DEGREE PROJECT IN COMPUTER SCIENCE AND ENGINEERING, SECOND CYCLE, 30 CREDITS

STOCKHOLM, SWEDEN 2020

A Deep Learning Approach to Predicting the Length of Stay of Newborns in the Neonatal Intensive Care Unit

BAS THEODOOR STRAATHOF

KTH ROYAL INSTITUTE OF TECHNOLOGY
SCHOOL OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

A Deep Learning Approach to Predicting the Length of Stay of Newborns in the Neonatal Intensive Care Unit

BAS THEODOOR STRAATHOF

Master in Machine Learning
Date: July 11, 2020
Supervisor: Kevin Smith
Examiner: Pawel Herman
School of Electrical Engineering and Computer Science
Swedish title: En Djupinlärningsstrategi för att Förutsäga Längden för Nyfödda Vistelser i Intensivvårdsenheten

Abstract

Recent advancements in machine learning and the widespread adoption of electronic health records have enabled breakthroughs for several predictive modelling tasks in health care. One such task that has seen considerable improvements brought by deep neural networks is length of stay (LOS) prediction, in which research has mainly focused on adult patients in the intensive care unit. This thesis uses multivariate time series extracted from the publicly available Medical Information Mart for Intensive Care III database to explore the potential of deep learning for classifying the remaining LOS of newborns in the neonatal intensive care unit (NICU) at each hour of the stay. To investigate this, this thesis describes experiments conducted with various deep learning models, including long short-term memory cells, gated recurrent units, fully-convolutional networks and several composite networks. This work demonstrates that modelling the remaining LOS of newborns in the NICU as a multivariate time series classification problem naturally facilitates repeated predictions over time as the stay progresses and enables advanced deep learning models to outperform a multinomial logistic regression baseline trained on hand-crafted features. Moreover, it shows the importance of the newborn's gestational age and binary masks indicating missing values as variables for predicting the remaining LOS.

Keywords: *Deep Neural Networks; Electronic Health Records; Length-of-Stay Prediction; Multivariate Time Series Classification*

Sammanfattning

Framstegen inom maskininlärning och det utbredda införandet av elektroniska hälsoregister har möjliggjort genombrott för flera prediktiva modelleringsuppgifter inom sjukvården. En sådan uppgift som har sett betydande förbättringar förknippade med djupa neurala nätverk är förutsägelsens av vistelsetid på sjukhus, men forskningen har främst inriktats på vuxna patienter i intensivvården. Den här avhandlingen använder multivariata tidsserier extraherade från den offentligt tillgängliga databasen Medical Information Mart for Intensive Care III för att undersöka potentialen för djup inlärning att klassificera återstående vistelsetid för nyfödda i den neonatala intensivvårdsavdelningen (neonatal-IVA) vid varje timme av vistelsen. Denna avhandling beskriver experiment genomförda med olika djupinlärningsmodeller, inklusive *long short-term memory*, *gated recurrent units*, *fully-convolutional networks* och flera sammansatta nätverk. Detta arbete visar att modellering av återstående vistelsetid för nyfödda i neonatal-IVA som ett multivariat tidsserieklassificeringsproblem på ett naturligt sätt underlättar upprepade förutsägelser över tid och gör det möjligt för avancerade djupa inlärningsmodeller att överträffa en multinomial logistisk regressionsbaslinje tränad på handgjorda funktioner. Dessutom visar det vikten av den nyfödda graviditetsåldern och binära masker som indikerar saknade värden som variabler för att förutsäga den återstående vistelsetiden.

Nyckelord: *Djupa Neurala Nätverk; Elektroniska Hälsoregister; Klassificering av Multivariat Tidsserie; Förutsägelse av Vistelsetid*

Svensk titel: *En Djupinlärningsstrategi för att Förutsäga Vistelsetiden för Nyfödda i Neonatala Intensivvårdsavdelingen*

Acknowledgements

I would like to express my gratitude to my thesis supervisor associate professor Kevin Smith for guiding me through the thesis process and for providing valuable feedback. I would also like to express special thanks to associate professor Pawel Herman, for being the examiner of my work and for his thoughtful insights and feedback. And many thanks to PhD candidate Lennart van der Goten, for his support and great advise throughout the thesis project.

Thanks to Master student Yue Song for peer reviewing my thesis report, and to Master student Beata Rystedt for reviewing the *sammanfattning* and Swedish title.

Special thanks to Katlin Kreamer-Toning and Philip Jean Hartout for their loving support, proof-reading and motivation.

List of Abbreviations

ANN	**A**rtificial **N**eural **N**etwork
APACHE-II	**A**cute **P**hysiology **A**nd **C**hronic **H**ealth **E**valuation **II**
BN	**B**ayesian **N**etwork
CNN	**C**onvolutional **N**eural **N**etwork
CSV	**C**omma **S**eparated **V**alues
DRG	**D**iagnosis **R**elated **G**roups
DT	**D**ecision **T**ree
EHR	**E**lectronic **H**ealth **R**ecord
FCN	**F**ully **C**onvolutional **N**etworks
GA	**G**estational **A**ge
GRU	**G**ated **R**ecurrent **U**nit
ICU	**I**ntensive **C**are **U**nit
MLR	**M**ultinomial **L**ogistic **R**egression
LOS	**L**ength **O**f **S**tay
LR	**L**inear **R**egression
LSTM	**L**ong **S**hort-**T**erm **M**emory
MIMIC-III	**M**edical **I**nformation **M**art for **I**ntensive **C**are **III**
MAE	**M**ean **A**bsolute **E**rror
NICU	**N**eonatal **I**ntensive **C**are **U**nit
RAM	**R**andom **A**ccess **M**emory
ReLU	**Re**ctiefied **L**inear **U**nit
RF	**R**andom **F**orest
RNN	**R**ecurrent **N**eural **N**etwork
SE	**S**queeze and **E**xcitation
SVM	**S**upport **V**ector **M**achine
UCR	**U**niversity of **C**alifornia **R**iverside

Contents

Chapter 1

Introduction

1.1 Background

The neonatal intensive care unit (NICU) specializes in the care of ill or premature newborns. Accurately predicting the length of stay (LOS) of infants admitted to the NICU is important for two main reasons. Firstly, admission of a newborn infant to the NICU is a stressful experience for parents. Besides concerns about their child's survival and the risk of long-term impairments, parents usually want to have an indication of how much longer their baby will have to remain hospitalized. An accurate prediction of the anticipated remaining LOS of newborn infants would be a useful guide for expectation management and improve communication between clinicians and the parents of their patients. Secondly, a primary goal of hospital management is to allocate their resources in an optimal way. They aim to maximize the quality of patient healthcare while minimizing the costs associated with it. Precise LOS prediction of newborns is essential for the efficient use of the resources available in the NICU, such as staff, incubators and other medical devices. In addition, reliable insight into the LOS of newborns in the NICU is of general economic interest to hospital management, since NICU admissions are costly [1], [2] and LOS is largely correlated with hospital costs [3], [4]. The relevance of accurate LOS prediction in the NICU is thus two-fold: it is a useful aid for parent counseling and essential for effective resource management.

Previous research has acknowledged that reliable LOS prediction of patients in the NICU is of great societal importance and of substantial financial interest to the healthcare industry [5], [6]. However, there is a substantial difference in magnitude of research on predicting the LOS of adult patients in comparison with predicting the LOS of neonates. Predicting the LOS of NICU patients and adult intensive care unit (ICU) patients are often considered as separate tasks, since the physiology and pathology of newborns greatly differ from those of adults [7]. Rather than proposing models for predicting the LOS of newborns, most existing research on the topic has primarily focused on statistical analysis to identify indicators of

prolonged stay [8]–[10]. In contrast, the research on predicting the LOS of adult patients is more mature and has made large strides in the past decade [11], [12]. State-of-the-art research describes how recurrent neural networks (RNNs) can be applied to predicting the LOS of adult patients in the ICU [11], [13]. These models perform significantly better than baselines such as multinomial logistic regression (MLR) and linear regression (LR), though there is still a large margin for improvement. RNNs, as well as fully convolutional networks (FCNs), come with the benefit that they enable approaching LOS prediction as a multivariate time series classification problem [11], [14]. This requires minimal modification of the raw clinical features and naturally facilitates the prediction of a patient's LOS at set time-intervals from the initial moment of admission until their discharge or end of life. In contrast, MLR and LR models require extensive feature engineering to be able to predict a patient's LOS at set time-intervals [11], [13]. The use of RNNs and FCNs for predicting the LOS of newborns in the NICU seems to be absent in the research literature.

The data used in this thesis comes from a large publicly available database called the Medical Information Mart for Intensive Care III (MIMIC-III) [15]. MIMIC-III is a comprehensive database of anonymized admissions, containing electronic health records (EHRs) of adult patients admitted to the regular ICU as well as those of newborns admitted to the NICU. The state-of-the-art research on LOS prediction has made extensive use of the adult cohort of MIMIC-III [11], [12], but to date no single study seems to exist that uses data from MIMIC-III to model the LOS of newborns.

1.2 Research Objective

This thesis studies how deep learning architectures such as RNNs and FCNs can be leveraged to advance the research on LOS prediction for newborn patients in the NICU. It compares these approaches to MLR and LR baselines and investigates the added value of including gestational age (GA) [1] as a variable and binary mask variables indicating missing values in the multivariate time series used to train the models. The thesis uses the MIMIC-III database as a representative data source to addresses the following two research questions:

1. *For the task of predicting a newborn's remaining LOS in the NICU at each hour of the stay, how do advanced RNN and FCN architectures using multivariate time series compare to MLR and LR models trained on hand-engineered features?*

2. *To what extent, if at all, do the GA variable and binary mask values offer extra information that can be used to predict the LOS of newborns in the NICU?*

This thesis makes several contributions to the research field on NICU LOS prediction. The main contributions are summarized as follows:

[1] The gestational age is the estimated age counting from the beginning of the mother's last menstrual period.

1. Modelling a selection of clinical NICU variables from the MIMIC-III database as a multivariate time series. This enables repeated predictions of the remaining LOS at every hour of the stay with enhanced performance using advanced deep learning models. Since there seem to be no published studies that have carried out a similar task on the same part of the MIMIC-III database, this thesis sets the bar for future research on LOS prediction for newborns in the NICU with MIMIC-III.

2. Exploring novel model architectures based on RNNs and FCNs for modelling NICU LOS as a multivariate time series classification problem.

3. Analysing the added value of the GA variable and binary mask indicator variables by refitting one of the best performing deep learning model twice: once on the training data set without the GA variable, and once on the training data without the binary mask variables.

1.3 Thesis Outline

The remainder of this thesis is organized as follows. Chapter 2 lays out the theoretical background of the research. It addresses multivariate time series analysis and provides a fundamental background of the workings of the RNNs and FCNs used in this thesis. Chapter 3 gives an overview of work related to this thesis. Chapter 4 is concerned with the methodology of this study. Chapter 5 explains the experiments conducted in this thesis and presents the results. Chapter 6 provides a discussion of the main findings of this thesis and highlights its limitations, ethics and sustainability. Additionally, it suggests directions for future research. Chapter 7 is the final chapter, which presents a general conclusion. The code to preprocess the data, and to construct, train and evaluate the models can be found on Github: github.com/bt-s/NICU-length-of-stay-prediction.

Chapter 2

Background

This chapter provides a preliminary background into the theory on which the models of this thesis are built. The theory is presented in three sections. The first section covers time series classification. The second section provides a brief introduction to RNNs and an in-depth explanation of the theory behind the long short-term memory (LSTM) and gated recurrent unit (GRU) cells, which are the type of all RNNs used in this thesis. The third section discusses the FCN, and architectures combining RNNs and FCNs for multivariate time series classification. This chapter justifies why these models are suitable architectures for predicting the LOS of newborns in the neonatal ICU on data that is represented as a multivariate time series.

2.1 Time Series Classification

This section explains the basic characteristics of time series and substantiates why this thesis chooses to model the remaining LOS as a discrete-time multivariate time series classification problem. Time series come in various forms, but the common denominator is that time series are collections of data points that are indexed across time. Time series most commonly represent a sequence of discrete-time data in which the individual data points are equally spaced in time. They can also be representations of real-valued continuous data or discrete symbolic data such as English text. The simplest form of time series are univariate time series, which, as the name suggests, consist of a single time-dependent variable in which each value can depend on its past values. In contrast to univariate time series, multivariate time series comprise two or more time-dependent variables. Besides depending on their past values, the variables in multivariate time series can be inter-dependent (i.e. the value of a variable at a given time point can impact the values of one or more of the other variables). Time series are used across a wide range of domains such as signal processing, econometrics, weather forecasting and statistics. In fact, time series are used in any domain of engineering or applied science that concerns data of temporal nature. This thesis represents the clinical data produced in the NICU as multivariate

time series of finite length with discrete one-hour intervals, spanning from the start of the NICU stay to the discharge or death of the patient.

Time series are applicable to a manifold of information extraction problems. Common use cases are to characterize or model the system that generates the time series signal, and to forecast future values in the time series. Another application is time series classification, which is the practice of assigning time series to one of multiple categories. The time series used in this thesis are exploited to model the dynamics of NICU LOS of individual patients. The main goal of LOS prediction is to predict at any discrete time step t how many more time steps will be observed before the series expires (i.e. it predicts the remaining LOS at time step t in hours). This is a regression problem in the case that the objective of the model is to predict the exact number of future time steps. It is anticipated that such fine-grained prediction has large error margins and thus be of low practical value. The problem is therefore transformed into a time series classification problem by changing the model's objective to assign each time series pattern to one of a finite amount of classes. These classes, or buckets, describe all possible discrete numbers of future timestamps (e.g. a remaining LOS of less than two days, two days up to a week, and longer than a week).

Multivariate time series preserve the time dimension of the data, which is the main advantag of representing the clinical data associated with NICU stays this way. This enables sequential modelling with RNNs and prediction at arbitrarily fine-grained, equidistant time steps. Ample research has recognized that regularly predicting LOS based on a discrete-time multivariate time series yields superior performance in comparison to individual predictions at pre-determined timestamps such as 24 or 48 hours after the initial time of admission [11], [12], [16], [17]. Additionally, one-hour time intervals seem appropriate for the task at hand since it is an adequate compromise between missing values and multiple recordings per timestamp [11]–[13].

2.2 Recurrent Neural Networks

2.2.1 A Brief Introduction

This thesis assumes familiarity with standard feed-forward artificial neural networks (ANNs) such as the multi-layer perceptron (MLP) and the general procedure of training ANNs by using the back-propagation algorithm [18]. RNNs are a family of ANNs that address the modelling of sequential data. RNNs exhibit temporal behavior through connections between nodes between individual layers and recursively applying the same function when traversing the network's graph structure. An RNN can be represented as a traditional feed-forward ANN by unrolling its recurrent units (see Figure 2.1).

The input to an RNN is a data sequence (e.g. a multivariate time series) for each of $1, 2, \ldots, \tau$ discrete time steps t: $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_\tau\}$ where $\mathbf{x}_t \in \mathbb{R}$. Commonly, RNNs maintain a hidden vector representation $\mathbf{h}_t$ of the input sequence up to the current time step t

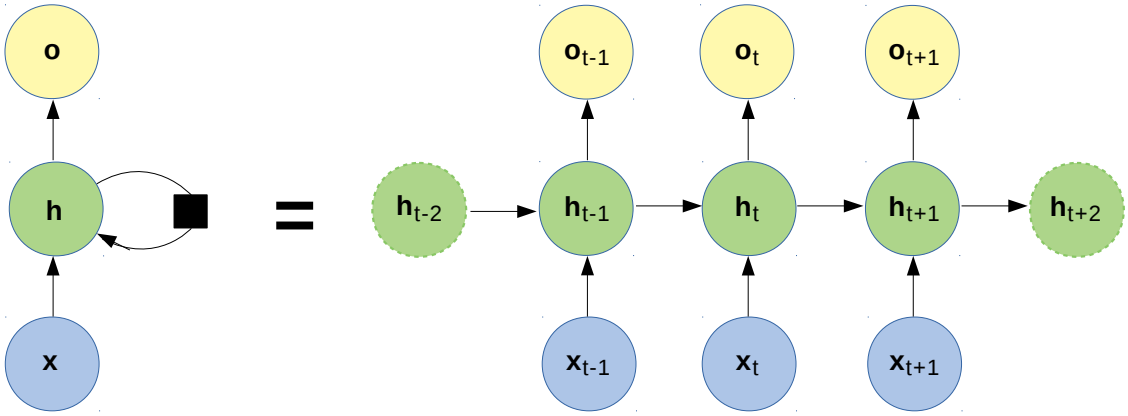

Figure 2.1: A regular RNN (left) and its time-unrolled visualization (right).

which is defined by a function f of the following form:

$$\mathbf{h}_t = f\left(\mathbf{h}_{t-1}, \mathbf{x}_t; \boldsymbol{\theta}\right)$$

In this function, $\mathbf{x}_t$ is the input vector at time t and $\boldsymbol{\theta}$ are the time-independent hyper-parameters of f. Commonly, the initial hidden state $\mathbf{h}_0$ of the RNN is assumed to be given and $\mathbf{h}_t$ is computed using a non-linear function such as the hyperbolic tangent function ϕ_h:

$$\phi\left(\mathbf{x}\right) = \frac{e^{2x} - 1}{e^{2x} + 1}$$

This is how the hidden state vector is computed:

$$\mathbf{h}_t = \phi_h\left(\mathbf{W}\mathbf{h}_{t-1} + \mathbf{U}\mathbf{x}_t + \mathbf{b}\right)$$

In the above equation, assuming that $\mathbf{x}_t \in \mathbb{R}^d$ and $\mathbf{h}_t \in \mathbb{R}^m$, $\mathbf{W}$ is an $m \times m$ weight matrix, $\mathbf{U}$ is an $m \times d$ matrix and $\mathbf{b}$ is a bias vector of size $m \times 1$. At each time step t, a prediction $\mathbf{o}_t$ can be made based on the corresponding hidden state $\mathbf{h}_t$, a $k \times m$ weight matrix $\mathbf{V}$, and a bias vector $\mathbf{d}$ of size $k \times 1$, where k is the number of predictable classes:

$$\mathbf{o}_t = \Sigma\left(\mathbf{V}\mathbf{h}_t + \mathbf{d}\right),$$

where:

$$\Sigma\left(\mathbf{x}\right)_i = \frac{e^{x_i}}{\sum_{j=1}^{k} e^{x_j}} \quad \text{for} \quad i = 1, \ldots, k \quad \text{and} \quad \mathbf{x} = \left(x_1, \ldots, x_k\right) \in \mathbb{R}^k$$

The softmax function Σ is used to normalize the k output predictions into a probability distribution comprising k probabilities corresponding to the normalized exponentials of the

output predictions. After applying the softmax function, each prediction will be in the $(0, 1)$ interval and the sum over all probabilities is 1.

The procedure for training RNNs is similar to the training procedure of feed-forward ANNs. The general training scheme is to define a loss function l_t between the target $\mathbf{y}_t$ and the prediction $\mathbf{o}_t$ for each time step t, sum the loss over all time-steps and use back-propagation to minimize the aggregated loss. A common choice of loss function l_t for multi-class classification tasks with RNNs is the categorical cross-entropy loss $H(\mathbf{o}_t, \mathbf{y}_t)$. The cross-entropy loss measures the performance of a model whose output is a probability distribution. For $C > 2$ classes, the categorical cross-entropy loss is the sum over the separate loss calculated for each class label for the observation at time t:

$$H(\mathbf{o}_t, \mathbf{y}_t) = -\sum_{c=1}^{C} \mathbf{y}_{t,c}\, log(\mathbf{o}_{t,c}),$$

where $\mathbf{y}_{t,c}$ is a binary indicator whether class label c is the correct classification for the observation at time t, and $\mathbf{o}_{t,c}$ is the predicted probability that the observation at time t is of class c. If the predicted probability $\mathbf{o}_t$ diverges from the actual label $\mathbf{y}_t$, the cross-entropy loss increases. Note that the cross-entropy loss is an adequate choice of loss function since the softmax function returns a probability distribution over all possible classes.

2.2.2 Long Short-Term Memory

Two of the major shortcomings of regular RNNs are that their gradients are prone to vanishing or exploding, which makes them inadequately equipped to capture long-term dependencies. In 1997, Hochreiter and Schmidhuber proposed a novel RNN architecture called the LSTM network that addresses these problems by a set of gating functions that control what information should be added or removed at each time step t [19]. Their capacity for capturing long-term dependencies has been demonstrated repeatedly [20]. In addition to the architecture of regular RNNs, LSTMs maintain a memory cell state vector $\mathbf{c}_t$. This vector controls what information from the current input vector $\mathbf{x}_t$, the previous hidden state vector $\mathbf{h}_{t-1}$ and the previous cell state vector $\mathbf{c}_{t-1}$ should be used to generate the new hidden state vector $\mathbf{h}_t$. Figure 2.2 illustrates the general structure of the time-unfolded LSTM network.

