HF) (Ponikowski et al., 2016). A patient that has been treated for HF and has symptoms and signs that have mostly stayed the same for at least one month is said to be 'stable'. The term 'de-compensated' is used to describe a patient with Chronic HF in which the symptoms have suddenly deteriorated. De-compensation usually leads to a hospitalisation. In some cases of HF the

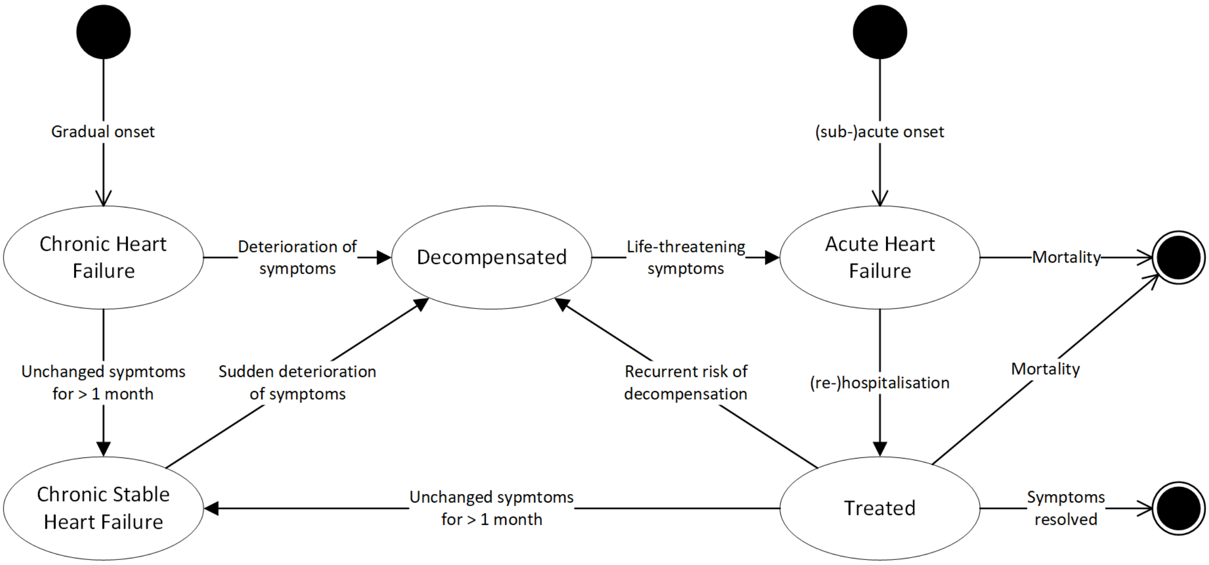

Gradual onset
(sub-)acute onset
Chronic Heart Failure
Deterioration of symptoms
Decompensated
Life-threatening symptoms
Acute Heart Failure
Mortality
Unchanged sypmtoms for > 1 month
Sudden deterioration of symptoms
Recurrent risk of decompensation
(re-)hospitalisation
Mortality
Chronic Stable Heart Failure
Unchanged sypmtoms for > 1 month
Treated
Symptoms resolved

Figure 3.1: Heart Failure state diagram. A patient can have a gradual or acute onset of symptoms. Suddenly deteriorating symptoms is known as de-compensation. If symptoms remain unchanged for longer than a month, the patient is considered to be stable. After treatment, the patient has a recurrent risk of de-compensation.

symptoms can completely resolve but in most cases there is a recurrent risk of de-compensation and/or death. Figure 3.1 shows the disease states and the transitions between them. For a summary on the diagnosis and common treatment options for HF see Appendix B.

3.2 Heart Failure Re-hospitalisation Prediction in Literature

Four literature review studies by Rahimi et al. (2014), Tripoliti et al. (2017), Ouwerkerk et al. (2014) and Ross et al. (2008) take a look at studies that describe a model used for predicting risk of re-hospitalisation for HF patients. Table 3.2 shows the outcome description, prediction window, model architecture, number of predictors and discriminative ability measured with the AUC[1]. Ouwerkerk et al. (2014) state that the mean AUC of models that aim to predict HF re-hospitalisation reviewed in their study was 0.68. The studies listed in table 3.2 have an average AUC of 0.66.

From the studies listed in the table there are 2 that have a significantly higher AUC score than the other studies. The study by Wang et al. (2012) included 198,640 patients with ≥ 1 diagnosis of HF (as defined by ICD-9 codes). The models developed in this study aimed to predict whether a patient would be hospitalized, die without hospitalisation or be event-free after 30 days. The authors of this study note that their accuracy is on the high end of the spectrum compared to other studies. However, they add that hospitalization may be easier to predict than re-hospitalization. They state that: “re-hospitalization risk could be more heavily influenced by hospital and physician level factors, whereas hospitalization risk might be inherently more related to patient level factors such as previous health care usage” (Wang et al., 2012).

The study by Koulaouzidis et al. (2016) stands out in the sense that it uses tele-monitoring device to track the diastolic blood pressure and the weight of the HF patients while they are at home. The study shows the promise of using tele-monitoring devices in predicting HF re-hospitalisation. However, the study lacks in sample size ($n = 308$) and only tries to predict the re-hospitalisation event 1 day in advance.

[1] We provide a comprehensive description of the AUC measure in Section 4.3

Study	**Outcome description**	**Prediction Window**	**Model**	**# predictors**	**AUC**
Yamokoski et al. (2007)	All-cause	6-month	LR	2	0.519
Kang et al. (2016)	HF	60-day	DT	7	0.59
Philbin and DiSalvo (1999)	HF	any	LR	11	0.6
Keenan et al. (2008)	All-cause	30-day	LR	37	0.6
Au et al. (2012)	All-cause	30-day	RF	4	0.6
Zolfaghar et al. (2013)	HF	30-day	LR	> 100	0.63
Postmus et al. (2012)	HF	18-month	LR	5	0.66
Watson et al. (2011)	All-cause	30-day	LR	3	0.67
Felker et al. (2004)	All-cause or death	60-day	LR	5	0.69
Basu Roy et al. (2015)	HF	30-day	HDC	91	0.69
Amarasingham et al. (2010)	All-cause	30-day	LR	29	0.72
Wang et al. (2012)	All-cause	30-day	LR	37	0.82
Koulaouzidis et al. (2016)	HF	1-day	NB	2	0.82

Table 3.2: HF re-hospitalisation prediction studies from literature. Models are abbreviated as follows: Logistic Regression (LR), Decision Tree (DT), Random Forest (RF), Hierarchical Dynamic Clustering (HDC) & Naive Bayes (NB).

Chapter 4

Electronic Health Records and Machine Learning

In this chapter we introduce the structure of Electronic Health Records data. We discuss the challenges in carrying out a machine learning project on Electronic Health Records. Finally, we introduce the Area Under the Receiver Operating Characteristic Curve as a way to measure the predictive performance of a machine learning model.

4.1 Electronic Health Records as a Temporal Sequence

Electronic Health Records (EHR) data can be regarded as a temporal sequence that describes the medical history of a patient as a number of hospital visits. EHR systems usually contain the types of data listed below (Lee et al., 2017). In the Dutch healthcare system, the healthcare providers and insurance companies make use of codes that describe a 'diagnose-behandelcombinatie' (DBC). A DBC represents standard treatment plan that is linked to a given diagnosis. These codes are used so that the healthcare providers can bill the insurance companies for the healthcare provided.

With the exception of the socio-demographic information, all data types have a temporal component that describes when something (e.g. a procedure or diagnosis) has happened. This temporal information makes it possible to see the EHR data as a temporal sequence. However, it is important to note that EHR data is not a fixed time-series because data is only recorded when a patient is at the hospital. Figure 4.1 shows an example of a data matrix that describes the medical history of a patient.

4.1.1 Formal EHR structure

A medical history can be regarded as a time-labelled sequence of observed variables. Let r be the number of variables. The medical history of the n-th patient (of N total patients) is then represented by a sequence of $T^{(n)}$ tuples $(t_i^{(n)}, x_i^{(n)}) \in \mathbb{R} \times \mathbb{R}^r$, where $i \in 1, ..., T^{(n)}$ and $T^{(n)}$ is number of hospital visits in the medical history of the n-th patient. $t_i^{(n)}$ denotes the time-stamp of the i-th visit in the medical

Data type
- Socio-demographic information
- Diagnoses
- DBC-codes
- Image data (such as an echo cardiograph)
- Lab Tests
- Procedures
- Medications
- Unstructured text data

Table 4.1: Data types that are usually stored in an EHR system.

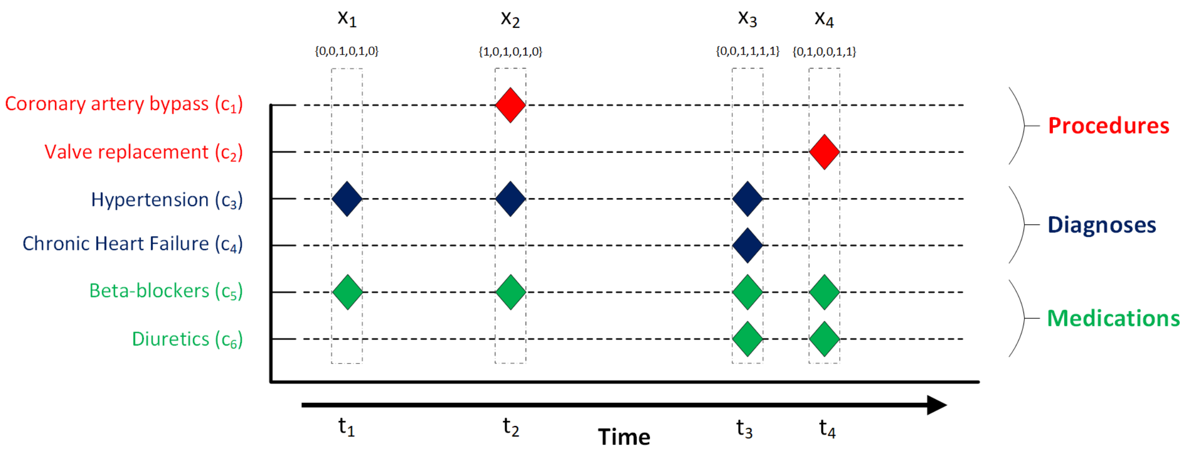


Figure 4.1: Example of EHR data. At every visit x_i, medical events such as procedures or medication prescriptions c_j and the time-stamp of their occurrence t_i are documented. The formal description of EHR structure is given in Section 4.1.1. Every visit x_i is encoded as a so-called 'one-hot' vector. This means that the j-th dimension of the vector is a 1 if the event c_j occurred in that visit.

history of the n-th patient. $x_i^{(n)}$ denotes the visit information of the i-th visit. Each visit is represented by a set of medical codes $\{c_1^{(n)}, c_2^{(n)}, ..., c_r^{(n)}\}$, where c_j is the j-th code from the vocabulary C. From this it follows that $r = |C|$ and $x_i^{(n)} \in \{0,1\}^r$ where the value one in the j-th coordinate means that c_j was observed in the i-th visit of the n-th medical history. See Figure 4.1 for an example of a medical history with the x_i's, t_i's and c_j's annotated. Hereafter we will drop the superscript (n) if it is unambiguous and describe the algorithms as if for the medical history of a single patient.

4.2 Challenges for Machine Learning using EHR data

Lee et al. (2017) describe several characteristics of EHR data that make machine learning more challenging. The first of these is high-dimensionality and sparsity. For instance in the case of diagnoses, the ICD-10[1] contains more than 14,000 diagnosis codes (Organization, 1993). In addition, a single patient is usually only diagnosed with a very small subset of the total number of possible diagnoses. The problems associated with increasing dimensionality are increased model complexity (and thus need for greater computational resources) and decrease in predictive power (with a fixed number of training examples). The next challenge has to do with the irregular time interval between data points. This is due to EHR data only being recorded when a patient visits the hospital and not when the patient is at home. In addition the number of recorded visits can also vary greatly between patients. Another challenge has to do with missing data. This can be the result of a data collection problem where patients are not checked for a condition, or a data documentation problem where the patients are checked for the condition but the outcome is not recorded (for whatever reason). The next challenge is that of noise in the data due to inconsistent coding or naming conventions. Finally, the last challenge has to do with bias that arises due to the fact that more data is usually recorded (e.g. more lab tests are performed) for a patient that is perceived to be more ill. In other words, the availability (or missingness) of data is related to the how ill the patient is. This is known as data Not Missing At Random (Scheffer, 2002) .

4.3 Area Under the Receiver Operating Characteristic Curve

The Area Under the Curve (AUC) –sometimes referred to as the C-statistic– is used as a measure to evaluate the discriminative ability of a model with a binary outcome. It ranges from 0.5 as the lowest

[1] International Statistical Classification of Diseases and Related Health Problems. It is a medical classification list maintained by the World Health Organization.

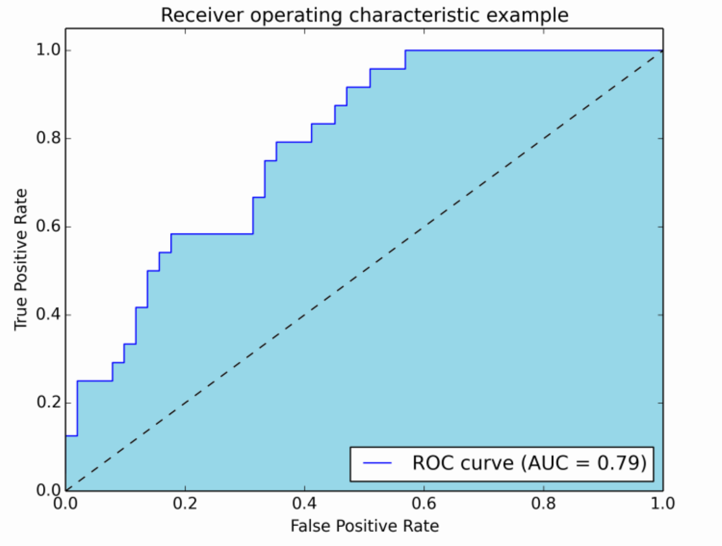


Figure 4.2: Receiver Operating Characteristic curve and Area Under the Curve. The true positive rate (y-axis) is plotted against the false positive rate (x-axis) at different classification thresholds. Resulting is the Receiver Operating Characteristic curve. The portion of the graph under the ROC curve is the AUC score.

possible value to 1 as the highest possible value. It is calculated by considering a range of classification thresholds (from 0 to 1) and plotting the false positive rate versus the true positive rate. The result of this is known as a Receiver Operating Characteristic (ROC) curve (shown in figure 4.2). Then, the AUC score is the fraction of the graph that is under the plotted ROC curve.

The AUC represents the probability that a randomly selected patient who experienced an event (e.g. re-hospitalisation within 30 days) had a higher risk score (according to the model) than a patient who has not experienced this event. It is most often used in situations where there is a class imbalance (e.g. there more patients that are not re-hospitalised within 30 days than patients who are). In these situations the AUC is more useful than the simple accuracy measure [2] because a simple model that classifies all patients as the majority class may score quite well on the accuracy measure but will not at all be able to discriminate between classes (and thus score low on the AUC measure). An alternative to the AUC score is the Area Under the Precision Recall Curve (AUPRC). Instead of plotting the true positive rate versus false positive rate, the AUPRC curve is calculated by plotting precision versus recall.

[2] Accuracy $= \frac{\text{\# of correctly classified patients}}{\text{\# of patients}}$

Chapter 5

Machine Learning Approaches

In this chapter we describe several machine learning approaches. We distinguish between approaches that take a single feature vector as input (Section 5.1), and those that are able to take a sequence of feature vectors as input (Section 5.2). Figure 5.1 highlights this distinction. The approaches considered in this chapter are included either because they are often used in the healthcare domain, or they are able to model sequential data.

5.1 Approaches that use a single feature vector

A way to use machine learning on EHR data is to construct aggregate features from the medical history of a patient (Wu et al., 2010; Wang et al., 2015). This means that the history of the patient with respect to a variable (i.e. a medical event) is denoted by a single variable value. To illustrate this, take the example of the hypertension diagnosis (figure 4.1). This is a diagnosis that is likely to occur multiple times in the medical history of a patient. To aggregate the medical history and represent it as a singe variable value, one could just count the amount of times the diagnosis was made and use that integer as the value for the variable 'hypertension'. An example of an aggregation of a numeric variable in the medical history is a measurement such as blood pressure. To aggregate this into a single value one could take the mean of the measures. Using aggregated features makes it possible to use 'traditional' data analysis methods such as Logistic Regression, Support Vector Machines, Decision Trees, etc. But a downside of using aggregated features is that the information about the sequence of medical events as well as the time between events are lost.

5.1.1 Logistic Regression

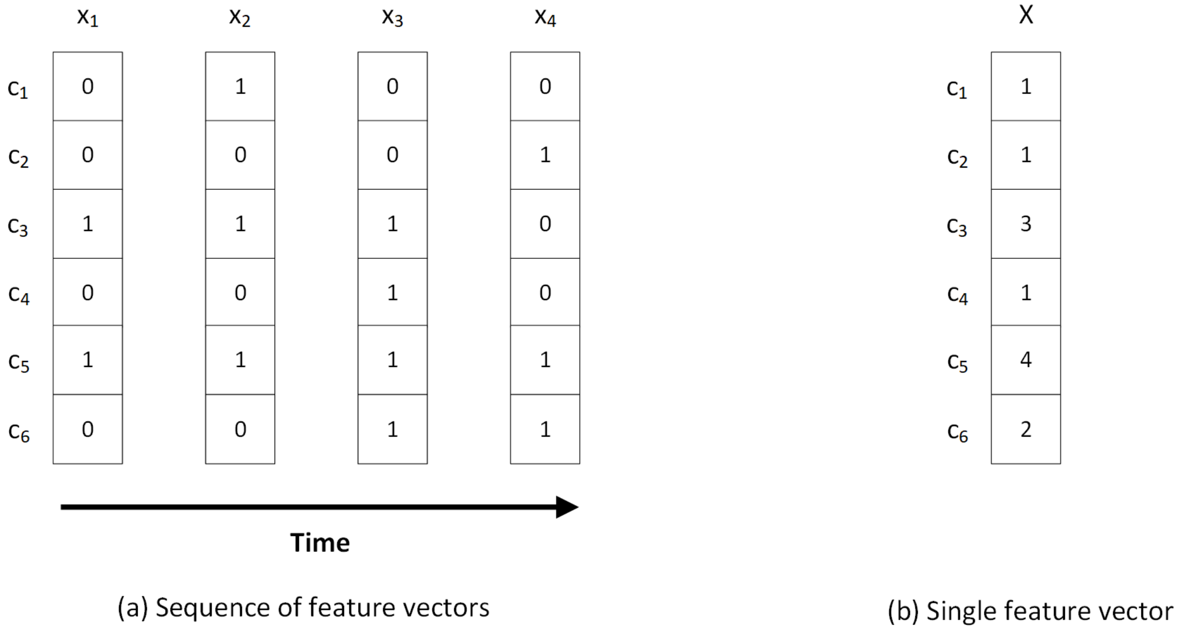
Logistic Regression (LR) is a model that is often used in the healthcare domain. Out of the 13 studies that aimed to predict Heart Failure re-hospitalisation from literature (Table 3.2), 9 used the LR model. The LR model can be represented as a layer of input nodes that have a weighted connection to the output node (see Figure 5.2). The input nodes contain the values of the feature vector. Then, the weighted connection from an input node to the output node determines influence of the input feature to the predicted risk score.

Trainable parameters	
$W_y \in \mathbb{R}^m$	Regression coefficients
$b_y \in \mathbb{R}$	Bias

m: input layer nodes (# of features)

$$\hat{y} = \text{sigmoid}(W_y x + b_y)$$

Table 5.1: Formal description and trainable parameters of LR. In this definition, the input features $(c_1, ..., c_j)$ are represented as a vector x, which has m dimensions. See Figure 5.2 for a graphical representation of the LR model. The sigmoid function is defined in Appendix E



	x_1	x_2	x_3	x_4
c_1	0	1	0	0
c_2	0	0	0	1
c_3	1	1	1	0
c_4	0	0	1	0
c_5	1	1	1	1
c_6	0	0	1	1

(a) Sequence of feature vectors

	X
c_1	1
c_2	1
c_3	3
c_4	1
c_5	4
c_6	2

(b) Single feature vector

Figure 5.1: Sequence versus Aggregate approaches. Two ways to represent EHR data. (a) is a sequence of feature vectors that does justice to the temporal dimension of EHR data as described in Section 4.1.1. (b) is a single feature vector which has ‘collapsed’ over the temporal dimension by summing up the occurrences of medical events. This single feature vector does not capture the temporal dimension.

