NOVA SCHOOL OF SCIENCE & TECHNOLOGY

DEPARTMENT OF PHYSICS

LUÍS EDUARDO XAVIER ARAÚJO

Bachelor of Science in Biomedical Engineering

THE CHALLENGE OF TIME IN CLINICAL AI: LONGITUDINAL PREDICTIONS WITH VARIABLE HISTORY

MASTER IN BIOMEDICAL ENGINEERING

NOVA University Lisbon
October, 2023

NOVA SCHOOL OF
SCIENCE & TECHNOLOGY

DEPARTMENT OF
PHYSICS

THE CHALLENGE OF TIME IN CLINICAL AI: LONGITUDINAL PREDICTIONS WITH VARIABLE HISTORY

LUÍS EDUARDO XAVIER ARAÚJO

Bachelor of Science in Biomedical Engineering

Adviser: Prof. Dr. Hugo Filipe Silveira Gamboa
Associate Professor, NOVA University Lisbon

Examination Committee

Chair: Dr. Célia Maria Reis Henriques
Associate Professor, NOVA University Lisbon

Members: Dr. Hugo Filipe Silveira Gamboa
Associate Professor with Aggregation, NOVA University Lisbon

Dr. Ricardo Nuno Pereira Verga e Afonso Vigário
Associate Professor with Aggregation, NOVA University Lisbon

MASTER IN BIOMEDICAL ENGINEERING

NOVA University Lisbon
October, 2023

The Challenge of Time in Clinical AI: Longitudinal Predictions with Variable History

This document was created with the (pdf/Xe/Lua)LaTeX processor and the NOVAthesis template (v6.10.17) [1].

Acknowledgements

It is impossible to put into words what the last five years have meant to me and it is even harder to explain my gratitude to the people that made it possible to have reached the end and to have written this dissertation. However, I'll try to do it anyway.

I want to thank my advisor, Professor Hugo Gamboa, for trusting me and helping me develop my interest in Machine Learning, which has helped me to discover my professional aspirations.

To my supervisor, Ricardo Santos, for his patience and tirelessness in guiding me through the process of writing this thesis.

To my friends and colleagues, who have been the ultimate inspiration and showed that you really can't graduate alone.

Finally, to my family, who have fully trusted and supported me in following my choices and truly shaping me into who I am.

I hope that all of them are as proud as I am to have made this journey with them.

Abstract

Deep Learning research in healthcare aims to handle the growing volume of data in Electronic Health Records. These algorithms present good performance for predicting outcomes on homogeneous longitudinal data. However, challenges arise when dealing with complex time-dependent data, which is common in real-world medical records.

This thesis aims to address these challenges by using Long-Short Term Memory (LSTM) networks to analyze multimodal longitudinal medical data.

The study included the preprocessing of a real-world clinical dataset from an Intensive Care Unit, to make it compatible with LSTM models. This initial step laid the foundation for subsequent analyses.

This research provides valuable insights into the complex domain of patient mortality prediction using LSTM neural networks. By addressing fundamental research questions, the study explored the intricacies of real-world medical data processing, predictive model construction, and the determination of optimal temporal window size for effective forecasts.

The findings highlight the importance of feature selection, particularly based on missing data, to reduce data dimensionality without sacrificing predictive performance. Additionally, the study compared patient mortality forecast performances with different prediction windows.

This research asserts the importance of a careful methodology in dealing with real-world medical data to develop more robust and trustworthy predictive models.

Keywords: Deep Learning, Long Short-Term Memory, Medical Prediction, Real-World Medical Data, Decision Support Systems

Resumo

A Aprendizagem Profunda na área da saúde visa lidar com o volume crescente de dados em Registos Eletrónicos de Saúde. Estes algoritmos apresentam bom desempenho na previsão em dados longitudinais homogéneos. No entanto, têm problemas ao lidar com dados com complexidade temporal, o que é comum em registos médicos reais.

Esta dissestação visa abordar estes desafios através da utilização de redes Long Short-Term Memory (LSTM) para analisar dados médicos longitudinais multimodais.

O estudo incluiu o pré-processamento e estruturação de um conjunto de dados clínicos do mundo real de uma Unidade de Cuidados Intensivos, de forma torná-lo compatível com modelos LSTM. Este passo inicial estabeleceu as bases para análises subsequentes.

O estudo tentou fornecer clareza no campo da previsão da mortalidade de pacientes usando redes neuronais LSTM. Ao abordar as questões fundamentais da pesquisa, o estudo explorou as complexidades do processamento de dados médicos reais, a construção de modelos preditivos e a determinação do tamanho ideal da janela temporal usada para previsões eficazes.

Os resultados sublinham a importância da seleção de features, particularmente com base nos dados em falta, para reduzir a dimensionalidade dos dados sem sacrificar o desempenho dos modelos. Além disso, o estudo comparou o comportamento da previsão de mortalidade de pacientes com diferentes janelas de previsão.

Esta dissertação afirma a importância de uma metodologia cuidadosa no tratamento de dados médicos do mundo real para desenvolver modelos preditivos mais robustos.

Palavras-chave: Aprendizagem Profunda, Long Short-Term Memory, Previsão Clinica, Dados Médicos Reais, Sistemas de Apoio à Decisão

Contents

List of Figures

List of Tables

Acronyms

AD Alzheimer's Disease *(p. 11)*
ADNI Alzheimer's Disease Neuroimaging Initiative *(p. 11)*
AI Artificial Intelligence *(pp. 1, 11)*
ANN Artificial Neural Network *(p. 6)*
API Application Programming Interface *(pp. 21, 26)*
ATC Anatomical Therapeutic Chemical *(pp. 21, 26)*
AUROC Area Under the Receiver Operating Characteristic *(pp. 6, 32, 35, 37)*

CDSS Clinical Decision Support System *(pp. 1, 2)*

DL Deep Learning *(pp. 2, 6, 7, 11, 13, 25, 26, 30)*
DRG Diagnosis Related Groups *(pp. 16, 19, 20, 26, 27)*

EHR Electronic Health Record *(pp. 1, 3, 11, 14, 24)*

FN False Negative *(p. 5)*
FP False Positive *(pp. 5, 31)*

ICD International Classification of Diseases) *(pp. 15, 21, 26, 27)*
ICU Intensive Care Unit *(pp. 3, 4, 13, 14, 17, 18, 23, 24, 37)*

LSTM Long Short-Term Memory *(pp. 2, 9, 11, 12, 23–25, 27–31, 33–37, 39)*

MDC Major Diagnostic Category *(pp. 3, 20, 24, 26, 40)*
MIMIC Medical Information Mart for Intensive Care *(pp. 2, 14, 16, 20, 22–25, 27, 39)*
ML Machine Learning *(pp. 2, 4–6)*

RNN Recurrent Neural Network *(pp. 8, 11, 12)*
ROC Receiver Operating Characteristic *(pp. 6, 31)*

SL Supervised Learning *(p. 4)*

TN True Negative *(p. 5)*
TP True Positive *(pp. 5, 31)*

UL Unsupervised Learning *(pp. 4, 5)*

1 Introduction

Context and Motivation

Over the last decades, healthcare services have been incorporating a growing number of technologies that rely more and more upon their digital resources and the use of the Electronic Health Record (EHR). The EHR is essentially a digital chart where a patient's medical data is stored — information about lab tests, surgeries, and medical notes, among others. Consequently, as research is implemented in clinical contexts, new types of practices are used and, following turn, new types of information are stored [2].

When diagnosing or treating a disease, medical staff submit the patient to several exams and consultations in addition to studying their medical history. All this information is often stored in the EHR granting it highly heterogeneous data that can span their lifespan or the duration of a treatment [2].

Additionally, the increase in life expectancy and the increasing number of clinical events as someone ages, grant EHRs large datasets that span multiple modalities and time windows. The complexity of this data makes it challenging for humans to analyse, which means that medical decision is often based on a small subset of all available information [3–5].

Artificial Intelligence (AI) research in healthcare has had the aim of addressing the growing volume of data. These algorithms use machine capabilities, statistical methods and pattern recognition to make use of all the EHR's information when performing data analysis.

This new possibility for pattern identification of medical data brought the opportunity of creating a new type of Clinical Decision Support System (CDSS) that, unlike previous systems, does not base its information on previous expert knowledge. Instead, CDSSs use AI's data analysis capabilities to learn from actual patterns in clinical information [6].