The LSTM uses three gate functions to control the contents of the memory cell vector $\mathbf{c}_t$, namely an input gate, a forget gate and an output gate. At each time step t the following computations are carried out:

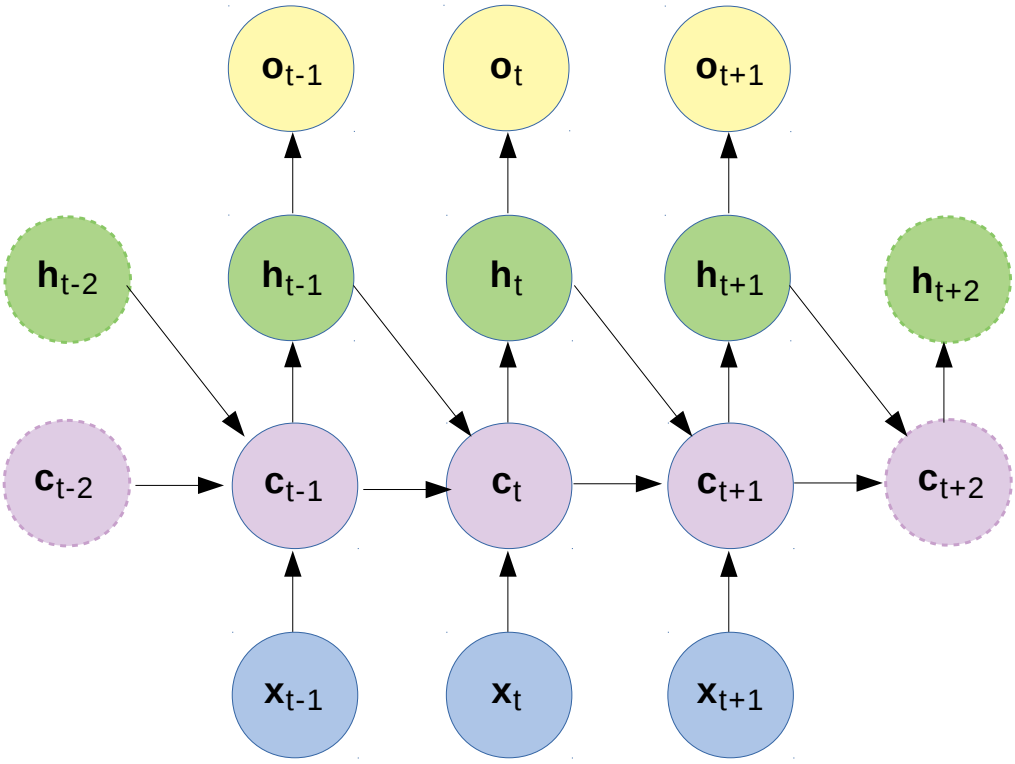

Figure 2.2: A high-level visualization of the time-unfolded LSTM.

$$
\begin{aligned}
\mathbf{g}_t^i &= \sigma\left(\mathbf{V}^i\mathbf{x}_t + \mathbf{W}^i\mathbf{h}_{t-1} + \mathbf{b}^i\right) \\
\mathbf{g}_t^f &= \sigma\left(\mathbf{V}^f\mathbf{x}_t + \mathbf{W}^f\mathbf{h}_{t-1} + \mathbf{b}^f\right) \\
\mathbf{g}_t^o &= \sigma\left(\mathbf{V}^o\mathbf{x}_t + \mathbf{W}^o\mathbf{h}_{t-1} + \mathbf{b}^o\right) \\
\mathbf{g}_t^c &= \mathbf{g}_t^f \odot \mathbf{c}_{t-1} + \mathbf{g}_t^i \odot \phi_h\left(\mathbf{V}^c\mathbf{x}_t + \mathbf{W}^c\mathbf{h}_{t-1} + \mathbf{b}^c\right) \\
\mathbf{h}_t &= \mathbf{g}_t^o \odot \phi_h\left(\mathbf{g}_t^c\right)
\end{aligned}
$$

In the above equations, $\mathbf{g}_t^i$, $\mathbf{g}_t^f$, $\mathbf{g}_t^o$ and $\mathbf{g}_t^c$ are the activation vectors of the input, forget, output, and cell state gates, respectively. The hidden state vector is represented by $\mathbf{h}_t$, and $\odot$ is the Hadamard product. The $\mathbf{V}$, $\mathbf{W}$ and $\mathbf{b}$ matrices portray the weights and biases corresponding to the various gates. The training scheme for LSTMs is the same as the one for regular RNNs as described above. The hyperbolic tangent function is represented by ϕ_h and σ depicts the sigmoid function:

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

The LSTM is one of the models used in this thesis because of its successful applications for predicting the LOS of adult patients [11], [17].

2.2.3 Gated Recurrent Units

The GRU is a modern alternative to the LSTMs. It was proposed by Cho et al. in 2014 [21], and its architecture is similar to the LSTM. The difference is that it utilizes the hidden state vector to transfer information instead of a cell state, and it only has two gates: a reset gate and an update gate. These gates decide what old information to retain and what new information to add. The high-level visualization diagram of the time-unfolded GRU looks like the traditional RNN (see Figure 2.1), with the reset and update gate directly acting on the hidden state unit. Its dynamics are dictated by the following set of equations:

$$\begin{aligned}
\mathbf{g}_t^u &= \sigma\left(\mathbf{V}^u \mathbf{x}_t + \mathbf{W}^u \mathbf{h}_{t-1} + \mathbf{b}^u\right) \\
\mathbf{g}_t^r &= \sigma\left(\mathbf{V}^r \mathbf{x}_t + \mathbf{W}^r \mathbf{h}_{t-1} + \mathbf{b}^r\right) \\
\mathbf{h}_t &= \mathbf{g}_t^u \odot \mathbf{h}_{t-1} + \left(1 - \mathbf{g}_t^u\right) \odot \phi_h(\mathbf{V}^h \mathbf{x}_t + \mathbf{W}^h(\mathbf{g}_t^r \odot \mathbf{h}_{t-1}) + \mathbf{b}_h\big)
\end{aligned}$$

In the above equations, $\mathbf{g}_t^u$ and $\mathbf{g}_t^r$ are the activation vectors of the update and reset gates, respectively. The Hadamard product is depicted by $\odot$ and $\mathbf{h}_t$ denotes the hidden state vector. The sigmoid function and hyperbolic tangent function are denoted by σ and ϕ, respectively. The trainable weight and bias matrices corresponding to the various gates are represented by the $\mathbf{V}$, $\mathbf{W}$ and $\mathbf{b}$ matrices.

It should be noted that the similar activation functions of the update and reset gate act on different weight matrices. If the activation of the reset gate is close to zero, this gate tells the GRU cell to forget the past hidden state and only retain the new information input. If the activation of the update gate is close to one, the GRU cell will carry over most of the previous hidden cell state and only allow a fraction of the new input to become part of the current hidden state. It is the intricate interplay of the update and reset gate that allows GRUs to both capture long-term and short-term dependencies. GRUs and LSTMs generally follow similar training schemes. An advantage of GRUs over LSTMs is that they have fewer parameters, which makes them computationally more efficient. GRUs are often on par with LSTMs in terms of performance [22], hence, this thesis investigates whether one is more suitable than the other for predicting the remaining LOS of newborns in the NICU.

2.3 Fully Convolutional Networks

2.3.1 A Concise Overview of CNNs and FCNs

In the past decade, convolutional neural networks (CNNs) have built a strong reputation in the deep learning community. They have been the driving forces behind significant improvements in recognition tasks, most famously in the image domain [23]. CNNs are deep learning architectures that consist of an input layer, multiple hidden layers, and an output layer. At least one of the hidden layers in a CNN is a convolutional layer, which uses convolution operations instead of general matrix multiplication. Convolutional layers try to learn representations of the input they receive by applying several convolutional kernels to compute different feature maps. Convolutional layers are often followed by a non-linear activation function such as the rectified linear unit (ReLU) [24] and a pooling layer to reduce the dimensions of the data before passing it on to the next layer. For an in-depth technical explanation of the components of CNNs, please refer to [25]. This is the ReLU function:

$$\text{ReLU}\,(x) = max\,(0, x)$$

While CNNs were originally developed to be applied to visual imagery, they have made their way into multiple other domains over time. To name a few, CNNs are now applied in speech recognition, natural language processing and time series classification [25]. The ubiquity of CNNs stems from the fact that they are capable of building hierarchical feature representation of raw data from a wide variety of sources. In 2014, Zheng et al. were one of the first to explore the utility of CNNs for supervised feature extraction in time series [26]. They proposed a multi-channel deep CNN for multivariate time series classification in which the convolutional filters are applied to each variable channel separately before being sent to a a fully-connected MLP with one hidden layer to carry out the classification.

One of the reasons why traditional CNNs are not applicable to time series is that they require fixed-size input tensors. FCNs are a recent development enabling convolutional operations on inputs of arbitrary size [27]. In 2017, Wang et al. demonstrated how FCNs outperform the state-of-the-art on the majority of the University of California Riverside (UCR) Benchmark data sets for univariate time series classification [28], [29]. The FCN archtitecture proposed by Wang et al. is displayed in Figure 2.3. Its main component is a convolutional block consisting of a convolution layer followed by a batch normalization layer [30] and a ReLU activation function. The kernels in the convolutional layer perform 1-D convolutions (i.e. they move along the time dimension) to extract features of various, though finite, locality. The finiteness of the context length of FCNs is a feature that distinguishes them from RNNs, which theoretically have an unbounded context length. The complete network consists of three such convolutional blocks with filter sizes $128, 256, 128$ and kernel sizes $8, 5, 3$, respectively. Note that no intermediate pooling layers are used, and that the purpose of the batch normalization is to speed up the

convergence and improve generalization. The output of the last convolutional block is sent to a global average pooling layer [31] which is followed by a softmax layer to perform classification. In comparison to fully connected layers, global average pooling significantly reduces the total number of model parameters and therefore lowers the risk of overfitting.

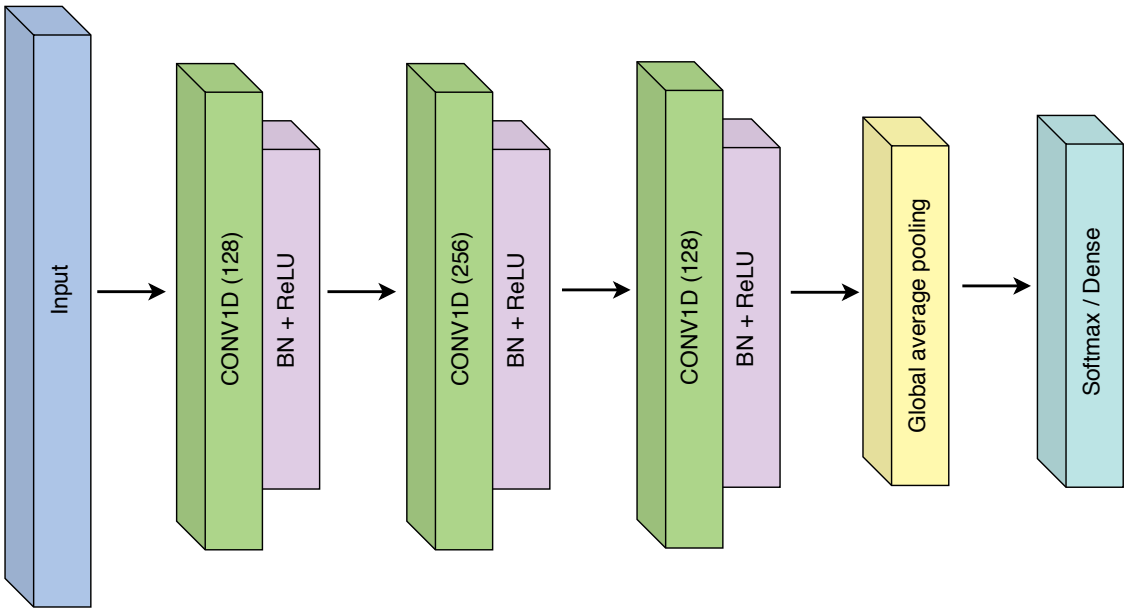

Figure 2.3: The FCN proposed by Wang et al. [29]. Note that for regression the output of the dense layer is the final output.

2.3.2 LSTM-FCNs

Karim et al. proposed an extension of the Wang et al.'s FCN for both univariate and multivariate time series classification with LSTM sub-modules [14], [32]. The addition of an LSTM module only nominally increases the size of the network, while causing a significant increase in performance on most of the UCR benchmark data sets. The architecture Karim et al. proposed for carrying out multivariate time series classification consists of a concatenation of an adapted version of Wang et al.'s FCN and an LSTM cell followed by a dropout layer [33]. The concatenation is subsequently fed to the output softmax layer. See Figure 2.4 for a complete overview of the network architecture.

As can be observed in Figure 2.4, the first two convolutional blocks in the FCN module are extended with a squeeze and excitation (SE) layer [34]. Karim et al. empirically verified that this addition is essential to the increase in classification performance on multivariate time series data sets [14]. The intuition behind this layer is that it iteratively recalibrates channel-wise feature responses by modelling dependencies between the different channels [34].

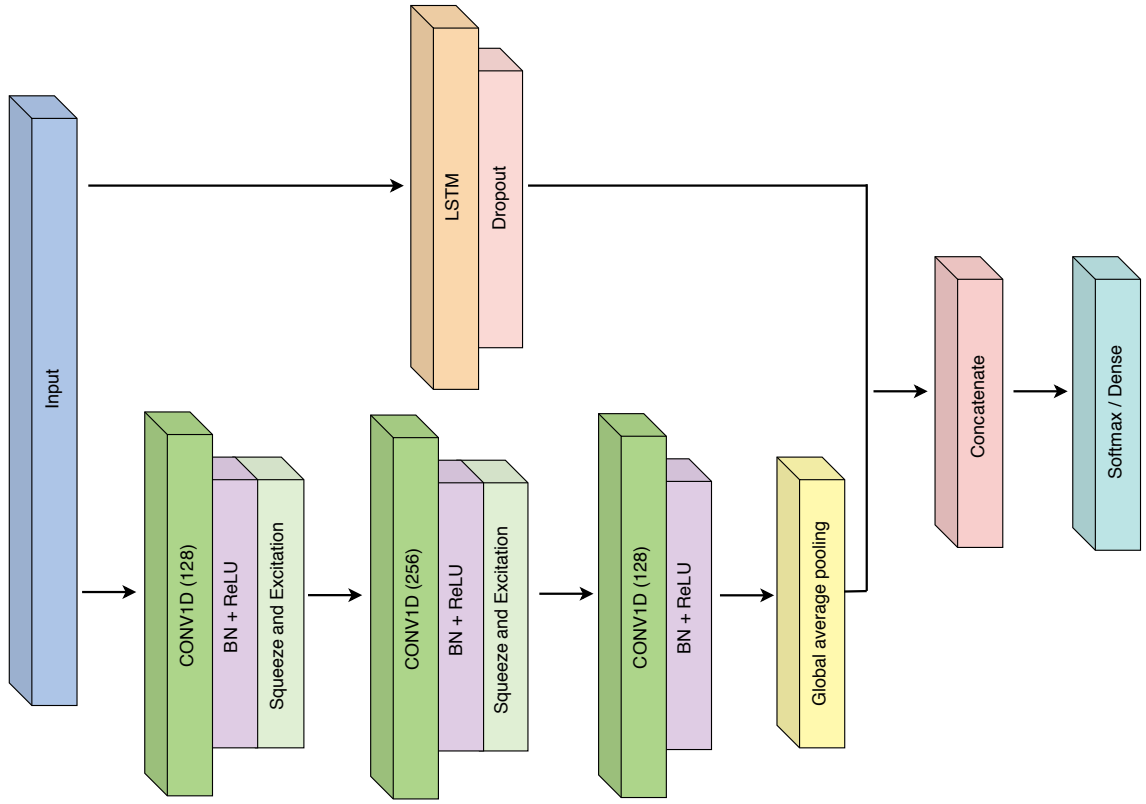

Figure 2.4: The FCN-LSTM proposed by Karim et al. [14] (with exception of the dimension shuffle layer, since it is not applicable to the tasks that this thesis deals with).

In other words, it captures the aforementioned inter-dependence between the variables in the multivariate time series. In regular feature transformations in CNNs, all convolutional channels are weighted equally when creating the output feature map. The SE layer leverages contextual information to select which local feature signals to enhance and which to suppress. An SE block can recalibrate the features for any 3D feature transformation (such as a convolution) of the following form:

$$\mathbf{F}_{tr} : \mathbf{X} \rightarrow \mathbf{U}, \mathbf{X} \in \mathbb{R}^{H' \times W' \times C'}, \mathbf{U} \in \mathbf{R}^{H \times W \times C}$$

In the case of multivariate time series, $H' \times W' = T'$ and $H \times W = T$, where T' and T denote the feature transformation's input and output dimensions corresponding to the series' time dimension. The number of output feature maps is denoted C, which is a hyper-parameter. The output $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_C]$ where:

$$\mathbf{u}_c = \mathbf{v}_c * \mathbf{X} = \sum_{s=1}^{C'} \mathbf{v}_c^s * \mathbf{x}^s$$

In the above equation, v_c^s denotes the 2-D convolutional kernels and * represents the convolutional operation. Note that the bias terms are omitted in the above and following equations to simplify the notation. The SE layer models the dependencies between the different channels of v_c and exists of two main components: a *squeeze* operation and an *excitation* operation (see Figure 2.5). The transformed features $\mathbf{U}$ are first passed through the *squeeze* operation to compress the feature maps of each of the channels into a single value by means of global average pooling. This operation enables the network to obtain global knowledge of the feature maps of the different convolutional channels. For multivariate time series, this means that the output of the feature transformation $\mathbf{U}$ are "squeezed" into the resulting vector $\mathbf{z} = [z_1, z_2, \ldots, z_C]^T$ across the time-associated dimension T, where:

$$\mathbf{z}_c = \mathbf{F}_{sq}(\mathbf{u}_c) = \frac{1}{T}\sum_{t=1}^{T} u_c(t)$$

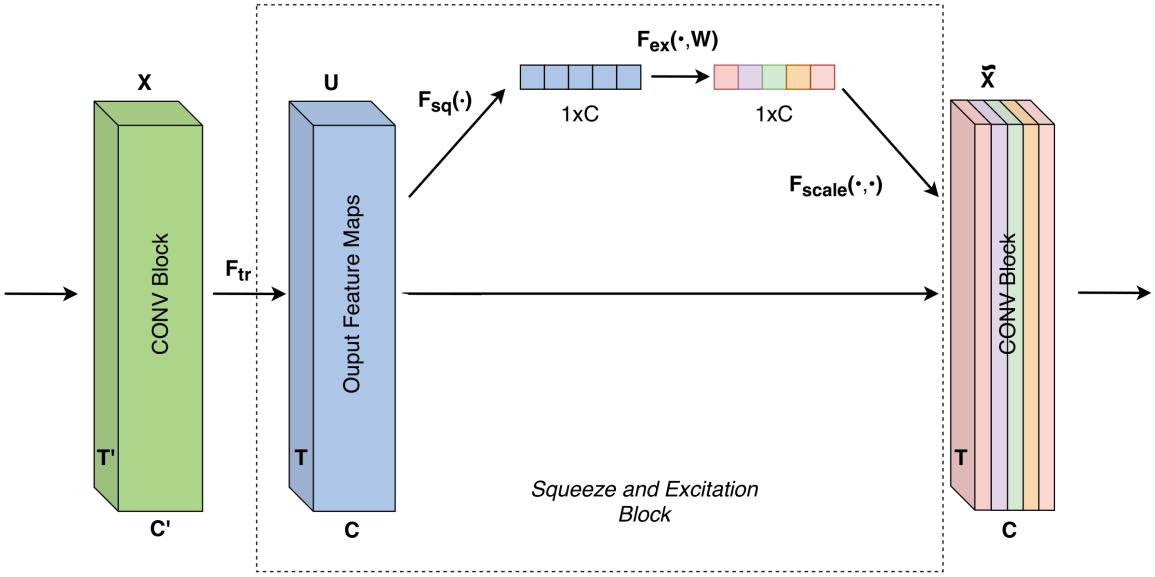

Figure 2.5: The computational flow of the SE block.

The compressed output feature maps $\mathbf{z}$ are subsequently passed through the *excitation* operation. This operation captures the dependencies between the channels such that they can be recalibrated before being passed to the next convolutional block. The vector $\mathbf{z}$ is "excited" by passing it through a simple gating mechanism with a sigmoid activation (to learn a non-linear interaction between the channels):

$$\mathbf{s} = \mathbf{F}_{ex}(\mathbf{z}, \mathbf{W}_1, \mathbf{W}_2) = \sigma\left(\mathbf{W}_2 \delta\left(\mathbf{W}_1 \mathbf{z}\right)\right),$$

where σ portrays the sigmoid function and δ refers to the ReLU function. Furthermore, $\mathbf{W}_1 \in \mathbb{R}^{\frac{C}{r} \times C}$ and $\mathbf{W}_2 \in \mathbb{R}^{C \times \frac{C}{r}}$ are two fully connected layers around the ReLU function. These layers act like a bottleneck with $\mathbf{W}_1$ performing dimensionality-reduction and $\mathbf{W}_2$ increasing the dimensionality. This limits model complexity and facilitates generalization through parameterization of the gating mechanism. The hyper-parameter r dictates the reduction ratio and thus the number of additional learnable parameters. Karim et al. set $r = 16$, which leads to a significant increase in performance while only expanding the amount of learnable parameters by $3\% - 10\%$.

Once the *excitation* operation has been carried out, the output of the SE block is rescaled to the dimension of its input $\mathbf{X}$:

$$\widetilde{\mathbf{X}}_c = \mathbf{F}_{scale}\left(\mathbf{u}_c, \mathbf{s}_c\right) = \mathbf{s}_c \cdot \mathbf{u}_c,$$

where $\widetilde{\mathbf{X}}_c = [\widetilde{\mathbf{x}}_1, \widetilde{\mathbf{x}}_2, \ldots, \widetilde{\mathbf{x}}_c]^T$ and $\mathbf{F}_{scale}\left(\mathbf{u}_c, \mathbf{s}_c\right)$ denotes the channel-wise multiplication between the output feature maps $\mathbf{u}_c \in \mathbb{R}^T$ and $\mathbf{s}_c$. This thesis uses the SE block in its experiments with FCNs, since it is an inexpensive addition that has been proven to enhance the model's classification performance [14], [34].

Interestingly, neither Wang et al. nor Karim et al. compared the performance of the FCN or LSTM-FCN against RNNs such as LSTMs or GRUs [14], [29], [32]. Instead, they compared their models against the reported state-of-the-art models on the UCR benchmark data set, such as the dynamic time-warping algorithm [35], random forest (RF) classifiers [36], support vector machines (SVMs) [37], and a data mining algorithm called WEASLE + MUSE [38]. Wang et al. compared their FCN against a MLP and a CNN but not against RNNs [29]. It should be noted that the objective of the studies by Wang et al. and Karim et al. was of a different nature than the goal of this study. Their objective was to find a general model that performs well on a wide variety of time series classification problems in which the data come in various modalities such as spectograms, images and sensory outputs [28]. This thesis aims to find the optimal model for a specific task with a particular data structure. It may be the case that RNNs are a poor candidate as a general model for time series classification, although it would be an interesting research endeavour to experiment with RNNs on the UCR benchmarks. It should also be noted that the UCR benchmark data sets are chosen in such a way that they require no or limited pre-processing; for example, they do not contain missing values. This makes them different from the hand-selected and largely imputed clinical time series used in this study. Nonetheless, experimenting with FCNs and LSTM-FCNs on those clinical time series provides insight into their versatility and relative discriminative power compared to RNN-based architectures.

This chapter has provided the historical and mathematical background of the main deep learning architectures used for modelling the remaining LOS of NICU patients. The exact training configurations of the models is delineated in Chapter 5.

Chapter 3

Related Work

This chapter opens with a succinct history of the research on LOS prediction before the deep learning era. Subsequently, it presents the improvements enabled by RNNs. The chapter concludes with an examination of the research specific to predicting the LOS of newborns in the neonatal ICU, highlighting the most prominent research gaps and delineating how this thesis attempts to fill some of them.

3.1 Before the Deep Learning Era

Statistical research on LOS prediction goes as far back as the late '60s. In 1968, Gustafson developed and evaluated five methods for LOS prediction, the most sophisticated being a Bayesian model that processed empirical data of symptomatic and demographic nature [39]. Gustafson's model used a tiny data set consisting of a sample of eight inguinal herniotomy patients stratified over four categories of LOS. A considerable limitation of Gustafson's method was its inadequate scalability, but as a proof of concept the method paved the way for future research. In the next decades, most research on LOS prediction focused on statistical analyses to identify indicators of prolonged stay. In 1972, Altman et al. used linear discriminant functions to identify the most salient of 35 indicators of prolonged stay in a sample of 3,004 psychiatric patients [40]. In 1984 Berki et al. showed with regression analysis that variation in LOS within various diagnosis-related groups (DRGs) can be modeled with several variables that are independent of the DRG's classification scheme [41]. This sparked interest in research on LOS prediction, since it demonstrated its potential as a tool for more effective management of hospital resources, as well as the shortcomings of the patient taxonomies used at the time.