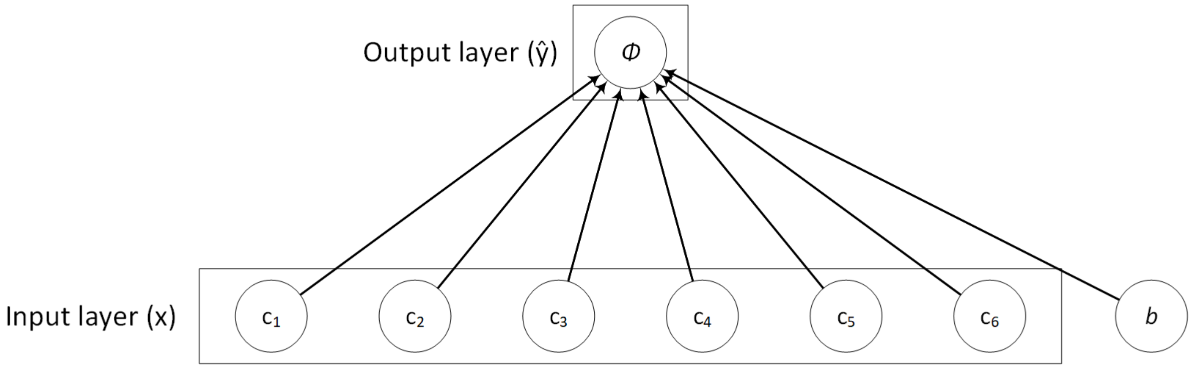


Figure 5.2: Logistic Regression (LR). The input layer x and the output layer y are shown. In this example, the number of input nodes m. ϕ represents the sigmoid activation function and b is the bias unit.

5.1.2 Cox Proportional Hazards Regression

Cox Proportional Hazards Regression analysis (CPHR) is an extension to Survival Analysis. Survival Analysis is a way to analyse datasets in which the dependent variable is the occurrence of an event (HF re-hospitalisation in our case) and we are interested in the expected time duration until that event will happen. Survival Analysis assumes that there is a 'survival function' that describes the predicted probability of survival (i.e. not re-hospitalised) after a given amount of time. In CPHR the aim is to take into account certain factors that influence survival probability other than just the amount of time that has passed. The result of a CPHR is a number of regression coefficients that correspond to a variable (just like in Logistic Regression). A coefficient represents the increase or decrease of the log of the hazard ratio relative to unit change in the variable that it corresponds to (holding all other predictors constant). The hazard ratio is similar to the odds ratio in Logistic Regression. The CPHR model can be written down as follows:

$$h(t) = h_0(t) \exp(b_1 X_1 + b_2 X_2 + \cdots + b_p X_p) \tag{5.1}$$

In this formula the b's are regression coefficients corresponding to the X's (the predictor variables). $h(t)$ is the hazard function at time t. To obtain the hazard ratio for a predictor variable one takes the exponent of the corresponding coefficient in the CPHR model. If the hazard ratio of a predictor variable is 1 it does not affect the probability of survival (i.e. the event happening). A hazard ratio smaller than 1 means that the presence (or unit increase of the value) of this variable increases the probability of survival (decreases the probability of the event happening). Conversely, a hazard ratio greater than 1 decreases the probability of survival. An important drawback of the CPHR model is the fact that it does not model the actual baseline hazard ($h_0(t)$) and can therefore not be used to perform an individualised prediction by 'plugging in' the values corresponding to a specific person. It can only be used to identify risk factors that increase/decrease the probability of survival. However with some additional work, Chin and Goldman (1997) and Krumholz et al. (2000) have used the CPHR model to predict risk of re-hospitalisation for HF patients. The main idea of their approach is to use CPHR to identify risk factors, and then look at the marginal distribution of re-hospitalisation given the identified risk factors.

The CPHR model assumes that the relative risk of two individuals with differing values for the coefficients is independent of the time that has passed (i.e. constant at all times). Babińska et al. (2015) have studied the case of a CPHR model in the context of survival from Acute Coronary Syndrome and have shown that it is improper to use a CPHR model without checking the proportional hazards assumption. The result of using a simple CPHR model in the case where the proportional hazards assumption is violated could result in the inclusion of seemingly 'time-independent' risk factors which are in fact time-dependent. The study by Chin and Goldman (1997) does not report whether they checked the proportional hazards assumption. Krumholz et al. (2000) state that "the proportional hazard assumption was confirmed graphically". See Appendix C for a more comprehensive description of these studies.

5.1.3 Multilayer Perceptron

A Multilayer Perceptron (MLP) is an interconnected group of nodes consisting of at least one input layer, an arbitrary number of hidden layers and one output layer (see Figure 5.3). Each node in a layer has a weighted connection to every node in the next layer. These weighted connections are optimized during the training of the network. The nodes in the input layer take the values of the predictor variables. In every node in the hidden layer there is an activation function[1] that takes as input the weighted sum of the output of all the nodes in the previous layer plus a bias term. The result of the function is further propagated to the nodes in the next layer. If the node is in the final layer, the result of the activation function is the output of the model.

Tu (1996) describe the advantages and disadvantages of using a MLP over LR for predicting clinical outcomes. The main advantage is the fact that a MLP with a single hidden layer with a finite number of nodes is a universal approximator (Csáji, 2001). This means that in theory, the model can approximate any measurable function to any desired degree of accuracy. In other words, a MLP can model highly complex and non-linear relationships between the predictor variables and the clinical outcome. This power comes from the nodes in the hidden layer that act as automatic feature detectors. In theory, LR is

[1] Activation functions that are often used in practice are the logistic sigmoid function, the hyperbolic tangent function (tanh) or the rectified linear function (ReLu). See Appendix E for the definition of these functions.

Trainable parameters

$W_h \in \mathbb{R}^{p \times m}$	Hidden layer weights
$b_h \in \mathbb{R}^p$	Hidden layer bias
$W_y \in \mathbb{R}^p$	Output layer weights
$b_y \in \mathbb{R}$	Output layer bias

m: input layer nodes (# of features)
p: hidden layer nodes

$$h = \phi(W_h x + b_h)$$
$$\hat{y} = \phi(W_y h + b_y)$$

Table 5.2: Formal description and trainable parameters of MLP. In this definition, the MLP has a single hidden layer and an output layer consisting of one node (like shown in Figure 5.3). The auxiliary function ϕ can be any activation function such as sigmoid or tanh. See Appendix E

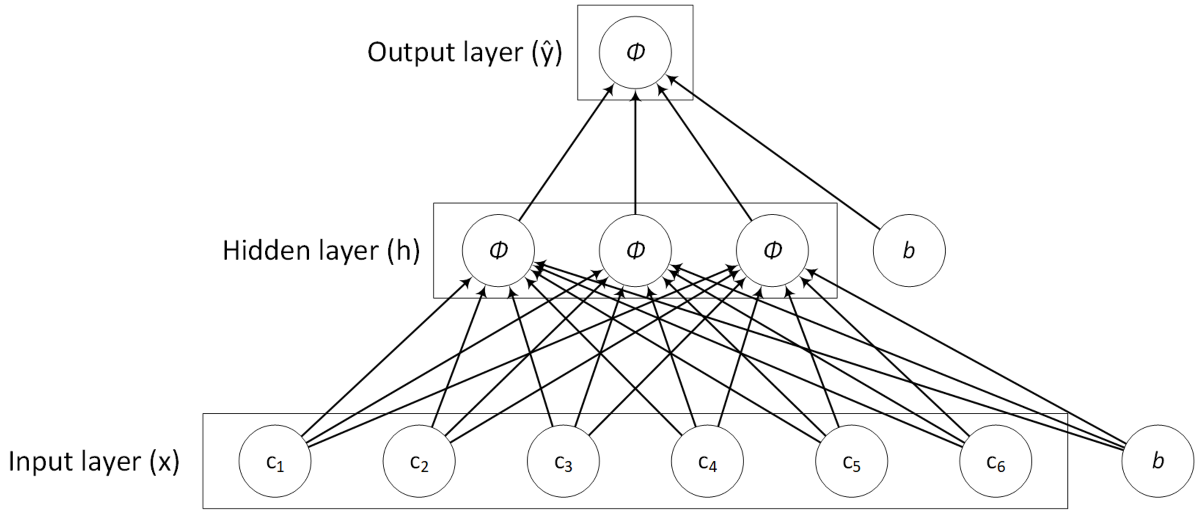


Figure 5.3: Multilayer Perceptron (MLP). A simple Multilayer Perceptron with a single hidden layer. The input layer x, the hidden layer h and the output layer y are shown. In this example, the number of input nodes m is 6 and the number of hidden layer nodes p is 3. ϕ represents the activation function and b is the bias unit.

also able to model complex and non-linear functions but this requires an explicit search by the model developer and may require complex transformations of the data. The same goes for interactions between predictor variables. The hidden layer in a MLP is able to implicitly detect these interactions while LR requires explicit modelling by the developer.

Unfortunately there are also some disadvantages of using a MLP over LR. The first of these is that MLPs require more computational resources. Due to the large number of connection weights that need to be optimized in an iterative fashion, the time needed to train a MLP can be considerable. The second disadvantage is that MLPs are prone to overfitting. This means that the MLP will perform really well on the dataset that is was trained on, but will perform a lot worse on new data. However, there are different strategies such as using dropout, regularization and early stopping that can be implemented to try and reduce the effects of overfitting (Caruana et al., 2001; Girosi et al., 1995; Srivastava et al., 2014). The third and most consequential disadvantage is the difficulty in the interpretation of the model (see section 6.1).

So far, MLPs have been used in a wide variety of contexts in the healthcare domain (Amato et al., 2013). The specific use of a MLP for predicting (re-)hospitalisation for HF patients has been reported by Atienza et al. (2000). However, this study suffers from the major drawback of developing the MLP model on only 123 patients and not using cross-validation or training/test set splitting. This makes it quite likely that the model has overfitted the training data. The reported sensitivity and specificity for re-hospitalisation for HF patients on the training set are 0.8 and 0.94 respectively.

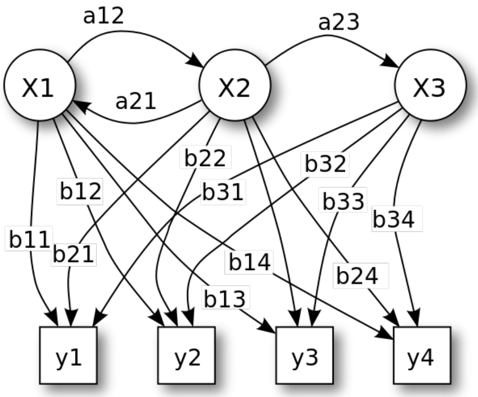


Figure 5.4: Hidden Markov Model (HMM). A simple Hidden Markov model with states (X), observations (y), state transition probabilities (a) and observation probabilities (b) (from wikipedia)

5.2 Approaches that use a sequence of feature vectors

One of the characteristics of clinical data as described in Chapter 4 is that there is an important temporal dimension. That is to say, the sequence of –and the times between medical events may contain information that could be useful in predicting a clinical outcome. The approaches considered up to this point do not effectively take this temporal dimension into account and may therefore be missing information that is useful in predicting a clinical outcome. In this chapter we describe two approaches that are better able to model the temporal dimension.

5.2.1 Hidden Markov Model

A way to model a sequence is with a Hidden Markov Model (HMM). This model assumes that the sequence it aims to model is a 'Markov process', which means that the probability of the next state (step) in the sequence is dependent only upon the current state and is independent of all the previous states of the sequence. The architecture of the HMM consists of states (that are hidden) and observations (that are not hidden). In addition the model describes transition probabilities between every state and output probabilities between every state and every observation. Figure 5.4 shows a HMM consisting of 3 hidden states and 4 observations.

Lipton et al. (2015a) cite a couple of problems that HMMs have in learning from sequence data. The first of these has to do with the computational complexity of the model, especially when the amount of hidden states (S) becomes large. This is because the HMM learning algorithm scales with $O(|S|^2)$ and the set of state transition probabilities is of size $|S|^2$. The other problem is the assumption that the sequence is a Markov process. This assumption is not likely to be valid in medical history data. As an example, the diagnosis of a myocardial infarct in the past of a patient will probably maintain predictive significance longer than just one step in the sequence. HMMs can be extended with a context window such that they incorporate more previous states. However, this procedure grows the state space S exponentially with the size of the context window thus compounding the problem of computational infeasibility.

Due to the problems mentioned above, we conclude that the HMM model is not a great option in the context of predicting re-hospitalisation for HF patients.

5.2.2 Recurrent Neural Network

A Recurrent Neural Network (RNN) is a type of neural network that is able to model sequential data. Oversimplified, it can be explained as a sequence of MLPs in which the nodes in the hidden layer of one MLP have weighted connection to the nodes in the previous MLP. In other words, there is a recurrent connection in the hidden layer of the network. This means that the hidden state at a given timestep is

Trainable parameters

$W_h \in \mathbb{R}^{p \times m}$	Hidden layer weights
$b_h \in \mathbb{R}^p$	Hidden layer bias
$W_r \in \mathbb{R}^{p \times p}$	Recurrent weights
$W_y \in \mathbb{R}^p$	Output layer weights
$b_y \in \mathbb{R}$	Output layer bias

m: input layer nodes (# of features)
p: hidden layer nodes

$$h_i = \phi(W_r h_{i-1} + W_h x_i + b_h)$$
$$y_t = \phi(W_y h_t + b_y)$$

Table 5.3: Formal description and trainable parameters of RNN. Given the input sequence $x_1, ..., x_t$ the RNN model outputs y_t. In this definition, the RNN has a single hidden layer and an output layer consisting of one node (like shown in Figure 5.5). The auxiliary function ϕ can be any activation function such as sigmoid or tanh (Appendix E).

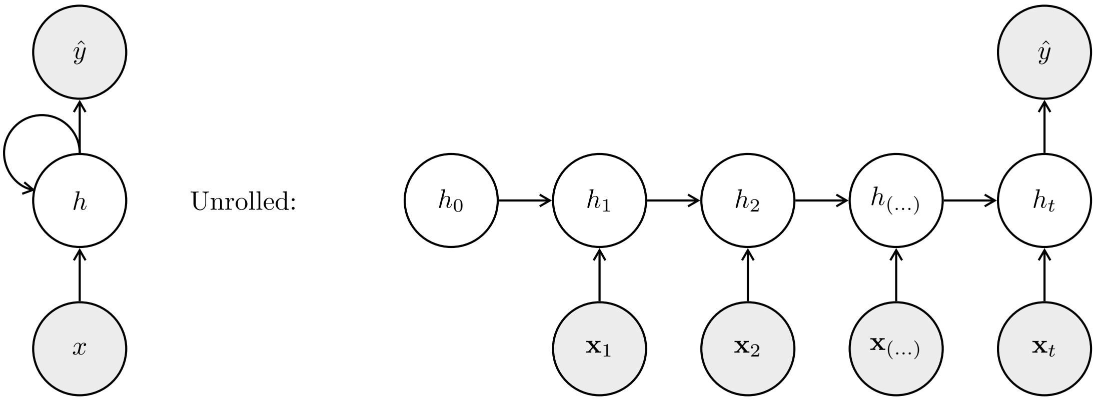


Figure 5.5: Recurrent Neural Network (RNN). A simple RNN architecture with input layer x, hidden recurrent layer h and output layer $\hat{y}$. Note that each x_i consists of m nodes and h_i consists of p nodes (as per Table 5.3).

dependent upon the input at that timestep, as well as the previous hidden state. Formally, given the sequence $x_1, ..., x_t$, the hidden layer of a RNN h_i is $\phi(x_i, h_{i-1})$. The activation function ϕ determines the information that is propagated further through the network. Because the RNN model is in essence a sequence of MLPs with a hidden layer to hidden layer connection, the same advantages and disadvantages that were mentioned for the MLP model in Section 5.1.3 apply to the RNN model. Also, the same kind of overfitting prevention strategies (e.g. dropout and regularization) can be used.

Figure 5.5 shows a graphical representation of the RNN model architecture. It is important to note that in this representation the circles denote *layers* of nodes of the network and not individual nodes. Table 5.3 shows the formal definition of a 'simple' RNN.

The RNN model has recently been applied in several healthcare applications. (Choi et al., 2016a; Lipton et al., 2015b; Esteban et al., 2016; Lipton et al., 2016). The RNN-based models consistently outperform models that use aggregate features (e.g. LR & MLP). A study aimed at specifically predicting re-hospitalisation for heart failure patients is not reported in literature. As previously mentioned, the RNN model is notoriously hard to interpret. However, the notion of interpretability requires further elaboration. We will discuss the need for interpretability and the properties of interpretable models in the next chapter.

Part III

Theory

Chapter 6

Model Interpretability

In the following chapter we discuss the interpretability taxonomy that was proposed by Lipton (2016). Then, we use the taxonomy to rank several models that were discussed in Chapter 5 on the interpretability continuum. Finally, we reflect on the apparent trade-off between predictive performance and interpretability.

6.1 Properties of Interpretable Models

Interpretability is hard to define because it can not be directly measured. However, in order to make any meaningful assertions regarding a models' interpretability, a definition is needed. Lipton (2016) suggest that interpretability is not a monolithic concept but propose a taxonomy that describes different techniques and model properties that are related to its interpretability. They are organized according to two categories. The first has to do with how the model works (**Transparency**). The second has to do with the extra information a model can provide to a user (**Post-hoc Interpretability**). We will discuss the elements of this taxonomy below.

6.1.1 Transparency

Transparency describes the degree to which the inner workings of a model can be easily understood by a human being. It is the antithesis of opacity or *blackbox-ness*. The transparency of a model can be assessed at different levels. These are: the whole model (**Simulatability**), its singular components (**Decomposability**) and its training algorithm (**Algorithmic Transparency**). We will use LR and MLP as examples in discussing the three transparency levels.

Simulatability

If a human being can (in a reasonable timespan) do the calculations that happen when a model makes a prediction, the model is considered to be simulatable. For example in the case of a small LR (with 4 input features), a human can easily look at the regression coefficients, make the calculations and simulate the prediction. However, the LR model becomes less simulatable as the number of input features grows because it would take a human more time to simulate the calculations of the model. In the case of an MLP with an input layer consisting of 20 nodes and a hidden layer consisting of 100 nodes, it becomes pretty clear that it not a simulatable model because of the number of calculations that have to be made to simulate the model. On the other hand, an MLP with only 2 input nodes and 4 hidden nodes can actually be easily simulated. This shows that simulatability is not necessarily intrinsic to a type of model but has more to do with the hyper-parameters of a model.

Decomposability

Decomposability has to do with the ability of each input, parameter and calculation of the model to be intuitively explained. Let's look again at LR. A single weighted connection can be easily explained as the contribution to the prediction of the input feature that it is connected to. subset of observations that satisfy a specific criteria (i.e. the split criteria). Also, the calculation at each node is quite simple to explain as it consists of a multiplication between the feature value and the connection weight. For a given parameter of a MLP (i.e. a weighted connection between two nodes), it is not at all easy to explain its relationship with the input and output of the network because it is unclear what happens in the hidden layer. The same holds for the calculations (i.e. node activations) in a MLP. Knowing the activation value of any given node in the network is not easily translatable to a link between the input and output of the model. With regards to the decomposability of the input the LR and MLP models are not a-priori different. The decomposability of the input has to do with the amount of feature engineering that has been performed. Both the LR and MLP models are able to use as input either 'raw' and simple features or heavily engineered features. One thing to note is that feature engineering is a less common practice for the MLP model because the hidden layer of a MLP performs implicit feature engineering anyway.

Algorithmic Transparency

Algorithmic transparency has to do with the mathematical guarantees that can be given regarding the solution of the training algorithm. The gradient descent algorithm (that is used for training LR and MLP) is a greedy search algorithm. It looks at the gradient of the loss function with respect to the weights in the network and incrementally updates the weights in the direction opposite to the gradient. In this sense, the algorithms of LR and MLP are less than fully transparent.