Nevertheless, AI models that use EHR's data have issues when dealing with high variability of data modalities and time-varying features [7]. Most models and algorithms use heterogeneous modalities without the longitudinality being taken into consideration [8] or don't learn the complex time-dependencies of longitudinal data [9].

Objective and Contribution

This thesis will explore how Machine Learning (ML) models, Deep Learning (DL) in particular, can leverage the complex longitudinal nature of medical data to make predictions in order to assist medical decisions.

The study was divided into two steps, starting with the processing of real-world medical data from the Medical Information Mart for Intensive Care (MIMIC) dataset [10] and transforming it into a structure that could be used by Long Short-Term Memory (LSTM)-based models.

The overarching objectives and the questions that will be tackled by this study are threefold:

1. **How can real-world medical data be processed and structured in order to be fed into DL models, specifically LSTM-based models?**

   The first objective delves into the intricacies of processing and structuring real-world medical data for compatibility with LSTM-based models. As medical datasets often exhibit high dimensionality and heterogeneity, this study seeks to establish methodologies that facilitate the effective transformation of raw medical data into a format suitable for deep learning models.

2. **How does dimensionality reduction influence the prediction capabilities of LSTM-based models?**

   The second objective investigates the impact of dimensionality reduction techniques on the predictive capabilities of LSTM-based models. By reducing the number of features, this research aims to assess how these techniques enhance model performance.

3. **What is the optimal prediction window for LSTM-based models in the context of medical data analysis?**

   The third objective is concerned with identifying the optimal prediction window for LSTM-based models in the context of medical data analysis. This entails examining the temporal aspects of medical data and determining the most suitable time frame for making accurate predictions, thus contributing to CDSS.

In pursuit of these objectives and research questions, this study hopes to contribute to the field of medical data analysis and facilitate the development of medical prediction using DL-based models as CDSSs and improving clinical practice and hospitalization outcomes.

The research work described in this dissertation was carried out in accordance with the norms established in the ethics code of Universidade Nova de Lisboa. The work described and the material presented in this dissertation, with the exceptions clearly indicated, constitute original work carried out by the author.

2

Theoretical Concepts

2.1 Electronic Health Records

EHR's organize data from multiple sources (i.e., different clinical events like lab tests or hospital admission) that cover the lifetime of a patient or even a population's health evolution through time. Since medical practice involves very distinct procedures, the contents of this data collection are very heterogenous — they have several formats like medical images, natural language (clinical notes), and numeric values (lab tests), among others) [11].

On one hand, demographic, lab results or medication data (called structured data because they have a fixed structure and representation inside the EHR) can be very easily analysed because of its inherent organisation. However, unstructured data like, for example, clinical notes or medical images need extra steps to analyse — e.g., clinical notes are in natural language which can often be influenced by context and may have semantic or grammatical errors [11].

Both structured and unstructured data types contain very rich information used by health professionals when studying a patient's history for a personal clinical health evaluation or when analysing a whole population's health status. When making any type of medical decision it is best to use as much information as possible [11].

2.2 Clinical prediction

In the high-stakes environment of an Intensive Care Unit (ICU), clinical decisions carry immense significance, as they directly impact patient outcomes and overall well-being. Every piece of information regarding a patient's health is meticulously examined and evaluated to ensure the best possible care and treatment.

2.2.1 Infectious and Parasitic Diseases in the ICU

Within this complex landscape, patients diagnosed with infectious and parasitic diseases affecting systemic or unspecified sites, classified under Major Diagnostic Category (MDC)

Code 18, represent a group for whom risk assessment and clinical decision-making are particularly crucial.

The significance of clinical decisions extends beyond the pre-operative phase. Patients with infectious and parasitic diseases are particularly vulnerable, and post-operative complications can have severe consequences, potentially hindering their quality of life or even becoming life-threatening. Hence, the importance of ongoing risk assessment remains pivotal in the post-operative phase. Early prediction of potential complications allows for proactive management, ensuring timely interventions to mitigate risks and provide the best possible care.

In essence, making clinical predictions with current and past information has a critical role in comprehensive risk assessment in ICU environments. For patients with infectious and parasitic diseases, this process is even more crucial due to their susceptibility to complications. Effective clinical decision-making not only optimizes treatment strategies but also ensures the allocation of resources where they are needed most, ultimately enhancing the overall care and outcomes of patients in this specific category.

2.3 Machine Learning

Murphy [12] defines ML as "a set of methods that can automatically detect patterns in data, and then use the uncovered patterns to predict future data, or to perform other kinds of decision". Analogously, some of ML's contribution to healthcare can be found in the detection of patterns for biomarker discovery and the prediction of clinical outcomes, or in the contribution to medical decision and practice in other domains such as medical image diagnosis or robotic surgery [3].

2.3.1 Supervised/Unsupervised Learning

ML systems can be classified into several types, based on human supervision during the learning process. In the context of this thesis, Supervised Learning (SL) and Unsupervised Learning (UL) are the most relevant ones:

Supervised Learning

In SL, the algorithm uses data with a label or target y for each instance or vector of features x. The objective is to predict y using the information in x and this can be done in tasks of classification (when predicting classes) or regression (when predicting values). Tasks of clinical risk prediction can utilise SL for the final prediction [13, 14].

Unsupervised Learning

In UL the data does not have a label or class. Some of the tasks performed by UL are Clustering, anomaly/novelty detection or dimensionality reduction [14]. Deep learning

tasks can learn the data probability distributions to perform synthesis or denoising of data. [13]. To this particular thesis, UL will have an important role in the representation of data according to its modality — some studies will employ models that learn the input values of information so that the algorithm can distinguish between similar or correlated events [15, 16].

2.3.2 Model Training and Validation

ML's goal is to learn from data, however, this process should avoid overfitting — i.e, the model should not adjust to every sample in the input data, but instead, learn the overall trend so that it makes correct predictions on future samples [12]. Consequently, model validation is required in order to evaluate its ability to model data without overfitting, additionally, it is used to identify the best-performing model.

Thus, before training the model, the dataset can be partitioned into two: training and test set. The former is used to fit the model and the latter will be used to evaluate its performance. These two sets can be used to compare models that have different hyperparameters or use different algorithms [12].

Cross-validation is also a possible alternative to the simple train-test partition. It consists of repeatedly and randomly splitting training and testing sets of the original dataset (e.g., K-fold Cross Validation) [13].

2.3.3 Performance Metrics

Performance metrics are used in both the training and testing stages to get an estimation of how the model behaves with unseen data, as well as to compare between models [17].

Confusion Matrix

Evaluating a classification model can be done by comparing the predicted label to the actual label for each instance in the data. This is called the Confusion Matrix (see Table 2.1 for an example of a binary classifier's Confusion Matrix) and uses the following notions:

- True Positive (TP) - Instance where positive label is correctly identified
- False Positive (FP) - Instance where positive label is incorrectly identified
- True Negative (TN) - Instance where negative label is correctly identified
- False Negative (FN) - Instance where negative label is incorrectly identified

	Actual Positive	Actual Negative
Predicted Positive	TP	FP
Predicted Negative	FN	TN

Table 2.1: Confusion Matrix for a binary classifier.

Other metrics that derive from the mentioned above can also be used:

Sensitivity

$$Sensitivity = \frac{TP}{TP + FN}$$

Specificity

$$Specificity = \frac{TN}{TN + FP}$$

Precision

$$Precision = \frac{TP}{TP + FP}$$

F-score

$$F - score = \frac{2 * precision * sensitivy}{precision + sensitivity}$$

AUROC - Area under the Receiver Operating Caracheteristic

The Sensitivity of a classifier is the ratio of instances where the label is correctly identified as positive. Specificity is the ratio where it is correctly identified as negative [18]. Thus, a good classifier has high values for both of these metrics. The Receiver Operating Characteristic (ROC) curve is plotted with $Sensitivity$ (x-axis) against $1 - Specificity$ (y-axis). Using the Area Under the Receiver Operating Characteristic (AUROC) (Figure 2.1) to measure a model's performance, the bigger the area, under the ROC curve, the better it is.