The '90s marked a time in which shallow artificial neural networks (ANNs) were first proposed as candidate models for LOS prediction, and the data sets used in such research became larger. In 1989, Weintraub et al. identified several predictors of prolonged stay using

logistic regression[1] analysis on a data set of 4,683 patients that had coronary bypass surgery [42]. Also applying a logistic regression model, in 1993 Knaus et al. were the first to conduct a study on predicting the LOS of a cohort of 17,105 ICU patients [43]. At the time, they reported state-of-the-art results for modeling the LOS in number of days based on a selection of demographic, physiologic and comborbidity variables recorded during the first day of a patient's stay in the ICU. In the same year, Tu and Guerriere were the first to suggest the use of ANNs for predicting hospital LOS [44]. Their model had 15 categorical input variables such as age group, sex, urgency of surgery and the presence of comorbid diseases. Using data of 713 patients following cardiac surgery, the model was trained to predict whether a patient would have an ICU stay longer than two days. The results of the model proposed by Tu and Guerriere's were promising considering the size of the data set. In 1998, the usefulness of ANNs for predicting LOS was corroborated by Walczak et al., who used a training set of 4,200 pediatric ICU patients [45]. Despite this success, it would take almost two decades for ANNs to rise in popularity again for LOS prediction, when the use of RNNs would become feasible and bring improvements.

In the late '90s and 2000s, several studies investigated the use of severity-of-illness scoring frameworks for predicting LOS. In 1996, Ruttimann and Pollack identified the Pediatric Risk of Mortality score as a useful indicator of prolonged LOS for pediatric ICU patients [46] and Osle et al. compared the Acute Physiology and Chronic Health Evaluation II (APACHE-II) versus the International Classification of Disease Illness Severity Score scores in 1998 [47]. In 2000, Marik and Hedman further investigated the use of APACHE-II and its successor, APACHE III [48]. A few years later, in 2003, Higgins et al. studied how the Simplified Acute Physiology Score correlated with prolonged LOS [49], and in a similar fashion Patterson et al. studied correlations of LOS with the Standardized Early Warning System score [50]. In 2008, Berry et al. studied predictors of the LOS of newborns admitted to the NICU, containing features such as the Score for Neonatal Acute Physiology Version II and the Score for Neonatal Acute Physiology Perinatal Extension Version II. In 2010, Liu et al. studied the potential improvement of including the Laboratory Acute Physiology Score and the Comorbidity Point Score [51]. Although several correlations have been identified between those scores and LOS, academic interest in their applicability for LOS prediction have shrunk following the rise of machine learning methods that often outperform them [11], [52]. The aforementioned traditional scores are usually only computed on the first 24 hours of a hospital stay, whereas the present work focuses on time series prediction. This complicates direct performance comparisons [11].

In the early 2010s, machine learning methods started to outperform statistical models. In 2012, Freitas et al. used logistic regression for flagging costly LOS outliers [53]. Their database contained over 9 million inpatient episodes across a wide range of diagnoses. In the

[1]Note that logistic regression and multinomial logistic regression (MLR) are not the same model. Logistic regression can only be used for binary classification, whereas MLR is a generalized form of logistic regression that is capable of multi-class classification.

same year, Azari et al. compared the performance of several machine learning models for predicting LOS, including decision trees (DTs), SVM, RF classifiers and a Bayesian Network (BN) [54]. Additionally, Azari et al. proposed that pre-clustering the training data can improve the performance of the subsequent predictive model. The way in which LOS prediction is approached in this thesis unfortunately does not allow for such pre-clustering. Hachesu et al. 2013 employed a DT classifier, SVM, and an ANN to predict short, intermediate and long stay of cardiac patients using a cleaned data set describing 2,064 hospitalizations [55]. A year later, Morton et al. used SVMs, MLR and RFs for discriminating long versus short stay of diabetic patients [56]. The experiments of both Hachesu et al. and Morton et al. indicated that SVMs performed best in comparison to the other methods tested [55], [56].

All of the research described above has approached LOS prediction as a one-time classification or regression problem. Their models predict the LOS only once per stay, typically after the first 24 or 48 hours of hospital admission. In the early 2010s, efforts were also made to work towards a more general, continuous form of LOS prediction. In 2012, Levin et al. proposed a logistic regression model for predicting the probability that a patient will be discharged within 72 hours, dividing the 72 hours in twelve 6-hour intervals. The data set they used consisted of 2,178 admissions of patients in the pediatric ICU spanning over a 16-month period [57]. The model's output can be updated over time, if a patient's conditions based on provider orders such as diet, activity, medication and laboratory tests have changed. In 2016, similar approaches were proposed by Barnes et al. [58] and Cai et al. [59]. Barnes et al. used logistic regression and RFs to predict whether a patient will be discharged before 2:00 p.m. or 11:59 p.m. today, using a variety of categorical variables. Cai et al. built a non-disease-specific BN model on the basis of data extracted from EHRs [59]. Cai et al. estimate the probability of a hospitalized patient being at home, in the hospital, or deceased for each of the next 7 days. These efforts were among the first to approach predicting LOS as a time series. One of the limitations of these studies was that the proposed techniques were not adequate for modeling long-term dependencies. Such modeling have become more tractable in the second half of the 2010s, with the emergence of sophisticated RNNs and more abundant computational resources.

3.2 Improvements Brought By RNNs

RNNs have pushed the state-of-the-art of LOS prediction in terms of predictive power. It has been demonstrated that they consistently out-perform statistical models and traditional machine learning models, although it should be noted that there is still much room for improvement. The main advantage of RNNs in comparison to models such as logistic regression and SVM is that they enable modeling time series that integrate the time dimension inherently present in clinical data associated with LOS instead of relying on snapshot measurements. LSTM cells have become the most prominent RNNs for predicting LOS with clinical data represented as a multivariate time series [11]. One of the reasons why LSTMs only came into favor for LOS

prediction in the second half of the 2010s is that missing data in EHRs – which is a common problem – used to render them intractable for this application. In 2016, the tractability of using LSTMs for LOS prediction became evident when Lipton et al. proposed a successful method for imputing missing data in clinical time series [13].

In the same year, the MIMIC-III database was published in the public domain and became the most comprehensive publicly available database of its kind [15]. MIMIC-III is a suitable database for experiments with fine-grained discrete-time modeling, since it only contains data from ICU admissions in which the patient is closely monitored by several medical devices and routine checkups. Moreover, prior to the publication of the MIMIC-III database, no real benchmark data set was available for research on LOS prediction. Most studies used to experiment on private data sets, which hindered direct comparison of the methods proposed. In 2017, Harutyunyan et al. proposed a benchmark based on the adult cohort of the MIMIC-III database [11]. They established a benchmark for four common clinical prediction and classification tasks: LOS prediction, in-hospital mortality prediction, decompensation prediction (i.e. predicting the rapid deterioration of a structure or system in the human body due to disease) and phenotype classification (i.e. the classification of medical conditions). The benchmark that they suggested consists of time series of 42,276 ICU stays, 15% of which is reserved for model testing. The resolution of the time series is one hour, and each time-stamp consists of 17 clinical features, such as body temperature, height, heart-rate, and various scores on the Glasgow coma scale. Harutyuntan et al. approached LOS prediction as a multi-class, multivariate time series classification problem. They divided the remaining length of stay at each time-stamp into 10 buckets (i.e. less than one day, one for each of 1-7 days, between 7 and 14 days, and longer than 14 days) and compared their LSTM to a MLR baseline, due to its demonstrated success in previous studies [13]. Their results indicated that the LSTM significantly out-performed the MLR model, but that multi-task learning did not yield more predictive power. In addition, their findings demonstrated that LOS prediction is a complex problem. The underlying dynamics of ICU stays are complex, which makes it challenging to model the remaining LOS in an accurate manner. This thesis takes a similar approach to transforming the MIMIC-III data into time series and dividing the remaining length of stay into target buckets.

In 2018, Song et al. used Harutyunyan et al.'s MIMIC-III benchmark to demonstrate that a masked self-attention network is on par with LSTMs for predicting the remaining LOS [12]. In the same year, the research of Xu et al. corroborated the applicability of attention modules for predicting LOS [16]. They performed multiple experiments on a subset of 10,282 EHRs from the MIMIC-III database that were matched and time-aligned with the MIMIC-III Waveform database to investigate the usability of waveform data (e.g. photoplethysmograms, electrocardiography, and ambulatory blood pressure) for LOS prediction[2] [15]. However, it is

[2]The newborn cohort of the MIMIC-III database is significantly smaller than the adult cohort and matching it with the MIMIC-III Waveform database would reduce its size even more, hence, this thesis does not explore the

difficult to compare the results by Xu et al. to those of Harutyunyan et al. or Song et al., since they used a subset of the MIMIC-III benchmark data set of Harutyunyan et al. and reported the quadratically weighted Cohen kappa statistic instead of the linearly weighted statistic. Another study published in 2018, by Rajkomar et al., confirmed the effectiveness of RNNs for predicting whether a patient's LOS will be longer than 7 days, considering data from only the first 24 hours of admission [52]. They used a private data set that is considerably larger than the MIMIC-III database, which contains clinical records from both ICU and non-ICU hospital stays, totalling 216,221 hospitalizations involving 114,003 unique patients. In 2018, another approach was proposed by Purushotham et al., a feed-forward neural network that incorporates GRUs [21], [60]. This architecture enabled the inclusion of 136 raw features of differing temporal and non-temporal modalities. The inclusion of this abundance of variables in a time series would result in many time-stamps containing a large amount of absent or static variables, making it intractable for modeling. Purushotham et al. thus resorted to only predicting the LOS twice: once after the initial 24 hours and once after the first 48 hours of the hospital stay. This thesis chooses to pursue the approach of predicting the remaining LOS at one-hour time intervals. It models the data as a multivariate time series, which rules out architectures such as the one that Purushotham et al. proposed.

3.3 Predicting the LOS of Neonates

There is a considerable difference in magnitude of research on predicting the LOS of adult patients versus predicting the LOS of newborns. Concerns have been voiced that there is too little research on LOS prediction of NICU patients [5], [61] and as of yet the use of RNNs for neonatal LOS prediction has not been investigated.

Seminal research on predicting the LOS of newborns in the NICU started in the '90s. In 1992, Powell et al. analyzed the LOS of 762 preterm babies using logistic regression and identified seventeen factors that are predictive of discharge date [8]. They found that the GA at birth was by far the most powerful predictor, explaining 40% of variability in LOS compared to 6% explainable by respiratory difficulties, the second most powerful predictor. In 1999, Zernikow et al. demonstrated the potential of an ANN regression model for predicting the LOS of newborns admitted to the NICU [62]. Their model's predictions, which used data from the first day of life of 2,144 newborns, showed promising correlations with the actual LOS. Nevertheless, the gap between the predicted and actual values was large, leaving room for improvement for future studies.

In 2010, Manktelow et al. used statistical analyses to find indicators of prolonged LOS of very preterm newborns in the NICU [9]. The main limitation of this study was that of the 5,528 newborns in the data set, all 558 who died in the hospital or did not follow normal care

use of waveforms for predicting LOS of newborns in the NICU.

were excluded. This relaxation of the problem is impossible in real-life scenarios. Moreover, the task was of a statistical nature and no predictive modeling was carried out. In 2013, Lee et al. conducted a similar statistical analysis to find indicators of prolonged stay in extremely low weight newborns admitted to the NICU [10]. A 2015 study by Temple et al. examined 26 features extracted or engineered from daily progress notes of 4,693 NICU patients [6]. The authors created separate RF models for each of 2, 4, 7 and 10 days to discharge, to predict discharge in a binary fashion. Per model, all values for which the date to discharge was not equal to the target were set to zero. A year later, the same authors slightly improved their model by incorporating features from daily progress notes using a bag-of-words technique [63]. In comparison to the state-of-the-art of predicting the LOS of adult patients in the ICU, the methods proposed by Temple et al. are outdated. Lastly, in 2019 Seaton et al. proposed a statistical model for estimating the median LOS (and death over time) of babies that were born very preterm [61]. Since the model they propose averages over a population of 21,631 infants admitted to the NICU, the study by Seaton et al. is of a different nature than this thesis, which predicts the LOS of individual patients.

As described above, there is only a limited amount of research available on the topic of LOS prediction of newborns. To make a valuable contribution to the research field on LOS prediction, this thesis mainly draws on insights from successful research dealing with predicting the LOS of adult ICU patients. In particular, this research borrows from the approaches of Lipton et al. [13] and Harutyunyan et al. [11] for creating time series, imputing missing values, bucketing targets and constructing baseline models. Although the benchmark data set of Harutyunyan et al. provide useful insights for this study, it should be noted that extracting features from the MIMIC-III database to create time series for modeling the LOS of newborn ICU patients is a challenge. Some variables used for predicting the LOS of adult patients (e.g. the Glasgow coma scale variables) are not directly, or not at all, applicable to newborn patients. Likewise, some variables exist (e.g. the GA) that are essential for prediction the LOS using data from newborns, whereas they might not be of use to the adult population. A more in-depth analysis of these differences and the data preprocessing pipeline of this study are explained in the next chapter.

Chapter 4

Methodology

The chapter begins with an explanation of how the clinical variables are extracted from the MIMIC-III database and how they are transformed into multivariate time series. The chapter then continues with a clarification of how hand-crafted features are distilled from the time series for the MLR and LR baselines. Afterwards, the methods for finding suitable hyper-parameters for the deep learning models are described. Lastly, the chapter justifies the evaluation methods to analyse the performance of the various models.

4.1 Data Set

4.1.1 MIMIC-III

The data used in this thesis is extracted from the Medical Information Mart for Intensive Care III (MIMIC-III) database [15]. The database contains de-identified data of 61,532 ICU stays, describing 8,100 NICU stays and 53,432 stays at the adult ICU. The database is comprised of 26 tables containing several types of data such as vital signs, laboratory measurements, clinical notes and observations, medications, survival data, and more. The clinical data were collected at a single care facility, the Beth Israel Deaconess Medical Center in Boston, Massachusetts. The de-identified data for adult ICU patients were amassed during an eleven year period spanning from June 2001 to October 2012. The database does not contain data about pediatric patients, but does contain clinical data associated with 7,830 newborns admitted to the NICU between 2001 and 2008. Whereas the adult cohort of MIMIC-III has been used extensively for research on LOS prediction, the data for NICU patients has not yet been used for this purpose [11], [12], [17]. The work described in this thesis is the first to take on the endeavour of predicting the LOS of NICU patients using the MIMIC-III database.

4.1.2 Preprocessing

After PhysioNet grants access to the MIMIC-III database, the raw data can be downloaded in comma-separated values (CSV) format. Data associated with the patients' de-identified demographic data, admissions and ICU stays are stored in the `PATIENTS.csv`, `ADMISSIONS.csv`, and `ICUSTAY.csv` files. For the purposes of this research, data has to be aggregated from each of these files. The time series consist of clinical data extracted from `CHARTEVENTS.csv`, `LABEVENTS.csv` and `NOTEEVENTS.csv`, in which each registered event has a timestamp associated with it.

This study's data preprocessing pipeline can be split into four stages: identifying NICU stays, event extraction, time series construction and baseline data set construction. The exact procedures are described in the next subsections.

Identifying NICU Stays and Data Selection

Not all NICU stays are considered for the LOS prediction task addressed in this thesis. Only stays of newborn patients that were transferred to the NICU immediately after vaginal or Cesarean delivery are regarded. The final data set thus contains a single stay per patient, ruling out re-admissions and admissions occurring well after birth. Further constraints are that the stay has chart events, lab events and notes associated with it, and that the stay lasted at least four hours. This approach has also been taken in the benchmark study on LOS prediction of the adult ICU patients in the MIMIC-III database [11] and ensures that for each stay at least some data is present that can be provided as training input to a machine learning model.

Premature birth is a common cause of a newborn's admission to the NICU [5], [61]. The patient's GA has therefore been described as an important feature for LOS prediction [5], [61]. The MIMIC-III database tabulates the estimated GA as a three-week window, which is a rather coarse estimate with dubious practical value. The general format of note-taking at the Beth Isreal Deaconess Medical Center usually allows for the extraction of a more precise indication of the GA in the notes associated with the ICU stay. These GA estimates, which can be extracted using a regular expression[1], are generally expressed as accurately as the exact number of weeks and days. In the event that only the number of weeks is present, the number of days is set by uniformly sampling from the range of integers from zero to six. The NICU stays of newborns for whom the GA cannot be identified in the notes are discarded (see Figure 4.1).

During the period in which the data of the MIMIC-III database were generated, the hospital at times suffered from insufficient capacity in the NICU leading to patients being transferred out for non-medical reasons. This bias can be partially mitigated. Stays for which the `ICUSTAYS.csv` table indicates that a transfer took place are discarded. Prematurely ended ICU stays because of bed unavailability identified in the clinical notes are discarded as well. These

[1]The regular expressions used in this thesis can be found here: *github.com/bt-s/NICU-length-of-stay-prediction/../reg_exps.py*

cases are identified by using a regular expression. Transfers due to specific medical conditions are not discarded. In some cases, the newborn was transferred to the operation room for surgery or to a specialized ICU such as the cardiac ICU, which marks the end of the NICU stay. These transfers complicate LOS prediction and may differ from facility to facility, depending on the technical expertise of the NICU in question. This thesis considers transfers to be an integral part of the task, since at the start of the NICU stay it is generally unknown that a transfer will take place. The complexity of such logistical dynamics make LOS prediction for NICU patients a challenging task.

Enforcing the above constraints on the initial 8,100 NICU stays in the MIMIC-III database results in a data set of 4,171 NICU stays of newborns, of which 54 resulted in death. Predicting the LOS of newborns in the NICU is difficult, since a short stay sometimes means good health whereas in other cases it means a medical urgency requiring a transfer or death. Refer to Figure 4.1 for the exact quantification of the data selection procedure.

Event Extraction

Medical equipment measures vital signs such as the respiratory rate, blood pressure, heart rate and body temperature at regular intervals. Other parameters, such as weight, length and physical appearances are reported at intervals that may vary from patient to patient. Lab tests are mostly carried out when a specific medical condition is suspected or has to be ruled out.

The MIMIC-III database lists thousands of event identifiers corresponding to an abundance of chart and lab events [15]. Many clinical variables such as body temperature and blood pressure have multiple event identifiers associated with them. To predict the LOS for adult patients in the ICU, Harutyunyan et al. selected 17 clinical variables that are measured regularly enough across the population to be useful for time series modelling with RNNs [11]. A subset of 11 of these variables seem adequate for LOS prediction of NICU patients, which include: *capillary refill rate* (i.e. the time required for color to return to external capillaries after pressure is applied to cause blanching), *diastolic blood pressure* (i.e. the force of blood against the artery walls while the heart relaxes), *systolic blood pressure* (i.e. the force of blood against the artery walls while the heart contracts), *heart rate*, *respiratory rate*, *temperature*, *pH* (of the blood), *fraction inspired oxygen* (i.e. the percentage or oxygen in the air that a person inhales), *oxygen saturation* (i.e. the extent to which hemoglobin in the blood is saturated with oxygen), *height* and *weight*. The remaining 6 are not selected in this study because of a variety of reasons. The *mean blood pressure* is not selected since it is implicitly present in the combination of the *diastolic blood pressure* and *systolic blood pressure* variables. The variable *glucose* is not selected since it is only measured in a small fraction of the NICU stays. Finally, the Glasgow coma scale variables (i.e. *Glasgow coma scale eye opening*, *Glasgow coma scale motor response*, *Glasgow coma scale total* and *Glasgow coma scale verbal response*) are not applicable to newborn patients.

MIMIC-III DATABASE RAW DATA

7,863	Neonatal ICU admissions
7,830	Neonatal ICU patients
8,100	Neonatal ICU stays
49,766,294	Chart events
620,115	Lab events
419,756	Clinical notes

EXCLUDING

7,863 - 7,818 =	45	Non-newborn admissions
7,818 - 7,722 =	96	Admissions without chart events
8,100 - 8,029 =	71	Stays with transfers
8,029 - 7,804 =	225	Stays that weren't the first admission
7,804 - 7,797 =	7	Stays with undefined LOS
7,797 - 5,482 =	2,315	Stays shorter than 4 hours
5,482 - 5,352 =	130	Stays with missing admission info
5,352 - 5,351 =	1	Stays of non-newborns
5,351 - 5,250 =	101	Stays without lab events
5,250 - 5,205 =	45	Stays without notes events
5,205 - 4,238 =	967	Stays without identifiable GA
4,238 - 4,171 =	67	Stays with a capacity-related transfer

RESULTS IN

4,171	Neonatal ICU stays
45,843,483	Clinical events (chart and lab)

EXCLUDING

16,281	Events with incorrect chart time
319,488	Events with no value
110	Events with incorrect HADM_ID and ICUSTAY_ID
44,069,974	Events corresponding to non-selected variables

RESULTS IN

4,111	Neonatal ICU stays
1,437,630	Clinical events (chart and lab)
360,113	Clinical notes

AFTER TRAIN/TEST SPLITTING

Train (64%)	Val (16%)	Test (20%)	
2,632	657	822	Neonatal ICU stays
925,570	233,975	278.085	Clinical events

Figure 4.1: The data selection pipeline.

In addition to the 11 clinical variables described above, *bilirubin direct*, *bilirubin indirect* and *gestational age in days* are selected for predicting the LOS of NICU patients. These are measurements taken at birth, which are not typical for adult patients. Hyperbilirubinemia (i.e. elevated bilirubin levels) can indicate a variety of medical issues in newborns, most commonly jaundice [64]. The direct bilirubin test measures the bilirubin that is conjugated with glucuronic acid, and the indirect bilirubin test measures the unconjugated bilirubin that attaches to albumin. The direct and indirect bilirubin test are usually carried out simultaneously to obtain a full picture. An overview of the 14 clinical variables used in this study is listed in Table 4.1. A list of the MIMIC-III item IDs that correspond to these clinical variables can be found in Appendix A. For each subject, only values that fall within the valid ranges are kept in the lab and chart event tables. A variable could be invalid due to human (e.g. a typo or using the wrong measurement unit) or unpredictable machine errors. A justification of the valid ranges and the imputation values can be found in Appendix B.

Variable	Unit	Valid range	Imputation value
Bilirubin (direct)	μmg/dL	$[0, 30]$	0.0
Bilirubin (indirect)	μmg/dL	$[0, 30]$	0.0
Blood pressure (diastolic)	mmHg	$[0, 100]$	37
Blood pressure (systolic)	mmHg	$[0, 170]$	67
Capillary refill rate	n.a.	$\{0..1\}$	0
Fraction inspired oxygen	n.a. (%)	$[21, 100]$	21
Gestational age	days	$[161, 301]$	n.a.
Heart rate	BPM	$[0, 400]$	156
Height	cm	$[0, 70]$	(see Table 4.2)
Oxygen saturation	n.a. (%)	$[0, 100]$	97
pH	n.a.	$[6.5, 8]$	7.3
Respiratory rate	BRPM	$[0, 150]$	43
Temperature	Celsius	$[20, 44]$	36.2
Weight	kg	$[0.4, 7.0]$	(see Table 4.2)

Table 4.1: The 14 clinical variables selected for this study, their valid range and the value used for imputation. Please refer to Figure 4.4 for their frequency.