6.1.2 Post-hoc Interpretability

Besides looking at the inner workings of a model and considering its interpretability, one could look at smart techniques to derive extra information from a trained model and present it to an end-user. These 'post-hoc explanations' can allow less transparent models (such as a MLP) to become more interpretable. In some sense, a good analogy is the human brain (as a black-box system) that provides post-hoc explanations for its behaviour even though we know that at the mechanical level of the brain these explanations do not make any sense. One important thing to note about post-hoc explanations is that they do not necessarily reflect the true inner workings of the model. Therefore they can potentially mislead humans into thinking the model works in some way while in reality it does not.

Text Explanations

A common way for a human to confer an explanation to another human is by giving a chain of reasons that can be written down as text. Understanding this type of explanation comes naturally to a human so therefore it makes sense to try and generate textual explanations for the predictions a model makes. One approach for this is to concurrently train two models. One prediction model, and another that uses the internal representations of the prediction model to produce a textual explanation.

Visualization

Humans can grasp complex entities with their visual system and pattern recognition capabilities. Granted, these systems are not perfect but given the right visualization, they allow humans to almost instantly gain a better understanding of a dataset and/or prediction model. Visualization techniques might be best suited models that learn rich representations of the data (because there is more to visualize). One way to use a visualization to interpret a model is to manipulate the input to the model to see how the visualization changes.

Local Explanations

Instead of trying to explain the whole model, efforts can be made to try and explain a smaller (local) part of the model. Although an explanation for the whole model is of course more desirable than just a local explanation. However, if a global explanation is unattainable, a local explanation might still be quite useful. For instance, Ribeiro et al. (2016) have proposed a technique to make a local sparse linear approximation to be used as an explanation for a model's predictions.

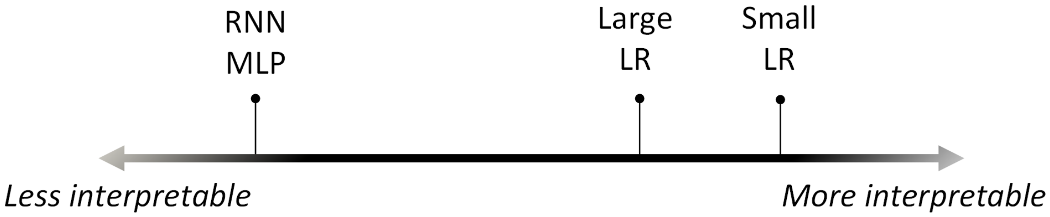


Figure 6.1: Model Interpretability Continuum. The models more to the left are less interpretable (e.g. RNN & MLP) than the models more to the right (LR).

Explanation by Example

Another way in which humans confer explanations to each other is by an analogy. They justify some decision by explaining that the situation was quite similar to another situation in which the same decision turned out to be a good one. In the same vein, a model could show the user the predicted label along with an observation (for which the label is known) that is most similar to the observation that gave rise to the prediction. Similarity can for example be calculated using a proximity measure in the space of the hidden representation.

6.2 The Interpretability Continuum

Typically, as the number of parameters of a model increases, it becomes increasingly harder to simulate and therefore more difficult for a human to interpret. For example, a LR with a small number of input features is more interpretable than a LR with a large number of input features. This is because with a small LR, it is easier to simulate the model by taking the input and simulating the calculations.

The MLP and RNN model have a reputation for being difficult to interpret. This is because the hidden layers learn increasingly abstract (hidden) features from the data that are represented by a set of weighted connections between nodes. This set of weighted connections is not easily translated to an intuitive explanation (i.e. decomposed), especially as the number of hidden nodes becomes large.

Concluding, even though it is difficult to objectively measure interpretability, it is still possible compare models with regards to their interpretability using the taxonomy proposed by Lipton (2016). The lack of an objective measure means that there is not a clear dichotomy between models that are interpretable and models that are not interpretable. Therefore, we consider model interpretability as a continuum along which different models are located according to their transparency and the post-hoc explanation techniques they allow.

Figure 6.1 shows the models previously discussed placed on the interpretability continuum. The small LR is ranked most towards the right of the continuum because it is the most simulatable and decomposable of the models. The large LR is ranked just below that because with increasing tree size, the model becomes less simulatable. The MLP & RNN models are ranked more towards the left side of the continuum because a 'normally' sized model (i.e. tuned for predictive performance) quickly becomes non-simulatable and also its individual nodes/weights do not allow for an intuitive explanation (i.e. non-decomposable).

6.3 Interpretability / Predictive performance Trade-off

The bias-variance decomposition describes the elements that contribute to the prediction error of a model on test data (Friedman, 1997). The bias-term is related to the errors made because of improper assumptions built into the model. On the other hand, the variance-term is related to the errors that are due to the model capturing 'random noise' from the training data. Typically as one increases model complexity, the bias-term will decrease while the variance-term will increase. In order to obtain the best possible predictive performance, one must manage the bias-variance trade-off and pick a model (and its hyper-parameters) with the right amount of complexity. Looking at just the bias-term of the decomposition, we should expect more complex models to outperform less complex models. Also, looking at the concepts of complexity and simulatability we can see that increasing model complexity is bound to decrease its simulatability (and thus its interpretability). Therefore, we would expect to see a trade-off between interpretability and predictive performance.

Chapter 7

Extending the Recurrent Neural Network model

In this chapter we return to the RNN model and consider several extensions that aim to improve it. These extensions are the foundations of the 'interpretable' RNN-based models that will described in Chapter 8. The first three extensions aim to improve the predictive performance of the RNN model. The final extension aims to both improve predictive performance, as well as interpretability.

7.1 Advanced Cell Architectures

Due to the recurrent nature of the hidden layer to hidden layer connection, the model can in theory capture long range dependencies in the sequence. However in practice, in a simple RNN architecture the long-range dependencies may not actually be learned due to the problem of the 'vanishing gradient' (Hochreiter, 1998). This problem is the result of performing a squashing non-linear transformation (e.g. sigmoid or tanh) on the gradient of the error function at every timestep. In other words, the information about the gradient of the error shrinks at every timestep and becomes so small that the slope of the error function can not be detected and therefore the network can not learn from a mistake made more than a few timesteps in the past. Proposed solutions for this problem are Long-Short Term Memory (LSTM) networks (Hochreiter and Schmidhuber, 1997) and more recently the Gated Recurrent Unit (GRU) (Chung et al., 2014).

Hochreiter and Schmidhuber (1997) have proposed the LSTM architecture (Figure 7.2a). Reconsider the simple node from the MLP that takes as input the weighted values of its input nodes, applies a transformation function and outputs the value to all the nodes in the next layer. A node in a LSTM network is extended with an input gate, forget gate and output gate. These gates control the flow of information through them. This allows the network to learn what parts of the input to ignore (input gate), what to forget from its current hidden state (forget gate) and what parts of the hidden state to output to the next layer and timestep (output gate). The other RNN cell architecture that has recently become popular is the Gated Recurrent Unit (Chung et al., 2014) shown in Figure 7.2b. The main difference between the two is that the GRU combines the input and forget gates into an update gate thus reducing the complexity of the model slightly while having similar results. The LSTM and GRU solutions to the vanishing gradient problem are so effective that every implementation of a RNN described in the literature makes use of them. Some examples are Bajor and Lasko (2016) (LSTM & GRU), Choi et al. (2016e) (GRU) & Choi et al. (2016a) (GRU).

We apply the GRU-based RNN in this research because they are a bit simpler in terms of trainable parameters than the LSTM-based RNN. The calculation of a hidden state in the GRU architecture shown in Table 7.1.

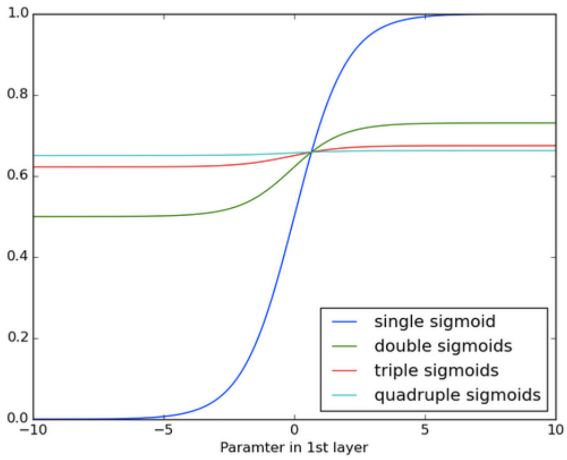
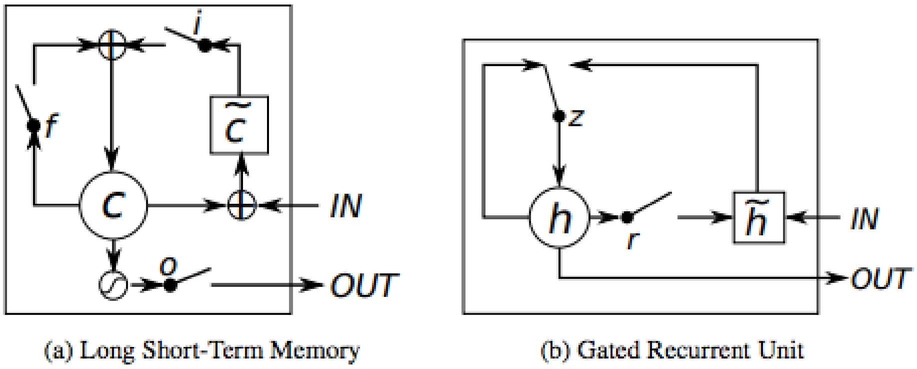


Figure 7.1: Multiple Sigmoids. The effect of multiple sigmoid transformations Deeplearning4j (2017). The error signal becomes increasingly small as multiple sigmoid transformations are performed to the point where the model is not able to 'learn from' mistakes made multiple timesteps in the past.



(a) Long Short-Term Memory

(b) Gated Recurrent Unit

Figure 7.2: LSTM and GRU. The LSTM (a) and GRU (b) cell architectures from Deeplearning4j (2017). The LSTM- and GRU-based RNN are better able to deal with the vanishing gradient problem and can therefore better capture long-range dependencies.

Trainable parameters

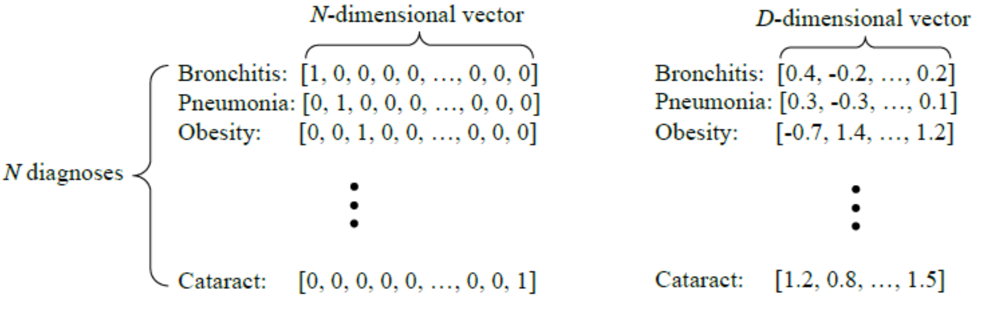
$W_r \in \mathbb{R}^{p \times m}$	Reset gate (input) weights
$U_r \in \mathbb{R}^{p \times p}$	Reset gate (hidden) weights
$b_r \in \mathbb{R}^{p}$	Reset gate bias
$W_z \in \mathbb{R}^{p \times m}$	Forget gate (input) weights
$U_z \in \mathbb{R}^{p \times p}$	Forget gate (hidden) weights
$b_z \in \mathbb{R}^{p}$	Forget gate bias
$W_h \in \mathbb{R}^{p \times m}$	Output gate (input) weights
$U_h \in \mathbb{R}^{p \times p}$	Output gate (hidden) weights
$b_h \in \mathbb{R}^{p}$	Output bias

m: input layer dimensions

p: hidden layer dimensions

$$
\begin{aligned}
r_t &= \text{sigmoid}(W_r x_t + U_r h_{t-1} + b_r),\\
z_t &= \text{sigmoid}(W_z x_t + U_z h_{t-1} + b_z),\\
\tilde{h}_t &= \tanh(W_h x_t + r_t \odot (U_h h_{t-1}) + b_h),\\
h_t &= (1 - z_t) \odot h_{h-1} + z_t \odot \tilde{h}_t,
\end{aligned}
$$

Table 7.1: Formal description and trainable parameters of GRU. The $\odot$ symbol denotes the element-wise multiplication operation. The auxiliary functions (such as sigmoid and tanh) are defined in Appendix E


N-dimensional vector
D-dimensional vector
N diagnoses
Bronchitis: [1, 0, 0, 0, 0, …, 0, 0, 0]
Pneumonia: [0, 1, 0, 0, 0, …, 0, 0, 0]
Obesity: [0, 0, 1, 0, 0, …, 0, 0, 0]
Cataract: [0, 0, 0, 0, 0, …, 0, 0, 1]
Bronchitis: [0.4, -0.2, …, 0.2]
Pneumonia: [0.3, -0.3, …, 0.1]
Obesity: [-0.7, 1.4, …, 1.2]
Cataract: [1.2, 0.8, …, 1.5]

(a) One-hot encoding for diagnoses (b) A better representation of diagnoses

Figure 7.3: Medical Concepts in Efficient Representation. Example of a tranformation from 'one-hot' into an efficient representation from Choi et al. (2016d). The efficient representation has a smaller dimensionality and captures semantic relatedness. An example of semantic relatedness can be seen in the diagnoses 'Bronchitis' and 'Pneumonia' (both lung-related) that are close to each other in the vector space. Both are further away from 'Obesity' and 'Cataract' (not lung-related).

7.2 Representation Learning

Another extension to RNN model is that of efficient representation learning. The goal here is to transform the 'one-hot' encoding of the input $x_i \in \{0,1\}^{|C|}$ to a more efficient representation $\mathbf{v}_i \in \mathbb{R}^m$ (see Figure 7.3). The representation is more efficient in the sense that it has a lower dimensionality but also in the sense that it captures a semantic relationship between the inputs. Semantic relatedness in this approach is characterized by the distance between points in the vector space. An example is shown in Figure 7.3. Choi et al. (2016d) and Choi et al. (2016b) have shown that learning such a representation improves the prediction performance for traditional methods (LR, KNN & SVM) as well for RNNs.

7.3 Bi-directional Recurrent Neural Networks

In addition to allowing the RNN to model the temporal sequence in one direction (from x_1 to x_t), it is also possible to model the sequence from x_t to x_1. This idea is implemented in the the Bi-directional Recurrent Neural Network (BRNN) as proposed by Schuster and Paliwal (1997). The BRNN architecture uses a forward and backward RNN to model the sequence in both directions. The forward RNN reads the input sequence from x_1 to x_t and results in a sequence of forward hidden states $(\overrightarrow{h_1}, ..., \overrightarrow{h_t})$. The backward RNN reads the input sequence from x_t to x_1 and results in a sequence of backward hidden states $(\overleftarrow{h_1}, ..., \overleftarrow{h_t})$. Concatenating the forward and backward hidden state results in the bi-directional hidden state $h_i = [\overrightarrow{h_i}; \overleftarrow{h_i}]$.

7.4 Neural Attention Mechanism

The neural attention mechanism has been proposed by Mnih et al. (2014) in the field of computer vision and was later refined by Bahdanau et al. (2014) for use in machine translation. It has been shown to be successful in different contexts (Yang et al., 2016) , (Ba et al., 2014) and (Chorowski et al., 2015). The attention mechanism can be explained as allowing the model to look at every element of a sequence (e.g. a sequence of hidden states in a RNN and) 'pay attention' to the elements that are important. Given a sequence of length T, with elements $h_1, ..., h_T$, we generate attention values α_i for $i = 1, ..., T$. Then, the elements are multiplied by their attention value and summed to generate the 'context vector' $\mathbf{c} = \sum_i \alpha_i h_i$. The idea is that the context vector contains the information from every hidden state that is important for making a correct prediction.

In the context of a medical history (such as shown in Figure 4.1), the attention mechanism can be used on two levels. The first of these is on a visit-level. In this case the attention mechanism learns which

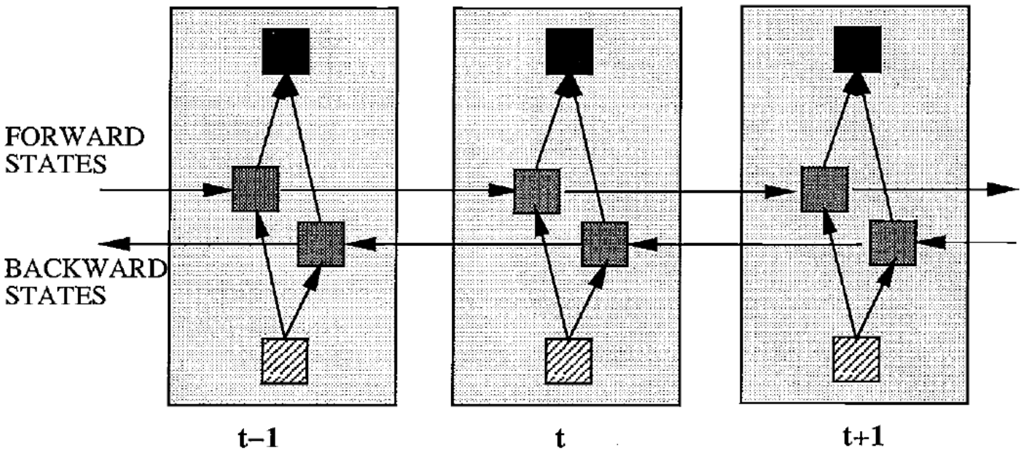


Figure 7.4: Bi-directional Recurrent Neural Network (BRNN). An illustration of the BRNN model from Schuster and Paliwal (1997). The input sequence is modelled in both directions resulting in a forward and backward hidden state. The forward and backward hidden states are concatenated to obtain the final hidden state at each timestep.

encoded visits are most important for making a correct prediction. The other level at which an attention mechanism can operate is on the event-level. In this case the attention mechanism learns which medical events are most important for making a correct prediction.

See Figure 7.5 for a graphical representation of the attention mechanism in a sentence summarization task. The distribution of attention values over the sentence shows that words ‘win’ and ‘victorious’ are important when predicting the word ‘beat’.

7.4.1 A note on correlation and causation

Like with any supervised machine learning algorithm, the ‘interpretable’ RNN-based models that will be discussed in Chapter 8 make predictions on the basis of correlations in the training data. It may be tempting to interpret the distribution of attention as a describing a causal relationship between the elements of the medical history of a patient and the re-hospitalisation event. When data shows that events A and B are correlated, any of the following relationships are possible (Novella, 2009):

- A causes B
- B causes A
- A and B are caused by C (which may be unobserved)
- There is no causal relation between A and B (the correlation is a coincidence)

These possible relationships show that it is unwarranted to infer a causal relationship simply from correlation. Having said that, looking for correlations can still be useful as they can provide a trigger for more comprehensive investigation into the causal relationship between events.

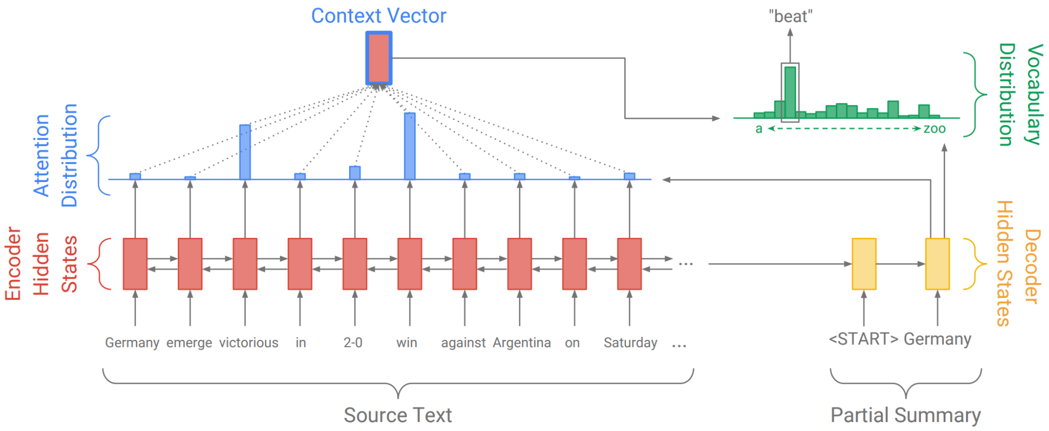


Figure 7.5: Neural Attention Mechanism. The attention mechanism in a sentence summarization task from See et al. (2017). The attention distribution shows that words like 'win' and 'victorious' are more important than others when predicting the word 'beat'.