2.4 Deep Learning

Traditional ML algorithms such as the Logistic Regression [19] or the Naive Bayes [20] utilize a machine's computational power to perform calculations that are harder for humans to perform but are usually easily programmable by simple logical or mathematical rules. However, to perform more complex tasks (e.g., distinguishing an object in an image or voice recognition) DL is necessary [13].

2.4.1 Artificial Neural Networks

Essentially, DL is an Artificial Neural Network (ANN) with several layers. An ANN architecture resembles the way neurons work and communicate with each other, meaning that each node (analogous for the neuron — see Figure 2.2) has a transfer function that processes inputs — these are influenced by parameters called weights that determine the information's contribution to the final decision — and outputs a value [21].

Figure 2.3 represents a simple architecture for an ANN where nodes are organized into layers. The input and output layers consist of the nodes that, respectively, receive

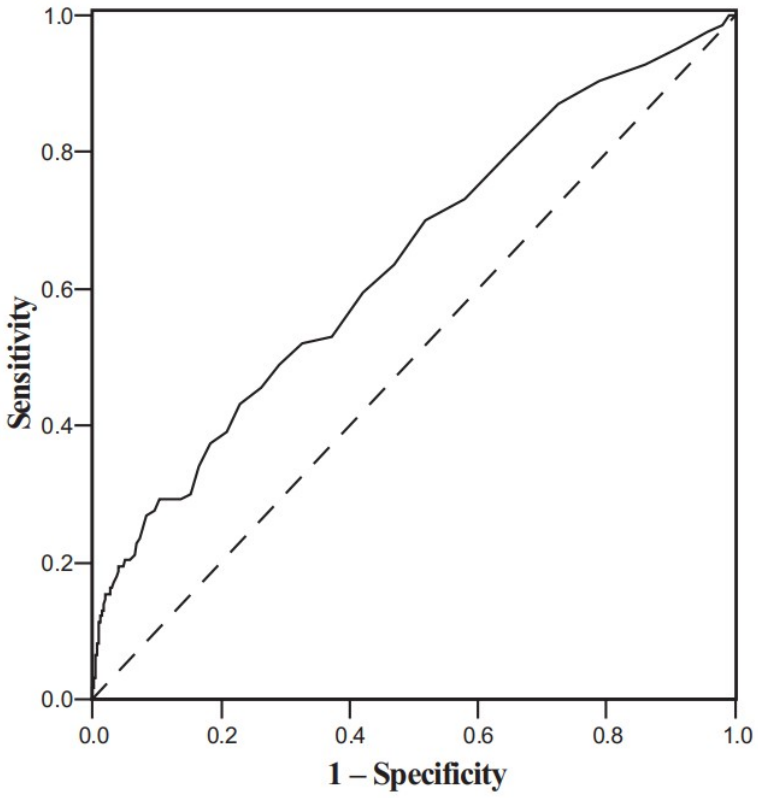

Figure 2.1: ROC curve (retrieved from [18]).

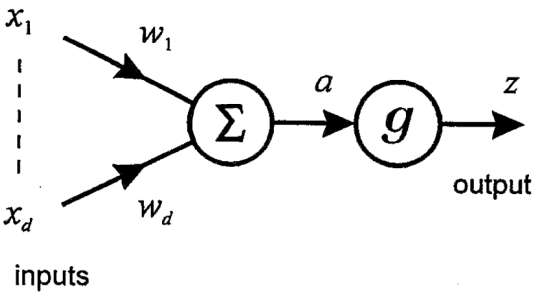

Figure 2.2: Neuron model, represented by the weighted inputs (x1 w1, xd wd) and transfer (Σ) and activation (g) functions (image retrieved from [22]).

the data and compute the system's response, the hidden layer connects both the input and output nodes and has no direct contact with data [23]. DL's name comes from the addition of multiple layers into the network.

Learning Process

Determining the input weight values is called the learning process, its goal is to reach a set of these parameters that guarantee the model completes its task most efficiently [22].

To determine these values — i.e., fitting the model — the error between the predicted and the actual class/value is taken into consideration. The weight values that minimize the error are chosen as optimal.

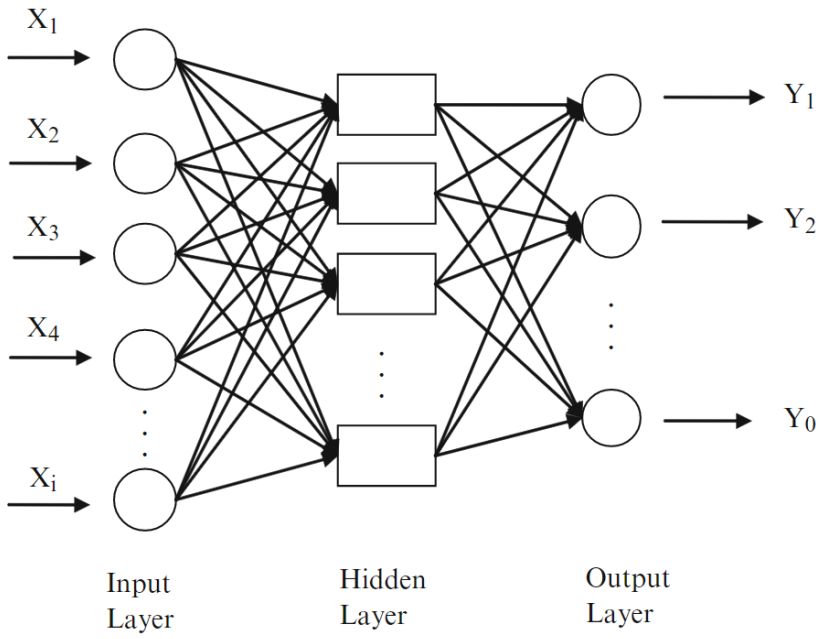

Figure 2.3: Artificial Neural Network Architecture, represented by Input, Hidden and Output layers as well as input (X) and output (Y) values (retrieved from [21]).

2.4.2 Recurrent Neural Networks

A Recurrent Neural Network (RNN) expands on other types of Neural Networks [21] (e.g., Convolutional Neural Networks [24]) by being able to process input sequences of different sizes. In RNNs (e.g., simple recurrent network in Figure 2.4) connections between nodes have two different natures, feedforward — moving information along the structure — and feedback — giving sequential information and context to other nodes [25]. RNNs benefit from the latter connection in sharing parameters between distinct parts of the model [13].

RNNs are used when dealing with sequential information (e.g., time-series) because a node can communicate with itself, meaning that the present value of a variable will influence how the node acts with the future value of that same variable [13] — Figure 2.5 shows an example of a recurrent node.

Vanishing and Exploding Gradient Problem

Despite representing a breakthrough when it comes to analyzing the evolution of a sequence over time, simple RNN's application is limited to smaller-sized data. This limitation when processing longer sequences derives from what is called the vanishing and exploding gradient problem. It refers to the fact that multiplying the recurring

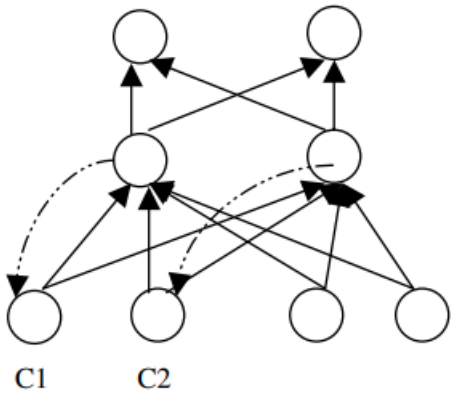

Figure 2.4: Example of a simple Recurrent Network (retrieved from [25]).

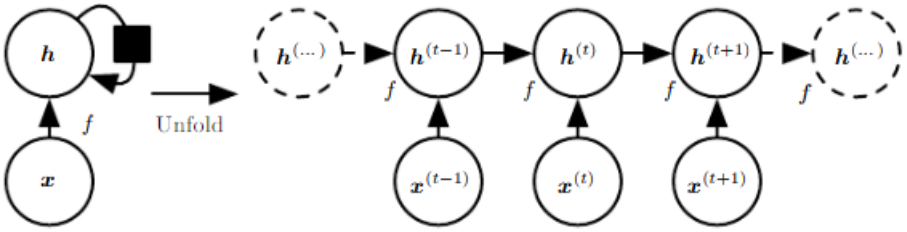

Figure 2.5: Recurrent network with no outputs, exemplifies how the node h is influenced by the sequence of values of the input x (retrieved from [13]).

parameter over a long sequence will either make it explode — reaching greater values — or vanish — getting closer to zero [13]. In the learning process, Gradient Descent is used as an optimizing tool — this is a common algorithm that aims to find the optimal model by iteratively changing the parameter values in order to minimize an error function [14]. Thus, vanishing and exploding gradients mean that this parameter update either slows downs until it stops or starts accelerating, preventing the optimization from reaching the optimal values.