Some of the variables need extra cleaning besides removing those with a variable outside the allowed range or an error value. In the MIMIC-III database, four strings are used to describe the capillary refill rate. The strings *brisk* and *Normal < 3 secs* are mapped to 0 whereas *delayed* and *abnormal > 3 secs* are mapped to 1. In some cases, the temperature is reported in Fahrenheit, in which case it is converted to Celsius. If the weight is given in grams, it is converted to kilograms. In the event that after the cleaning process no valid measurements

GA (weeks)	Height (cm)	Weight (kg)
22	31.6	0.60
23	32.6	0.68
24	33.6	0.77
25	34.6	0.85
26	35.6	0.93
27	36.6	1.02
28	37.6	1.11
29	38.6	1.22
30	39.9	1.37
31	41.1	1.54
32	42.4	1.73
33	43.7	1.90
34	45.0	2.11
35	46.2	2.35
36	47.4	2.59
37	48.6	2.87
38	49.8	3.13
39	50.7	3.36
40	51.2	3.48
41	51.7	3.57
42	51.5	3.51
43	51.3	3.42
44	51.0	3.38

Table 4.2: The imputation values for weight and height corresponding to GA.

remain, the corresponding NICU stay is discarded.

All events are validated to finalize the event selection process. Events with an incorrect chart time (i.e. a time occurring before the timestamps marking the beginning of the NICU stay or the timestamp signifying the end of the stay), events with no value, and events with incorrect hospital IDs (i.e. `HADM_ID`) or ICU stay IDs (i.e. `ICUSTAY_ID`) are excluded (see Figure 4.1).

There is good reason to consider LOS prediction of adult and newborn ICU patients as separate tasks. First of all, there is a substantial difference in the distribution of the LOS of NICU patients versus adult ICU patients. It can be observed from Figure 4.2 that the variance and range of the LOS and remaining LOS distributions for NICU patients is much larger. The 95 percentile (i.e. the shortest 95% of stays) of NICU stays include stays up to approximately

2000 hours (approx. 83 days), whereas for adults the longest stay in the 95 percentile is around 500 hours (about 21 days). The biggest outlier for NICU patients is a stay of 4,120 hours (or 172 days when rounded to the closest day). When the bucketing approach of Harutyunyan et al. [11] is applied to the NICU stays, the class distribution is much more imbalanced than for adult ICU patients (see Figure 4.2). The fact that the remaining LOS distribution has such a long and heavy tail, in combination with cases of premature death and unplanned hospital transfers, makes accurately predicting the LOS for newborns in the NICU a more challenging task.

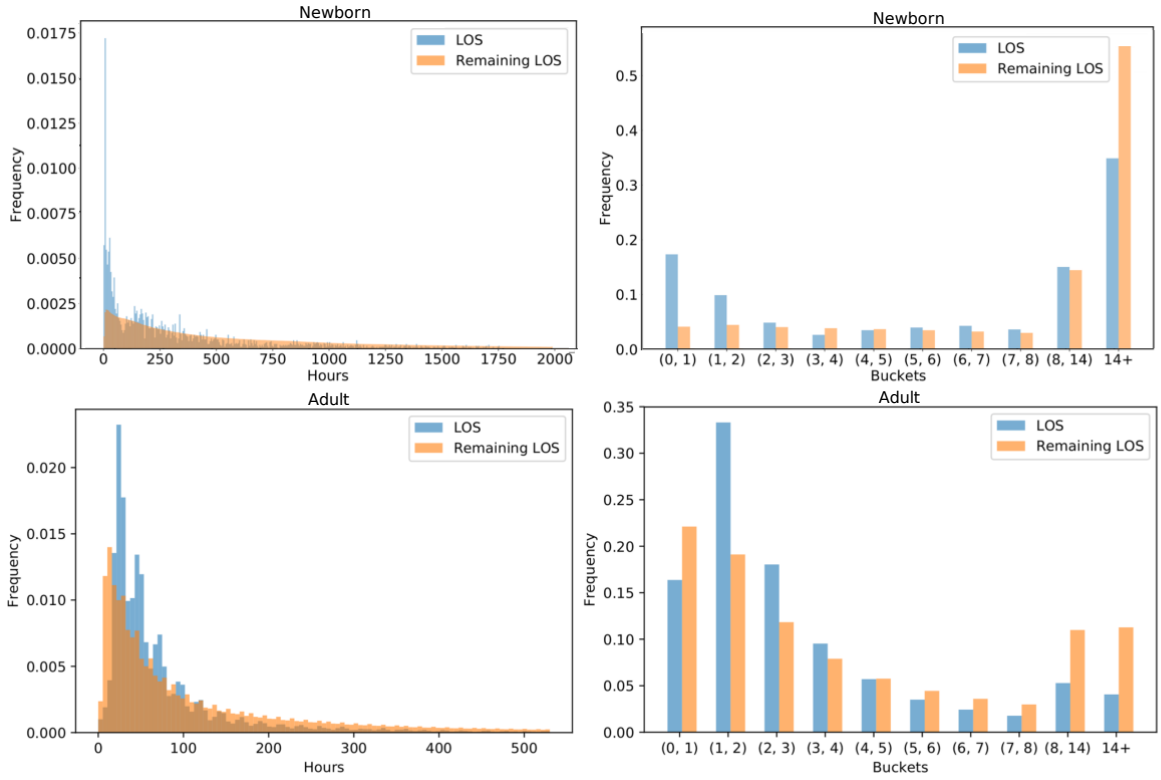

Figure 4.2: The distribution of the LOS and remaining LOS. The 95 percentile in hours for newborn patients (top left) and adult patients [11] (bottom left). In buckets: for newborn patients (top right) and adult patients (bottom right).

Another reason for addressing LOS prediction for newborns and adult ICU patients as separate problems, is that the physiology and pathology of newborns differ from those of adults [7]. There is a large discrepancy between the normal values for the heart rate, respiratory rate and blood pressure for newborns in comparison to those of adults.

The relatively small size of the data set – prior to filtering the MIMIC-III database contains over six times more data on adult ICU stays than on NICU stays – and the complexity of NICU LOS prediction makes the bucketing strategy proposed by Harutyunyan et al. [11] inappropriate for the task at hand. Preliminary experiments showed that applying their bucketing approach

leads to unstable results, since some buckets contain too few examples. Instead, a coarser strategy that involves partitioning the targets into three buckets is applied: a bucket for remaining LOS shorter than two days (i.e. less than 48 hours), a bucket for remaining LOS between two days and a week, and a bucket for remaining LOS longer than a week. In spite of leading to coarser predictions, this bucketing strategy still has much practical value. It provides insight into what resources may be vacated within two days and which are likely to remain occupied for longer than a week. Its use in parent counselling will be similar to the use of the strategy proposed by Harutyunyan et al, and it may provide insight into the medical state of the newborn: a prediction of the stay ending within two days for an infant in distress may signify a medical urgency, whereas such a prediction for an infant that is doing well may mean that it is almost ready to be discharged from the NICU. The distribution of this bucketing approach is shown in Figure 4.3.

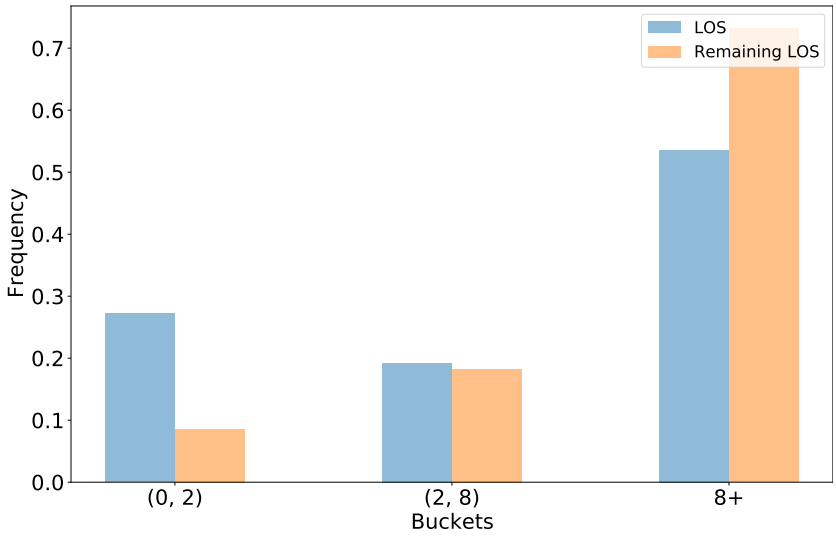

Figure 4.3: The normalized distribution of the target classes after coarse bucketing.

Time Series Construction

Once the chart and lab events have been filtered and cleaned, they can be used to create time series. A time resolution of one hour, besides being an intuitive unit, strikes a good balance between missing measurements and multiple recordings per time step [11], [13]. For each NICU stay, the chart times associated with the measurements of the selected clinical variables are rounded up to the nearest hour. The same is done for the time steps that mark the beginning and the end of the stay. When this results in a stay with no valid measurements, the stay is discarded. For the time series that are considered up to this point, if a timestamp consists of multiple measurements for a variable, only the last one is kept. If the first timestamp of a series

does not contain a valid value for *weight* and/or *height*, it is filled with the first such recording if it exists, or with the imputation value corresponding to the GA at the start of the stay taken from Table 4.2. For each timestamp in the time series, the GA can be calculated with the value of the GA at the start of the stay and the chart time corresponding to the timestamp. The remaining LOS for each timestamp and the corresponding target bucket are calculated in a similar manner.

After creating all time series, the data are split into a training and test set. There exists a single time series per newborn subject, since only the first NICU stay is considered. The subjects are split in an 80% training set and 20% test set. This is performed using stratification over the total LOS in hours, to ensure that the target distribution of both sets are similar. Applying the same stratified splitting strategy, 20% of the training data is reserved for model validation. As a result, there are 2,632 (i.e. 64%) training time series, 657 (i.e. 16%) validation time series and 822 (i.e. 20%) time series for testing.

Missing data in the time series are imputed such that the time series can be modelled with an RNN. The chosen imputation strategy is forward filling, due to its success for predicting the LOS of adult patients [11], [13]. This means that for each variable v at time step $t = 0$ that does not have a value, the corresponding imputation value in Table 4.1 is used. For each time step $t > 0$, if a variable v does not contain a value, the value of v at $t - 1$ is imputed. This imputation strategy is applied to each of the training, validation and the test sets. An insightful observation is that the pattern of missing measurements can be a feature on its own [13]. For example, recurrent requests for bilirubin lab tests might indicate that the newborn is still suffering from jaundice, whereas if no such tests are carried the infant is unlikely to suffer from such a condition. The pattern of missing measurements is preserved by creating a binary indicator feature for each of the clinical variables, which contains ones for imputed values and zeros for values that were present in the original time series [11], [13].

Some clinical measurements are recorded much more frequently than others (see Figure 4.4). For instance, in approximately 82.5% of the time steps the oxygen saturation rate is recorded, whereas on average the bilirubin test results are recorded approximately once every 100 time steps. More important than the absolute recording frequency is the fraction of patients for which at least one measurement recorded. All of the selected variables have at least one recording in 40% of the stays, many of which are measured at least once in the majority of the stays (see Figure 4.5). The objective is that the forward filling imputation technique results in time series with a discriminative pattern in the case that measurements were recorded. For stays in which a certain variable was never mentioned, the corresponding imputation value listed in Table 4.1 is used for all time steps.

To finalize the multivariate time series, the training and validation data after imputation are used to calculate the mean and standard deviation of each of the variables in Table 4.1. For each variable in the multivariate time series of the training, validation and test data sets, its values are standardized by subtracting the corresponding mean and dividing by the standard

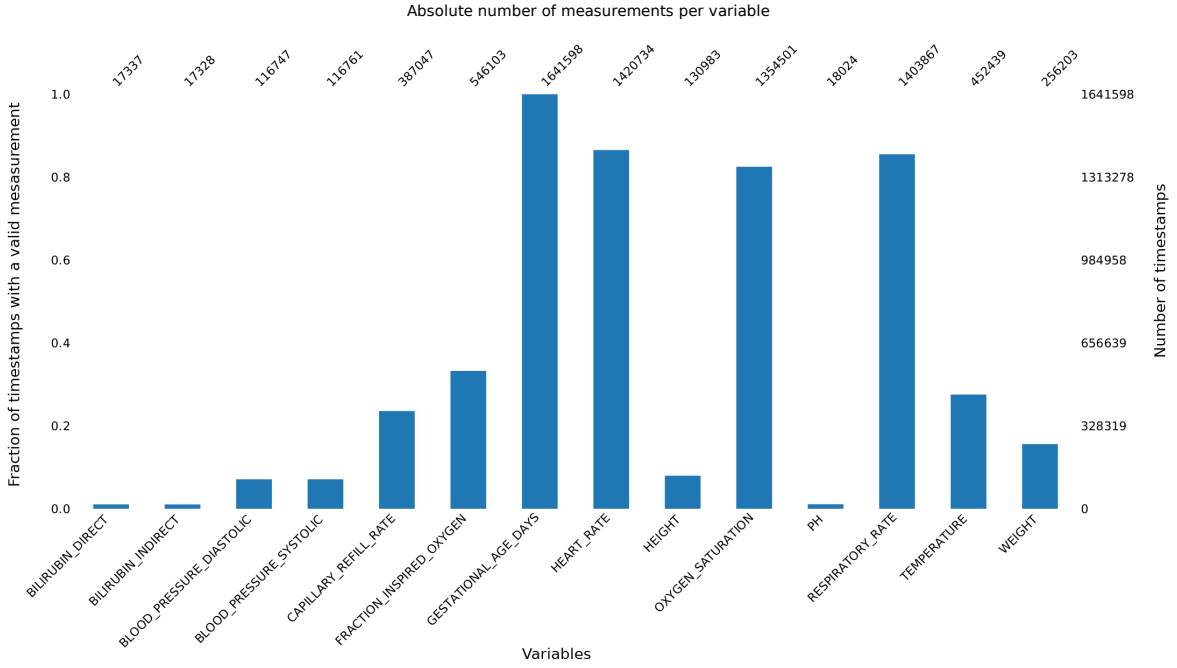

Figure 4.4: The fraction (left y-axis) of the 1,641,598 time steps of all time series combined that have a recorded measurement for each of the selected clinical variables. The absolute number of measurements for a clinical variable is mentioned above each corresponding bar.

deviation that are calculated using the data from the training and validation set.

One hurdle to overcome before the multivariate time series can be used as input to the deep learning models, is that the hospital stays vary in length. The RNN and FCN models are trained in mini-batches that require all multivariate time series in one such mini-batch to be of the same length. A fast and memory-efficient method for grouping multivariate time series into mini-batches in which all time series are close in length is explained in section 5.1.2. To ensure that all multivariate time series in a mini-batch are of exactly the same length, shorter time series are zero-padded to the length of the longest time series in the mini-batch. Before a mini-batch is passed as input to a deep learning model, a mask is applied to ensure that the model does not consider the padded zeros as features. This mask is propagated through the whole model, such that each layer skips the calculations corresponding to the zero-padded time steps. The masking procedure is of vital importance for the reliability and validity of the models. If the mask were not to be applied, predictions for individual data examples at inference time would be inconsistent if carried out in batches. Different batch sizes lead to different amounts of zero-padding for individual data examples in the batch, hence, it is imperative that the zero padding be ignored during the training process. See Figure 4.6 for a diagram exemplifying how a single multivariate time series corresponding to the hospitalization of an

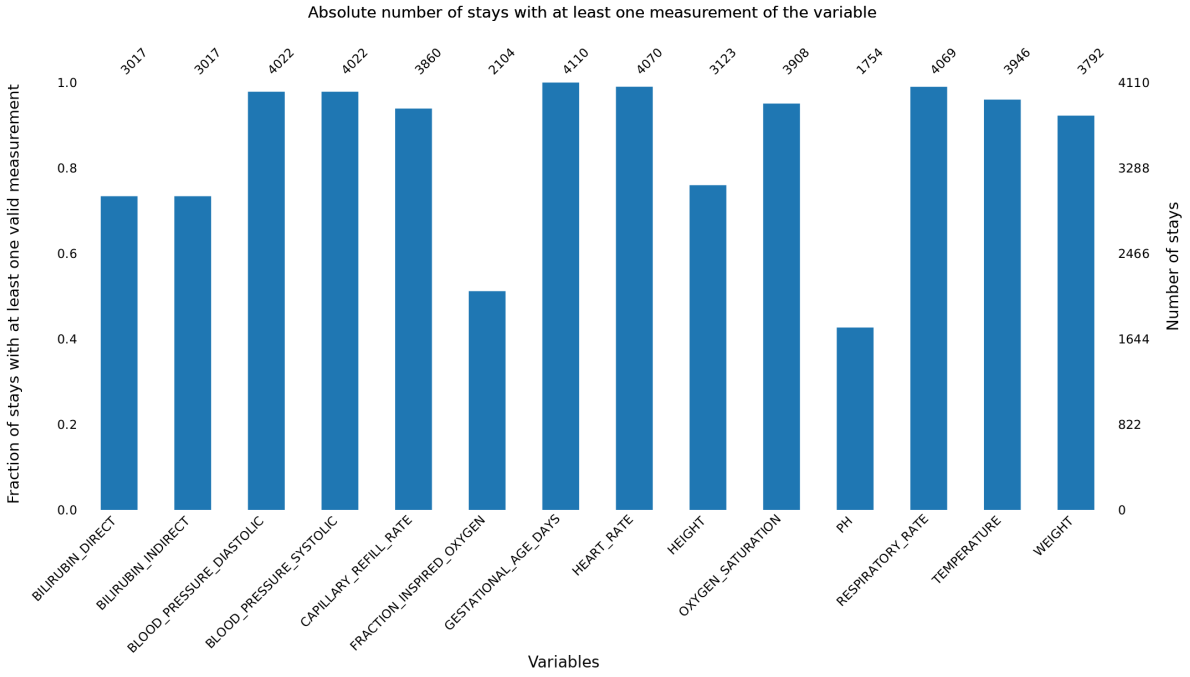

Figure 4.5: The fraction (left y-axis) of 4,110 NICU stays that have at least one recorded measurement for each of the 14 clinical variables. The absolute number of stays for which a variable was recorded at least once is mentioned above each corresponding bar.

individual newborn is passed to a learning model for classification or regression.

4.2 Baselines

This thesis constructs two baseline models to compare the RNNs and FCNs. An MLR model serves as a baseline for classification, and LR is used as a baseline for regression. To train these models, all sub-sequences of the multivariate time series originating from the start of the stay that are at least four hours long are considered. There are 1,629,763 such sub-sequences for the 4,110 unique time series. On average, the sub-sequences are 615 timestamps (or hours) long. Previous research has demonstrated the importance of down-scaling the dimensionality of the data to be able to train the MLR and LR baselines effectively [11], [13]. Based on the practices of Lipton et al. [13] and Harutyunyan et al. [11], each of the time series sub-sequences is represented by a fixed-size feature vector that is constructed by computing statistics over slices of the sub-sequence. This is necessary, since the MLR and LR cannot model variable-length inputs directly. See Figure 4.7 for a schematic overview of how the baseline feature vectors

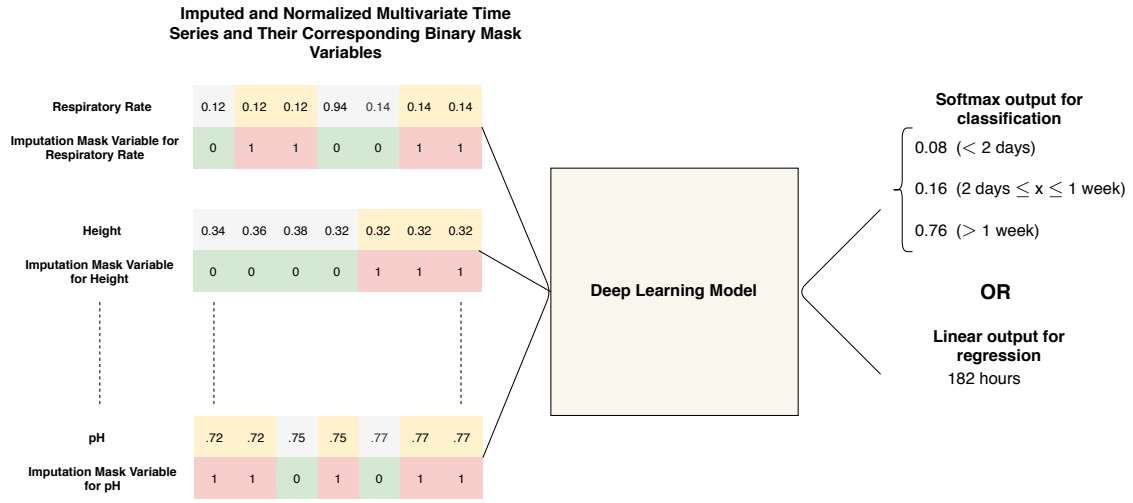

Figure 4.6: A conceptual diagram illustrating how a single data point (i.e. the multivariate time series corresponding to a part of a NICU stay of a single patient) is passed as input to any of the RNN and FCN deep learning models that this thesis explores. The deep learning model gives one of two types of output, depending on whether the task is classification or regression. For classification, the output of the deep learning model is the softmax probability distribution over the three classes (i.e. a remaining LOS shorter than two days, between days and a week, and longer than a week). For regression, the model's output is a prediction of the exact remaining LOS in hours for the input time series. In the visualization of the binary mask variables a 1 means that the value above was imputed, whereas a 0 means that the value is an original measurement. Note that the various RNNs and FCNs are trained in mini-batches containing multiple multivariate time series, in which some may be zero-padded such that all multivariate time series in the mini-batch are of the same length. By means of masking, it is ensured that the deep learning models disregard the padded zeros in their computations.

are created. For each sub-sequence, seven slices are taken: the full sub-sequence, the first 10%, 25% and 50% of the sub-sequence, and the last 50%, 25% and 10% of the sub-sequence. Statistics are computed over these slices that are sensitive to various time localities in the sub-sequence, and capture trends, variability and extremes. The statistics are computed per clinical variable, without considering the binary masks. For each variable in the slice, the following eight statistics are computed: the first and last value, the minimum and maximum value, the mean and standard deviation of the variable in the slice, the skew of the variable in the slice and the length of slice. The resulting vector $\mathbf{x}$ consists of $7 \times 8 \times 14 = 784$ features, since there are 14 clinical variables. This is similar to the method adopted by Harutyunyan et al., who created vectors of 714 features per time series sub-sequence [11].