Chapter 8

‘Interpretable’ Recurrent Neural Networks

Choi et al. (2016c), Ma et al. (2017) and Sha and Wang (2017) have proposed RNN-based model architectures for diagnosis prediction in the healthcare domain (Figure 8.1). These models have been specifically developed in order to allow for better interpretation than a ‘regular’ RNN would. They aim to achieve this by implementing an attention mechanism that is able to tell the user what the model focuses on when making a prediction. This improves the decomposability of the RNN model because it provides an element of the model – the attention weights – that can be used to give an intuitive explanation. In this chapter we will outline these models and show how they implement the RNN extensions that were described in Chapter 7. Then, we revisit the interpretability continuum described in Section 6.2 and include the RNN-based models outlined in this chapter.

In the following sections we provide for each model its formal description and its trainable parameters. In addition we provide an ‘unrolled’[1] overview of what the model would look like if the medical history from Figure 4.1 was the input of the model. Also, in order to keep the description simple we have left out how we incorporated the temporal information. A full description of how we incorporated the temporal information can be found in Appendix F.

8.1 Diagnosis Prediction Model (Dipole)

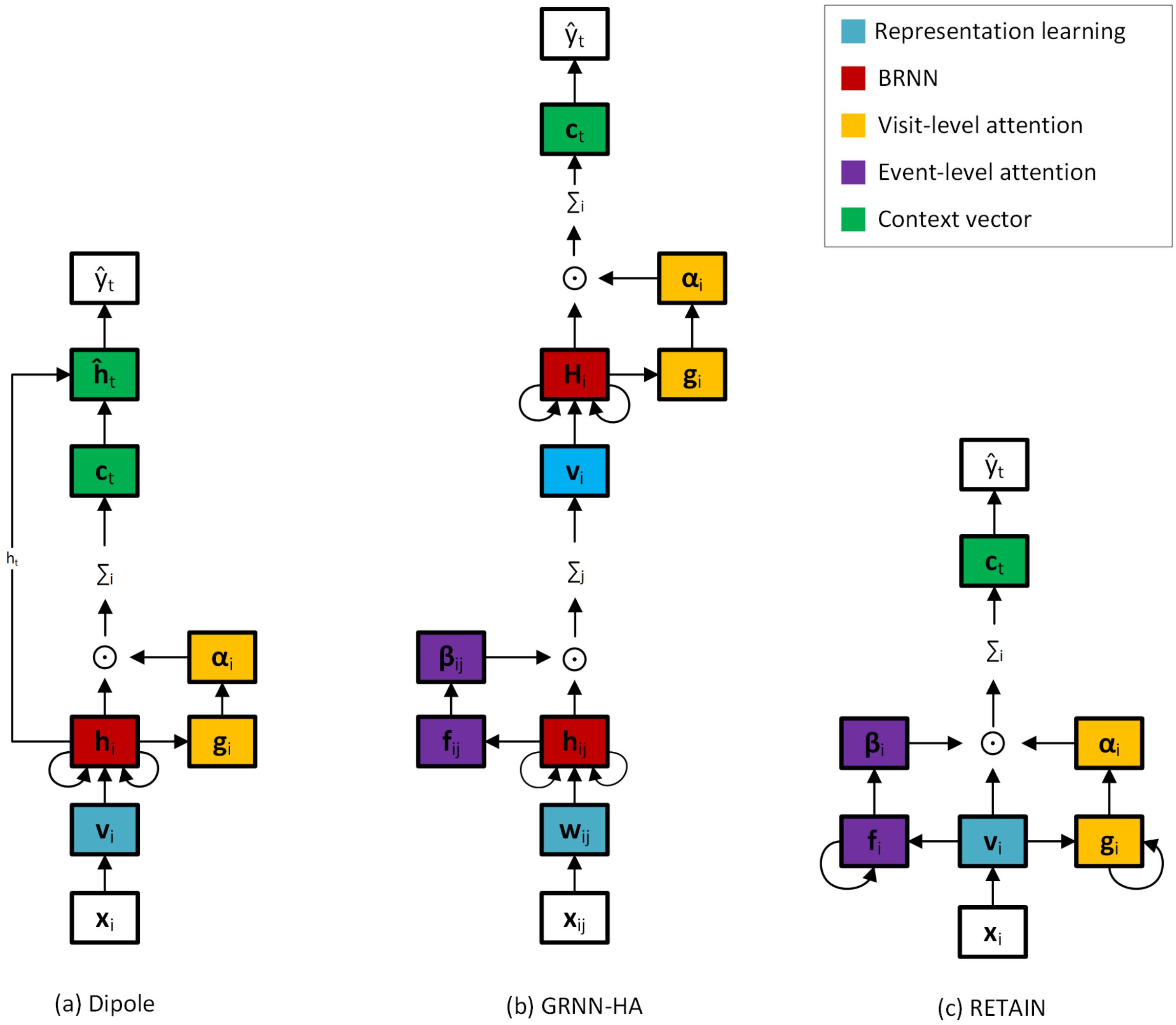
Figure 8.1a shows the high-level architecture of the Dipole model as proposed by Ma et al. (2017). Figure 8.2 shows the ‘unrolled’ overview of the model. Table 8.1 contains the formal description and the trainable parameters of Dipole. The authors describe three attention mechanisms but state that the ‘location-based’ attention mechanism performed better than the other two. In our description of the Dipole model we have implemented this attention mechanism.

The attention mechanism of Dipole allows the model to explain how much attention each timestep in the sequence of hidden states $h_1, ..., h_t$ receives. The amount of attention on an h_i can be interpreted as its importance to the calculation of the output y_t. However, due to the transformation of v_i by the bi-directional recurrent layer into h_i, one is not justified in using the attention on h_i as a basis for a claim about the importance of solely v_i. This is because h_i is the result of weights that are connected to h_{i-1}, h_{i+1} as well as v_i. In turn, the the hidden state h_{i-1} is connected to v_{i-1} and h_{i-2} and so on. In addition, because the context vector c_t is concatenated with the last hidden state h_t in the sequence, it becomes less clear what the relative impact of the states $h_1, ..., h_{t-1}$ versus h_t is.

8.2 GRU-based RNN with Hierarchical Attention (GRNN-HA)

Figure 8.1 shows the high-level architecture of the GRNN-HA model as proposed by Sha and Wang (2017). Table 8.2 contains the formal description and the trainable parameters of GRNN-HA. The authors state that they used the word2vec algorithm to learn the medical event embedding matrix. However, we used a

[1] An unrolled visualisation of a RNN removes the cycle (recurrence) and instead shows every timestep of the input.



(a) Dipole

(b) GRNN-HA

(c) RETAIN

Figure 8.1: 'Interpretable RNN'. High-level overview of 'interpretable' RNN-based models for medical event prediction. Dipole (Section 8.1), GRNN-HA (Section 8.2) and RETAIN (Section 8.3). The color coding illustrates the implementation of the concepts described in Chapter 7. Given a sequence of hospital visits $x_1, ..., x_t$ (with a single visit indexed as x_i), the models produce the prediction y_t. The description of the other elements of the models are given in Table 8.1, 8.2 & 8.3 respectively. The unrolled overview of the models is given in Figure 8.2, 8.3 & 8.4 respectively.

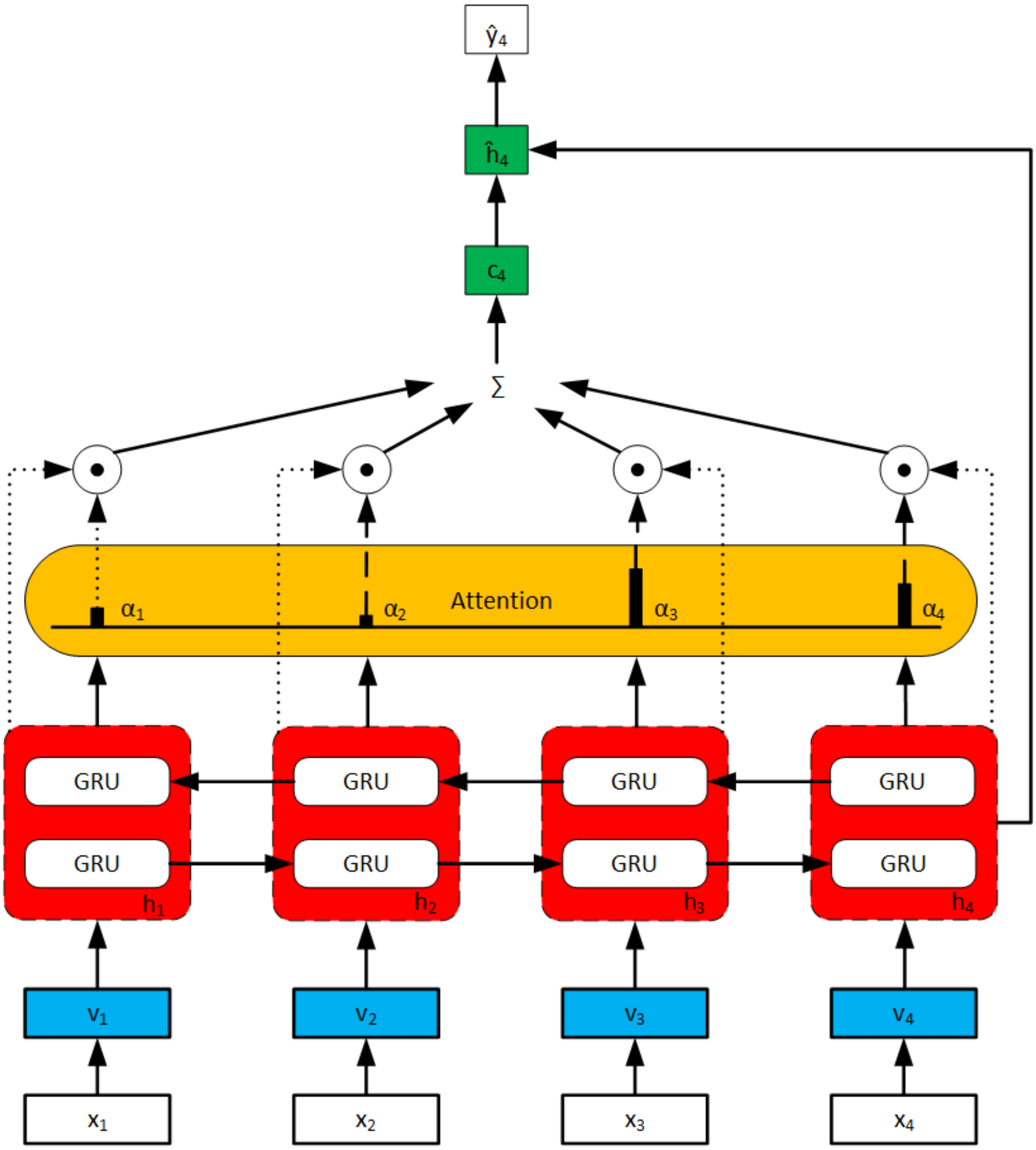


Figure 8.2: Dipole Architecture. Unrolled overview of the Dipole model with an input sequence of 4 visits (x_1, x_2, x_3, x_4). Each visit x_i is embedded into an efficient representation v_i. Then each embedded visit is transformed by a BRNN into hidden state h_i. The visit-level attention mechanism 'pays attention' to the hidden states. Each hidden state h_i is multiplied with its attention value α_i. The sum of the 'attended' hidden states is the context vector c_t. The context vector and the last hidden state are concatenated into $\tilde{h_t}$ which is used to generate prediction y_t.

Trainable parameters

$W_{emb} \in \mathbb{R}^{m \times r}$	Input embedding weights
$b_{emb} \in \mathbb{R}^{m}$	Input embedding bias
$W_{\alpha} \in \mathbb{R}^{2p}$	Visit-level attention weights
$b_{\alpha} \in \mathbb{R}$	Visit-level attention bias
$W_{c} \in \mathbb{R}^{c \times 4p}$	Context weights
$W_{y} \in \mathbb{R}^{c}$	Output weights
$b_{y} \in \mathbb{R}$	Output bias

r: input dimensions
m: embedding dimensions
p: visit-level RNN dimensions
c: context vector dimensions

$$\begin{aligned}
v_i &= \text{ReLU}(W_{emb}x_i + b_{emb}), \quad \text{for } i = 1, ..., t \\
h_1, ..., h_t &= [\overrightarrow{GRU}(v_1, ..., v_t); \overleftarrow{GRU}(v_t, ..., v_1)], \\
g_i &= W_{\alpha}^{\top} h_i + b_{\alpha}, \quad \text{for } i = 1, ..., t \\
\alpha_1, ..., \alpha_t &= \text{Softmax}(g_1, ..., g_t), \\
c_t &= \sum_{i}^{t} \alpha_i h_i, \\
\tilde{h}_t &= \tanh(W_c[c_t; h_t]), \\
\hat{y}_t &= \text{sigmoid}(W_y \tilde{h}_t, +b_y),
\end{aligned}$$

Table 8.1: Formal description and trainable parameters of Dipole. *GRU* refers to the equations described in Table 7.1. The definitions of ReLU, Softmax, tanh and sigmoid are given in Appendix E

Trainable parameters

$W_{emb} \in \mathbb{R}^{m \times r}$	Input embedding weights
$b_{emb} \in \mathbb{R}^{m}$	Input embedding bias
$W_{\beta} \in \mathbb{R}^{2p \times 2p}$	Event-level attention weights
$u_{\beta} \in \mathbb{R}^{2p}$	Event-level context weights
$b_{\beta} \in \mathbb{R}^{2p}$	Event-level bias
$W_{\alpha} \in \mathbb{R}^{2q \times 2q}$	Visit-level attention weights
$u_{\alpha} \in \mathbb{R}^{2q}$	Visit-level context weights
$b_{\alpha} \in \mathbb{R}^{2q}$	Visit-level bias
$W_{y} \in \mathbb{R}^{2q}$	Output weights
$b_{y} \in \mathbb{R}$	Output bias

r: input dimensions
m: embedding dimensions
p: event-level RNN dimensions
q: visit-level RNN dimensions

$$\begin{aligned}
w_{ij} &= \text{ReLU}(W_{emb}x_{ij} + b_{emb}), \\
h_{1j}, ..., h_{tj} &= [\overrightarrow{GRU}(w_{1j}, ..., w_{tj}); \overleftarrow{GRU}(w_{tj}, ..., w_{1j})], \\
f_{ij} &= \tanh(W_{\beta}^{\top} h_{ij} + b_{\beta})^{\top} u_{\beta}, \\
\beta_{1j}, ..., \beta_{tj} &= \text{Softmax}(f_{1j}, ..., f_{tj}), \\
v_i &= \sum_{j}^{k} \beta_{ij} h_{ij}, \\
H_1, ..., H_t &= [\overrightarrow{GRU}(v_1, ..., v_t); \overleftarrow{GRU}(v_t, ..., v_1)], \\
g_i &= \tanh(W_{\alpha}^{\top} H_i + b_{\alpha})^{\top} u_{\alpha}, \\
\alpha_1, ..., \alpha_t &= \text{Softmax}(g_1, ..., g_t), \\
c_t &= \sum_{i}^{t} \alpha_i H_i, \\
\hat{y}_t &= \text{sigmoid}(W_y c_t, +b_y),
\end{aligned}$$

Table 8.2: Formal description and trainable parameters of GRNN-HA. The definitions of ReLU, Softmax, and sigmoid are given in Appendix E

MLP to learn the embedding matrix (just like Dipole). The GRNN-HA model is formally described in Table 8.2. One thing to note is that the GRNN-HA model also takes into account the temporal dimension within a visit. For the GRNN-HA model the visit x_i is a sequence of vectors $x_{i1}, ..., x_{ik}$ with $x_{ij} \in \{0, 1\}^r$ and only one dimension in the in the vector having the value 1.

GRNN-HA is quite similar to Dipole in that it uses a BRNN with an attention mechanism. However, GRNN-HA has a hierarchical structure. Which means it first uses a BRNN with attention to encode every medical event within a visit resulting in a visit encoding. Then, GRNN-HA uses another BRNN with attention on every encoded visit, resulting in an encoded sequence (i.e. the context vector). GRNN-HA improves upon the shortcoming of Dipole by implementing an attention generating mechanism on the event-level as well as the visit-level

On the visit-level, the GRNN-HA model is quite similar to Dipole and is therefore (non-)decomposable to the same degree. However, GRNN-HA adds the BRNN and attention mechanism on the event-level. This makes it so that the β_{ij} values can be used to explain the importance of hidden state h_{ij}. Just like with Dipole, one is not justified in relating β_{ij} strictly to the medical event embedding v_{ij} because h_{ij} is influenced by the next and previous hidden states in the sequence. However, one is justified in claiming that a hidden state h_{ij} is more important than h_{ik} if $\beta_{ij} > \beta_{ik}$. In addition, one can also make comparisons between hidden states of different visits (e.g. h_{12} and h_{32}) by weighting the β attention

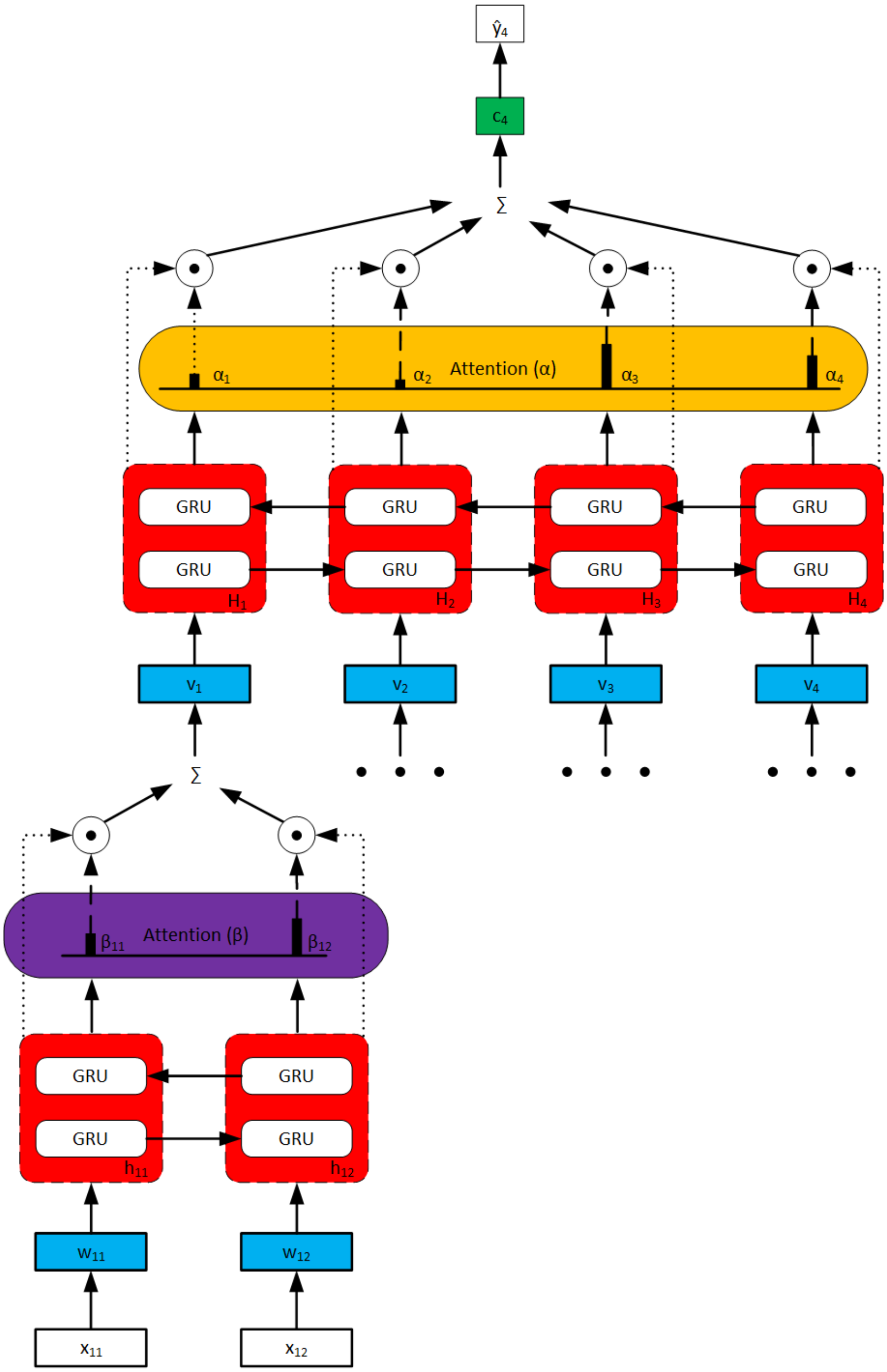


Figure 8.3: GRNN-HA Architecture. Unrolled overview of the GRNN-HA model with an input sequence of 4 visits (x_1, x_2, x_3, x_4) with x_1 consisting of two medical events x_{11} & x_{12}. The medical events observed during the other visits are not shown in the visualisation but are encoded just like x_{11} & x_{12}. Each medical event x_{ij} is embedded into an efficient representation w_{ij}. Then, each embedded medical event w_{ij} is transformed by a BRNN into hidden state h_{ij}. The event-level attention mechanism 'pays attention' to the hidden states. Each hidden state h_{ij} is multiplied with its attention value β_{ij}. The sum of the 'attended' hidden states on the event-level is the visit representation v_i. Each visit representation v_i is transformed by a BRNN into hidden state H_i. The visit-level attention mechanism values α_i are multiplied with H_i and summed over to obtain the context vector c_t which is used to generate prediction $\hat{y}_t$.