To explore long-term dependencies in data sequences Gated Units are the most effective algorithms. They produce smaller changes in each step of the Gradient Descent algorithm, which help in reaching the optimized solution [13, 26]. Also, these gates essentially control how the current inputs affect the network's memory as well as avoiding that an irrelevant current memory influences future computations [27].

2.4.3 Long Short-Term Memory

LSTM [27] offers an architecture where the hidden layer contains what is called a memory cell and its accompanying gate units (see Figure 2.6). The input gate's (in_j) activation function controls access to the memory cell and the output gate (out_j) can prevent the

corresponding cell from influencing other units by controlling its output.

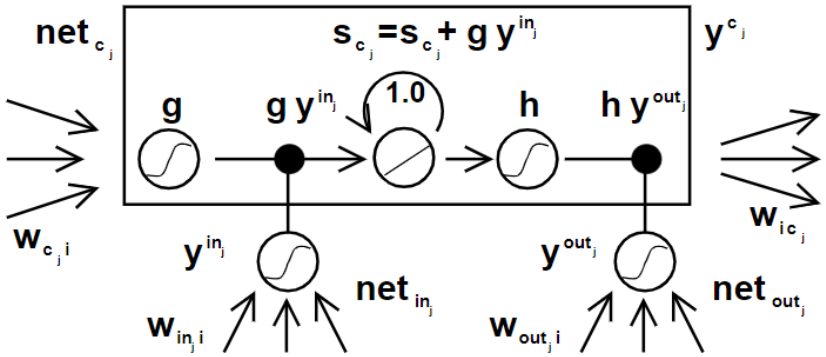

Figure 2.6: Memory cell represented by the box and the accompanying inputs (net_{c_j}), outputs (y^{c_j}) and respective weights ($w_{c_j i}$ and $w_{i c_j}$. The figure also represents the gate units net_{in_j} and net_{out_j} (retrieved from [27]).

The memory cell itself has connections to other layers' units while gate units connect only to the corresponding cell or other units in the same layer (see Figure 2.7) [27].

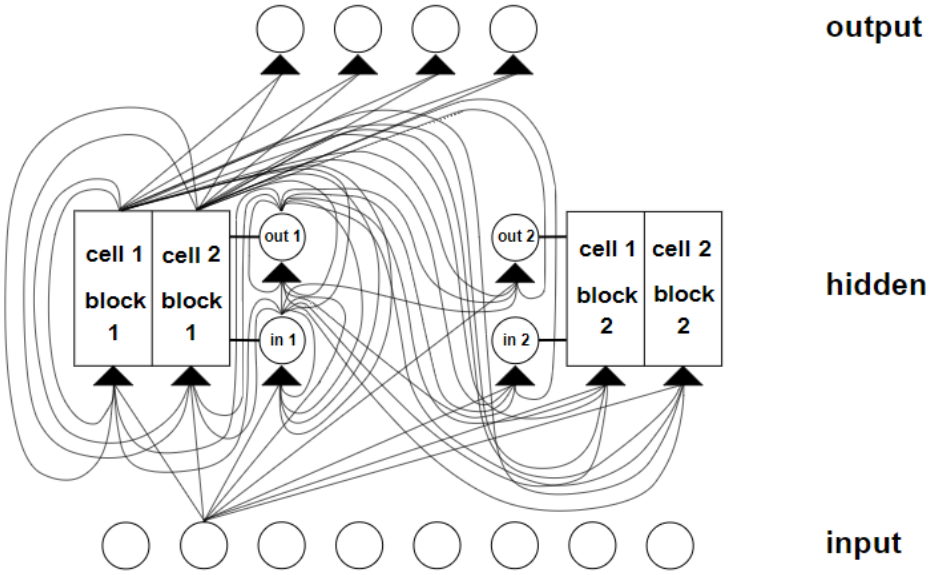

Figure 2.7: LSTM network example (retrieved from [27]).

Additionally, incorporating a forget gate to the recurring self-loop enables the memory cell to learn to release its information. Consequently, this makes processing long input streams with reoccurring events possible without the necessity of pre-decomposing the original sequence into the corresponding events. The forget gate learns to distinguish these events while the model is fit [28].

3

State of the Art

Medical event prediction is the objective of plenty of works that try to utilise AI to analyse clinical events and their data. Natural Language Processing has had a lot of development in healthcare. Specifically, dementia and geriatric mental health have been identified as areas with good potential for DL feature extraction algorithms [29]. Other works model longitudinal information for disease prediction [30] but make assumptions of time heterogeneity, which means that the complex time dependencies inherent to EHR data are not taken advantage of.

Multimodal data with heterogeneous temporal characteristics have had some scientific focus lately. The classic RNN model does not react well to this kind of data. However, new iterations based on this architecture have had good responses to this problem [31].

RNN's ability to learn from sequence data has led this model to be used extensively in EHR data research with the task of disease prediction [32]. The following studies have found model variations and data representation techniques that focus on learning from multimodal longitudinal data from medical practice and disease progression.

RETAIN [33] implements RNN and attention mechanism architectures to emulate medical practice by reading data in reverse — from later to earlier entries. RETAIN showed comparable performances to other RNN versions with the task of Heart Failure prediction. Additionally, the model shows a degree of interpretability that was later [34] expanded upon by adding visual tools that can help health practitioners and researchers understand how medical events influence clinical event prediction.

RNN architectures and attention mechanisms have been used in multimodal and longitudinal studies on Alzheimer's Disease (AD) [16, 35–38]. Specifically, they use Alzheimer's Disease Neuroimaging Initiative (ADNI)'s data to train their proposed models with the objective of predicting disease progression. The use of all the available modalities — e.g, demographic, neuroimage or biomarkers — proves to be beneficial to the prediction performance [16, 35, 38]. Also, LSTM-based algorithms can be made to manage missing values in both the input and output layers. Backpropagation through time works with a layer of LSTM units that model the sequence, also, the network weights are adaptable to the missing input and output data in each point. This method outperformed other

models when dealing with, at most, 74% missing data. When dealing with higher values, replacing the lacking instances with the training data median showed better results [37].

Liu et al. [7] expand on the LSTM model with a proposal to study time heterogeneity in clinical events in order to make predictions in two tasks (death and lab results). Their model (HE-LSTM) is based on both clinical event representation and a novel network architecture. Firstly, before being fed into the network, an event is structured into a vector containing information on its category (e.g., lab test parameters, vital signals, medical procedures, etc) and the corresponding value (categorical or numerical). HE-LSTM's architecture adds an event gate to the LSTM model, which is composed of an event filter and a phase gate [9]. This gate's purpose is to capture the event's category, creating memory cells that are specific to a set of related events.

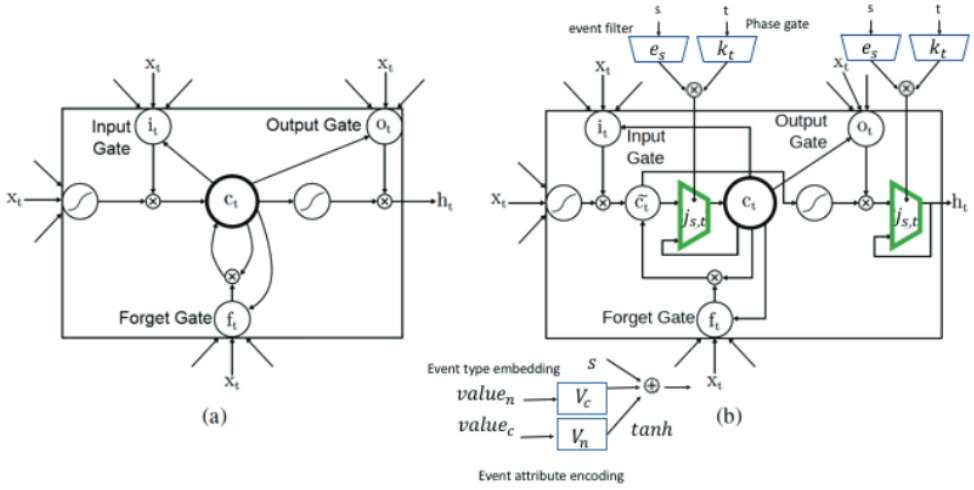

Figure 3.1: Retrieved from [7], LSTM architecture (left) next to HE-LSTM (right).