Two types of baseline data sets are created: $\mathbf{X}_{raw}$ computed over sub-sequences of the raw time series, and $\mathbf{X}_{pre-imputed}$ computed over sub-sequences of the imputed time series.

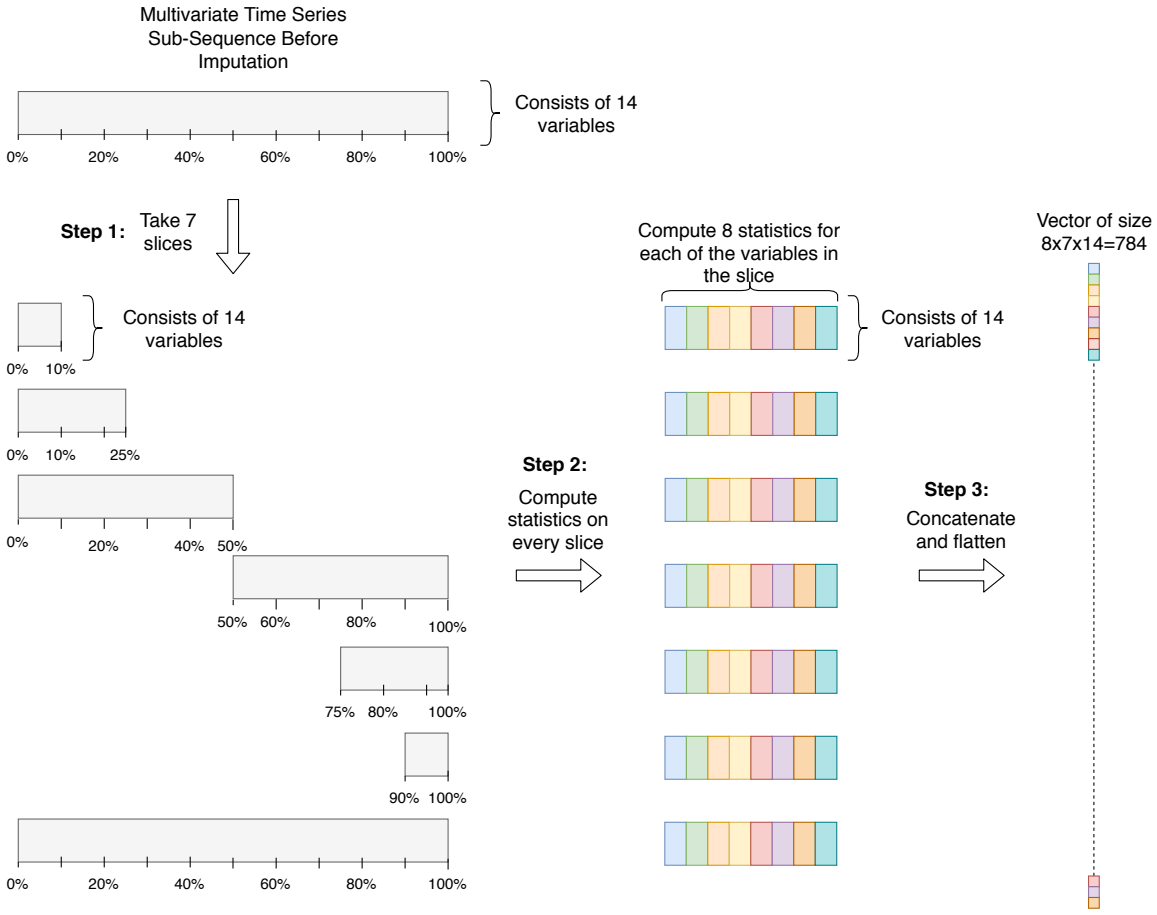

Figure 4.7: This diagram demonstrates how any given sub-sequence of a raw multivariate time series representing an individual NICU stay from the start of the stay until any hour between the fourth hour and the last hour before the patient's discharge or end of life is processed into a vector of 784 hand-engineered features for training the MLR and LR baseline models. This process consists of three steps: (1) taking seven slices that represent different time localities of the sub-sequence, (2) computing eight statistics that capture trends, variability and extremes in each of the slices, and (3) concatenating and flattening the computed statistics for each of the slices into a feature vector of size $7 \times 8 \times 14 = 784$.

In both cases the feature vectors $\mathbf{x}$ are all initialized with zeros. Some sub-sequences are so short that two or more of the slices are empty. In that case, the corresponding features in $\mathbf{x}$ remain zero. Since the raw time series are sparse, the corresponding feature vectors $\mathbf{x}_r \in \mathbf{X}_{raw}$ are sparse as well. For each $\mathbf{x}_r$, missing features are imputed with the mean value of that feature computed over all $\mathbf{x}_r$ in the training data set. The feature vectors $\mathbf{x}_p \in \mathbf{X}_{pre-imputed}$ do not require imputation since they are dense. Both $\mathbf{X}_{pre-imputed}$ and $\mathbf{X}_{raw}$ are standardized by

subtracting the mean and dividing by the standard deviation per feature, which are respectively computed from the $\mathbf{x}_r$ and $\mathbf{x}_p$ in the training data set.

The models are also compared against two naive baselines. The first always predicts the mean remaining LOS; the second always predicts the median remaining LOS. The mean and median remaining LOS are computed over the training data.

4.3 Hyper-Parameter Calibration

Due to time and resource limitations, no systematic hyper-parameter searches are performed in this thesis. Instead, the hyper-parameter configurations described in Section 5.1.2 were found during preliminary experiments. In these preliminary experiments, the hyper-parameters were initially set to values reported to perform well in related work and carefully adjusted towards models with a better bias-variance trade-off using the training and validation data sets. The hyper-parameter settings were optimized towards classification and kept constant for the regression task, with exception of the FCN and LSTM-FCN models. It was clear that the FCN and LSTM-FCN regression models required different regularization than their classification counterparts, hence, the dropout rates for these models are changed. The exact hyper-parameter configurations and the ranges of hyper-parameters that were explored are detailed for each individual model in Section 5.1.2. The intermediate calibration results of the preliminary experiments are not explicitly reported.

4.4 Evaluation

For the classification models, the classification accuracy is an unsuitable evaluation metric due to the imbalance of the target label distribution. If for a given time series the true target is bucket class 0 (i.e. a stay of less than two full days), a false prediction of class 1 (i.e. a stay between two days and a week) is less wrong than a false prediction of class 3 (i.e. a stay longer than a week). An evaluation metric that can account for the ordinality of the target classes is Cohen's linearly weighted kappa coefficient [65]. Cohen's kappa coefficient measures the level of agreement between the true and predicted target labels and inherently takes imbalances in the distribution of the target classes into account. The coefficient is a real number between -1 and 1 and is defined as:

$$\kappa = \frac{(p_o - p_e)}{(1 - p_e)},$$

where p_o is the relative observed agreement and p_e is the probability of agreement that can be expected by chance. A value of 1 means complete agreement between the predictions and true values. A value of 0 or lower means that the agreement happened by chance or that the

agreement is worse than random. Although some controversy exists around the interpretability of Cohen's kappa statistic, multiple sources cite a score of over 0.40 as an acceptable agreement for most tasks [66], [67]. The main argeement for adopting this evaluation metric is that it has been used in similar studies on classifying the remaining LOS of adult ICU patients [11], [12].

The main metric to evaluate the regression models is the mean absolute error (MAE). The MAE corresponds to the expected value of the absolute error loss, which is calculated as:

$$MAE\left(\mathbf{y}, \hat{\mathbf{y}}\right) = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i|,$$

where $\mathbf{y}$ are the true target labels, $\hat{\mathbf{y}}$ are the predicted target labels and n is the number of samples in the data set. The MAE is an intuitive metric for LOS prediction, since it explicitly communicates the average error in number of hours.

A statistical test framework is required to test whether advanced deep learning models based on RNNs and FCNs fit on multivariate time series can outperform MLR and LR baselines in a statistically significant way. The same is true for testing the importance of the GA and binary mask indicator variables. The Friedman test with a significance level of 5% is used to compare the performance of the different models [68], [69]. The Friedman test is a non-parametric statistical test that can be used to detect differences in model performance across multiple test set attempts [70]. It is a null hypothesis test that indicates whether any of k models consistently performs better than the others. Given the limited scope of this thesis, it is not possible to refit each of the models more than once on the training and test set. To be able to obtain multiple test samples per model, the test set is divided into 20 non-overlapping partitions instead. Each partition contains the multivariate time series sub-sequences corresponding to the NICU stays of a given set of NICU patients. For a given patient, all sub-sequences of the multivariate time series associated with their stay fall in exactly one partition. The test set comprises of 822 NICU stays, hence, 18 partitions contain the sub-sequences of 41 stays and two contain the sub-sequences of 42 stays. The stays are randomly distributed over the 20 test partitions and because the stays vary in length some of the partitions contain more sub-sequences than others. If the Friedman test's null hypothesis is rejected, it is established that a significant statistical difference exists within the performance of the various models. If this is the case, a pairwise post-hoc analysis is carried out using the Wilcoxon signed-rank test [71]. The results of the post-hoc tests are visually represented with critical difference diagrams [70], [72]. See Figure 4.8 for an example.

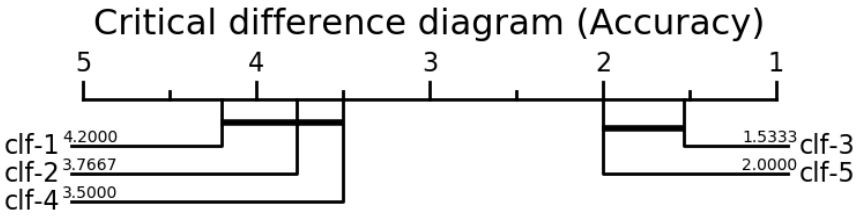

Figure 4.8: Example critical difference diagram comparing the results of a fictitious post-hoc analysis on the classification accuracy of five imaginary classifiers. The average ranks of the classifiers obtained from the post-hoc analysis are plotted on the main axis. A lower rank signifies a better classifier in terms of classification accuracy. A thick horizontal line connecting two or more classifiers means that there is no significant difference between their performance. In this example, the classification accuracy of classifiers 3 and 5 is significantly better than the classification accuracy of classifiers 1, 2 and 4. Example taken from Fawaz et al. [70].

Chapter 5

Experiments and Results

5.1 Experimental Set-Up

5.1.1 Objective

To recapitulate, this thesis aims to investigate how various RNN and FCN architectures trained on multivariate time series compare to MLR and LR models trained on hand-engineered features for the task of predicting the remaining time that a patient will spend in the NICU at each hour in the stay, after the first four hours of the stay. In addition, this thesis assesses the amount of extra information that can be extracted from the GA variable and binary mask variables when predicting the remaining LOS of NICU patients. Accurately predicting the remaining LOS of newborns in the NICU is important for scheduling and resource allocation in the hospital, and can be used as an aid in parent counselling. This thesis addresses this task mainly as a classification problem in which the models predict one of three classes: shorter than two days, between two days and a week, and longer than a week. Regression models are also constructed, but they are secondary to the classification task; it is expected that they are of much less practical value [11], [12]. The model architectures for the regression task are similar to those for classification. The only difference is that the regression models use the raw output of the last layer to make predictions, whereas the classification models first pass the output through a softmax activation function.

5.1.2 Models

This thesis explores 12 different models for predicting the LOS of NICU patients (see Table 5.1). The labels produced by the coarse strategy introduced in Chapter 4 are referred to as coarse targets $\tau_{coarse} \in \{1, 2, 3\}$. For regression, the targets are referred to as $\tau_{regression}$, which are positive integer values (i.e. each stay is rounded to the closest number of hours). The

multivariate time series for each NICU stay can be seen as 14 pairs $(\{v_t^{(i)}\}_{t\geq 1}^{T}, \{m_t^{(i)}\}_{t\geq 1}^{T})$, where $v_t^{(i)}$ is the normalized recorded or imputed value of variable i at timestamp t (i.e. the t^{th} hour of the stay), $m_t^{(i)}$ is the corresponding binary mask value at that timestamp, and T is the last timestamp in the series. The inputs of the deep learning models are sub-sequences of the multivariate time series and their corresponding target labels. These sub-sequences consist of the concatenation of all $(\{v_t^{(i)}\}_{t\geq 1}^{\tau}, \{m_t^{(i)}\}_{t\geq 1}^{\tau})$, denoted $\{\mathbf{x}_t\}_{t\geq 1}^{\tau}$ and the corresponding target label is one of τ_{coarse} or $\tau_{regression}$. Note that $4 \leq \tau \leq T$, i.e. any hour between the fourth hour of the NICU stay and the hour that marks the end of the stay.

Model	Classification	Regression
LR ($\mathbf{X}_{pre-imputed}$)	✗	✓
LR ($\mathbf{X}_{raw}$)	✗	✓
MLR ($\mathbf{X}_{pre-imputed}$)	✓	✗
MLR ($\mathbf{X}_{raw}$)	✓	✗
LSTM	✓	✓
LSTM w/o mask variables	✓	✓
LSTM w/o gestational age variable	✓	✓
GRU	✓	✓
Channel-wise LSTM	✓	✓
Channel-wise GRU	✓	✓
FCN	✓	✓
LSTM-FCN	✓	✓

Table 5.1: The machine learning models applied to the task of predicting the remaining LOS of NICU patients. MLR and LR serve as baselines for classification and regression, respectively.

In this thesis, there is no clear notion of an epoch since for each NICU stay in the training data set, all its possible sub-sequences $\{\mathbf{x}_t\}_{t\geq 1}^{\tau}$, where $4 \leq \tau \leq T$, are considered as individual training points. The 2,632 NICU stays in the training data set comprise of 1,048,583 such sub-sequences; there are 258,975 and 322,220 such sequences for the validation and test data, respectively. Although each sub-sequence is passed to the network as an individual data example when training, it should be noted that the sub-sequences associated with a single NICU stay are all subsets or supersets of each other. This requires a different training scheme; passing all data at once makes it hard to validate the model on the validation set. Rather than using training epochs, the models are trained in "rounds". Per training round, a fixed amount of mini-batches is randomly sampled from all sub-sequences in the training data set and used for training. After each training round, the model is validated on a fixed number of validation examples. This amount of training and validation samples is carefully chosen, such that the

training and validation loss curves are smooth and reliable enough to observe when the model starts overfitting while preventing the validation procedure from becoming an undesirable bottleneck concerning overall training time. After experimenting with various settings, it was confirmed that sampling 2,048 training mini-batches of size 8 and 4,096 validation mini-batches of the same size yields a stable learning process and an acceptable training speed for the FCN, LSTM-FCN and standard GRU and LSTM networks. The channel-wise LSTM and channel-wise GRU, which is explained in more detail later in this section, use mini-batches containing 16 examples. Per training round, they process 1,024 training and 2,048 validation batches such that each model samples the same amount of training and validation data observations per training round.

In Section 4.1.2 it was communicated that the deep learning networks constructed in this thesis are trained in mini-batches that expect all their time series sub-sequences to be of equal length. It has been explained how the multivariate time series sub-sequences $\{\mathbf{x}_t\}_{t\geq 1}^{\tau}$ vary in length, but can be zero-padded to create mini-batches in which all sub-sequences are of equal length. The padded zeroes can subsequently be masked out to ensure that deep learning models do not consider them as input features. The amount of zero-padding can be minimalized by efficiently sorting the $\{\mathbf{x}_t\}_{t\geq 1}^{\tau}$ before each training round. This speeds up the training procedure and reduces the the amount of memory required for carrying out the computations. It is intractable to read and sort all training and validation data examples into random access memory (RAM), due to their size and quantity. To overcome this issue, the original 4,111 multivariate time series – which vary in length and each represent a complete, single NICU stay – are sorted by the total LOS in hours and partitioned into 100 buckets of equal size. See Figure 5.1 for a conceptual diagram of this sorting procedure. In this way, each bucket contains time series of a length that fall within a unique range. For each of the buckets, the length of the longest time series that it contains constitutes the upper boundary that separates it from any buckets containing longer time series, thus there are 100 unique boundaries in total. At the start of each training round, the training data (i.e. all possible multivariate time series sub-sequences of at least four hours long) are shuffled, and the training generator is passed the list of bucket boundaries and the requested mini-batch size. The training generator iteratively reads the randomly shuffled training examples and places them in buckets corresponding to the provided bucket boundaries. Each time that a bucket contains a number of data examples (i.e. multivariate time series sub-sequences representing part of a NICU stay) that is equal to the mini-batch size, it emits the mini-batch of training examples in that bucket to the main training loop and empties the bucket. Upon emitting a mini-batch, all training examples in the mini-batch are zero-padded to the longest example in the mini-batch such that all examples in the mini-batch are of the same dimensions. This bucketing procedure is memory-efficient and ensures that emitting a mini-batch from any bucket is equally likely, thus respecting the target distribution of the data. The choice of 100 buckets is somewhat arbitrary, but it ensures that each bucket contains many more samples for both the training set and the validation set than

the mini-batch sizes used in this thesis (i.e. 8 and 16), while leading to limited zero-padding. At the end of each training round, a similar procedure is carried out with the validation data generator.

All deep learning models are trained using the Adam optimizer with an initial learning rate of 0.001, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-7}$ [73], except for the channel-wise LSTM and GRU for regression, which pass a learning rate scheduler to the Adam optimizer to speed up learning during the first couple of training rounds. The schedule sets a learning rate of 0.01 for the first two rounds, a learning rate of 0.005 for the next three rounds, and a learning rate of 0.001 for all subsequent rounds. For classification and regression, the objective loss functions to be minimized are the categorical cross-entropy and the MAE, respectively.

The Scikit-learn Python library is used to build the LR and MLR baseline models [74]; TensorFlow 2.0 is used for building the deep learning architectures [75]. The models are trained on a machine with 126 GB RAM, an Intel(R) Core(TM) i7-6950X CPU @ 3.00GHz and four NVIDIA GeForce RTX 2080 GPUs, although all models can be trained using a single NVIDIA GeForce RTX 2080 GPU with 8 GiB of GPU memory.

Linear Regression

Two LR models are trained. The first uses baseline feature vectors $\mathbf{X}_{raw}$ that are extracted from the raw time series; the second uses feature vectors $\mathbf{X}_{pre-imputed}$ that are derived from the pre-imputed time series (see section 4.2). The objective of the LR model is to fit a linear model with coefficients $\mathbf{w} = (w_1, \ldots, w_p)$ that minimizes the residual sum of squares between the true target labels and the targets predicted by the linear approximation. Other than the number of input features, this model does not have any hyper-parameters to tune, hence, the model is trained on the training and validation combined.

Multinomial Logistic Regression

Logistic regression is a linear model for classification that, in the simplest case, uses the sigmoid function to model the probabilities describing the possible outcome of a binary classification problem. To classify the remaining LOS, a generalization of logistic regression called multinomial logistic regression (MLR) is applied instead. MLR uses the softmax function to minimize the multinomial loss across the probability distribution of all target classes.

Two MLR models are trained to predict the coarse targets τ_{coarse}, one using the raw features $\mathbf{x}_r \in \mathbf{X}_{raw}$ and the other using the pre-imputed features $\mathbf{x}_p \in \mathbf{X}_{pre-imputed}$. For each model, a grid search is performed to discover the optimal hyper-parameter settings. The hyper-parameters for MLR are the regularizer λ and the C parameter that dictates the inverse of the regularization strength. Twelve hyper-parameter configurations are explored, namely all possible pairs of $\lambda = \ell_1$-norm or $\lambda = \ell_2$-norm with one of six distinct values of C (i.e. 1.0, 0.1, 0.01, 0.001,

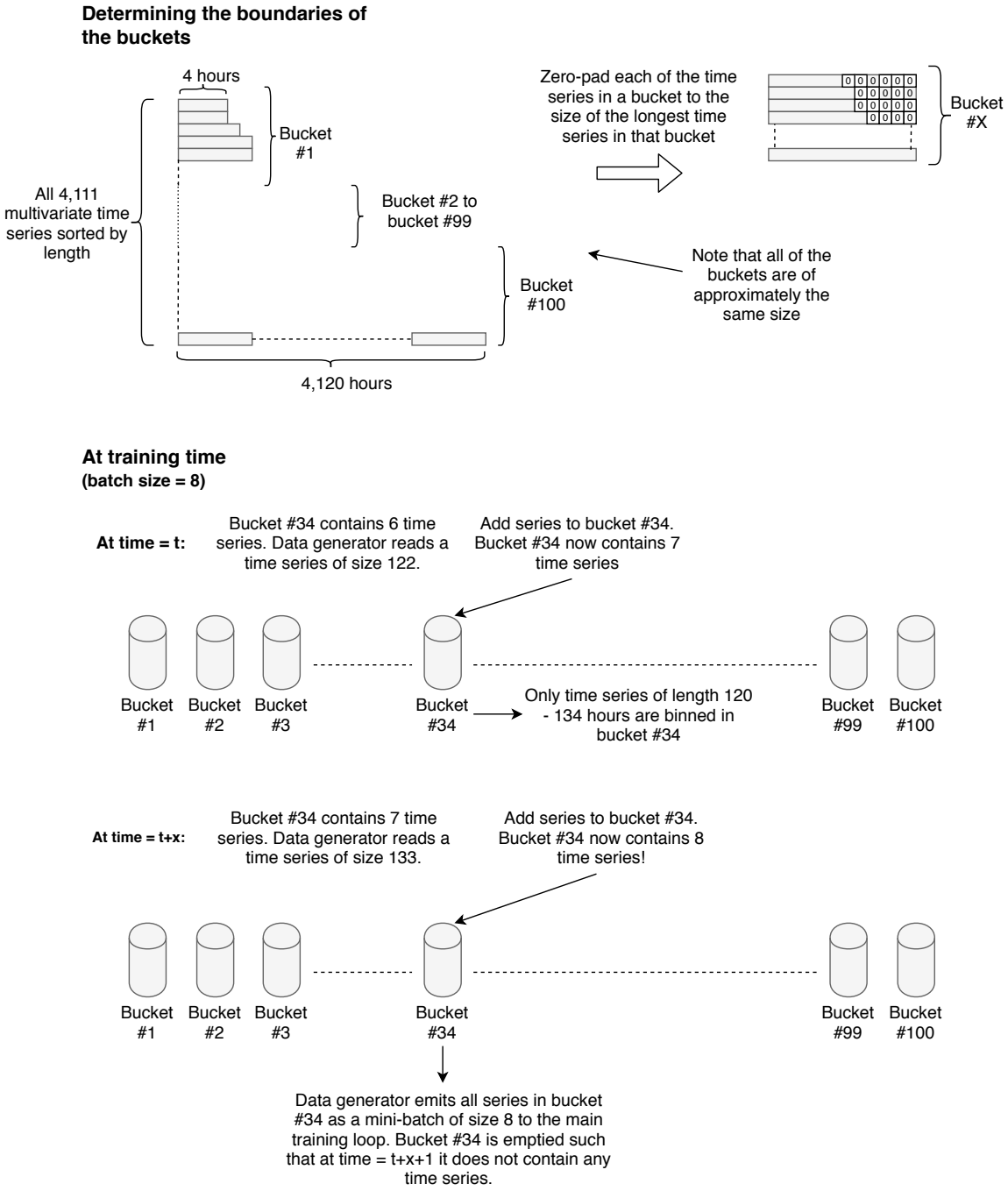

Figure 5.1: Conceptual diagram of the bucketing procedure that ensures memory-efficient reading and emission of time series sub-sequences of similar length to the main training loop. Assume that the boundaries of bucket #34 are 120 and 134, i.e. all time series sub-sequences of length between 120 and 134 hours are binned in bucket #34. Furthermore, assume that the data generator does not read any time series of length between 120 and 134 between time step t and time step $t + x$. Each of the time series sub-sequences in a bucket are zero-padded to the size of the maximum sequence length that the bucket accepts, i.e. the value of the bucket's upper boundary. All training examples are randomly shuffled before each training round, which ensures that each bucket is as likely to emit a mini-batch at any point during the training phase.