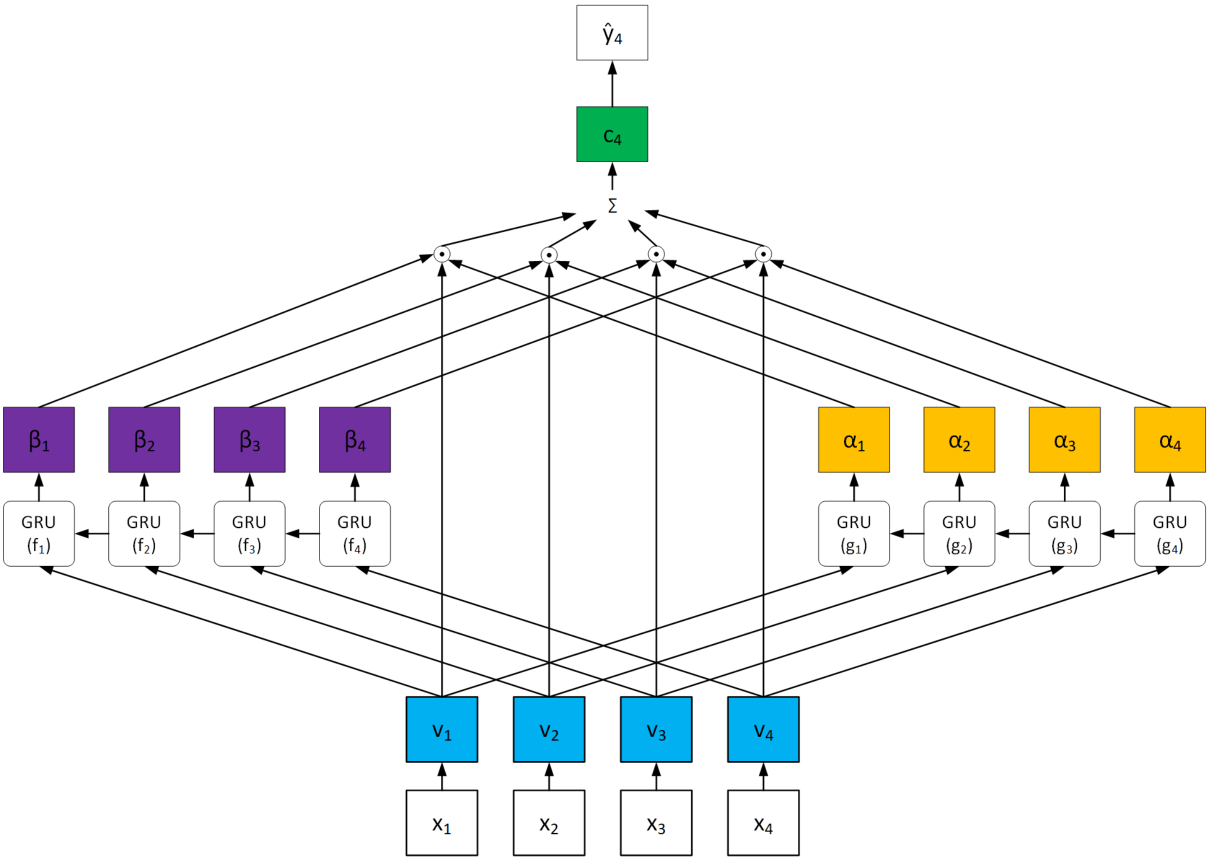


Figure 8.4: RETAIN Architecture. Unrolled overview of the RETAIN model with an input sequence of 4 visits (x_1, x_2, x_3, x_4). Each visit x_i is tranformed into visit representation v_i. Then, the visit representation v_i is input for the Reverse-Time RNNs f and g. The hidden state f_i is used to generate the event-level attention β_i. The hidden state g_i is used to generate the visit-level attention α_i. The attention vectors α_i and β_i are combined with the visit representation v_i into the context vector c_t which is used to generate the prediction $\hat{y}_t$.

values by their respective α attention weight.

8.3 Reverse Time Attention model (RETAIN)

Figure 8.1 shows high-level overview of the RETAIN model architecture as proposed by Choi et al. (2016c). Table 8.3 contains the formal description and the trainable parameters of RETAIN. Unlike Dipole and GRNN-HA, it does not use a BRNN structure but processes input only in reverse-time order so as to mimic a physicians' practice of looking at medical events from present to past.

Another difference between RETAIN and Dipole/GRNN-HA is that the visit embeddings are not transformed by recurrent hidden layer before attention is applied. Instead, two RNNs (RNNα and RNNβ) use the visit embeddings to learn the α and β attention vectors, which is then directly applied to the visit embeddings.

The α attention mechanism of RETAIN allows the model to explain how much attention each timestep in the sequence of embedded visits $v_1, ..., v_t$ receives. In addition, the β attention mechanism allows the model to explain how much attention each dimension in the embedded visit receives. Because the visit embeddings are combined with the attention without being transformed by a recurrent layer, one is justified in relating the α_i and β_i values to the v_i visit embedding.

The benefit of not transforming the input with an RNN is that the single contribution of each medical event can be calculated. Equation 8.1 shows how to calculate this contribution ω to the predicted risk

Trainable parameters

$W_{emb} \in \mathbb{R}^{m \times r}$	Input embedding weights
$W_{\beta} \in \mathbb{R}^{m \times q}$	Event-level attention weights
$b_{\beta} \in \mathbb{R}^{m}$	Event-level attention bias
$W_{\alpha} \in \mathbb{R}^{p}$	Visit-level attention weights
$b_{\alpha} \in \mathbb{R}$	Visit-level attention bias
$W_{y} \in \mathbb{R}^{m}$	Output weights
$b_{y} \in \mathbb{R}$	Output bias

r: input dimensions
m: embedding dimensions
p: RNN$_{\alpha}$ dimensions
q: RNN$_{\beta}$ dimensions

$$\begin{aligned}
v_i &= W_{emb} x_i, \\
g_t, ..., g_1 &= \overleftarrow{GRU}(v_t, ..., v_1), \\
e_i &= W_{\alpha}^{\top} g_i + b_{\alpha}, \\
\alpha_1, ..., \alpha_t &= \text{Softmax}(e_1, ..., e_t), \\
f_t, ..., f_1 &= \overleftarrow{GRU}(v_t, ..., v_1), \\
\beta_i &= \tanh(W_{\beta} f_i + b_{\beta}), \\
c_t &= \sum_{i}^{t} \alpha_i \beta_i \odot v_i, \\
\hat{y}_t &= \text{sigmoid}(W_y c_t + b_y)
\end{aligned}$$

Table 8.3: Formal description and trainable parameters of RETAIN. The definitions of Softmax, tanh and sigmoid are given in Appendix E

score $\hat{y}_t$ of each medical event x_{ij}. α_i represents the attention on visit embedding v_i as a whole. β_i contains the attention on every dimension of the embedded visit v_i. W_y and W_{emb} represent the output and embedding weight matrices respectively. From Equation 8.2 it follows that a negative ω value decreases the risk score, while a positive value increases the risk score. The contribution values can be explained quite intuitively because they can be used to simulate the prediction as is done in Equation 8.2. In order to go from the contribution values to the prediction, one needs to sum up all the contributions, add the bias term of the output layer b_y and use the sigmoid function (defined in Appendix E).

$$\omega(\hat{y}, x_{ij}) = \alpha_i W_y (\beta_i \odot W_{emb[:,j]}) \tag{8.1}$$

$$\hat{y}(x_1, ..., x_t) = \text{sigmoid}\left(\left(\sum_{i}^{t} \omega(\hat{y}, x_{ij}) \right) + b_y \right) \tag{8.2}$$

8.4 The Interpretability Continuum Revisited

Having discussed the interpretability taxonomy (in Section 6.1) and described three RNN-based models that claim to be interpretable, we will now revisit the interpretability continuum. We will use the interpretability taxonomy to justify the placement of the RNN-based models, as well as the MLP and LR models, on the continuum.

LR is placed the most towards the right side of the continuum because it more simulatable and decomposable than the MLP and RNN-based models. Then, the three 'interpretable' models described in this chapter are placed more towards the right than the regular RNN-based models because they allow the post-hoc visualization of attention weights.

Because MLP and the RNN-based models use the same type of training algorithm (i.e. some variation on gradient descent) they will not score differently on algorithmic transparency. With regards to the simulatability property of the models we previously alluded to its relationship with hyper-parameters of the model (e.g. number of nodes in a hidden layer). This means that any of the models can be made more simulatable than the others by making the number of nodes very small. In addition, any RNN that is tuned for predictive performance is very likely to have so many nodes that it will take an unreasonable amount of time for a human to simulate anyway. Finally, although the 'interpretable' RNN-based models all allow for a post-hoc visualization of attention weights, the attention mechanisms are implemented differently and are therefore also decomposable to a different degree.

For the reasons listed above, we use the **decomposability** criteria for the relative ordering of the 'interpretable' RNN-based models on the interpretability continuum.

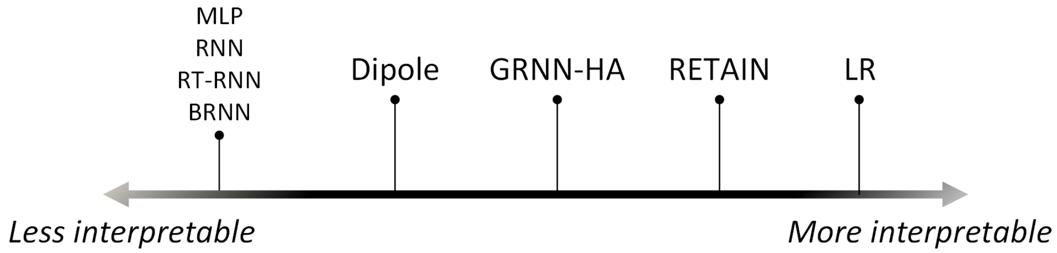


Figure 8.5: Model Interpretability Continuum Revisited. The placement of the models is based on the **decomposability** criteria. The implementation of the neural attention mechanism improves the decompsability of the models described in Chapter 8. Of these models, RETAIN is the most interpretable, followed by GRNN-HA and then Dipole.

8.4.1 Decomposability

With the healthcare domain and the proposed RNN-based models as our context, we assume that at least the input (a sequence of hospital visits with recorded events as shown in Figure 4.1) and output (risk of re-hospitalisation within 30 days) are terms that are understandable to the user (i.e. medical professional). Thus, if a model can provide an explanation these terms, we consider the explanation to be intuitive. To summarize, we operationalize the decomposability criteria in the following way:

> The **decomposability** of a model is linked to its ability to provide an explanation that relates hospital visits and medical events to the predicted re-hospitalisation risk.

The main improvement in decomposability of the 'interpretable' RNN-based models is in the implementation of the attention mechanism (Section 7.4). This makes the models more decomposable than the 'regular' and bi-directional RNNs because it offers an explanation as to what the model focuses on. However, the three models implement the attention mechanism differently which results in explanations that are 'intuitive' to a different degree.

Looking at the attention mechanisms and the explanations they allow, GRNN-HA and RETAIN are able to provide a more intuitive explanation than Dipole. This is because Dipole does not provide attention on the event-level. Dipole is able to explain which part of the sequence of hidden states receive the most attention, but it is not able to intuitively relate the events within a visit to the output. GRNN-HA and RETAIN are able to provide an explanation in terms of the input that is more specific, namely in terms of the events that occurred within a visit.

Between RETAIN and GRNN-HA, RETAIN allows for a more intuitive explanation and is therefore the most intuitive of the three models. This is because the attention mechanisms of RETAIN relates its attention directly to the (embedded) inputs while the GRNN-HA attention mechanisms relate their attention to a timestep of a hidden state sequence. The result of this is that the RETAIN model is able to calculate the exact contribution (positive or negative) to the risk score as per Equation 8.1. On the other hand, GRNN-HA is only able to use the attention weights to explain which elements of the input are important to making the prediction, but it cannot quantify the exact contribution.

Part IV

Results

Chapter 9

Experiments

In this chapter we describe the experimental setting in which we compared the ‘interpretable’ RNN-based models with the RNN baselines and two models that use aggregate features (LR & MLP). We continue with an evaluation of the predictive performance. Finally we showcase the interpretation the attention-based RNN models allow.

9.1 Experimental Setting

Source of data
Data was extracted from the Elecotronic Health Records system at the UMC Utrecht. Admission and DBC data was obtained along with the time-stamp of occurrence. Table 9.1 contains some descriptive statistics of the dataset the was used. Selected patients satisfy the following criteria:

- At least one hospital admission at the cardiology specialism between 2007 and 2017.
- Heart Failure (or synonyms) mentioned in a clinical letter. See Appendix G for a list of the synonyms we used.
- At least one DCB (Diagnosis-Treatment-Code) related to cardiovascular disease

Admission events are represented by the concatenation of the admission specialism, type of admission and origin of admission. Table 9.2 shows the different type and origin codes and what they represent. DBC events are simply represented by the description of the DBC.

Implementation details
We implemented all models using Python 3.6 and Keras 2.1.2 (Chollet et al., 2015) (using the TensorFlow 1.2.1 backend (Abadi et al., 2015)). For training the models we used the Adam optimization algorithm (Kingma and Ba, 2014) with a batch size of 512. The training was done on the High Performance Computing cluster at the UMC Utrecht.

Baselines
As a comparison with the ‘interpretable’ RNN-based models, we implemented several baselines shown in Figure 9.1. In order to create features for the LR and MLP models (that do not take a sequence of feature vectors as input), we used counts of the number of occurrences of each medical event. Then

Descriptive Statistics

# patients	4,930	# sequences (discharges)	37,287
# visits	531,624	Avg # visits per sequence	14.257
# events	1,167,152	Avg # events in a visit	2.195
# event types	589	Next admission < 30 days	% 31

Table 9.1: Descriptive statistics of the UMCU EHR dataset.

Specialism	Type	Origin
CAR = Cardiology	D = Same day therapy	P = via outpatient clinic
ONC = Oncology	K = Inpatient	W = via waiting list
URO = Urology	B = Outpatient	A = via another Hospital
HEA = Hematology	I = Intensive Care AZU	S = via emergency ward
CHI = Surgery	L = Long lasting observation	H = via home
CTC = Cardiothoracic Surgery	O = pre-operative screening	E = via elsewhere
KNO = Throat-, nose- en earsurgery	S = IC admission from emergency	I = via nursing home
GER = Geriatrics	Y = Psychiatry GGZ	
NEU = Neurology	C = Cytostatica AZU	
....		

Table 9.2: Explanation of admission event codes. An example admission event code is CAR (specialism) K (type) S (origin). This means that the person was admitted at the cardiology specialism, it was a clinical admission and the person came in as an emergency case.

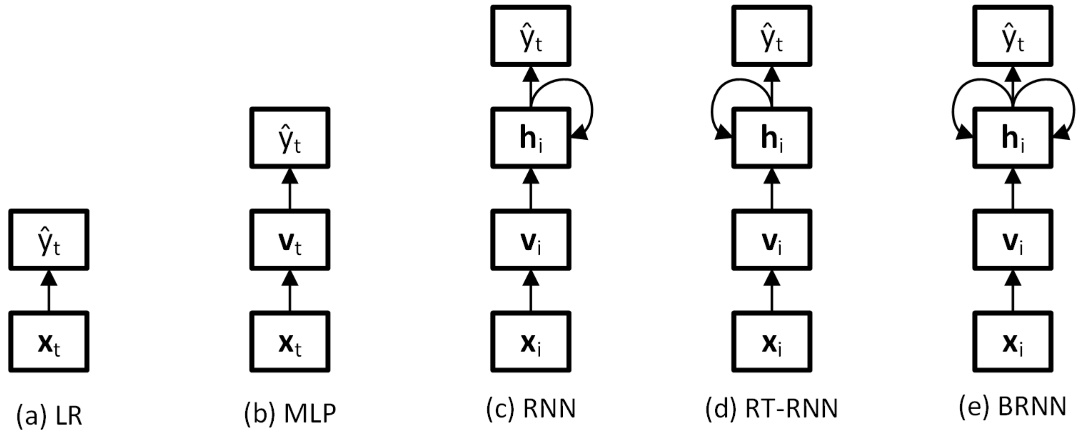


Figure 9.1: Model Baselines. (a) Logistic Regression, (b) Multilayer Perceptron, (c) Recurrent Neural Network, (d) Reverse-Time Recurrent Neural Network & (e) Bi-directional Recurrent Neural Network.

we normalized the resulting single feature vector to have mean 0 and unit variance (i.e. 1). For the RNN-based models we used the same data preparation as for the 'interpretable' RNN-based models.

Objective

Given a sequence of visits $x_1, x_2, ..., x_t$ to the hospital and a date of discharge, the objective is to predict the probability of a re-hospitalisation within 30 days. We use the ground truth labels of $y_i \in \{0, 1\}$ for x_{t+1}. Where a 1 represents the case of the patient being re-hospitalised within 30 days of the discharge (and 0 otherwise). We used the binary cross-entropy function 9.1 as the loss function to minimize during training.

$$L_i(p_i, y_i) = -(y_i \log(p_i) + (1 - y_i) \log(1 - p_i)) \tag{9.1}$$

Cohort Construction

For every selected patient we generated the training observations by looking at each discharge and taking the medical history of the patient up until that point. We labelled the observation with a 1 if the next admission date was fewer than 31 days after the discharge and 0 otherwise (see Figure 9.2). We used a maximum number of 10 visits in the past. 31% of the observations were labelled 1. We disregarded the last discharge in all patients because the labels for these sequences are unknown.