Comparing HE-LSTM to other RNN-based models Liu et al. [7] concluded that a joint representation of clinical event's categories and values eases temporal correlation analysis and leads to better prediction performances.

Liu L. et al. [15] use the previous event representation of [7] and describe a model that uses attention mechanisms to initially partition a patient's whole event sequence — event attention — and finally to learn long-term dependencies — temporal attention. Liu L. et al. find that this model outperforms other RNN based models. Additionally, despite both attention mechanisms improving prediction scores the former has greater importance.

Patients regularly have a number of different but often related pathologies, and analysing the compared risk of different diseases can offer medical insight that helps decide treatments and manage resources. Dynamic-DeepHit [39] can distinguish between competing risks without resorting to statistical models like the Cox proportional hazard [40]. Dynamic-DeepHit also uses a RNN structure and attention mechanism to deal with the temporal information, additionally, the model's architecture consists of a

shared network — that deals with learning data's time dependencies — and cause-specific sub-networks — that distinguishes each competing risk. The authors showed that this approach had better performance in discriminating between risks when compared to other studies using statistical models.

MCF-LSTM [41] (multi-channel fusion LSTM) is a model that, unlike most others, is highly influenced by the event types most associated with the prediction task (e.g., drug test in the drug recommendation task). Multi-channel fusion projects the auxiliary channels to the primary channel through a gated layer, this fusion module can learn the correlations between the two types of channels. When compared with other models, MCF-LSTM shows better performances in several tasks (e.g., diagnosis prediction, lab test prediction, medication prediction). It is noted by the authors that the proposed approach needs to be tested in other different scenarios with the integration of clinical notes or medical images. Also, time dependencies and temporal interval heterogeneities are only considered in each event type, hence Liu S. et al. suggest improvement by making time dependencies taken into account using the whole data.

An et al. [42] propose to model the clinical record's time irregularity in addition to a type-fusion attention mechanism. The heterogeneous longitudinal information is learned through representation models that process each data type. The represented time-dependent features are then fed into a model that, using an attention mechanism, will learn the correlations between the types. TERTIAN's authors conclude, comparing its performance against other DL algorithms, that the longitudinal patterns and type structuring help this approach outperform the competition in the task of death prediction.

The goal of this thesis will be to take the latest developments in medical event prediction using DL and join their different conclusions to build an algorithm that can leverage multiple modalities and use information inherent to the complex and variable timed event sequences. This model will be used to predict death risk for patients in an ICU — it will be trained and validated with data from patients with infectious and parasitic diseases.

4

Real-World medical Data Processing

4.1 MIMIC Database

This section will provide an overview of MIMIC, its data sources, and the types of information. MIMIC [10] is an open-access database consisting of EHR from patients admitted to the Beth Israel Deaconess Medical Center in Boston, MA, USA. Around 15% of total admissions include a stay in the ICU.

	Admissions	Length of Stay (days)			Mortality		Age		Gender			
		Mean	Standard Deviation	Max	#	%	Mean	Standard Deviation	Female		Male	
									#	%	#	%
Hospital	431231	4.46	6.63	295.98	8598	2	48.5	20.8	158553	52.9	141159	47.1
ICU	66239	3.45	4.92	110.23	6958	10						

Table 4.1: MIMIC Statistics for Hospital and Intensive Care Unit admissions

The dataset is organized in tables that fall into two categories: *Hosp* and *ICU*. Additionally, data is also de-identified, through the substitution of patient identifiers with random integers and the shifting of the date and time information. The dataset is structured as a relational database where the connection between tables is established through the use of unique patient and admissions identifiers, known as 'subject_id' and 'hadm_id', respectively. These identifiers play a crucial role in linking and querying the data across different tables, allowing for comprehensive analysis and insights into patient journeys within the healthcare system. It's important to note that despite this structured organization in a comma-separated values format, the freely-accessible data is not normalized or cleaned to fully reflect the complexities of a real-world clinical dataset.

4.1.1 Hosp Module

The *Hosp* module comprises data from the various clinical departments regarding a patient's hospital stay and outpatient information (e.g. laboratory tests). The latter was not used in this study since the objective is to study patient outcomes using patient information gathered during hospitalization. From this module, five tables were used: *patients*, *admissions*, *procedures*, *prescriptions*, *drgcodes*. These tables were chosen because they provide data that is relevant to the patient's stay (e.g. demographic data in *patients*,

the diagnosis that led to hospitalization in *drgcodes*) or, on the other hand, describes the patient's development during its course (e.g. prescribed medications in *prescriptions*).

patients

The *patients* table stores information that remains constant throughout a patient's lifetime (eg. gender or age), but, most importantly it contains the date of death for each patient, up to one year after hospital discharge. This was crucial in the task of mortality prediction since the death time information in the *admissions* table does not have information past the patient's discharge. Using the *patients* table's date of death, patients that have died after discharge but in the prediction window can be used in the study.

patients					
subject_id	gender	anchor_age	anchor_year	anchor_year_group	dod

Table 4.2: patients table columns

admissions

The *admissions* table is the first to be processed. It contains general information regarding each patient's admission to the hospital (eg. urgency classification, hospital location), and demographic information (insurance, language, marital status, ethnicity). This table also includes timing information (admission, discharge and death) that is used to create a time series.

admissions			
subject_id	hadm_id	admittime	dischtime
deathtime	admission_type	adm_provider_id	admission_location
discharge_location	insurance	language	marital_status
race	edregtime	edouttime	hospital_expire_flag

Table 4.3: admissions columns

procedures_icd

The *procedures_icd* table records information on medical procedures carried out while patients were hospitalized. International Classification of Diseases) (ICD) codes are used to identify procedures. ICD codes give academics and healthcare workers a standardized and systematic way to categorize a variety of medical procedures, making it simpler to interpret and assess the types of interventions patients undergo. This table includes ICD codes from two versions of the classification (ICD-9 and ICD-10). Procedures include surgeries, medical images, monitoring of physiological signals, etc. The table links these codes to the patient and admission information with "subject_id"and "hadm_id",

respectively, as well as to another table *d_icd_procedures* which contains the definition of the procedure.

procedures_icd					
subject_id	hadm_id	seq_num	chartdate	icd_code	icd_version

Table 4.4: procedures_icd table columns

prescriptions

The *prescriptions* table offers crucial details about the medications prescribed to patients during their hospital stays. Includes the names of the prescribed medications, their dosages, routes and timestamps of administration. Medications and pharmaceutical items in the United States are given a special identification number called the National Drug Code (NDC). It serves a variety of functions, including billing, inventory management, prescription tracking, and regulatory compliance and is a universal product identifier for pharmaceuticals.

The Generic Sequence Number (GSN) is given to particular generic medicine products. It is largely used for billing and reimbursement purposes, enabling medical professionals and pharmacies to appropriately bill for generic drugs.

Both identifiers play crucial roles in maintaining accurate records and facilitating the flow of information in healthcare systems.

prescriptions				
subject_id	hadm_id	pharmacy_id	poe_id	poe_seq
order_provider_id	starttime	stoptime	drug_type	drug
formulary_drug_cd	gsn	ndc	prod_strengh	form_rx
doe_val_rx	dose_unit_rx	form_val_disp	doses_per-24	route

Table 4.5: prescriptions columns.

drgcodes

It contains important information related to Diagnosis Related Groups (DRG). For the purposes of reimbursement and statistical analysis, DRG is a system used in the healthcare industry to divide patients into groups depending on their diagnosis and medical treatments.

MIMIC uses two DRG code types HCFA (from the former Health Care Financing Administration, now called the Centers for Medicare & Medicaid Services — CMS) and APR (All Patients Refined — divides patients into groups based on their illness severity and mortality risk as well as the reason they were admitted).

drgcodes				
subject_id	hadm_id	drg_type	drg_code	description

Table 4.6: drgcodes columns.