0.0001, 0.00001). Note that smaller values of C indicate stronger regularization and that these grid searches mainly serve to mitigate model overfitting. The SAGA solver is used to minimize the cross-entropy loss over a maximum of 100 iterations [76]. For both MLR models, the optimal regularizer is the ℓ_2-norm with a C-value of 0.0001.

LSTMs and GRUs

Due to their similar internal dynamics, LSTM cells can be replaced by GRU cells (and vice versa) without having to make other changes to the deep learning architecture in which they are used. Two types of RNN architecture are considered, both with an LSTM variant and a GRU variant.

The first architecture, which is referred to as the standard LSTM or standard GRU architecture, consists of a single LSTM or GRU cell and is visualized in Figure 5.2. Dropout is applied to the output of the cell to mitigate overfitting. Preliminary experiments were carried out with architectures containing one or two LSTM/GRU cells of 32, 64 and 128 hidden units regularized with with dropout rates in the $0.0 - 0.5$ range. It was found that for both the LSTM and GRU architectures a single RNN cell of 64 hidden units with a dropout rate of 0.3 applied to its linear transformation of the input, followed by a 0.3 dropout rate applied to the output leads to a good bias-variance trade-off.

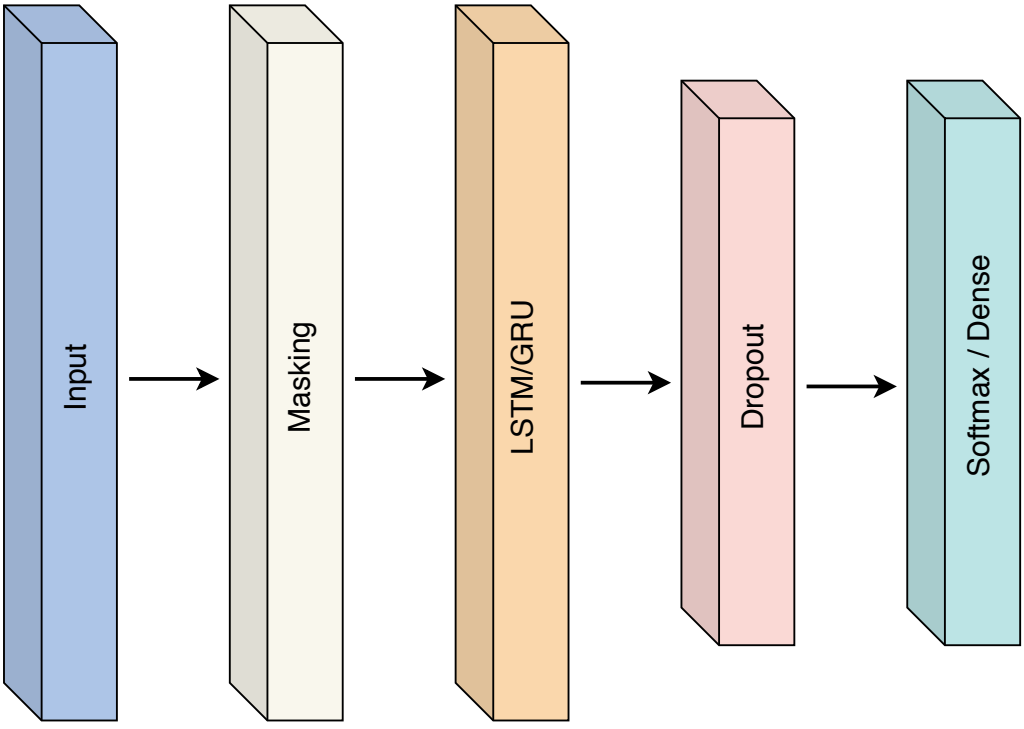

Figure 5.2: The standard RNN architecture with a single LSTM or GRU cell.

The second type of LSTM and GRU architecture is similar to what Harutyunyan et al. dubbed the channel-wise RNN [11]. See Figure 5.3 for a schematic of this architecture. A channel is defined as a pair of time series of length τ for a specific variable i and its corresponding binary mask variable: $(\{v_t^{(i)}\}_{t\geq 1}^{\tau}, \{m_t^{(i)}\}_{t\geq 1}^{\tau})$. Each such channel in the input $\{\mathbf{x}_t\}_{t\geq 1}^{\tau}$ is individually pre-processed by a single bidirectional LSTM or GRU cell. The intuition behind this pre-processing is that the model explicitly learns to associate the correct pattern of missing measurements for each variable. Moreover, by pre-processing each channel separately, the model gets the chance to extract useful features from each variable in an independent manner. The output of each of the bidirectional LSTM or GRU cells is concatenated and sent to a second (unidirectional) LSTM or GRU cell. In preliminary experiments models with bidirectional LSTMs/GRUs with 8, 16 and 32 hidden units were tested in combination with a final unidirectional LSTM/GRU cell with 64 or 128 units. To regularize the models, various combinations of dropout with rates in the range $0.0 - 0.4$ were explored. The results of these experiments indicated that the following hyper-parameters yield a model of proper bias-variance trade-off: 16 hidden units in the bidirectional LSTMs/GRUs with no internal dropout, a final unidirectional LSTM/GRU cell with 64 units and a dropout rate of 0.2 for the linear transformations of the input and a dropout rate of 0.2 for the output.

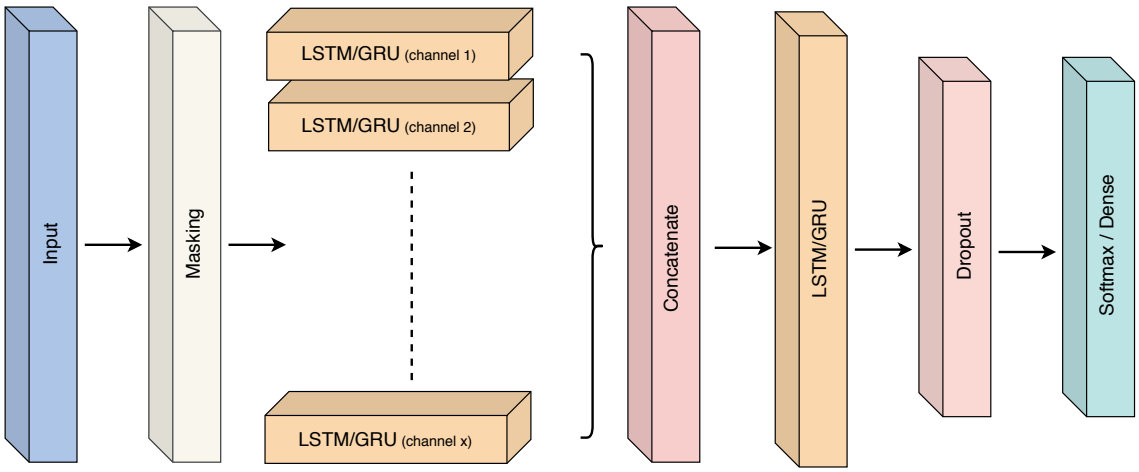

Figure 5.3: The channel-wise architecture with LSTM or GRU cells and x input channels.

Due to limited time and resources, not all variable combinations can be tested. Therefore, a full ablation study cannot be carried out. This thesis limits the ablation to the assessment of the importance of the binary mask variables and the GA variable, because of its prominence in the literature [5], [61]. By virtue of its relatively fast training time and promising results, the standard LSTM is used to carry out this ablation experiment. The LSTM model that performs best on the complete training data is refitted twice: once on the training data set without the

binary mask variables, and once on all training data except the GA variable.

FCNs and LSTM-FCNs

The FCN for univariate time series classification proposed by Wang et al. [29] can easily be extended to multivariate time series classification and regression. Preliminary experiments with the FCN and LSTM-FCN architectures as proposed by Wang et al. [29] (see Figure 2.3) and Karim et al. [14] (see Figure 2.4) showed that they are not suitable for the prediction of remaining LOS of newborns in the NICU with the training set-up of this thesis. The main issue is that the models overfit on the training data too quickly. The overfitting problem is partially resolved by making a small modification to the network architectures, namely by applying dropout after the ReLU activation and batch normalization in the first two convolutional blocks. An additional change is that the order of batch normalization and ReLU activation is swapped, since exploratory experiments indicated that performing ReLU activation before batch normalization leads to better results. Another approach would be to simplify the model architectures, for example by removing layers or by reducing the size of the convolutional filters. Due to the limited scope of this thesis, such experiments are left to explore in future studies. Like the other deep learning models, for both the FCN and LSTM-FCN a mask is computed over the input such that subsequent layers ignore the zero-padding. SE blocks are added to the FCN architecture due to their proven success in the experiments conducted by Karim et al. [32]. See Figures 5.4 and 5.5 for the final FCN and LSTM-FCN architectures used for the experiments.

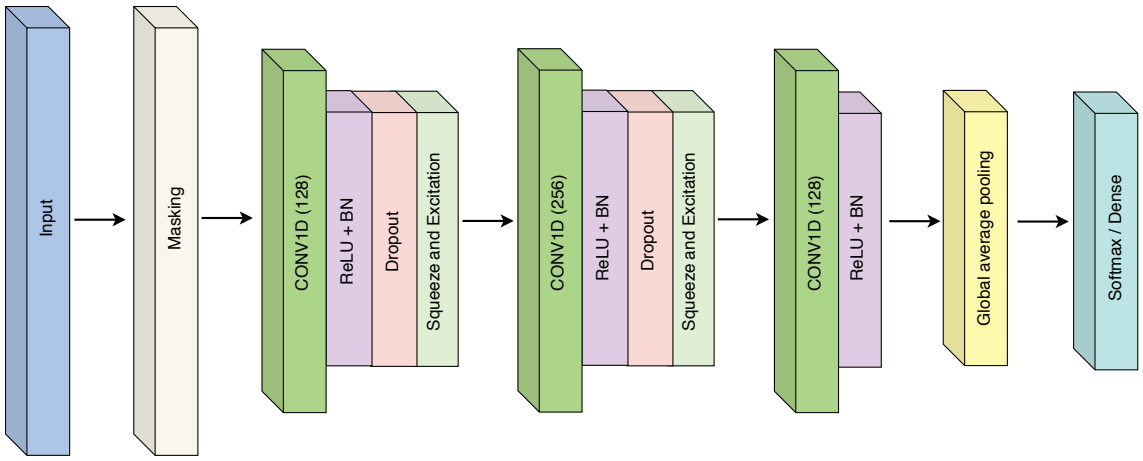

Figure 5.4: The FCN architecture with extra dropout layers.

Dropout is only applied to the last dimensions of the convolutional output (i.e. the dimension corresponding to the filter size of the convolutional layer) since the other dimensions vary per batch. During preliminary experiments, dropout rates in the $0.0 - 0.8$ range were investigated.

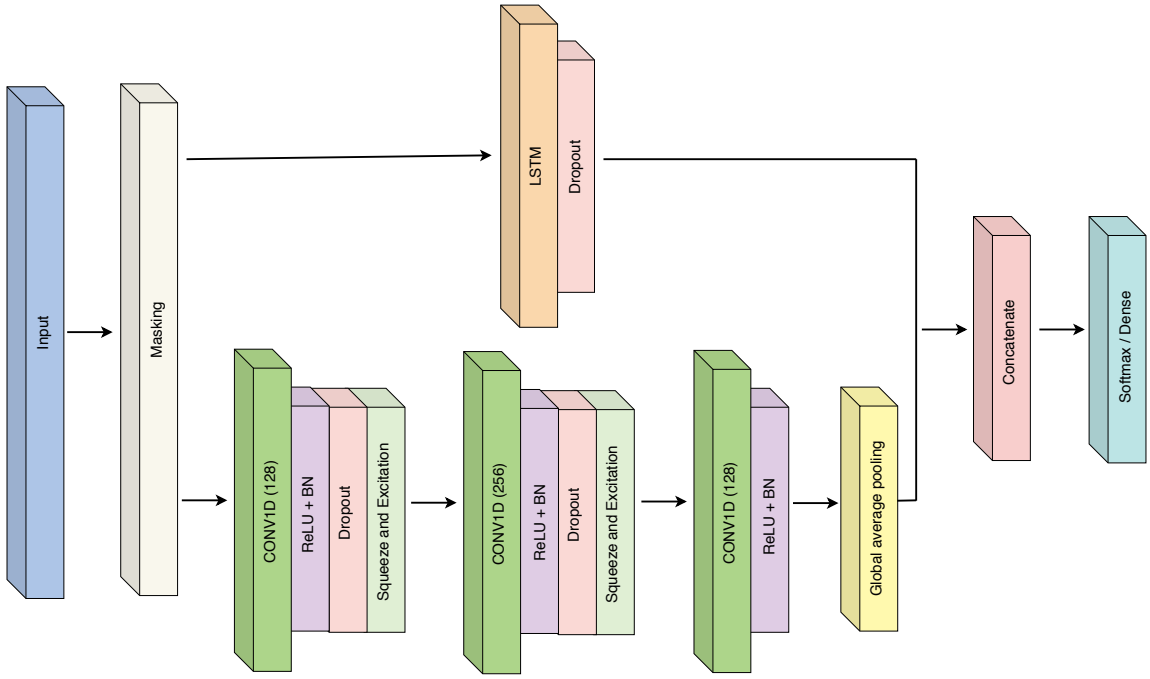

Figure 5.5: The LSTM-FCN architecture with extra dropout layers.

It was found that the following dropout rates are adequate for regularizing the FCN: 0.3 for the classification task and 0.7 for the regression task. The experiments demonstrated that applying a dropout rate of 0.8 to the output of the LSTM module for both classification and regression in the LSTM-FCN model and a dropout rate of 0.5 in the FCN module for classification and 0.3 for regression leads to a model that performs well. Moreover, the preliminary experiments with the LSTM-FCN architecture showed that an LSTM module with 16 hidden units performed better than an LSTM module with 8 hidden units. The remaining hyper-parameter settings are set to the values proposed by Karim et al. [14]. That is to say, the convolutional layers have 128, 256 and 128 filters and kernels of size 8, 5, and 3, respectively, and the r parameter in the SE blocks is set to 16.

5.2 Results

5.2.1 Performance Analysis

The results of the classification and regression experiments are displayed in Table 5.2. These results are obtained through training the models until the training round that yields the optimal validation loss with the hyper-parameter configurations and training set-ups as described in the previous section. Each model is evaluated independently on a hold-out test set. In Table 5.2,

the mean, 2.5 percentile and 97.5 percentile of a bootstrapping procedure in which the test set is resampled 1,000 times are reported. Each bootstrap sample is of the same size as the original test set.

Model	Kappa coefficient	MAE (in hours)
Naive baseline (median)	0.0 (0.0, 0.0)	453 (451, 454)
Naive baseline (mean)	0.0 (0.0, 0.0)	490 (489, 492)
LR ($\mathbf{x}_{pre-imputed}$)	n.a.	321 (320, 322)
LR ($\mathbf{x}_{raw}$)	n.a.	312 (311, 313)
MLR ($\mathbf{x}_{pre-imputed}$)	0.298 (0.295, 0.302)	n.a.
MLR ($\mathbf{x}_{raw}$)	0.297 (0.293, 0.300)	n.a
Standard LSTM	0.361 (0.358, 0.364)	305 (304, 306)
Standard GRU	0.324 (0.321, 0.327)	319 (319, 321)
Standard LSTM w/o mask variables	0.264 (0.261, 0.267)	316 (315, 317)
Standard LSTM w/o gestational age variable	0.303 (0.300, 0.306)	314 (313, 315)
Channel-wise LSTM	**0.371** (0.366, 0.376)	315 (314, 317)
Channel-wise GRU	0.331 (0.327, 0.335)	316 (314, 317)
FCN	0.308 (0.305, 0.311)	308 (307, 309)
LSTM-FCN	0.316 (0.313, 0.320)	**304** (303, 306)

Table 5.2: Results for classification and regression. For all models the mean score and the 2.5 and 97.5 percentiles are reported by resampling the test set 1,000 times with replacement. The size of each of the samples is equal to the size of the original test set. Accordingly, a given sample contains duplicate observations while other observations are not present in the sample at all. The bold-faced scores are the highest reported on the task. Note that a higher kappa coefficient or a lower MAE means better model performance.

Naive Baselines

As expected, naively predicting the median or mean class for classification or the median or mean value for regression leads to poor results. Therefore, they are not analyzed within the statistical framework introduced in Section 4.4. They only serve to underscore the difficulty of the task: the range of possible values is large, and the data set has a large variance.

Classification

Performing the Friedman test on all classifiers with respect to Cohen's kappa coefficient yields a rejection of the null hypothesis with a significance threshold of 5% (p-value $= 5.24 \times 10^{-20}$). After performing a post-hoc analysis of the results of the Friemand test with the Wilcoxon signed-rank test with a significance threshold of 5%, the following claims can be made. Refer to

Figure 5.6 for the critical difference diagram of the post-hoc analysis and to Table 5.3 for the p-values for all pair-wise Wilcoxon signed-rank tests. Within the proposed statistical framework, all models trained on the complete multivariate time series significantly outperform the MLR baselines on the classification task. There is no significant difference between the MLRs trained on the raw (i.e. $\mathbf{X}_{raw}$) and pre-imputed (i.e. $\mathbf{X}_{pre-imputed}$) baseline features. From Table 5.2 it can be observed that the channel-wise LSTM performed best on the classification task. With exception of the standard LSTM, the channel-wise LSTM is significantly better than all other models. No significant difference in performance is observed between the standard GRU, channel-wise GRU and LSTM-FCN models, though they are significantly better than the FCN classifier. Another observation is that the LSTM trained on the complete set of selected clinical variables is significantly better than the LSTMs trained without using the GA variable or the binary indicator mask variables. Training without masks is significantly worse than training without the GA variable. Without the mask variables, the LSTM did not manage to significantly outperform the MLR baseline models.

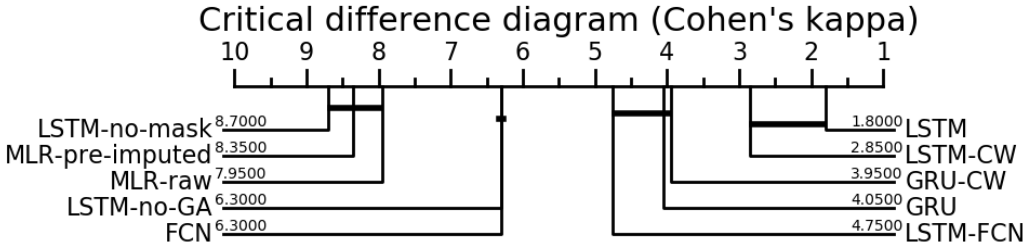

Figure 5.6: Critical difference diagram comparing the results of the post-hoc analysis of all classifiers with respect to their Cohen's kappa coefficient. A smaller rank means better performance.

Regression

The Friedman test with a significance threshold of 5% also rejects the null hypothesis when performed on the results of the regression models (p-value $= \ 6.81 \times 10^{-5}$). The critical difference diagram corresponding to the post-hoc analysis of this test is shown in Figure 5.7. Note, however, that for the regression models a higher rank signifies better performance. This is because of the fact that a smaller MAE means better performance. For the regression models, the p-values of the pair-wise Wilcoxon signed-rank test with a significance threshold of 5% are displayed in Table 5.4. The LSTM-FCN is the regression model with the lowest MAE reported in Table 5.2. However, the post-hoc analysis of the Friedman test for the regression models indicates that there is no significant difference between the LR baselines and the best deep learning models. According to the pair-wise Wilcoxon signed-rank test, the standard LSTM, FCN and LR trained on $\mathbf{X}_{raw}$ all have similar performance. The difference in performance

	LSTM	LSTM no GA	LSTM no mask	LSTM channel-wise	GRU	GRU channel-wise	FCN	LSTM-FCN	MLR pre-imputed
LSTM no GA	8.86E-05								
LSTM no mask	8.86E-05	1.03E-04							
LSTM channel-wise	5.50E-01	3.38E-04	8.86E-05						
GRU	2.54E-04	5.93E-04	1.03E-04	5.73E-03					
GRU channel-wise	8.86E-05	1.94E-03	8.86E-05	3.04E-02	5.26E-01				
FCN	8.86E-05	8.81E-01	2.54E-04	5.93E-04	3.19E-03	5.93E-04			
LSTM-FCN	1.03E-04	1.69E-02	1.03E-04	3.59E-03	3.91E-01	7.93E-02	1.37E-02		
MLR pre-imputed	1.63E-04	5.11E-03	6.27E-01	1.40E-04	1.16E-03	6.81E-04	4.05E-03	6.81E-04	
MLR raw	1.20E-04	4.55E-03	6.01E-01	1.03E-04	4.49E-04	5.17E-04	1.37E-02	5.17E-04	8.23E-01

Table 5.3: Pair-wise Wilcoxon signed-rank test comparison (p-values) of all classification models. Red cells denote that the null hypothesis (based on a significance threshold of 5%) cannot be rejected. For these models, the claim is made that they have similar performance in terms of Cohen's kappa coefficient.

between the LR trained on $\mathbf{X}_{pre-imputed}$ versus $\mathbf{X}_{raw}$ is significant; the LR trained on $\mathbf{X}_{raw}$ is significantly better. Furthermore, the analysis yields a significant performance deterioration when the LSTM is trained without employing the GA variable or the the binary mask indicator variables.

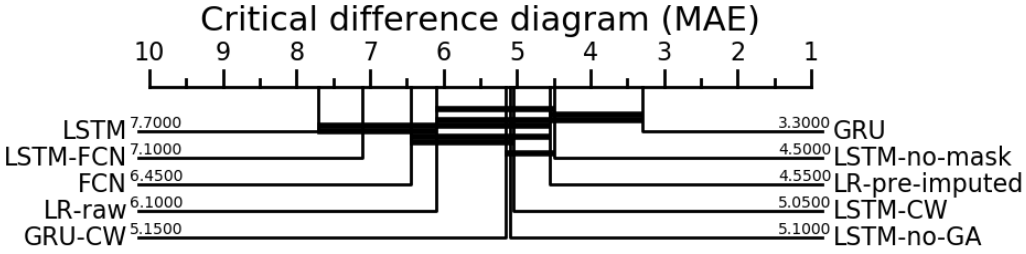

Figure 5.7: Critical difference diagram comparing the results of the post-hoc analysis of all regression models with respect to their MAE. Note that in this case a higher rank means better performance, since the goal is to minimize the MAE.