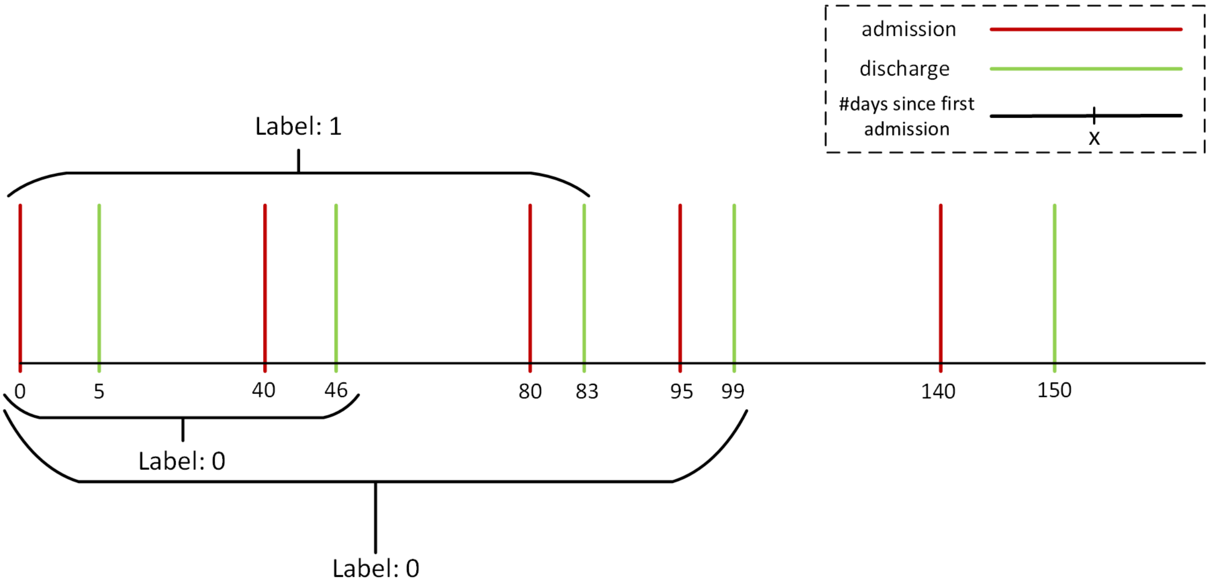


Figure 9.2: Medical history label assignment. The assignment of labels to medical history sequences. If the next admission was within 30 days of discharge, the sequence is labelled 1 and 0 otherwise.

Training Details
We randomly divided the data into train/validation/test sets (0.7/0.15/0.15). In order to try and prevent overfitting, we used (recurrent) dropout, regularization and early stopping. These strategies all require the tuning of several hyper-parameters. Also, the numbers of nodes in each layer were hyper-parameters that needed to be optimized.

The hyper-parameter optimization was done using the training data while evaluating the Area Under the ROC Curve (AUC) on the validation data. Appendix A shows (for each model) the parameters that were tuned, as well as the setting that was used during the experiments. Early stopping with a patience of 5 epochs was used during the tuning. After tuning the parameters, we used 8-fold cross-validation (on the training + validation data) to record the average number of epochs the model needed to converge. Then, the average number of epochs was used for training the final model. For this we used train + validation data to train the model and recorded its final AUC score on the test set as an estimate of the performance of the model. In addition to the AUC, we also recorded the Area Under the Precision Recall Curve (AUPRC).

9.2 Evaluation of Predictive Performance

Table 9.3 and Figure 9.3 show that the RNN-based models clearly outperform the LR and MLP baselines. It also shows the interpretability score of the models based on the nominal ordering along the interpretability continuum as described in Section 8.4. Out of the ‘interpretable’ RNN-based models, GRNN-HA had the highest predictive performance as measured by the AUC and AUPRC metrics. RETAIN performed slightly worse than the other RNN-based models but still a lot better than LR and MLP. Out of the RNN-based baselines, RT-RNN performed the best with regards to the AUC metric. It was also the best performing model overall, outperforming the ‘interpretable’ models.

Considering the interpretability / predictive performance trade-off, we would expect to see the non-interpretable baseline models to outperform the ‘interpretable’ models. We can partly observe the trade-off when looking at the results. RETAIN, offering the best interpretation of the RNN-based models, underperformed compared to the others. Also, a non-interpretable model (RT-RNN) had the highest predictive performance as measured by the AUC metric. However, it is also true that GRNN-HA (which is interpretable to some degree) outperformed the non-interpretable models RNN and BRNN. In addition, while Dipole is less interpretable than GRNN-HA, it was still outperformed by GRNN-HA. It could be

Model	AUC	AUPRC	Interpretability Score	Epochs	# Trainable Parameters
MLP	0.728110	0.588377	0	7	151,297
RNN	0.768283	0.674035	0	5	893,697
RT-RNN	**0.774042**	0.688009	0	8	1,878,529
BRNN	0.768664	0.679695	0	8	3,454,977
Dipole	0.770069	0.685232	1	9	1,616,514
GRNN-HA	0.773293	**0.689235**	2	11	875,521
RETAIN	0.766619	0.679100	3	13	1,613,442
LR	0.691240	0.552255	4	23	590

Table 9.3: Experiment Results. RT-RNN performed the best with regards to the AUC measure. GRNN-HA ranked first according to the AUPRC measure. The interpretability score is based on the ordinal ranking on the interpretability continuum as described in Section 8.4.

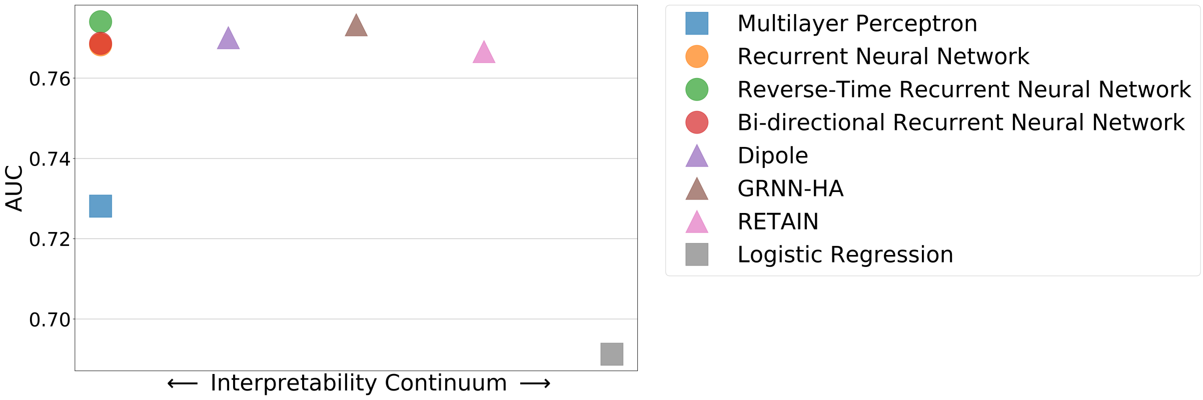


Figure 9.3: Area Under the Curve (AUC) versus placement on the interpretability continuum. The models are ordered along the interpretability continuum based on the decomposability criteria as described in Section 8.4. The models denoted with by a □ use aggregate features, the models denoted by a ○ are the RNN baselines and the the models denoted by a △ are the 'interpretable' RNN-based models.

that modelling the temporal dimension within a visit (as only GRNN-HA does) is the reason for the model to perform slightly better.

One might be tempted to compare the performance of the LR and MLP models with the RNN-based models and draw an inference about the interpretability /predictive performance trade-off. However, we feel that this is not warranted given that LR and MLP use aggregate features while the RNN-based models use the medical history as a temporal sequence. Nonetheless, a comparison between LR and MLP is warranted and the trade-off is definitely noticeable in this comparison.

Overall, the main takeaway is that the difference in performance between the RNN-based models is quite small. On the other hand, the difference in interpretability between the baselines and 'interpretable' models is substantial (outlined in Section 8.4). Therefore we would argue that the 'interpretable' RNN-based models are a better option to use, especially in an application domain where interpretability is paramount (i.e. healthcare).

9.3 Model Interpretation

9.3.1 Local interpretation

In this section we take one medical history from our experiment as an example to showcase how the attention weights of the three models can facilitate local interpretation of the 30-day re-hospitalisation

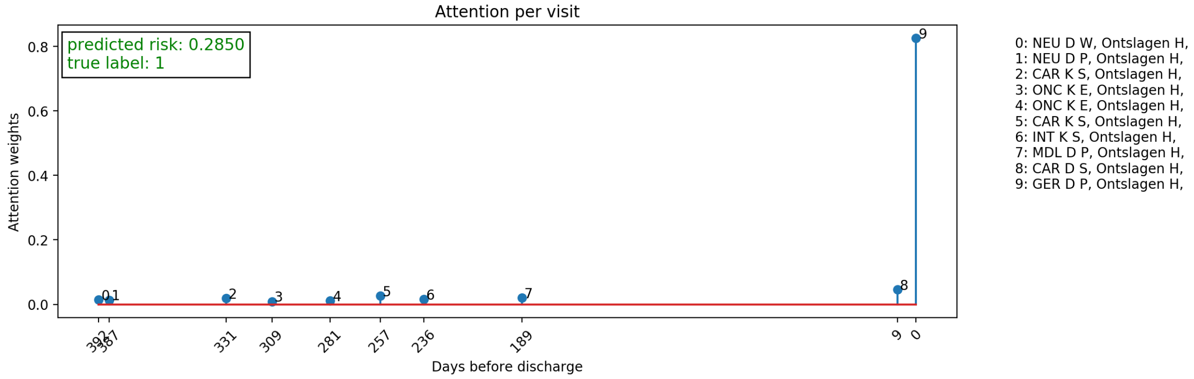


Figure 9.4: Dipole Attention. Visualisation of encoded visit attention coefficient.

risk score of a single patient.

Dipole

In Figure 9.4 we can see the relative attention paid to a hospital visit (that has been transformed by a BRNN). The attention weight corresponds to the α values as as calculated in Table 8.1. If $\alpha_i > \alpha_j$, we can conclude that the hidden state h_i is more important than h_j in calculating the risk score. As a clear example, the attention on the 10th visit (with index 9) receives more way more attention than the other visits. From this we can see that Dipole uses mainly the information in the last visit to the hospital to predict the re-hospitalisation risk score.

GRNN-HA

In Figure 9.5 we see the β (event-level) attention weights weighted by their respective α (visit-level) attention weights. The calculation of these weights is shown in Table 8.2. With this visualisation we can compare the importance of the hidden states of the model – both on the event-level and on the visit-level – to making the prediction of the 30-day re-hospitalisation risk. We can see that the final visit receives the most attention, but we can also see relative attention paid to single events. If $(\alpha_i * \beta_{ij}) > (\alpha_h * \beta_{hl})$ we conclude that that event j is more important that event l in predicting the re-hospitalisation risk score. As an example we can see that event 18 (GER D P) is more important than event 4 (CAR K S) in the calculation of the re-hospitalisation risk of this patient.

RETAIN

From Figure 9.6 we can easily see the medical events that increase or decrease the re-hospitalisation risk score. The contributions are calculated by Equation 8.1. Given the contribution of every medical event in the visit sequence, we can calculate the predicted risk using 8.2. From the figure we see that the events 6 (ONC K E) and 8 (ONC K E) greatly increase the re-hospitalisation risk score. It is interesting to that the three models differ in their assessment of risk score of this patient, and that they focus on different parts of the input sequence. Where Dipole almost only focuses on the last hospital visit, GRNN-HA has a more evenly distributed attention but is still most focused on the last visits. RETAIN (in this example) focuses on events further in the past.

9.3.2 Global interpretation

Here we look at GRNN-HA and RETAIN and look at the events that receive the most (and least) attention averaged over the whole dataset. In the case of GRNN-HA we show the average β attention weight corresponding to the hidden state of a medical event in Table 9.4. For RETAIN we show the average contribution to the predicted risk score in Table 9.5. In both cases we leave the α (visit-level) weights out of consideration. Because of the lack of an event-level attention mechanism, we are not able to perform the same kind of interpretation with Dipole.

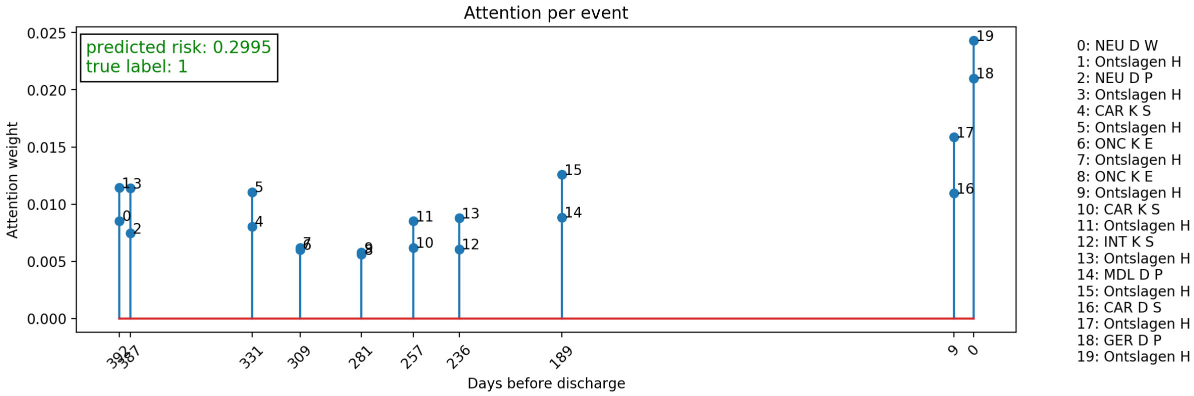


Figure 9.5: GRNN-HA Attention. Visualisation of encoded event attention coefficient.

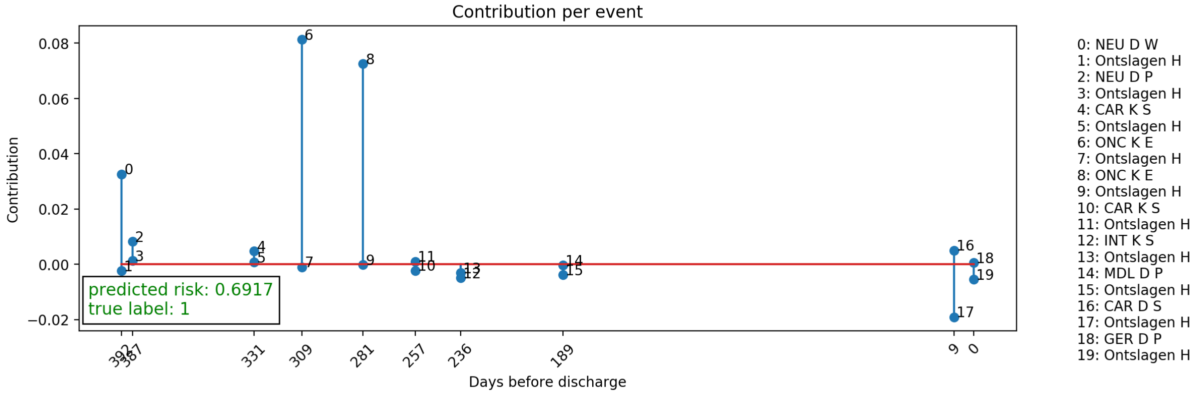


Figure 9.6: RETAIN Attention. Visualisation of event contribution to predicted risk score.

Event	**Avg. attention**	**# occurrences**
PSY D W	0.1259	1,427
Ontslagen H	0.1180	243,825
Pericarditis	0.1138	135
KNO O W	0.1130	404
CAR O W	0.1114	783
....	....	....
CAR D W	0.0640	15,360
CAR D P	0.0670	17,166
ONC K S	0.0678	242
CAR B W	0.0682	424
HAE B W	0.0694	154

Table 9.4: GRNN top- and bottom 5 medical events. The events are ranked by average attention weight (events with more than 100 occurrences)

Event	Avg. contribution	# occurrences
KHA K W	0.4314	128
PSY D W	0.3881	1427
Pericarditis	0.3822	135
CTC I S	0.3597	112
Screening harttransplantatie	0.3430	723
....	....	....
OOG K S	-0.7913	119
PLA K H	-0.7297	101
VAS D H	-0.6839	117
ORT D P	-0.6414	376
ORT K H	-0.5735	311

Table 9.5: RETAIN top- and bottom 5 medical events. The events are ranked by average attention weight (events with more than 100 occurrences)

Chapter 10

Discussion

Interpretable machine learning models are important in a context where the aim is to achieve a goal that is difficult to define formally or when multiple goals trade-off. The healthcare domain is such a context. There are traditional machine learning models (e.g. Logistic Regression) that are interpretable but suffer from the fact that they use aggregate features and as a consequence ignore the temporal relationship between features. The Recurrent Neural Network (RNN) is able to model the temporal dimension but offers limited interpretability. Here we return to the main research question and the sub-questions that were posed in Chapter 2. We conclude the document with a consideration of the limitations of this study and some pointers for future research.

10.1 Sub-questions

1. **What constitutes an interpretable model?**

   An interpretable model has (at least some of) the properties described in Section 6.1. The model should be **transparent** on multiple levels. These levels are the whole model (**simulatability**), its single components (**decomposability**) and its training algorithm (**algorithmic transparency**).

   Simulatability has to do with the ability of a human to simulate the calculations of the model in a reasonable amount of time. Decomposability has to do with single components of the model allowing for an intuitive explanation. An explanation is intuitive if it is given in terms that are understandable to the user. In the context of the healthcare domain with a medical professional as the user, we have assumed that the medical history – consisting of hospital visits and observed medical events of a patient (visualised in Figure 4.1 and formalized in Section 4.1.1) – are terms that are understandable. Algorithmic transparency has to do with the mathematical guarantees that the training algorithm can provide with regards to its solution.

   The interpretability is also tied to its **post-hoc interpretability**. These are explanations that do not necessarily reflect the true inner workings of the model but give the user some useful explanation in the form of a **textual explanation**, **visualization**, **local explanation** or an **explanation by example**.

2. **How can the regular RNN model be adapted in order to allow for better predictive performance and interpretation?**

   Chapter 7 describes several extensions to the regular RNN model that aim to improve its predictive performance or interpretability. The **advanced cell architectures** (Section 7.1) are better able to capture long-range dependencies in a sequence. **Representation learning** techniques (Section 7.2) transform the 'one-hot' encoding of medical events into a more efficient representation that captures semantic relatedness between medical events. **Bi-directional RNN** (Section 7.3) model the input sequence in both time and reverse-time directions. This allows the model to take into account events from the past as well as the future when deciding what information to keep.

   In addition, Section 7.4 describes the **neural attention mechanism** that aims to improve predictive performance by allowing the model to look at the hidden state of every timestep and 'pay attention' to the parts that are most important for making a correct prediction. Furthermore, the attention mechanism can be used to provide an explanation as to what the model focusses on when making a prediction. This improves the decomposability – and therefore the interpretability – of the RNN model.

The extensions described in Chapter 7 are implemented in the **Dipole** (Section 8.1), **GRNN-HA** (Section 8.2) and **RETAIN** (Section 8.3) model architectures. All three models use representation learning and the GRU cell architecture. Dipole and GRNN-HA also use a BRNN architecture while RETAIN uses a reverse-time RNN. The implementation of the attention mechanism is different in the three models. Dipole implements a visit-level attention mechanism on the hidden layer of the network. GRNN-HA implements event-level attention on the hidden layer to create a visit representation, then implements visit-level attention on the visit representations. RETAIN implements event- and visit-level attention on the input layer of the network.

3. **How do the 'interpretable' RNN-based models compare against each other, the traditional approaches and the regular RNN model with regards to predictive performance and interpretability?**

Predictive Performance
Section 9.2 shows the results of a case study on a cohort of HF patients at the UMC Utrecht. In the case study, the RNN-based models solidly outperform the traditional approaches (i.e. logistic regression and multilayer perceptron). Also, the RNN baselines (RNN, RT-RNN & BRNN) show similar performance to the 'interpretable' RNN-based models. Within the 'interpretable' RNN-based models, GRNN-HA slightly outperforms the other two.

Interpretability
Section 8.4 describes how the 'interpretable' RNN-based models compare with each other and the RNN baselines on the interpretability continuum. The models differ in their **decomposability**, therefore we used this criteria as the justification for the placement of the models on the continuum. Section 9.3 shows the explanations the 'interpretable' RNN-based models can offer to a human user.

Out of the RNN-based models, RETAIN is the most interpretable because of the visit- and event-level attention that can be used to calculate the contribution of each medical event to the predicted risk score. GRNN-HA is second because of the visit- and event-level attention that can be used to indicate the relative importance of a medical event to the predicted risk score (but not the exact contribution). Dipole is ranked third because of the visit-level attention that can be used to indicate the relative importance of a hospital visits to the calculation of the risk score.