4.1.2 ICU Module

the *ICU* module, is designed to gather and preserve a wealth of information about patients admitted to the ICU facilities. It provides a thorough and in-depth look at the clinical care given to seriously ill individuals while they are hospitalized. This module is also organized into several tables. The majority of patients who are admitted to the ICU are in serious condition and require continuous observation, extensive medical care, and medication. Patients are continuously watched, resulting in regular updates on vital signal data, laboratory tests, and other indicators. ICU patients commonly have a variety of treatments and interventions, including invasive monitoring, dialysis, and mechanical breathing. All these characteristics of the complex medical practice in an ICU grant this module extensive information and greater data volume when compared to the *Hosp* module.

All of the following tables have a column *itemid* which is used to link to another table in this module. The *d_items* table has an entry for each unique value of *itemid* and the associated description of the specific clinical measurement or observation being recorded.

chartevents

The table stores most of the available charted data for a patient's ICU stay (e.g. vital signs, laboratory tests, ventilator settings, mental status, and so on). It is a fundamental and comprehensive part of the MIMIC database, capturing a wide range of clinical measurements and observations made during a patient's hospital stay. It provides valuable insights into the patient's condition over time and is crucial for clinical research, monitoring patient health, and supporting clinical decision-making.

chartevents			
subject_id	hadm_id	stay_id	caregiver_id
storetime	itemid	value	charttime
valuenum	valueuom		warning

Table 4.7: chartevents columns.

inputevents

inputevents includes information for intravenous fluid administration. These inputs include fluids delivered intravenously, medications (including dosages and administration routes),

blood transfusions or blood-related products, and even contrast agents used for imaging or radiological procedures.

The table serves as a valuable resource for understanding patient treatment regimens, tracking medication administration, and monitoring fluid balance.

inputevents			
subject_id	hadm_id	stay_id	caregiver_id
starttime	endtime	itemid	amount
amountuom	rate	rateuom	orderid
ordercategoryname	secondaryordercategoryname	formulary_drug_cd	ordercategorydescription
patientweight	totalamount	isopenbag	continueinnexdept
statusdescription	originalamount	storetime	linkorderid
	originalrate		

Table 4.8: inputevents columns

ingredientevents

ingredientevents is associated with *inputevents*. Each intravenous administration recorded in the latter table has associated ingredients in the former (e.g. water content, caloric information, etc.). This table is designed to support nutrition research.

outputevents

This table measures various outputs taken from patients in a hospital setting. Although they typically refer to urine output, these outputs can sometimes include other fluid outputs such as drainage from surgical wounds or chest tubes. This table contains useful data to examine a patient's fluid balance, which can be important when treating conditions like heart failure or renal disease.

outputevents				
subject_id	hadm_id	stay_id	caregiver_id	charttime
storetime	itemid	value	valueuom	

Table 4.9: output columns

procedureevents

The table captures and stores data related to medical procedures and interventions performed on patients hospitalized in the ICU. These procedures can range from surgical operations and diagnostic tests to therapeutic interventions and other medical treatments.

procedureevents					
subject_id	hadm_id	stay_id	caregiver_id	starttime	endtime
storetime	itemid	value	valueuom	location	locationcategory
orderid	linkorderid	ordercategoryname	ordercategorydescription	ndc	isopenbag
	continueinnextdept	statusdescription	ORIGINALAMOUNT	ORIGINALRATE	

Table 4.10: procedureevents columns

4.2 Data Preparation

4.2.1 Data Structure: creating a time series

In order to study a patient's health throughout their stay in the hospital it was necessary to create a new data format to store every available information in a time sequence using the timestamps in the original dataframes.

This section will focus on the process of creating time series from the raw data in the MIMIC database.

hadm_id	admittime	dischtime	deathtime	admission_type	admission_location	insurance	language	marital_status	race
22595853	2180-05-06 22:23:00	2180-05-07 17:15:00		URGENT	TRANSFER FROM HOSPITAL	Other	ENGLISH	WIDOWED	WHITE
22841357	2180-06-26 18:27:00	2180-06-27 18:49:00		EW EMER.	EMERGENCY ROOM	Medicaid	ENGLISH	WIDOWED	WHITE
25742920	2180-08-05 23:44:00	2180-08-07 17:50:00		EW EMER.	EMERGENCY ROOM	Medicaid	ENGLISH	WIDOWED	WHITE
29079034	2180-07-23 12:35:00	2180-07-25 17:55:00		EW EMER.	EMERGENCY ROOM	Medicaid	ENGLISH	WIDOWED	WHITE
26184834	2131-01-07 20:39:00	2131-01-20 05:15:00	2131-01-20 05:15:00	OBSERVATION ADMIT	EMERGENCY ROOM	Medicare	ENGLISH	MARRIED	BLACK/AFRICAN AMERICAN

Table 4.11: Sample from the admissions table with the original data — here are only visible the columns that were used.

Firstly, the *admissions* (table 4.11) and *drgcodes* data frames were processed in order to, respectively, use the timestamps for admission, discharge and death to create a new data frame (table 4.12) and retrieve the DRG codes to use only the patients that where hospitalized for a particular set of diseases (e.g. Circulatory system, Injuries or Poisonings, etc.). To put it in another way these codes were not put into the resulting time-series because all resulting samples were from the same set of conditions.

This new data frame's rows represented a time interval, meaning that the indexes were equally-spaced date-time stamps that started with the interval that contained the stored admission timestamp and ended in the interval that contained the discharge or death timestamp.

The new data frame's index was also used to create a second data frame with the target label, indicating if the patient died within each time interval.

	admission_type_URGENT	admission_type_EW EMER.	...	marital_status_WIDOWED	marital_status_MARRIED
2180-05-06 22:00:00	1	0	...	1	0
2180-05-07 00:00:00	1	0	...	1	0
...	...	...	...	...	...
2180-05-07 14:00:00	1	0	...	1	0
2180-05-07 16:00:00	1	0	...	1	0

Table 4.12: time-series sample of a single admission after its creation and the addition of that admission's table information.

The remaining information from the *admissions* data frame columns — *admission_type*, *admission_location*, *insurance*, *language*, *marital_status*, and *race* — was stored in columns

according to their timestamps.

As the remaining data frames (of the two modules — *Hosp* and *ICU*) were processed, new columns with their respective values were added to this new data frame — there were some additional steps to this process that will be addressed in the next section.

4.2.2 Encoding Data

MIMIC data is complex in the sense that each data frame has data from different origins and with different representations.

The easiest type to process is numeric data since it can be copied from the original data frames into the new time series. The problem with the numeric data type in MIMIC was that it was often associated with another categorical column's value that gave meaning to the numerical value. Therefore, inserting the original information into the new data structure required a step of one-hot encoding the data, meaning that for each categorical value in the original column, a new column was created that indicated the presence of each of those different categories with 1 and 0 otherwise (you can see how this was performed to the categorical columns *admission_type* and *marital_status* in table 4.12).

Then, in the case of the numerical columns, the newly created rows with value 1 were substituted by their values.

The next subsections will deal in detail with each different encoding technique that was employed to handle the various tables and their specificities.

drgcodes

To structure the patient data effectively for analysis, a two-fold data encoding process was undertaken. First, the DRG codes associated with each patient admission were encoded into MDC categories. DRG codes are fundamental in classifying hospital cases into clinically meaningful groups. In this research, we utilized a predefined list that mapped the HCFA-type DRG codes to MDC. These MDCs represent broader medical diagnostic categories, facilitating a more comprehensive analysis of patient outcomes within the chosen subset.

The second aspect of data encoding involved mapping the encoded MDC categories back to the patient admission records. This step involved associating each admission with its corresponding MDC category based on the DRG code linked to that admission. This encoding process enhanced the granularity of our dataset, allowing us to explore mortality predictions within the context of groups of diseases. Furthermore, it enabled us to account for the variations in disease severity and treatment responses that may occur across different MDCs.

The encoding of patient data based on MDC categories provided a structured and focused approach to the research. By narrowing the analysis to a specific MDC, the aim was to gain deeper insights into patient outcomes when analysing a specific group of diseases.

procedures_icd

The *procedures_icd* table's ICD codes were stored in two different versions, therefore it was required the use of an ICD-9 to ICD-10 mapping table — this way all rows in the dataframe have become represented by the most recent version of the code.