	LSTM	LSTM no GA	LSTM no mask	LSTM channel-wise	GRU	GRU channel-wise	FCN	LSTM-FCN	LR pre-imputed
LSTM no GA	1.63E-04								
LSTM no mask	1.40E-04	2.79E-01							
LSTM channel-wise	1.03E-04	9.70E-01	4.78E-01						
GRU	8.86E-05	4.55E-04	2.06E-02	2.51E-02					
GRU channel-wise	1.63E-04	6.54E-01	3.13E-01	6.54E-01	4.38E-02				
FCN	8.52E-01	1.00E-01	2.76E-02	6.74E-02	2.82E-03	7.31E-2			
LSTM-FCN	5.75E-01	3.19E-02	4.49E-04	4.55E-02	1.89E-04	3.59E-02	1.79E-01		
LR pre-imputed	3.70E-01	7.37E-01	7.94E-01	7.94E-01	7.65E-01	7.94E-01	4.33E-01	3.70E-01	
LR raw	8.23E-01	6.01E-01	5.26E-01	6.54E-01	3.51E-01	6.01E-01	9.40E-01	7.09E-01	2.06E-02

Table 5.4: Pair-wise Wilcoxon signed-rank test comparison (p-values) of all regression models. Red cells denote that the null hypothesis (based on a significance threshold of 5%) cannot be rejected. For these models, the claim is made that they have similar performance in terms of the MAE.

5.2.2 Training Dynamics

It can be observed in Figures 5.8, 5.9 and 5.10 that the training dynamics for both the standard and channel-wise LSTM and GRU architectures are similar for both the classification and the regression tasks. The channel-wise GRU and LSTM models take longer to train than the standard GRU and LSTM. This is because the channel-wise architectures contain many more parameters. When trained on all clinical variables, the standard GRU and LSTM have approximately 18 thousand and 24 thousand learnable parameters, respectively. In contrast, the channel-wise GRU and channel-wise LSTM respectively have around 65.5 thousand and 84 thousand learnable parameters.

In spite of the measures taken to improve the generalizability of the FCN and LSTM-FCN models, their training process was still more irregular than the training process of the RNN models (see Figure 5.11).

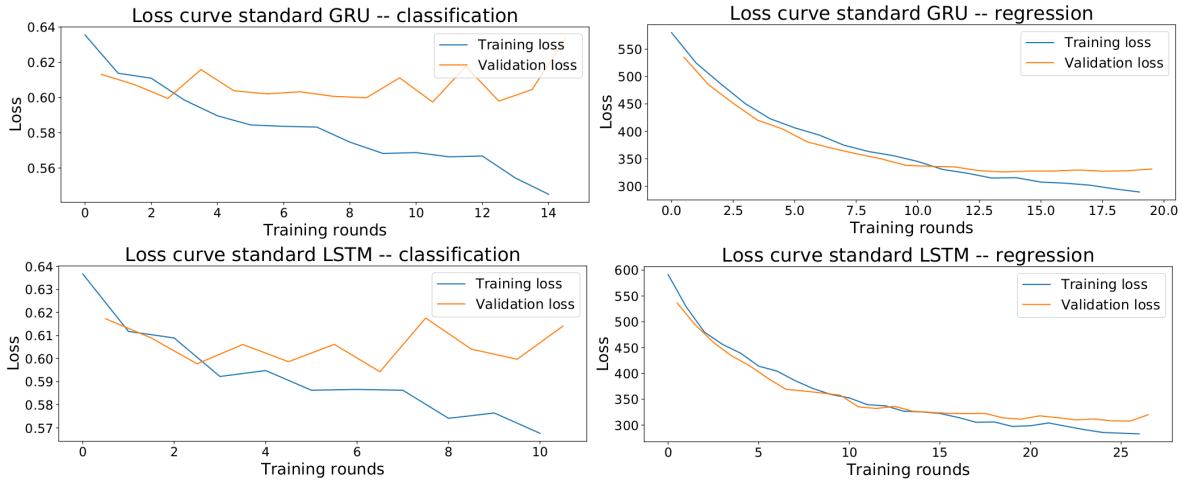

Figure 5.8: The training and validation loss curves for classification (top-left) and regression (top-right) with the standard GRU, and for classification (bottom-left) and regression (bottom-right) with the standard LSTM.

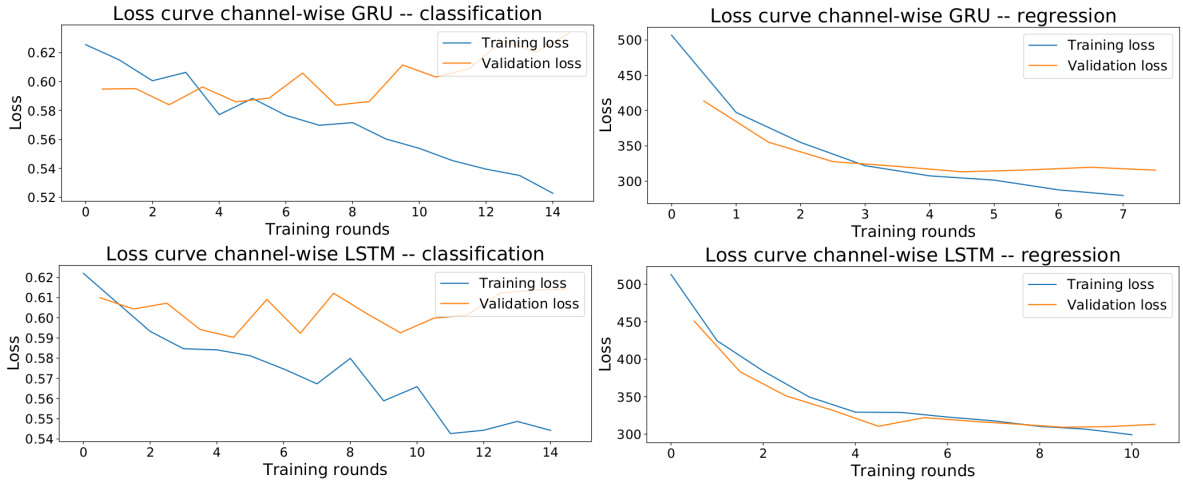

Figure 5.9: The training and validation loss curves for classification (top-left) and regression (top-right) with the channel-wise GRU, and for classification (bottom-left) and regression (bottom-right) with the channel-wise LSTM.

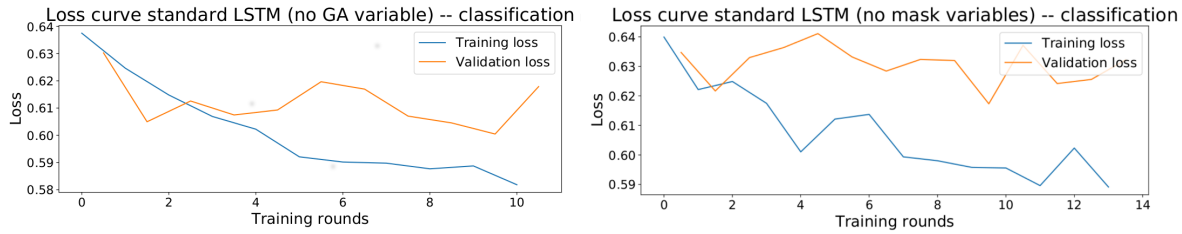

Figure 5.10: The training and validation loss curves for classification with the standard LSTM model trained on all variables except for the GA (left) and trained on all variables except for the the mask variables (right).

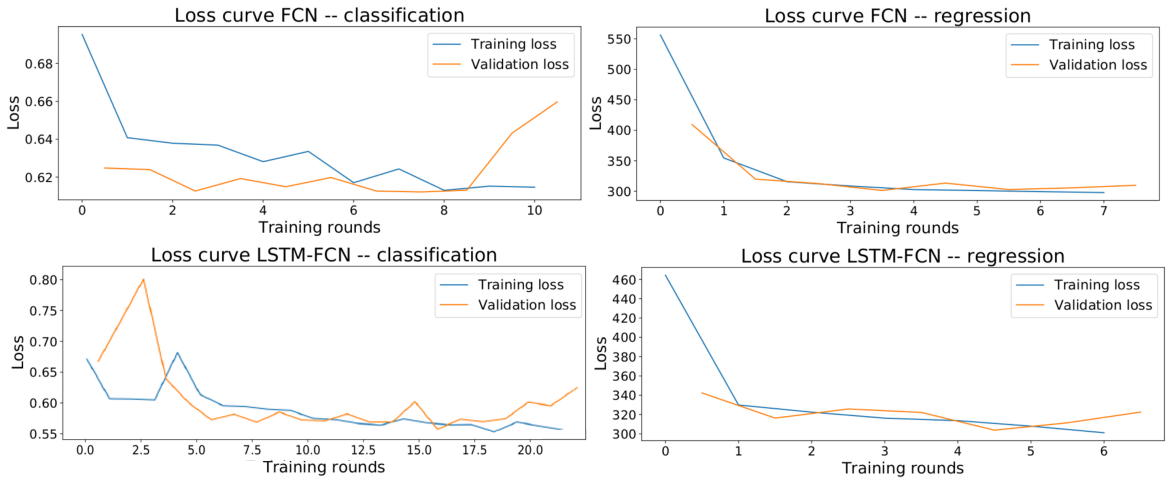

Figure 5.11: The training and validation loss curves for classification (top-left) and regression (top-right) with the FCN, and for classification (bottom-left) and regression (bottom-right) with the LSTM-FCN.

Chapter 6

Discussion

6.1 Key Findings

6.1.1 Outperforming the Baseline

The standard and channel-wise LSTMs are the best performing models for classifying the remaining LOS. This is in line with the results of Harutyunyan et al. for predicting the remaining LOS for adult ICU patients [11], who also tested a standard LSTM and channel-wise LSTM architecture. Harutyunyan et al. reported linearly weighted kappa scores in the 0.43 - 0.45 range for their best models for predicting the LOS of adult ICU patients in the MIMIC-III database [11]. The fact that these scores are higher than the ones obtained in this thesis can most likely be attributed to the MIMIC-III database containing many more data on adult ICU stays (53,432) versus NICU stays (8,100).

Although performing worse than their LSTM counterparts, the standard GRU and channel-wise GRU models also significantly outperformed the MLR baselines. The relative superiority of the LSTM-based models may be due to the fact that the LSTM cell is more complex than the GRU cell, which enables it to be more sensitive to subtle patterns in the clinical data.

The LSTM-FCN also outperformed the MLR baselines, though the gap between the best performing LSTM-FCN and the best channel-wise and standard LSTM models for the classification task is large. In comparison to MLR and RNNs such as the GRU and LSTM cell, the FCN and LSTM-FCN are relatively new architectures. As explained in section 5.1.2, several changes had to be made to the LSTM-FCN architecture proposed by Karim et al. [14] to make it perform well on the classification task. To mitigate the tendency of the FCN and LSTM-FCN to overfit, this thesis used large dropout rates. Other regularization strategies such as decreasing the number of convolutional layers or the number of filters or kernels per convolutional layer have not been explored in this thesis because of scope limitations. Karim et al. reported state-of-the-art results on the UCR benchmark with their LSTM-FCN architecture

[14]. This suggests that the multivariate time series extracted from clinical data could be fundamental different in comparison to the multivariate time series in the UCR benchmark. The main differences seem to be that the clinical time series used in this thesis are discrete rather than continuous, and that they contain a substantial amount of imputed values. This thesis recommends exploring different regularization strategies in future research than the ones adopted in this thesis, since it is unclear how much performance improvement is still to be gained.

The results seem to indicate that a combination of a FCN module and a LSTM module yields better results, though no strictly significant improvement was observed. It may be that FCNs and RNNs distill different features from their input data; perhaps they complement each other because they operate on different time scales. This study has not tried to explore many different configurations of the LSTM module in the LSTM-FCN architecture, but investigating various configurations may have some potential. In the current setup, the LSTM module only contains 16 hidden units, whereas the standard LSTM and standard GRU models contain 64 hidden units. Furthermore, the tendency of the FCN module to overfit may also signify that it has a greater ability to learn complex patterns, which may only become evident when training on a larger data set.

As previously stated, regression is considered as a secondary task in this thesis. The models are not optimized towards regression, hence, there may be much room for improvement of the regression results. The best score reported in Table 5.2 is a mean MAE of 304, produced by the LSTM-FCN. This result, which signifies that the best model is on average 304 hours off (i.e. approximately 12.7 days), underscores the limited practical use of direct remaining LOS predictions. The MAE being so high is mainly rooted in the fact that the target distribution is wide, with a large variance and relatively fat tail (see Figure 4.2). The data set is not large enough to represent an adequate amount of samples for each segment of the target distribution. It is because of this result and similar results reported in the literature on LOS prediction [11] that this thesis chose to focus on classification.

6.1.2 The Importance of GA and Indicator Masks

The ablation experiments with the standard LSTM model corroborate the claims made in recent literature that the GA is an important variable for predicting the LOS of newborns in the NICU [5], [61], and that the use of mask variables that indicate imputed values has a large effect on model performance [11], [13]. When the GA variable is not considered for training or no mask variables are used, the performance of the standard LSTM classifier is significantly lower (see Table 5.2 and Figure 5.6). The choice of including a precise estimate of the GA as a feature in this study led to an approximate 18.6% reduction in data (see Figure 4.1). The results suggest that this was a legitimate choice and that hospital management should attempt to report the GA in a precise and complete manner if they plan on modelling the LOS of newborns in the NICU.

If future studies are to be carried out with a larger data set, it is recommended to conduct a similar ablation study to corroborate the results presented in this thesis.

6.2 Limitations and Suggestions for Future Work

The findings of this thesis illustrate the usefulness of modelling the LOS of newborns in the NICU as a multivariate time series classification problem. This approach intuitively allows for predictions at one hour intervals. Moreover, it serves as an exploration into how the LOS dynamics in the NICU can be modelled using various deep learning models. It is unclear whether the performance of the designed models is high enough for them to be deployed in a clinical setting – the highest kappa coefficients lie below the general viability threshold of 0.4 [66], [67]. However, solely considering the threshold for clinical viability of 0.4 for Cohen's kappa coefficient may not be enough to reject their clinical feasibility. To make a proper assessment of their usefulness as a decision support system, they should be compared to the performance of expert clinicians. The main recommendation for future studies is to conduct the research in collaboration with neonatologists and create a human baseline. Besides, the MIMIC-III database describes relatively few stays of newborns in the NICU; collecting more data and retraining the models may improve the performance.

The goal of this thesis was not to create a model or system that is ready for deployment in a clinical environment. However, it is worthwhile to explore how the models designed in this thesis could be used in a real-life setting. Cohen's kappa coefficient is a suitable metric for model comparison and model selection, however, it is not an intuitive metric to be used in the hospital. A more insightful representation of the model performance is the normalized confusion matrix. The normalized confusion matrix of the best LSTM model applied to the test data is displayed in Figure 6.1. The confusion matrix is normalized across its rows, hence, the numbers in the cells signify the fraction of cases that a data example is classified as a certain class with respect to its true label. The fraction of correct predictions per class is displayed along the diagonal of the confusion matrix. The confusion matrix indicates that the model excels at classifying a remaining LOS of longer than a week. One of the reasons for this result could be that the remaining LOS longer than a week is the dominant class in the data set (see Figure 4.3). Collecting a larger data set may lead to improved scores for the two minority classes. Depending on the objective of the hospital management or neonatologists, higher performance on the minority classes can be achieved on the current data set by oversampling data samples of the minority classes. An alternative is to assign higher weight to the loss of misclassified minority examples in the categorical cross-entropy loss function. These two strategies come at the expense of a reduced performance on the majority class and may lead to much lower kappa coefficients.

In a clinical setting, it may be advantageous to work with probabilistic output rather than with specific class predictions. This is readily accomplished with the deep learning models

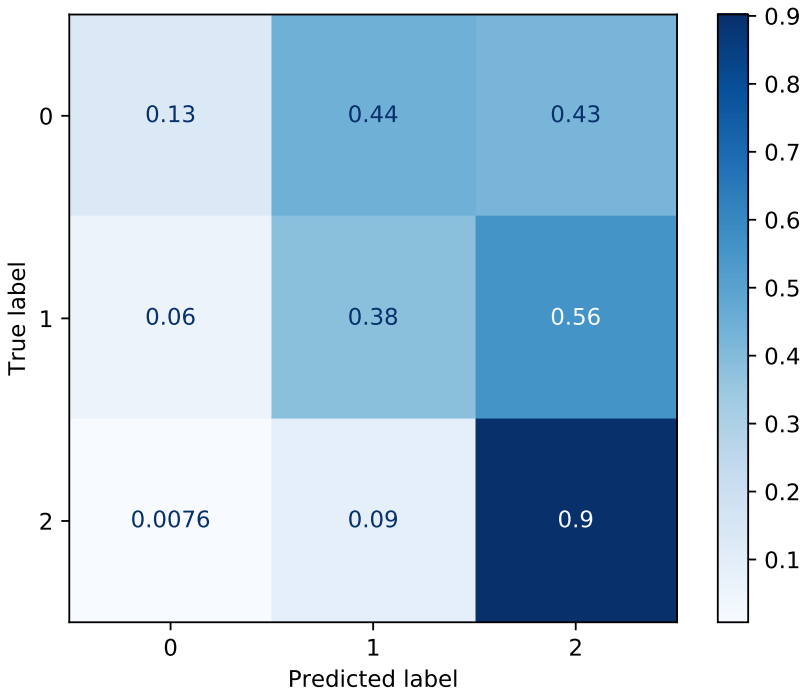

Figure 6.1: The normalized confusion matrix of the best LSTM applied to the test set. The axis labels indicate the following classes: (0) less than two days, (1) between two days and a week, (2) more than a week.

presented in this thesis, since the softmax function outputs a normalized probability distribution over all possible target classes. When the clinical data are represented as a multivariate time series that incrementally grows per hour as new data becomes available, a visualization of the probabilistic output of the softmax function in the form of a heat map may provide valuable insights. Such a heat map reflects the confidence of the model at each time step and shows how this confidence may change over time. Figure 6.2 shows the heat map of the probabilistic predictions of the best LSTM model for a particular time series from the test set. Even though at many of the time steps the model does not classify the remaining LOS correctly, a global trend is clearly visible. When the tint of the color for one category starts to fade while the shade of another starts to darken, this may indicate a class transition.

The main obstacle that seems to stand in the way of the models performing better is that there are multiple reasons for a short stay. Some short stays result in the transfer of a healthy infant to the newborn nursery. During other stays, a medical urgency is detected that requires a transfer to a different facility or that leads to the newborn's passing away. A newborn in critical situation can either lead to a short stay, if the newborn passes away, or in a long stay requiring much medical help. Yet other stays are short due to the newborn being transferred because of

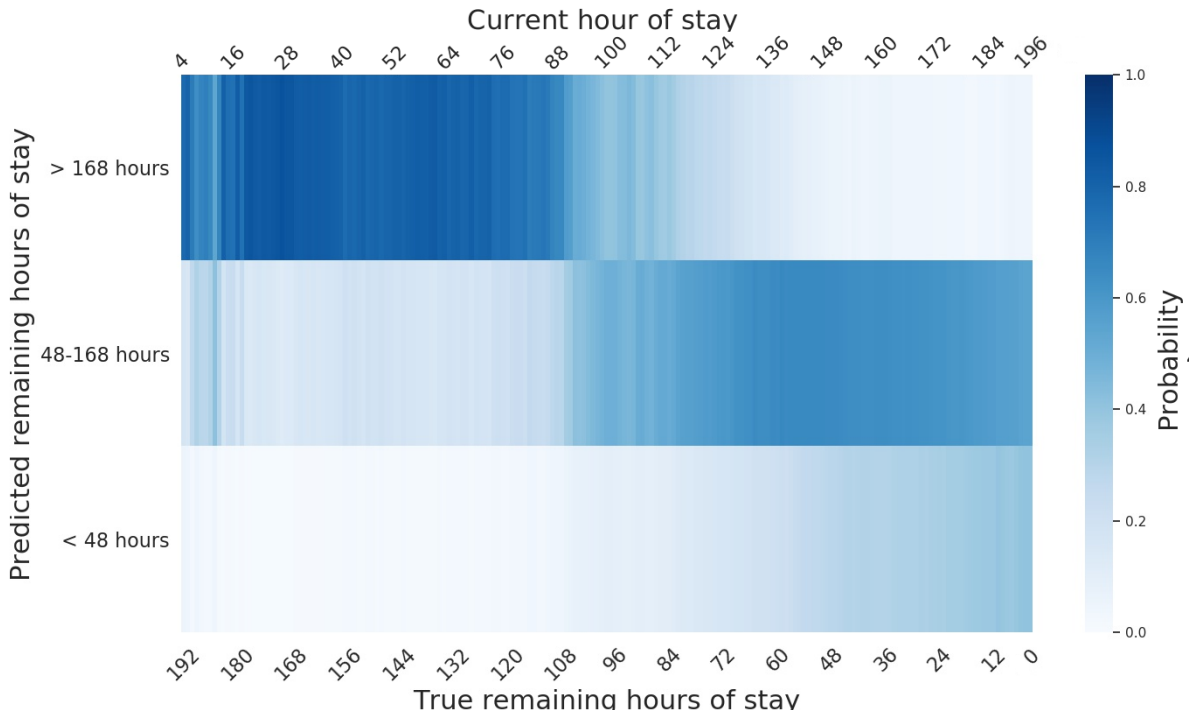

Figure 6.2: Probability heat map of the predictions of the best LSTM model at every hour of a particular stay. The total stay lasted 196 hours (i.e. a little over 8 days) and the model started predicting after the first four hours of the stay. The time series data of the NICU stay used to generate this plot is taken from the test data set. Note that this is a particular time series that is not necessarily representative for the the entire data set.

the NICU not having enough capacity at the time of birth. The reason for admission to the NICU is often unclear when interpreting the MIMIC-III database. Some stays have discharge summaries associated with them, which may shed some light on these unknown dynamics, however, these are not considered in this thesis since they were only written after the stay had ended. In some cases it may be evident for the medical staff when a transfer will take place, for example when a transfer to a different care facility is planned due to limited capacity in the NICU, or when the infant will be transferred for a planned surgery. The data set used in this thesis does not contain such annotations. If future studies can be conducted in collaboration with neonatologists, models could be trained exclusively on the data about which the medical staff was unclear as to how long the stay would last.

There are several systematic biases that complicate accurate LOS prediction. All data used in this study were generated at a single medical facility, which may not accurately reflect the dynamics of similar facilities in the region, let alone in the country or internationally. Some NICUs are better equipped or more specialized; others have more bed capacity or staff available.