In comparison with the traditional approaches that use aggregate features we feel that RETAIN comes close to LR with regards to decomposability. However, LR is still more interpretable because of the simulatability criteria. In comparison with MLP we feel that the 'interpretable' models score about the same on the simulatability criteria. However, they are better decomposable because of the explanation they allow.

Interpretability / Predictive performance trade-off
Looking at the placement of the models on the interpretability continuum (Section 8.4) and the results of the case study (Sectionr̃efsec:results), we can observe a slight trade-off between interpretability and predictive performance. The most interpretable model (RETAIN) is outperformed by all the other RNN-based models. In addition, a non-interpretable model (RT-RNN) had the highest predictive performance (measured by AUC). However, GRNN-HA outperformed Dipole and is also more interpretable.

In comparing LR with MLP, the trade-off between interpretability and predictive performance is more clear. LR is more interpretable than MLP but performs worse. We feel that comparing LR/MLP to the RNN-based models and attributing the difference to the interpretability / predictive performance trade-off is unwarranted because the LR/MLP models use aggregate features (i.e. ignore the temporal dimension).

10.2 Main Research Question

In our work we review and compare three RNN-based models – RETAIN, Dipole and GRNN-HA – that make use of the neural attention mechanism to improve interpretability. The implementations of the neural attention mechanism differ, resulting in models that are interpretable to a different degree. The review of these models aimed to answer the main research question that was formulated as follows:

> *Can the Recurrent Neural Network model be improved such that it can provide an accurate and interpretable prediction of (re-)hospitalisation risk for Heart Failure patients?*

The review and comparison of the RNN-based models shows that the interpretability of the RNN model can be improved. Also, with regards to the predictive performance of models that were reported in literature (Table 3.2) the RNN model shows state-of-the-art predictive performance. However, this work also shows that it matters how the neural attention mechanism is implemented and that not all implementations are equally interpretable. In addition, interpretability and predictive performance seem to slightly trade-off. Although overall, the differences in predictive performance between the models is quite small. The difference in AUC between the best performing (RT-RNN) and worst performing (RETAIN) is only 0.008. On the other hand, in a context where interpretability is important –such as the healthcare domain– interpretable models like RETAIN and GRNN-HA are clearly preferable over the regular RNN model even though they have slightly less predictive performance.

10.3 Limitations

The patients selected for this study fulfil the requirements listed in Section 9.1. However, this does not necessarily mean that all selected patients were actually diagnosed with Heart Failure. It may be the case that some of the patients selected did have some cardiac related disease, but it was somewhere noted in the clinical letters that the patient did **not** have Heart Failure.

Another limitation of this study lies in the comparison of models with many hyperparameters and that their settings may not have been optimal. The hyper-parameter optimization was done using a grid search with pre-defined values for every parameter. However, it is of course possible that are other parameters values that would have resulted in better predictive performance.

Finally, using medical event counts as features is just one way to collapse over the temporal dimension. There may be other feature engineering techniques that could result in different predictive performance for the LR and MLP models. An example of a different feature engineering approach might be to only take the medical events of the last visit into consideration.

10.4 Future Research

Combining the properties of GRNN-HA and RETAIN

During this study we found an interesting problem that warrants future research. This problem has to do with the way that the occurrence of medical events within a visit are encoded by Dipole and RETAIN. Although the Dipole and RETAIN model the temporal dimension **between** visits, they collapse over the temporal dimension **within** visits. Resulting in two problems. The first is that the temporal dimension within a visit may contain useful information and could – if modelled – improve prediction performance. In our experiments, GRNN-HA outperformed Dipole and RETAIN, hinting that modelling the temporal dimension within a visit does indeed improve predictive performance. The second problem is that Dipole and RETAIN are not really able to handle a medical event occurring multiple times within a single visit. One could encode multiple events by counting the number of times the event occurred and using the count as the feature. But what to do when the medical event itself has multiple features consisting of real numbers. An example of this is an ECG test which has more than 20 real-valued features. In the case of an ECG test occurring multiple times in a visit, it is not clear how to encode the information.

GRNN-HA does not have the two problems mentioned above. Therefore we argue that there is a need for a model that combines the properties of GRNN-HA and RETAIN. GRNN-HA allows multiple occurrences of the same event within a visit and also models the temporal dimension of events within a visit. However, RETAIN offers the best interpretation with its contribution values for every event which GRNN-HA does not.

Computational cost

Another difference between GRNN-HA and RETAIN is that training GRNN-HA is computationally more expensive. This is because GRNN-HA models the temporal dimension within a visit. Introducing the factor of computational cost to the trade-off between predictive performance and interpretability might be another interesting avenue for future research.

Quantifying interpretability

The second avenue for future research is in the quantifying of model interpretability. It would be incredibly useful if there was a principled way to measure the interpretability of a model and represent it as a real number. With such a measure, comparing the interpretability of two models would be less dependent upon an argument about the 'intuitiveness' of explanations or the ability for a human to simulate the model in a 'reasonable timespan'. Although I would not argue that these kinds of arguments are vacuous, concepts such as 'intuitiveness' and a 'reasonable timespan' are quite imprecise and leave a lot of room open for debate.

Model validation

Chapter 9.3 shows that in our case study, the three interpretable RNN models seem to pay attention to different elements of the medical history that was used as an example. Also the predicted risk score that the three models calculate differs[1]. This results prompts further research into how these models can focus on different elements but still have similar AUC scores. It may be that some models are better able to handle certain types of medical histories than others.

[1] Dipole: 0.2850, GRNN-HA: 0.2995 & RETAIN: 0.6917

Bibliography

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Amarasingham, R., Moore, B. J., Tabak, Y. P., Drazner, M. H., Clark, C. A., Zhang, S., Reed, W. G., Swanson, T. S., Ma, Y., and Halm, E. A. (2010). An automated model to identify heart failure patients at risk for 30-day readmission or death using electronic medical record data. *Medical care*, 48(11):981–988.

Amato, F., López, A., Peña-Méndez, E. M., Vaňhara, P., Hampl, A., and Havel, J. (2013). Artificial neural networks in medical diagnosis.

Atienza, F., Martinez-Alzamora, N., De Velasco, J. A., Dreiseitl, S., and Ohno-Machado, L. (2000). Risk stratification in heart failure using artificial neural networks. In *Proceedings of the AMIA Symposium*, page 32. American Medical Informatics Association.

Au, A. G., McAlister, F. A., Bakal, J. A., Ezekowitz, J., Kaul, P., and van Walraven, C. (2012). Predicting the risk of unplanned readmission or death within 30 days of discharge after a heart failure hospitalization. *American heart journal*, 164(3):365–372.

Ba, J., Mnih, V., and Kavukcuoglu, K. (2014). Multiple object recognition with visual attention. *arXiv preprint arXiv:1412.7755*.

Babińska, M., Chudek, J., Chełmecka, E., Janik, M., Klimek, K., and Owczarek, A. (2015). Limitations of cox proportional hazards analysis in mortality prediction of patients with acute coronary syndrome. *Studies in Logic, Grammar and Rhetoric*, 43(1):33–48.

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Bajor, J. M. and Lasko, T. A. (2016). Predicting medications from diagnostic codes with recurrent neural networks.

Basu Roy, S., Teredesai, A., Zolfaghar, K., Liu, R., Hazel, D., Newman, S., and Marinez, A. (2015). Dynamic hierarchical classification for patient risk-of-readmission. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1691–1700. ACM.

Black, A. D., Car, J., Pagliari, C., Anandan, C., Cresswell, K., Bokun, T., McKinstry, B., Procter, R., Majeed, A., and Sheikh, A. (2011). The impact of ehealth on the quality and safety of health care: a systematic overview. *PLoS medicine*, 8(1):e1000387.

Braunwald, E. (2015). The war against heart failure: the lancet lecture. *The Lancet*, 385(9970):812–824.

Caruana, R., Lawrence, S., and Giles, C. L. (2001). Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping. In *Advances in neural information processing systems*, pages 402–408.

Chaudhry, B., Wang, J., Wu, S., Maglione, M., Mojica, W., Roth, E., Morton, S. C., and Shekelle, P. G. (2006). Systematic review: impact of health information technology on quality, efficiency, and costs of medical care. *Annals of internal medicine*, 144(10):742–752.

Chin, M. H. and Goldman, L. (1997). Correlates of early hospital readmission or death in patients with congestive heart failure. *The American journal of cardiology*, 79(12):1640–1644.

Choi, E., Bahadori, M. T., Schuetz, A., Stewart, W. F., and Sun, J. (2016a). Doctor ai: Predicting clinical events via recurrent neural networks. In *Machine Learning for Healthcare Conference*, pages 301–318.

Choi, E., Bahadori, M. T., Searles, E., Coffey, C., Thompson, M., Bost, J., Tejedor-Sojo, J., and Sun, J. (2016b). Multi-layer representation learning for medical concepts. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1495–1504. ACM.

Choi, E., Bahadori, M. T., Sun, J., Kulas, J., Schuetz, A., and Stewart, W. (2016c). Retain: An interpretable predictive model for healthcare using reverse time attention mechanism. In *Advances in Neural Information Processing Systems*, pages 3504–3512.

Choi, E., Schuetz, A., Stewart, W. F., and Sun, J. (2016d). Medical concept representation learning from electronic health records and its application on heart failure prediction. *arXiv preprint arXiv:1602.03686*.

Choi, E., Schuetz, A., Stewart, W. F., and Sun, J. (2016e). Using recurrent neural network models for early detection of heart failure onset. *Journal of the American Medical Informatics Association*, 24(2):361–370.

Chollet, F. et al. (2015). Keras. `https://keras.io`.

Chorowski, J. K., Bahdanau, D., Serdyuk, D., Cho, K., and Bengio, Y. (2015). Attention-based models for speech recognition. In *Advances in neural information processing systems*, pages 577–585.

Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Csáji, B. C. (2001). Approximation with artificial neural networks. *Faculty of Sciences, Etvs Lornd University, Hungary*, 24:48.

Deeplearning4j (2017). A Beginner's Guide to Recurrent Networks and LSTMs. `https://deeplearning4j.org/lstm.html`. Accessed: 2017-10-12.

Doshi-Velez, F. and Kim, B. (2017). Towards a rigorous science of interpretable machine learning.

Esteban, C., Staeck, O., Baier, S., Yang, Y., and Tresp, V. (2016). Predicting clinical events by combining static and dynamic information using recurrent neural networks. In *Healthcare Informatics (ICHI), 2016 IEEE International Conference on*, pages 93–101. IEEE.

Felker, G. M., Leimberger, J. D., Califf, R. M., Cuffe, M. S., Massie, B. M., Adams, K. F., Gheorghiade, M., and O'Connor, C. M. (2004). Risk stratification after hospitalization for decompensated heart failure. *Journal of cardiac failure*, 10(6):460–466.

Friedman, J. H. (1997). On bias, variance, 0/1—loss, and the curse-of-dimensionality. *Data Mining and Knowledge Discovery*, 1(1):55–77.

Gheorghiade, M. and Pang, P. S. (2009). Acute heart failure syndromes. *Journal of the American College of Cardiology*, 53(7):557–573.

Girosi, F., Jones, M., and Poggio, T. (1995). Regularization theory and neural networks architectures. *Neural computation*, 7(2):219–269.

Hochreiter, S. (1998). The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(02):107–116.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Jha, A. K., DesRoches, C. M., Campbell, E. G., Donelan, K., Rao, S. R., Ferris, T. G., Shields, A., Rosenbaum, S., and Blumenthal, D. (2009). Use of electronic health records in us hospitals. *New England Journal of Medicine*, 360(16):1628–1638.

Jones, S. S., Rudin, R. S., Perry, T., and Shekelle, P. G. (2014). Health information technology: an updated systematic review with a focus on meaningful use. *Annals of internal medicine*, 160(1):48–54.

Kang, Y., McHugh, M. D., Chittams, J., and Bowles, K. H. (2016). Utilizing home health care electronic health records for telehomecare patients with heart failure: a decision tree approach to detect associations with rehospitalizations. *Computers, informatics, nursing: CIN*, 34(4):175.

Keenan, P. S., Normand, S.-L. T., Lin, Z., Drye, E. E., Bhat, K. R., Ross, J. S., Schuur, J. D., Stauffer, B. D., Bernheim, S. M., Epstein, A. J., et al. (2008). An administrative claims measure suitable for profiling hospital performance on the basis of 30-day all-cause readmission rates among patients with heart failureclinical perspective. *Circulation: Cardiovascular Quality and Outcomes*, 1(1):29–37.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Koudstaal, S., Pujades-Rodriguez, M., Denaxas, S., Gho, J. M., Shah, A. D., Yu, N., Patel, R. S., Gale, C. P., Hoes, A. W., Cleland, J. G., et al. (2016). Prognostic burden of heart failure recorded in primary care, acute hospital admissions, or both: a population-based linked electronic health record cohort study in 2.1 million people. *European journal of heart failure*.

Koulaouzidis, G., Iakovidis, D., and Clark, A. (2016). Telemonitoring predicts in advance heart failure admissions. *International journal of cardiology*, 216:78–84.

Krumholz, H. M., Chen, Y.-T., Wang, Y., Vaccarino, V., Radford, M. J., and Horwitz, R. I. (2000). Predictors of readmission among elderly survivors of admission with heart failure. *American heart journal*, 139(1):72–77.

Lee, C., Luo, Z., Ngiam, K. Y., Zhang, M., Zheng, K., Chen, G., Ooi, B. C., and Yip, W. L. J. (2017). Big healthcare data analytics: Challenges and applications. In *Handbook of Large-Scale Distributed Computing in Smart Healthcare*, pages 11–41. Springer.

Lipton, Z. C. (2016). The mythos of model interpretability. *arXiv preprint arXiv:1606.03490.*

Lipton, Z. C., Berkowitz, J., and Elkan, C. (2015a). A critical review of recurrent neural networks for sequence learning. *arXiv preprint arXiv:1506.00019.*

Lipton, Z. C., Kale, D. C., Elkan, C., and Wetzell, R. (2015b). Learning to diagnose with lstm recurrent neural networks. *arXiv preprint arXiv:1511.03677.*

Lipton, Z. C., Kale, D. C., and Wetzel, R. (2016). Modeling missing data in clinical time series with rnns. *Machine Learning for Healthcare.*

Ma, F., Chitta, R., Zhou, J., You, Q., Sun, T., and Gao, J. (2017). Dipole: Diagnosis prediction in healthcare via attention-based bidirectional recurrent neural networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1903–1911. ACM.

Mnih, V., Heess, N., Graves, A., et al. (2014). Recurrent models of visual attention. In *Advances in neural information processing systems*, pages 2204–2212.

Novella, S. (2009). Evidence in medicine: Correlation and causation. `https://sciencebasedmedicine.org/evidence-in-medicine-correlation-and-causation/`.

Organization, W. H. (1993). *ICD-10 Classification of Mental and Behavioural Disorders (The): Diagnostic Criteria for Research.* ICD-10 classification of mental and behavioural disorders / World Health Organization. World Health Organization.

Ouwerkerk, W., Voors, A. A., and Zwinderman, A. H. (2014). Factors influencing the predictive power of models for predicting mortality and/or heart failure hospitalization in patients with heart failure. *JACC: Heart Failure*, 2(5):429–436.

Philbin, E. F. and DiSalvo, T. G. (1999). Prediction of hospital readmission for heart failure: development of a simple risk score based on administrative data. *Journal of the American College of Cardiology*, 33(6):1560–1566.

Ponikowski, P., Voors, A. A., Anker, S. D., Bueno, H., Cleland, J. G., Coats, A. J., Falk, V., González-Juanatey, J. R., Harjola, V.-P., Jankowska, E. A., et al. (2016). 2016 esc guidelines for the diagnosis and treatment of acute and chronic heart failure: The task force for the diagnosis and treatment of acute and chronic heart failure of the european society of cardiology (esc) developed with the special contribution of the heart failure association (hfa) of the esc. *European heart journal*, 37(27):2129–2200.

Postmus, D., Veldhuisen, D. J., Jaarsma, T., Luttik, M. L., Lassus, J., Mebazaa, A., Nieminen, M. S., Harjola, V.-P., Lewsey, J., Buskens, E., et al. (2012). The coach risk engine: a multistate model for predicting survival and hospitalization in patients with heart failure. *European journal of heart failure*, 14(2):168–175.

Rahimi, K., Bennett, D., Conrad, N., Williams, T. M., Basu, J., Dwight, J., Woodward, M., Patel, A., McMurray, J., and MacMahon, S. (2014). Risk prediction in patients with heart failure. *JACC: Heart Failure*, 2(5):440–446.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144. ACM.

Ross, J. S., Mulvey, G. K., Stauffer, B., Patlolla, V., Bernheim, S. M., Keenan, P. S., and Krumholz, H. M. (2008). Statistical models and patient predictors of readmission for heart failure: a systematic review. *Archives of internal medicine*, 168(13):1371–1386.

Scheffer, J. (2002). Dealing with missing data.

Schuster, M. and Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

See, A., Liu, P. J., and Manning, C. D. (2017). Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*.

Sha, Y. and Wang, M. D. (2017). Interpretable predictions of clinical outcomes with an attention-based recurrent neural network. In *Proceedings of the 8th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, pages 233–240. ACM.

Shekelle, P., Morton, S. C., and Keeler, E. B. (2006). Costs and benefits of health information technology.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Tripoliti, E. E., Papadopoulos, T. G., Karanasiou, G. S., Naka, K. K., and Fotiadis, D. I. (2017). Heart failure: Diagnosis, severity estimation and prediction of adverse events through machine learning techniques. *Computational and structural biotechnology journal*, 15:26–47.

Tu, J. V. (1996). Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes. *Journal of clinical epidemiology*, 49(11):1225–1231.

Wang, L., Porter, B., Maynard, C., Bryson, C., Sun, H., Lowy, E., McDonell, M., Frisbee, K., Nielson, C., and Fihn, S. D. (2012). Predicting risk of hospitalization or death among patients with heart failure in the veterans health administration. *The American journal of cardiology*, 110(9):1342–1349.

Wang, Y., Ng, K., Byrd, R. J., Hu, J., Ebadollahi, S., Daar, Z., Steinhubl, S. R., Stewart, W. F., et al. (2015). Early detection of heart failure with varying prediction windows by structured and unstructured data in electronic health records. In *Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE*, pages 2530–2533. IEEE.

Watson, A. J., O'Rourke, J., Jethwani, K., Cami, A., Stern, T. A., Kvedar, J. C., Chueh, H. C., and Zai, A. H. (2011). Linking electronic health record-extracted psychosocial data in real-time to risk of readmission for heart failure. *Psychosomatics*, 52(4):319–327.

Writing, G. M., Mozaffarian, D., Benjamin, E., Go, A., Arnett, D., Blaha, M., Cushman, M., Das, S., de Ferranti, S., Després, J., et al. (2016). Heart disease and stroke statistics-2016 update: A report from the american heart association. *Circulation*, 133(4):e38.

Wu, J., Roy, J., and Stewart, W. F. (2010). Prediction modeling using ehr data: challenges, strategies, and a comparison of machine learning approaches. *Medical care*, 48(6):S106–S113.

Yamokoski, L. M., Hasselblad, V., Moser, D. K., Binanay, C., Conway, G. A., Glotzer, J. M., Hartman, K. A., Stevenson, L. W., and Leier, C. V. (2007). Prediction of rehospitalization and death in severe heart failure by physicians and nurses of the escape trial. *Journal of cardiac failure*, 13(1):8–13.

Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., and Hovy, E. (2016). Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489.

Zolfaghar, K., Meadem, N., Teredesai, A., Roy, S. B., Chin, S.-C., and Muckian, B. (2013). Big data solutions for predicting risk-of-readmission for congestive heart failure patients. In *Big Data, 2013 IEEE International Conference on*, pages 64–71. IEEE.