There are seven alphabetic characters in each code, and it is structured in a way that the first carachter is a section and each subsequent caracter is a subsection of the previous one.

In order to limit the creation of new features — to work around computational constraints — the instances were grouped by their ICD code's first 3 characters and the frequency of each unique value was counted. Values with more than 200,000 counts remained and the rest were filtered and replaced by their first charecter as to create less overall groups.

prescriptions

The data processing pipeline leveraged the RxNorm[43] Application Programming Interface (API) , a comprehensive resource provided by the National Library of Medicine.

This API allowed the retrieval of drug information, including RxNorm Concept Unique Identifiers (RxCUIs) and Anatomical Therapeutic Chemical (ATC) codes — used for classifying and measuring the consumption of drugs and other medical substances. It is maintained by the World Health Organization (WHO).

The API took *drug_name* – a free-text descriptions of the medication — from the *prescriptions* table as input and retrieved the ATC classification systen codes.

Analogously to the ICD, ATC codes are represented by 5 caracters (i.e. organized into groups at five different levels). The first level — represented by the first letter — signifies the anatomical main group of which there are 14. The first character classification for each drug was then added to the original instances of the *prescriptions*.

Alongside the encoding of the drug groups by their descriptions, it was also necessary to transform the original dose information in the table. This process ensured consistent numeric formating and dealt with the exceptions where a prescribed dose was represented by an interval. In the latter case, the mean of the two values was assigned to the admnistered medication feature.

chartevents, inputevents, ingredientevents, outputevents, procedureevents

These tables' processing pipeline can explained all at once due to their internal structured being similar. Since each instance is defined by its itemid, which links to *d_items* where it is defined.

In these cases, specially in *chartevents* the problem was its size — originally, this table had 2.3 Gb which would increase with the several one-hot encoding processes it had to go throught. Aditionally, and unlike the rest of the *ICU* tables, *chartevents* also has categorical

variables as its *itemid* associated values (the rest of tables' instances are just an itemid associated with a numerical value).

In order to not increase the size of the data, the *itemid*s with less value counts than 1 000 000 were discarded to compensate with faster processing times.

4.2.3 Parallel Processing

Medical datasets like MIMIC are very large and take time to process with traditional sequential processing methods. Parallel processing is crucial for its faster processing speed. This is achieved by dividing the main task into smaller, more manageable and simultaneously-processable subtasks. Also, this efficiently utilizes the multiple processing units that computers and servers come with. In other words, the large data frames are divided into smaller data frames to be processed simultaneously. In this study, this division was made by admission, which means that all information in a data frame that refers to a specific patient's hospital stay was processed by the same unit. Avoiding the issue of concurrent processes interfering with each other and corrupting data.

The parallel processing was executed with the Python library *multiprocessing*, specifically with its' *map* method that takes care of managing the creation and managing of the parallel processes when fed an iterable — object that can be looped over or iterated upon, allowing you to access its elements one at a time — that goes over each original data frame.

5 Methodology

5.1 General Aproach

This study aimed to investigate how to retrieve valuable information from real-world medical data and use it to feed LSTM-based models for sequential data analysis.

The following sections will detail the methodologies and techniques employed to address the three research questions of how to process and structure real-world medical data, how dimensionality reduction and the LSTM model influence prediction and finally how to choose a prediction window.

5.2 MIMIC as Data Source

The first step dealt with the choice of MIMIC as the source of the data, including its relevance to the study.

5.2.1 MIMIC Dataset Overview

The foundation of this research lies in the comprehensive and clinically rich dataset MIMIC. It is a freely accessible, de-identified database containing a wealth of patient data collected from the hospital stay — including ICU admissions — of the Beth Israel Deaconess Medical Center in Boston, Massachusetts. This dataset is renowned for its scale, diversity, and suitability for conducting critical care research. Chapter 4.1 has an in-depth breakdown of the information that composes the dataset and its internal structure.

One of the distinctive features of the MIMIC dataset is its rawness——the data was collected directly from patient records and captures the real-world complexity of medical cases. This unfiltered nature of the data was a crucial asset, as it allowed working with a true representation of clinical scenarios, making it particularly valuable for predictive modelling in the medical field.

The primary objective of this research is to leverage the raw and comprehensive nature of the MIMIC dataset to develop predictive models for death in the ICU. Hopefully, this investigation can be of service in developing tools for enhancing clinical decision-making

and patient care in the critical care setting, where timely and accurate predictions can make a significant difference in patient outcomes.

5.2.2 Selection of Patient Data

Within the expansive MIMIC database, the choice of patient data is a pivotal step in ensuring the relevance and effectiveness of this research. To address the primary objective of predicting patient mortality while considering the inherent challenges posed by high data availability variability in time-series medical data, a specific subset of patients was selected.

The focus of this research concentrated on patients diagnosed with MDC code 18: *INFECTIOUS & PARASITIC DISEASES, SYSTEMIC OR UNSPECIFIED SITES*. This particular MDC code was chosen deliberately, given its unique characteristics and relevance to the study's objectives.

The selection of MDC code 18 is underpinned by two key considerations:

1. **Data Availability**

   One of the fundamental challenges in working with EHR, especially in the context of ICU data, is the inherent variability in data availability over time. This variability is often influenced by the patient's medical condition and the necessity for frequent monitoring and interventions in the ICU. MDC code 18, encompassing infectious and parasitic diseases, exhibits a relatively consistent pattern of data availability. Due to the nature of these diseases, patients are typically subjected to rigorous monitoring and treatment protocols, resulting in a more uniform and densely populated temporal dataset compared to other MDC codes within the MIMIC database. This consistency in data availability is crucial for training robust machine learning models, such as LSTM networks, which thrive on temporal dependencies in the data.

2. **Clinical Relevance**

   The choice of MDC code 18 aligns with the clinical relevance of the study. Infectious and parasitic diseases can have a profound impact on a patient's prognosis, and predicting patient outcomes in such cases is of paramount importance. Furthermore, the decision to use MDC code 18, which has a high ratio of positive samples (see Table 5.1), is particularly advantageous as it ensures a sufficient number of target instances for training and evaluation, enhancing the model's ability to capture mortality patterns effectively.

	Admissions	Length of Stay (days)			Mortality		Age		Gender			
		Mean	Standard Deviation	Max	#	%	Mean	Standard Deviation	Female		Male	
									#	%	#	%
Hospital	16175	8.05	8.68	152.71	1695	10.4	62.4	17.6	6430	48.6	6795	51.4
ICU	6920	4.02	4.57	66.27	1527	22.1						

Table 5.1: MIMIC Statistics for Hospital and Intensive Care Unit admissions with patients with MDC 18

5.3 Data Processing

The effective utilization of the MIMIC dataset for predictive modelling required extensive data preprocessing to create a structured, time-series dataset that could serve as the foundation for building LSTM models. This section outlines the steps taken to prepare the data for modelling.

Section 4.2 has greater details on the creation of the data structure and its encoding from the raw dataset. That section outlined the steps taken to transform raw medical data into structured time series data for analysis, including data processing — data extraction and structuring, handling numeric and categorical data types and interpreting missing values —, encoding — transforming the original data into a meaningful structure that can be fed into DL models —, and parallel processing techniques.

5.3.1 Extraction of Medical Event Sequences

To study a patient's health throughout their hospital stay, a new data format was created to represent all available information in a time sequence using the timestamps from the original data frames. Initially, the *admissions* and *drgcodes* data frames were processed to extract key information.

The timestamps for admission, discharge, and death were used to create a new data frame representing time intervals. Each row in this new data frame represented a time interval, with equally-spaced date-time stamps starting from the interval containing the admission timestamp and ending in the interval containing the discharge or death timestamp. Additionally, a second data frame was created with the target label, indicating whether the patient died within each time interval. This target data frame had its timestamps shifted in time in order for every prediction window to match the patient status in the future.

Information from other data frames — both from the *Hosp* and *ICU* modules — was stored in columns corresponding to their timestamps. This approach allowed for the creation of a time-series dataset that captured patient information throughout their hospitalization.