Differences also exist in the way in which data is generated: not all hospitals use the same monitoring devices, laboratory testing may be more or less frequent or available depending on the facility, and the various hospital information systems and protocols may lead to differences in documentation. Factors unrelated to the hospital that can introduce biases include the health and diet of the mother before and during the pregnancy, complications during birth, as well as the mother's age, gravidity and parity and whether the pregnancy was a single or multiple gestation. Other factors such as the insurance of the newborn and the socioeconomic status of the parent(s) may also have an influence on the LOS. Due to a lack of data, the models proposed in this thesis can unfortunately not account for such biases. If the privacy policies of the care facilities involved allow for it, it is recommended that future studies use data from multiple care facilities.

One of the preliminary experiments conducted in this study involved the use of clinical notes to predict the remaining LOS. The idea was to use the `BioSentVec` [77] word embeddings to represent notes in matrix format and to extend the models with a CNN module to extract features from these embeddings. Under the assumption that recent notes contain most relevant information but may relate to past notes, the idea was to preserve the time dimension by computing a weighted linear combination of past and present notes in which present notes weigh more. This strategy did not lead to a tractable learning set-up. Less than 20% of the timestamps have a note associated with them and many time series sub-sequences have no notes associated with them at all. This, combined with the fact that notes are of different length, leads to excessively sparse feature matrices that are hard to align with the multivariate time series. In the absence of well-curated notes, this thesis does not recommend the pursuit of this research direction.

Two potentially fruitful research directions are to consider attention networks to model the multivariate time series and to use continuous waveforms instead of discrete measurements. The potential of attention modules for predicting the LOS of adult ICU patients has been indicated in previous work [12], [16]. As described in chapter 3, Xu et al. matched a subset of the MIMIC-III database with the MIMIC-III Waveform database and reported promising results for predicting the LOS of adult ICU patients based on the waveforms. [16]. If future studies manage to find a waveform database that can be associated and time-aligned with NICU stays, it may be a worthwhile research direction to use these for LOS prediction.

6.3 Ethics and Sustainability

The clinical data in the MIMIC-III database is de-identified in such a way that it should not be possible to trace back the data to specific individuals. Nonetheless, the authors of the database demand that the data be treated with care and respect, due to the sensitive nature of detailed medical information about patients. Access to the data has to be requested by means of a formal application on the PhysioNet website [78]. The main requirement of the application is

that the applicant has successfully completed the CITI "Data or Specimens Only Research" online training course[1] This application procedure has been successfully completed for this thesis.

The goals of this thesis are in line with Sustainable Development Goal III by the United Nations: *Good Health and Well-being*[2].

[1] See: about.citiprogram.org
[2] See: un.org/development/

Chapter 7

Conclusion

This thesis explored deep learning approaches to modelling the remaining LOS of newborn patients in the NICU. One of the most significant findings to emerge from this thesis is the superior performance of various deep learning architectures in comparison to a MLR baseline, most notably a standard LSTM and a channel-wise LSTM. Moreover, it demonstrated how modelling clinical data with deep learning models in which the data is represented as a multivariate time series naturally facilitates repeated predictions over time while the stay progresses, which can be visualized in an intuitive way. The results and analysis of this thesis showed the relevance of the GA variable for accurate predictions of the remaining LOS of a newborn in the NICU. Additionally, the results corroborate the importance of passing binary indicator variables as input to the deep learning models to make patterns of missing measurements in the data more explicit. This thesis also explored the novel approach of predicting the remaining LOS with FCNs and LSTM-FCNs, which yields mixed results but requires more investigation in future studies. This thesis has demonstrated that deep learning is a promising candidate for the challenging task of modelling the LOS of newborns in the NICU. The results of the models considered in this thesis can hopefully serve as a framework for future studies.

Bibliography

[1] J. Rogowski, "Measuring the cost of neonatal and perinatal care", *Pediatrics*, vol. 103, no. Supplement E1, pp. 329–335, 1999.

[2] T. J. Johnson, A. L. Patel, B. J. Jegier, J. L. Engstrom, and P. P. Meier, "Cost of morbidities in very low birth weight infants", *The Journal of pediatrics*, vol. 162, no. 2, pp. 243–249, 2013.

[3] J. R. Lave and S. Leinhardt, "The cost and length of a hospital stay", *Inquiry*, vol. 13, no. 4, pp. 327–343, 1976.

[4] P. S. Romano, P. Hussey, and D. Ritley, *Selecting quality and resource use measures: A decision guide for community quality collaboratives*. Citeseer, 2010.

[5] S. E. Seaton, L. Barker, D. Jenkins, E. S. Draper, K. R. Abrams, and B. N. Manktelow, "What factors predict length of stay in a neonatal unit: A systematic review", *BMJ open*, vol. 6, no. 10, e010466, 2016.

[6] M. W. Temple, C. U. Lehmann, and D. Fabbri, "Predicting discharge dates from the nicu using progress note data", *Pediatrics*, vol. 136, no. 2, e395–e405, 2015.

[7] R. M. Kliegman, R. E. Behrman, H. B. Jenson, and B. M. Stanton, *Nelson textbook of pediatrics e-book*. Elsevier Health Sciences, 2007.

[8] P. Powell, C. Powell, S. Hollis, and M. Robinson, "When will my baby go home?", *Archives of disease in childhood*, vol. 67, no. 10 Spec No, pp. 1214–1216, 1992.

[9] B. Manktelow, E. Draper, C. Field, and D. Field, "Estimates of length of neonatal stay for very premature babies in the uk", *Archives of Disease in Childhood-Fetal and Neonatal Edition*, vol. 95, no. 4, F288–F292, 2010.

[10] H. C. Lee, M. V. Bennett, J. Schulman, and J. B. Gould, "Accounting for variation in length of nicu stay for extremely low birth weight infants", *Journal of Perinatology*, vol. 33, no. 11, pp. 872–876, 2013.

[11] H. Harutyunyan, H. Khachatrian, D. C. Kale, G. Ver Steeg, and A. Galstyan, "Multitask learning and benchmarking with clinical time series data", *Scientific data*, vol. 6, no. 1, pp. 1–18, 2019.

[12] H. Song, D. Rajan, J. J. Thiagarajan, and A. Spanias, "Attend and diagnose: Clinical time series analysis using attention models", in *Thirty-second AAAI conference on artificial intelligence*, 2018.

[13] Z. C. Lipton, D. C. Kale, and R. Wetzel, "Modeling missing data in clinical time series with rnns", *arXiv preprint arXiv:1606.04130*, 2016.

[14] F. Karim, S. Majumdar, H. Darabi, and S. Harford, "Multivariate lstm-fcns for time series classification", *Neural Networks*, vol. 116, pp. 237–245, 2019.

[15] A. E. Johnson, T. J. Pollard, L. Shen, H. L. Li-wei, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. A. Celi, and R. G. Mark, "Mimic-iii, a freely accessible critical care database", *Scientific data*, vol. 3, p. 160 035, 2016.

[16] Y. Xu, S. Biswal, S. R. Deshpande, K. O. Maher, and J. Sun, "Raim: Recurrent attentive and intensive model of multimodal patient monitoring data", in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 2565–2573.

[17] S. Khadanga, K. Aggarwal, S. Joty, and J. Srivastava, "Using clinical notes with time series data for icu management", *arXiv preprint arXiv:1909.09702*, 2019.

[18] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors", *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[19] S. Hochreiter and J. Schmidhuber, "Long short-term memory", *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[20] M. Sundermeyer, R. Schlüter, and H. Ney, "Lstm neural networks for language modeling", in *Thirteenth annual conference of the international speech communication association*, 2012.

[21] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation", *arXiv preprint arXiv:1406.1078*, 2014.

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

[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks", in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[24] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines", in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.

[25] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, *et al.*, "Recent advances in convolutional neural networks", *Pattern Recognition*, vol. 77, pp. 354–377, 2018.

[26] Y. Zheng, Q. Liu, E. Chen, Y. Ge, and J. L. Zhao, "Time series classification using multi-channels deep convolutional neural networks", in *International Conference on Web-Age Information Management*, Springer, 2014, pp. 298–310.

[27] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[28] Y. Chen, E. Keogh, B. Hu, N. Begum, A. Bagnall, A. Mueen, and G. Batista, "The ucr time series classification archive", 2015.

[29] Z. Wang, W. Yan, and T. Oates, "Time series classification from scratch with deep neural networks: A strong baseline", in *2017 International joint conference on neural networks (IJCNN)*, IEEE, 2017, pp. 1578–1585.

[30] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift", *arXiv preprint arXiv:1502.03167*, 2015.

[31] M. Lin, Q. Chen, and S. Yan, "Network in network", *arXiv preprint arXiv:1312.4400*, 2013.

[32] F. Karim, S. Majumdar, H. Darabi, and S. Chen, "Lstm fully convolutional networks for time series classification", *IEEE access*, vol. 6, pp. 1662–1669, 2017.

[33] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting", *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[34] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[35] D. J. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series.", in *KDD workshop*, Seattle, WA, vol. 10, 1994, pp. 359–370.

[36] A. Liaw, M. Wiener, *et al.*, "Classification and regression by randomforest", *R news*, vol. 2, no. 3, pp. 18–22, 2002.

[37] C. Cortes and V. Vapnik, "Support-vector networks", *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[38] P. Schäfer and U. Leser, "Multivariate time series classification with weasel+ muse", *arXiv preprint arXiv:1711.11343*, 2017.

[39] D. H. Gustafson, "Length of stay: Prediction and explanation", *Health Services Research*, vol. 3, no. 1, p. 12, 1968.

[40] H. Altman, H. V. Angle, M. L. Brown, and I. W. Sletten, "Prediction of length of hospital stay", *Comprehensive Psychiatry*, vol. 13, no. 5, pp. 471–480, 1972.

[41] S. Berki, M. L. Ashcraft, and W. C. Newbrander, "Length-of-stay variations within icda-8 diagnosis-related groups", *Medical Care*, pp. 126–142, 1984.

[42] W. Weintraub, E. Jones, J. Craver, R. Guyton, and C. Cohen, "Determinants of prolonged length of hospital stay after coronary bypass surgery.", *Circulation*, vol. 80, no. 2, pp. 276–284, 1989.

[43] W. A. Knaus, D. P. Wagner, J. E. Zimmerman, and E. A. Draper, "Variations in mortality and length of stay in intensive care units", *Annals of Internal Medicine*, vol. 118, no. 10, pp. 753–761, 1993.

[44] J. V. Tu and M. R. Guerriere, "Use of a neural network as a predictive instrument for length of stay in the intensive care unit following cardiac surgery.", in *Proceedings of the Annual Symposium on Computer Application in Medical Care*, American Medical Informatics Association, 1992, p. 666.

[45] S. Walczak, W. E. Pofahl, R. J. Scorpio, *et al.*, "Predicting hospital length of stay with neural networks.", in *FLAIRS conference*, 1998, pp. 333–337.

[46] U. E. Ruttimann and M. M. Pollack, "Variability in duration of stay in pediatric intensive care units: A multiinstitutional study", *The Journal of pediatrics*, vol. 128, no. 1, pp. 35–44, 1996.

[47] T. M. Osler, F. B. Rogers, L. G. Glance, M. Cohen, R. Rutledge, and S. R. Shackford, "Predicting survival, length of stay, and cost in the surgical intensive care unit: Apache ii versus iciss", *Journal of Trauma and Acute Care Surgery*, vol. 45, no. 2, pp. 234–238, 1998.

[48] P. E. Marik and L. Hedman, "What's in a day? determining intensive care unit length of stay", *Critical care medicine*, vol. 28, no. 6, pp. 2090–2093, 2000.

[49] T. L. Higgins, W. T. McGee, J. S. Steingrub, J. Rapoport, S. Lemeshow, and D. Teres, "Early indicators of prolonged intensive care unit stay: Impact of illness severity, physician staffing, and pre–intensive care unit length of stay", *Critical care medicine*, vol. 31, no. 1, pp. 45–51, 2003.

[50] R. Paterson, D. MacLeod, D. Thetford, A. Beattie, C. Graham, S. Lam, and D. Bell, "Prediction of in-hospital mortality and length of stay using an early warning scoring system: Clinical audit", *Clinical Medicine*, vol. 6, no. 3, p. 281, 2006.

[51] V. Liu, P. Kipnis, M. K. Gould, and G. J. Escobar, "Length of stay predictions: Improvements through the use of automated laboratory and comorbidity variables", *Medical care*, pp. 739–744, 2010.

[52] A. Rajkomar, E. Oren, K. Chen, A. M. Dai, N. Hajaj, M. Hardt, P. J. Liu, X. Liu, J. Marcus, M. Sun, *et al.*, "Scalable and accurate deep learning with electronic health records", *NPJ Digital Medicine*, vol. 1, no. 1, p. 18, 2018.

[53] A. Freitas, T. Silva-Costa, F. Lopes, I. Garcia-Lema, A. Teixeira-Pinto, P. Brazdil, and A. Costa-Pereira, "Factors influencing hospital high length of stay outliers", *BMC health services research*, vol. 12, no. 1, p. 265, 2012.

[54] A. Azari, V. P. Janeja, and A. Mohseni, "Predicting hospital length of stay (phlos): A multi-tiered data mining approach", in *2012 IEEE 12th International Conference on Data Mining Workshops*, IEEE, 2012, pp. 17–24.

[55] P. R. Hachesu, M. Ahmadi, S. Alizadeh, and F. Sadoughi, "Use of data mining techniques to determine and predict length of stay of cardiac patients", *Healthcare informatics research*, vol. 19, no. 2, pp. 121–129, 2013.

[56] A. Morton, E. Marzban, G. Giannoulis, A. Patel, R. Aparasu, and I. A. Kakadiaris, "A comparison of supervised machine learning techniques for predicting short-term in-hospital length of stay among diabetic patients", in *2014 13th International Conference on Machine Learning and Applications*, IEEE, 2014, pp. 428–431.

[57] S. R. Levin, E. T. Harley, J. C. Fackler, C. U. Lehmann, J. W. Custer, D. France, and S. L. Zeger, "Real-time forecasting of pediatric intensive care unit length of stay using computerized provider orders", *Critical care medicine*, vol. 40, no. 11, pp. 3058–3064, 2012.

[58] S. Barnes, E. Hamrock, M. Toerper, S. Siddiqui, and S. Levin, "Real-time prediction of inpatient length of stay for discharge prioritization", *Journal of the American Medical Informatics Association*, vol. 23, no. e1, e2–e10, 2016.

[59] X. Cai, O. Perez-Concha, E. Coiera, F. Martin-Sanchez, R. Day, D. Roffe, and B. Gallego, "Real-time prediction of mortality, readmission, and length of stay using electronic health record data", *Journal of the American Medical Informatics Association*, vol. 23, no. 3, pp. 553–561, 2016.

[60] S. Purushotham, C. Meng, Z. Che, and Y. Liu, "Benchmarking deep learning models on large healthcare datasets", *Journal of biomedical informatics*, vol. 83, pp. 112–134, 2018.

[61] S. E. Seaton, L. Barker, E. S. Draper, K. R. Abrams, N. Modi, and B. N. Manktelow, "Estimating neonatal length of stay for babies born very preterm", *Archives of Disease in Childhood-Fetal and Neonatal Edition*, vol. 104, no. 2, F182–F186, 2019.

[62] B. Zernikow, K. Holtmannspötter, E. Michel, F. Hornschuh, K. Groote, and K.-H. Hennecke, "Predicting length-of-stay in preterm neonates", *European journal of pediatrics*, vol. 158, no. 1, pp. 59–62, 1999.

[63] M. W. Temple, C. U. Lehmann, and D. Fabbri, "Natural language processing for cohort discovery in a discharge prediction model for the neonatal icu", *Applied clinical informatics*, vol. 7, no. 01, pp. 101–115, 2016.

[64] A. A. of Pediatrics Subcommittee on Hyperbilirubinemia *et al.*, "Management of hyperbilirubinemia in the newborn infant 35 or more weeks of gestation.", *Pediatrics*, vol. 114, no. 1, p. 297, 2004.

[65] J. Cohen, "A coefficient of agreement for nominal scales", *Educational and psychological measurement*, vol. 20, no. 1, pp. 37–46, 1960.

[66] J. L. Fleiss, B. Levin, M. C. Paik, *et al.*, "The measurement of interrater agreement", *Statistical methods for rates and proportions*, vol. 2, no. 212-236, pp. 22–23, 1981.

[67] J. R. Landis and G. G. Koch, "The measurement of observer agreement for categorical data", *biometrics*, pp. 159–174, 1977.

[68] M. Friedman, "The use of ranks to avoid the assumption of normality implicit in the analysis of variance", *Journal of the american statistical association*, vol. 32, no. 200, pp. 675–701, 1937.

[69] ——, "A comparison of alternative tests of significance for the problem of m rankings", *The Annals of Mathematical Statistics*, vol. 11, no. 1, pp. 86–92, 1940.

[70] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, "Deep learning for time series classification: A review", *Data Mining and Knowledge Discovery*, vol. 33, no. 4, pp. 917–963, 2019.

[71] F. Wilcoxon, "Individual comparisons by ranking methods", in *Breakthroughs in statistics*, Springer, 1992, pp. 196–202.

[72] J. Demšar, "Statistical comparisons of classifiers over multiple data sets", *Journal of Machine learning research*, vol. 7, no. Jan, pp. 1–30, 2006.

[73] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization", *arXiv preprint arXiv:1412.6980*, 2014.

[74] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, *et al.*, "Scikit-learn: Machine learning in python", *the Journal of machine Learning research*, vol. 12, pp. 2825–2830, 2011.

[75] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, *et al.*, "Tensorflow: A system for large-scale machine learning", in *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, 2016, pp. 265–283.

[76] A. Defazio, F. Bach, and S. Lacoste-Julien, "Saga: A fast incremental gradient method with support for non-strongly convex composite objectives", in *Advances in neural information processing systems*, 2014, pp. 1646–1654.

[77] Q. Chen, Y. Peng, and Z. Lu, "Biosentvec: Creating sentence embeddings for biomedical texts", in *2019 IEEE International Conference on Healthcare Informatics (ICHI)*, IEEE, 2019, pp. 1–5.

[78] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley, "PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals", *Circulation*, vol. 101, no. 23, e215–e220, 2000 (June 13).

[79] M. L. Porter and M. B. L. Dennis, "Hyperbilirubinemia in the term newborn", *American family physician*, vol. 65, no. 4, p. 599, 2002.

[80] S. Fleming, M. Thompson, R. Stevens, C. Heneghan, A. Plüddemann, I. Maconochie, L. Tarassenko, and D. Mant, "Normal ranges of heart rate and respiratory rate in children from birth to 18 years of age: A systematic review of observational studies", *The Lancet*, vol. 377, no. 9770, pp. 1011–1018, 2011.

[81] E. L. Haksari, H. N. Lafeber, M. Hakimi, E. P. Pawirohartono, and L. Nyström, "Reference curves of birth weight, length, and head circumference for gestational ages in yogyakarta, indonesia", *BMC pediatrics*, vol. 16, no. 1, p. 188, 2016.

[82] N. Di Rollo, D. Caesar, D. A. Ferenbach, and M. J. Dunn, "Survival from profound metabolic acidosis due to hypovolaemic shock. a world record?", *Case Reports*, vol. 2013, bcr2012008315, 2013.

Appendix A

MIMIC-III Event Identifiers

Table A lists the mappings between the 14 clinical variables used in this thesis and the MIMIC-III event identifiers.

Variable	MIMIC-III Identifiers)
Bilirubin (direct)	50883, 803
Bilirubin (indirect)	50884, 3765
Blood pressure (diastolic)	8502, 8503, 8504, 8506, 8507
Blood pressure (systolic)	3313, 3315, 3317, 3323, 3321
Capillary refill rate	3348
Fraction inspired oxygen	3420, 3422
Gestational age	n.a.
Heart rate	211
Height	4188
Oxygen saturation	834, 50817, 8498
pH	50820, 51491, 860, 4753, 3839, 4202, 1673, 50831, 51094
Respiratory rate	3603
Temperature	3655, 3654
Weight	3580, 3693, 3723, 4183

Table A.1: MIMIC-III event IDs to clinical variables mappings.

Appendix B

Variables: Valid Ranges and Imputation Values

It is a challenge to determine the imputation values and valid ranges of the clinical variables. NICU patients are not a representative sample of the overall population of newborns, hence, imputing the mean of the training data or the normal values presented in the literature may be inadequate. The following list serves as a justification for the choice of valid ranges and imputation values:

- **Bilirubin (direct and indirect)**: Instead of keeping a certain percentile, the upper limit of 30 mg/dL was determined by inspecting the data distributions and extrapolating from the chart published by Porter and Dennis [79]. Without explicitly stating so, it seems that some values in the MIMIC-III database were reported in μmol/L instead of mg/dL, which yields much higher values. It is challenging to decide on concrete imputation values, since the bilirubin levels both depend on the gestational age and varies a lot over the first couple of days of life. Since low bilirubin values are generally harmless, both the direct and indirect bilirubin are imputed at 0.0 mg/dL.

- **Blood pressure (diastolic and systolic)**: Since the data contains patients who died during their NICU stay, the blood pressure values can be relatively extreme. After checking the blood pressure distributions for impossible outliers, the maximum diastolic blood pressure was set to 100 mm/Hg and the systolic to 170 mm/Hg. The mean values of 37 mm/Hg for diastolic pressure and 67 mm/Hg for systolic pressure were chosen as imputation values.

- **Capillary refill rate**: The capillary refill rate is reported as a binary value. A normal capillary refill rate (i.e. 0) is set as the default.

- **Fraction inspired oxygen**: This fraction corresponds to a percentage. Assuming that no purified air is administered to the newborn if no measurements are recorded, the imputation value is set to 21%.
- **Gestational age**: Only the NICU stays for which a valid gestational age could be extracted from the clinical notes are considered, hence, this value is always present.
- **Heart rate**: Similar to the blood pressure, extreme values for the heart rate are not uncommon. The heart rate is capped at 300 BPM, and the mean heart rate of 155 BPM is used for imputation since it is close to the reported mean of a larger population [80].
- **Height and weight**: The values for height and weight corresponding to the gestational age in weeks are determined using the charts of Haksari et al. [81].
- **Oxygen saturation**: The oxygen saturation is a percentage. The mean value of 97% oxygen saturation is used for imputation.
- **pH**: Because of the fact that some newborns die in the NICU, the valid range for pH is set slightly larger than the pH range that is compatible with life [82]. The mean value of 7.3 in the training data is used for imputation.
- **Respiratory rate**: Owing to the fact that some newborns are seriously or fatally ill, the valid range of the respiratory rate is wide. The cut-off point of 150 BRPM was chosen after visually inspecting the distribution. Since many newborns are experiencing respiratory distress, the mean of 49.7 BRPM of the training data may not be representative. According to Tveiten et al., the median respiratory of 953 term infants during their first day of life was 43 BRPM. This value is used as the imputation value, under the assumption that the newborn's breathing pattern is normal if no measurements are recorded.
- **Temperature**: The valid range for body temperature is set from room temperature 20 °C to 44 °C. The mean of the training data is used for imputation, which is 36.2 °C.

TRITA-EECS-EX-2020:582