List of Figures

List of Tables

Appendices

Appendix A

Hyper-parameter optimization

Parameter	**Settings**
Embedding Nodes	[128, 256, 512]
Embedding Dropout	[0, 0.2, 0.4, 0.6, 0.8]
Recurrent Nodes	[128, 256, 512]
Recurrent Dropout	[0, 0.1, 0.2]
Context Dropout	[0, 0.2, 0.4, 0.6, 0.8]
L2 Regularization	[0, 0.0001, 0.001]

Table A.1: Hyper-parameters grid

Logistic Regression	
Overall	
L2 Regularization	0.1

Multilayer Perceptron	
Hidden Layer	
Hidden Nodes	256
Hidden Dropout	0
L2 Regularization	0.0001
Output Layer	
L2 Regularization	0.1

Table A.2: Logistic Regression and Multilayer Perceptron hyper-parameter settings

RNN	
Embedding Layer	
Embedding Nodes	512
Embedding Dropout	0
Recurrent Layer	
Recurrent Nodes (→)	256
Recurrent Dropout	0
Overall	
L2 Regularization	0

RT-RNN	
Embedding Layer	
Embedding Nodes	512
Embedding Dropout	0
Recurrent Layer	
Recurrent Nodes (←)	512
Recurrent Dropout	0.1
Overall	
L2 Regularization	0.0001

BD-RNN	
Embedding Layer	
Embedding Nodes	512
Embedding Dropout	0
Recurrent Layer (visit)	
Recurrent Nodes (→)	512
Recurrent Nodes (←)	512
Recurrent Dropout	0.1
Overall	
L2 Regularization	0.0001

Table A.3: Baseline RNN-based models hyper-parameter settings

Dipole

Embedding Layer	
Embedding Nodes	512
Embedding Dropout	0
Recurrent Layer	
Recurrent Nodes (→)	256
Recurrent Nodes (←)	256
Recurrent Dropout	0.1
Context Layer	
Context Dropout	0.4
Overall	
L2 Regularization	0

RETAIN

Embedding Layer	
Embedding Nodes	256
Embedding Dropout	0.8
Recurrent Layer (α)	
Recurrent Nodes	128
Recurrent Dropout	0.1
Recurrent Layer (β)	
Recurrent Nodes	512
Recurrent Dropout	0
Context Layer	
Context Dropout	0.6
Overall	
L2 Regularization	0

GRNN-HA

Embedding Layer	
Embedding Nodes	256
Embedding Dropout	0.1
Recurrent Layer (visit)	
Recurrent Nodes (→)	128
Recurrent Nodes (←)	128
Recurrent Dropout	0.05
Recurrent Layer (sequence)	
Recurrent Nodes (→)	128
Recurrent Nodes (←)	128
Recurrent Dropout	0.05
Context Layer	
Context Dropout	0
Overall	
L2 Regularization	0

Table A.4: ‘Interpretable’ RNN-based models hyper-parameter settings

Appendix B

Diagnosis and treatment of Heart Failure

B.1 Chronic Heart Failure

There are a bunch of symptoms and signs that are typical for Heart Failure (table B.1) Ponikowski et al. (2016). However, the symptoms are often non-specific which makes it hard to discriminate between HF and other problems. Most of the signs are more specific but they suffer from being harder to detect and reproduce. Being elderly, obese or having chronic lung disease makes it even harder to identify and interpret the symptoms and signs of HF.

Symptoms

Typical
Breathlessness
Orthopnoea
Paroxysmal nocturnal dyspnoea
Reduced exercise tolerance
Fatigue, tiredness, increased time to recover after exercise
Ankle swelling
Less Typical
Nocturnal cough
Wheezing
Bloated feeling
Loss of appetite
Confusion (especially in the elderly)
Depression
Palpitations
Dizziness
Syncope
Bendopnea

Signs

More Specific
Elevated jugular venous pressure
Hepatojugular reflux
Third heart sound (gallop rhythm
Laterally displaced apical impulse
Less Specific
Weight gain (> 2 kg/week)
Weight loss (in advanced HF)
Tissue wasting (cachexia)
Cardiac murmur
Peripheral oedema (ankle, sacral, scrotal)
Pulmonary crepitations
Reduced air entry and dullness to percussion at lung bases (pleural effusion)
Tachycardia
Irregular pulse
Tachypnoea
Cheyne Stokes respiration
Hepatomegaly
Ascites
Cold extremities
Oliguria
Narrow pulse pressure

Table B.1: Symptoms and Signs of Heart Failure

There are several initial tests that can be performed on a patient that is suspected to have HF. The first of these is to measure the plasma concentration of natriuretic peptides (NPs). If a person has a normal plasma NP concentration value, that person is unlikely to have HF and other hypotheses should be investigated. In contrast however, the positive predictive value of the plasma NP concentration is not that high. This means that having abnormal plasma NP concentration does not imply that the patient has HF. Therefore, "the use of NPs is recommended for ruling-out HF, but not to establish the diagnosis". Ponikowski et al. (2016)

Another initial test used to rule out HF is the electrocardiogram (ECG). An ECG registers the electric activity within the heart during a heartbeat. Patients that have normal ECG are again unlikely to have HF. In addition to being used to rule out HF, some abnormalities observed on an ECG can provide information about the underlying cause of the HF.

If the plasma concentration NPs and the ECG both show abnormalities, echocardiography is the recommended test to perform in order to establish the HF diagnosis. Echocardiography allows for the measurement of the left ventricular ejection fraction (LVEF) which is essential in the diagnosis of HF. Furthermore it can be used to find out about most types of structural and/or functional cardiac abnormalities.

Before moving on to management and prevention there are some interventions that can prevent or delay the onset of CHF. The ESC notes the following interventions with the highest class of recommendation and level of evidence Ponikowski et al. (2016).

Pre-onset interventions for HF prevention/delay

1. Treatment of hypertension (i.e. reduce blood pressure) with anti-hypertensive drugs like diuretics, angiotensin receptor blockers and beta-blockers.

2. Treatment with statins (cholesterol lowering drugs) for patients with or at high risk of coronary artery disease.

3. Treatment with angiotensin-converting enzyme inhibitors for patients with asymptomatic left ventricular dysfunction and a history of myocardinal infarction.

Once a patient has been diagnosed with HF the physicians has the difficult task to manage the clinical status and functional capacity of his/her patients. Ultimately the goal is to improve or maintain the quality of life and prevent hospitalisation and mortality. For CHF the physician has the option prescribe certain medications (pharmacological treatment) or the placement of an electrical device. The ESC again notes the following interventions with the highest class of recommendation and level of evidence Ponikowski et al. (2016).

Pharmacological treatment for CHF patients

1. Angiotensin-converting enzyme inhibitors (ACEI) to reduce mortality and morbidity in patients with HFrEF.

2. Beta-blockers to reduce mortality and morbidity in HFrEF patients by controlling high heart rate. Also, beta-blockers and ACEI's seem to be complementary and can/should be prescribed at the same time.

3. Mineralorcorticoid receptor antagonist for patients with LVEF $\leq 35\%$ to reduce mortality and HF hospitalisation.

Treatment with a device for CHF patients

1. Implantable cardioverter-defibrilator to reduce the risk of sudden death and all-cause mortality due to bradycardia and potentially lethal ventricular arrhythmias.

2. Cardiac resynchronization therapy to improve cardiac performance, symptoms and well-being as well as reduce mortality and morbidity

B.2 Acute Heart Failure

Acute Heart Failure (AHF) is the term used to describe a patient either with rapid onset of HF symptoms or with rapid deterioration of HF symptoms. AHF can be life-threatening and therefore usually results in hospitalisation. Multiple factors can be the trigger for AHF. Some of these factors are acute coronary syndrome, excessive rise in blood pressure, infection, non-adherence with salt/fluid intake or medications, toxic substances, drugs, surgery and perioperative complications.

In the urgent phase after first medical contact, the patient with suspected AHF investigated to see if he/she is in a cardiogenic shock or suffering from respiratory failure. If either of these is the case, the first objective is to stabilize the patient with circulatory or ventilatory support. After that the objective is to identify the acute aetiology (underlying cause) leading to the decompensation of the patient. Typical precipitants include the following.

Acute aetiology for patients with suspected AHF

1. Acute Coronary Syndrome
2. Hypertensive emergency
3. Rapid arrhythmias or sever bradycardia/conduction disturbance
4. Acute mechanical cause
5. Acute pulmonary embolism

After identification and stabilization of the acute aetiology, the diagnostic process should be started to confirm the AHF diagnosis. This process begins with looking at the medical history of the patient for signs and symptoms of HF. In addition, an assessment of signs and symptoms of congestion and hypoperfusion should be made by physical examination. After that the diagnosis should be confirmed with additional investigations like ECG, chest X-ray, laboratory assessment (looking for specific biomarkers) and echocardiography.

Once AHF has been confirmed a patient is usually categorized as being in one of the four categories based on the absence/presence of congestion and hypoperfusion (table B.2) as described below.

Congestion	**Hypoperfusion**
Pulmonary congestion	Cold sweated extremities
Orthopnoea/paroxysmal nocturnal dyspnoea	Oliguria
Peripheral (bilateral) oedema	Mental confusion
Jugular venous dilatation	Dizziness
Congested hepatomegaly	Narrow pulse pressure
Gut congestion, ascites	
Hepatojugular reflux	

Table B.2: Symptoms and Signs of congestion and hypoperfusion

1. **Warm-Dry**. Patients where congestion and hypoperfusion are both absent. In this category of patients the recommended treatment is an adjustment of the oral therapy (medications).
2. **Warm-Wet** Patients where congestion is present but hypoperfusion is absent. These type of patients should be treated with diuretics and/or vasolidators.
3. **Cold-Dry** Patients where congestion is absent but hypoperfustion is present. For these category of patients a fluid challenge or the administration of an inotropic agent is recommended to treat the hypoperfusion.
4. **Cold-Wet** Patients where congestion and hypoperfusion are both present. Treatment options differ based on the systolic blood pressure of the patients. For patients with systolic blood pressure <90 mm Hg, the administration of an inotropic agent, vasopressors, diuretics and mechanical circulatory support (if no response to drugs) are recommended. For patients with higher systolic blood pressure, the recommended treatment options are vasodilators, diuretics and an inotropic agent.

Appendix C

Cox Proportional Hazards Regression

Chin and Goldman (1997) have performed a study in which they use a survival analysis model. The goal of this study was to identify risk factors that correlated with early re-hospitalisation or death for HF patients. The authors used a CPHR model. The characteristics that were found to be significant in increasing the probability of early hospitalisation or death were ’single marital status’, ’Charlson Comorbidity Index score’, ’Systolic blood pressure $\leq$ 100 mm Hg’ and ’No ST-T-wave ECG changes’. Based on these risk factors the authors devised a risk score that categorized patients into groups ranging from low risk (0-20 %) to the highest risk (51-88 %) of re-hospitalisation or death within 60 days of discharge (see table C.1). A similar study has been performed by Krumholz et al. (2000) (see table C.2) that found patient characteristics upon which to base risk stratification. The significant predictors in this study were quite different from Chin and Goldman (1997). Namely, ’Creatinine >2.5 mg/dL at discharge’, ’Prior admission within 1 year’, Prior heart failure’ and ’Diabetes’.

The study by Chin and Goldman (1997) suffers from a small sample size of a mere 257 patients from only 1 hospital. Also (probably due to the small sample size), the authors did not use a separate dataset to validate their model. The study by Krumholz et al. (2000) suffers less from these drawbacks with a sample size of 2176 patients (from 18 hospitals) and an almost 50-50 split of the data for derivation and validation. In both studies the authors state that they were unable to identify low-risk patients. This is due to the small number of people in the lowest risk category in Chin and Goldman (1997) and due to the relative high risk (31%) of re-hospitalisation in the lowest risk group in Krumholz et al. (2000).

Risk Score	**Nr. of patients**	**% Re-hospitalised or dead within 60 days (95% CI)**
0–1	17	0 (0–20)
2–5	144	24 (17–31)
6–7	71	42 (31–55)
> 7	25	72 (51–88)

Significant correlates and their contribution to the risk score
Single marital status = 2 points
Charlson Comorbidity Index score = 1 point per Charlson point (maximum 4)
Initial systolic blood pressure $/leq100$ mm Hg = 3 points
No ST-T-wave ECG changes = 2 points

Table C.1: Risk Stratification with CPHR. Adapted from Chin and Goldman (1997)

Nr. of correlates	**Nr. of patients**	**% All-cause Re-hospitalised or dead within 6 months (validation sample)**
0	156	31
1–2	649	54
3–4	242	65

Significant correlates
Creatinine >2.5 mg/dL at discharge
Prior admission within 1 year
Prior heart failure
Diabetes

Table C.2: Risk Stratification with CPHR. Adapted from Krumholz et al. (2000)

Appendix D

Simulatability

With regards to the simulatability property of the models we previously alluded to its relationship with hyper-parameters of the model (e.g. number of nodes in a hidden layer). This means that any of the models can be made more simulatable than the others by making the number of nodes very small. However, leaving these hyper-parameters aside, we can still look at the minimal number of layers of each model and use that as a measure for simulatability. See Table D.1 for the minimal number of layers needed for the MLP and RNN-based models.

Model	**Layers**	**Nr.**
MLP	- Input Layer - Hidden Layer - Output Layer	3
RNN (& RT-RNN)	- Input Layer - Embedding Layer - (→ / ←) Recurrent Layer - Output Layer	4
BD-RNN	- Input Layer - Embedding Layer - (→) Recurrent Layer - (←) Recurrent Layer - Output Layer	5
Dipole	- Input Layer - Embedding Layer - (→) Recurrent Layer - (←) Recurrent Layer - Attention Layer - Output Layer	6
RETAIN	- Input Layer - Embedding Layer - (←) Recurrent Layer visit-level - (←) Recurrent Layer event-level - Attention Layer - Output Layer	6
GRNN-HA	- Input Layer - Embedding Layer - (→) Recurrent Layer event-level - (←) Recurrent Layer event-level - Attention Layer event-level - (→) Recurrent Layer visit-level - (←) Recurrent Layer visit-level - Attention Layer visit-level - Output Layer	9

Table D.1: Minimal number of layers for MLP, and RNN-based models

Appendix E

Auxiliary functions

$$\text{sigmoid}(x) = \frac{1}{1 + e^{-x}} \tag{E.1}$$

$$\text{ReLU}(x) = max(x, 0) \tag{E.2}$$

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{E.3}$$

$$\text{Softmax}(x_i) = \frac{e^{x_i}}{\sum_j^t e^{x_j}} \tag{E.4}$$

Appendix F

Using Time Information

As was described in Section 4.1.1, the medical history of a patient is represented by a sequence of T tuples $(t_i, x_i) \in \mathbb{R} \times \mathbb{R}^r$, where $i \in 1, ..., T$ and T is the number of visits to the hospital. x_i denotes the visit information of the i-th visit and t_i denotes the timestamp of the i-th visit. In this section we describe a way to incorporate the temporal information into the RNN-based models. The general idea is that the visit representation v_i is concatenated with the temporal information t_i in order to obtain v_i'. It is important to note that t_i is not constrained to be in a specific representation. It can be represented as days from the first visit, days until next hospitalisation, or anything else that describes the temporal dimension. In our experiments we used the natural log of the number of days until the next visit to the hospital. In the following tables we formally describe how the temporal information can be incorporated for each of the three interpretable RNN models.

$$\begin{aligned}
v_i &= \text{ReLU}(W_{emb}x_i + b_{emb}), \text{ for } i = 1, ..., t\\
v_i' &= [v_i, t_i],\\
h_1, ..., h_t &= [\overrightarrow{GRU}(v_1', ..., v_t'); \overleftarrow{GRU}(v_t', ..., v_1')],\\
g_i &= W_\alpha^\top h_i + b_\alpha, \text{ for } i = 1, ..., t\\
\alpha_1, ..., \alpha_t &= \text{Softmax}(g_1, ..., g_t),\\
c_t &= \sum_i^t \alpha_i h_i,\\
\tilde{h}_t &= \tanh(W_c[c_t; h_t]),\\
\hat{y}_t &= \text{sigmoid}(W_y \tilde{h}_t, +b_y),
\end{aligned}$$

Table F.1: Dipole formal description with temporal information. The temporal information of each visit t_i is concatenated with the visit embedding v_i to obtain v_i'. The visit embedding plus temporal information is then propagated further through the network.

$$\begin{aligned}
w_{ij} &= \text{ReLU}(W_{emb}x_{ij} + b_{emb}),\\
h_{1j}, ..., h_{tj} &= [\overrightarrow{GRU}(w_{1j}, ..., w_{tj}); \overleftarrow{GRU}(w_{tj}, ..., w_{1j})],\\
f_{ij} &= \tanh(W_\beta^\top h_{ij} + b_\beta)^\top u_\beta,\\
\beta_{1j}, ..., \beta_{tj} &= \text{Softmax}(f_{1j}, ..., f_{tj}),\\
v_i &= \sum_j^k \beta_{ij} h_{ij},\\
v_i' &= [v_i, t_i],\\
H_1, ..., H_t &= [\overrightarrow{GRU}(v_1', ..., v_t'); \overleftarrow{GRU}(v_t', ..., v_1')],\\
g_i &= \tanh(W_\alpha^\top H_i + b_\alpha)^\top u_\alpha,\\
\alpha_1, ..., \alpha_t &= \text{Softmax}(g_1, ..., g_t),\\
c_t &= \sum_i^t \alpha_i H_i,\\
\hat{y}_t &= \text{sigmoid}(W_y c_t, +b_y),
\end{aligned}$$

Table F.2: GRNN-HA formal description with temporal information. The temporal information of each visit t_i is concatenated with the visit representation that is obtained after the first BRNN layer. Similar to Dipole, v_i and t_i are concatenated into v_i' which is then propagated further through the network.

$$
\begin{aligned}
v_i &= W_{emb} x_i, \\
v'_i &= [v_i, t_i], \\
g_t, ..., g_1 &= \overleftarrow{GRU}(v'_t, ..., v'_1), \\
e_i &= W_\alpha^\top g_i + b_\alpha, \\
\alpha_1, ..., \alpha_t &= \text{Softmax}(e_1, ..., e_t), \\
f_t, ..., f_1 &= \overleftarrow{GRU}(v'_t, ..., v'_1), \\
\beta_i &= \tanh(W_\beta f_i + b_\beta), \\
c_t &= \sum_i^t \alpha_i \beta_i \odot v_i, \\
\hat{y}_t &= \text{sigmoid}(W_y c_t + b_y)
\end{aligned}
$$

Table F.3: RETAIN formal description with temporal information. Similar to Dipole an GRNN-HA, the temporal information t_i is concatenated with the visit embedding v_i to obtain v'_i. The visit embedding plus temporal information is then used to generate the attention values. However, unlike Dipole and GRNN-HA the visit embedding **without** temporal information is used to generate the context vector (in order to match dimensionalities).

Appendix G

Synonyms of Heart Failure

Hartfalen	**Hartfalen Links**	**Hartfalen Rechts**
Hart zwak	Linkerventrikelfalen	Rechtszijdig hartfalen
Falen cardiaal	Lnkszijdig hartfalen	Rechts decompensatio cordis
Hartinsufficientie	Links decompensatio cordis	Hartfalen rechts
Insufficientie hart	Linker ventriculaire insufficientie	Falen rechter harthelft
Insufficientie cardiaal	Linkerhartfalen	Rechter ventrikel decompensatie
Falen van de hartfunctie	Links decompensatie	Right heart failure
Zwak hart	Dalen linkervertrikel	Heart Failure, right-sided
Falen hart	Falen linker harthelft	Cardiac failure right
Cardiaal falen	Linker-ventrikeldecompensatie	Cardiac failure right heart
Hartdecompensatie	Left cardiac failure	
Hart insufficientie	Left ventricular failure	
Falende hartfunctie	Left heart failure	
Cardiaal insufficientie	Left sided heart failure	
Hartdecompensatie, niet gespecificeerd	Left ventricular insufficiency	
Weak heart		
Cardiac Failure		
Cardiac insufficiency		
Cardiac function failure		
Heart failure		
Heart insufficiency		
Cardiac function failed		
Acuut Hartfalen	**Cardiale Decompensatie**	**Chronisch Hartfalen**
Plots hartinsufficientie	Caridaal decompensatie	Chronisch hartfalen
Acute hartinsufficientie	Cardiale decompensatie	Chronic heart failure
Hartfalen acuut	Decompensatie hart	Chronic cardiac failure
Hartinsufficientie plots	Decompensatie cardiaal	Cardiac failure chronic
Acute Heart failure	Heart Decompensation	
Acute cardiac failure	Decompensation cardiac	
Cardiac failure acute		
Acute cardiac insufficiency		

Table G.1: Synonyms of Heart Failure.