5.3.2 Handling of Different Data Types

MIMIC data is complex, with each data frame originating from different sources and having different types of data. Handling these differences was a crucial step in the data

processing pipeline. Various strategies were employed to address missing values:

- **Numeric Data:** Numeric data from the original data frames were directly copied into the new time-series dataset.
- **Categorical Data:** Categorical columns underwent one-hot encoding. This process created new binary columns for each category within the original categorical column, indicating the presence (1) or absence (0) of each category. This encoding allowed for the retention of categorical information while making it suitable for DL models.

In cases where the absence of information could be interpreted as a numerical zero, a value of 0 was assigned. For instance, if no data was available for a particular medication or procedure during a specific time frame, it was assumed that none occurred, and 0 was recorded. However, for columns that represented physiological measures or other status evaluations, the timestamp received a value of NaN (Not-a-Number). These NaN values were retained for later handling within the data pipeline before the data was fed into the model.

5.3.3 Encoding

In the data preprocessing phase (Section 4.2.2), several encoding techniques were applied to enhance the dataset's usability:

- **DRG Codes:** A two-step process mapped HCFA-type DRG codes to broader MDC, facilitating a more comprehensive analysis of patient outcomes within disease groups.
- **ICD Codes:** ICD codes in the *procedures_icd* table were grouped based on their first three characters to reduce the number of distinct codes while preserving essential information.
- **Medication Data:** Medication information in the *prescriptions* table was processed using the RxNorm API to retrieve ATC codes, allowing for drug classification. The first character of the ATC code was added as a new feature, and dose information was standardized for consistency.
- **ICU Tables:** Tables like from the *ICU* module shared a common structured representation based on the original column *itemid*. To manage the large size of *chartevents* *itemid*s with fewer than 1 000 000 value counts were removed for efficiency. Similar preprocessing methods were applied to the other event tables to maintain dataset consistency.

These encoding techniques were instrumental in preparing the data for subsequent modelling and analysis, ensuring that the dataset had the data from the original data frames translated into a structure that could be handled by DL networks.

In summary, the data processing pipeline involved the extraction of medical event sequences, handling of missing data, encoding of DRG and ICD codes, and standardization of medication data. These steps were crucial in preparing the MIMIC dataset for LSTM model development while addressing the challenges posed by the dataset's complexity and variability. The resulting structured time-series dataset laid the foundation for subsequent modelling and analysis of patient mortality predictions.

5.4 Pre-Modelling Sample Adjustment

In the data preprocessing phase, following the retrieval and encoding steps, the dataset comprised separate files for each patient hospital admission. These individual samples were essential for capturing the distinct patient trajectories and events. However, a challenge arose from the inherent variability in the number and content of features across these samples due to differences in patient histories and available data.

5.4.1 Sample Adjustment

To address this issue, a two-step process was implemented to standardize the data across all admission samples.

1. Searching through each sample and collecting their unique columns, a list of all features was created.

2. After compiling the unique columns, a second pass through the sample files was performed. During this iteration, any missing columns in each sample were added to ensure uniformity across all samples. Missing values were filled with NaN to signify the absence of data for those specific columns. Additionally, the order of columns was standardized across all samples to create a consistent column structure.

This rigorous standardization process was imperative for maintaining data integrity and consistency when working with large and heterogeneous datasets. It ensured that all data sources had uniform column structures, enabling seamless subsequent analysis and modelling.

5.5 Windowing Process

In the data preprocessing pipeline, a crucial step involved dividing the original dataset into distinct windows. This section elaborates on the windowing process, emphasizing its significance in the context of medical data analysis and LSTM model compatibility. It also outlines the criteria used to determine the window size and any variations that were tested.

5.5.1 Rationale for Windowing

The utilization of windows in the dataset serves several pivotal purposes:

- **Temporal Analysis:**

  In the medical field, patient data often exhibits temporal dependencies, where the evolution of a patient's condition is closely tied to the passage of time. By dividing the data into discrete windows, we can capture these temporal patterns and allow the LSTM model to consider them when making predictions. This is particularly crucial in the context of intensive care, where timely interventions can be life-saving.

- **Sequencing:**

  LSTMs are inherently designed to work with sequential data. Splitting the data into windows effectively transforms the dataset into a sequence of observations. Each window encapsulates a segment of the patient's journey, allowing the LSTM to learn sequential relationships and dependencies between features.

5.5.2 Determining Window Size

The choice of window size is a critical decision in the windowing process. It determines the granularity at which the LSTM model operates and affects its ability to capture short-term or long-term dependencies. In this study, the following window sizes were explored: 12 (24 hours), 18 (36 hours), and 24 (48 hours) time intervals. These choices were based on several considerations:

- **Clinical Relevance:** Different medical conditions may exhibit varying time dependencies. A shorter window size (e.g., 12 hours) can capture rapid changes in a patient's condition, while a longer window size (e.g., 24 hours) allows the model to consider more extended trends and interventions.

- **Data Availability:** The availability of medical data, including the frequency of measurements and interventions, can influence the choice of window size. A smaller window size may be appropriate when data is collected at high frequencies, whereas a larger window may be suitable for less frequent measurements.

5.5.3 Target Adjustment

In the process of preparing the data, the target variable, which indicated patient mortality, required special consideration. Since the patient admissions were partitioned into windows for modelling purposes, a particular scenario emerged. In the case of patient deaths, some of the windows, specifically the last one within each admission, contained information that fell within the prediction window.

This posed a significant problem as the goal was to predict patient mortality a certain number of hours or days in advance. Having information from the same time frame as the

prediction window within the last window of a death-related sample would introduce data leakage, potentially leading to misleading model performance.

To address this leakage problem, a solution was implemented during the preprocessing phase. Data inside the prediction window was discarded from the last window of each admission, ensuring that the model received only information up to the point of prediction, preventing any inadvertent data leakage.

This targeted adjustment was crucial for ensuring the model's ability to make accurate predictions while adhering to the temporal constraints of the research question. It maintained the integrity of the prediction task by effectively removing information from the last window that could influence predictions beyond the desired forecasting horizon.

In summary, the windowing process played a pivotal role in preparing the dataset for LSTM modelling by capturing temporal patterns, enabling the analysis of patient data at different time intervals, and ensuring that predictions were made with the appropriate temporal context. Additionally, the selected window sizes were determined considering clinical relevance, data availability, and model complexity, ensuring a balanced approach to sequence-based prediction in the medical context.

5.6 Train, Validation, and Test Subsets

In the model development process, the initial step involves loading the dataset and the corresponding labels. The dataset is structured in the form of pickle files, with each file containing patient data. The issue arises with the underrepresentation of the *death* label in the dataset.

5.6.1 Class Imbalance Handling

Class imbalance is a significant challenge in many machine learning tasks, particularly in the healthcare domain. In the context of this study, class imbalance referred to the situation where one class *death* was markedly underrepresented — 1484 (9.3%) hospital admissions that culminated in patient death — compared to the other class that represented survival. To address this class imbalance, a strategy that computes class weights was implemented.

These class weights were determined in such a way that they were inversely proportional to the class frequencies. This means that the less frequent class *death* received a higher weight, emphasizing its importance during model training. The rationale behind this was to ensure that the model did not become biased towards the majority class and paid more attention to the minority class.

5.6.2 Stratified Split

The dataset was divided into three subsets: training, validation, and test sets. Stratified splitting was employed to guarantee that the distribution of classes within each subset closely mirrorred the overall class distribution in the entire dataset.

Stratification was pivotal because the study dealt with an imbalanced dataset. It safeguarded against situations where one of the subsets might have ended end up containing very few or even zero instances of the minority class. By maintaining the class distribution, it ensured that the model could effectively learn from both classes.

5.6.3 Weighted Random Sampler

A weighted random sampler was incorporated into the data loading process, designed to create data batches during model training. This sampler took into account the previously calculated class proportions.

During training, the weighted random sampler operated by selecting instances with a higher probability for the underrepresented class *death* and a lower probability for the overrepresented class — essentially repeating positive samples. This process served to balance the contribution of each class to the model's learning process.

These strategies held paramount significance in the context of dealing with imbalanced datasets, especially within this study where the accurate prediction of a rare event — patient death — was of utmost importance.

By effectively addressing class imbalance and guaranteeing representative data splits, this contributed to the development of a resilient and unbiased DL model for predicting patient outcomes.

5.7 LSTM Model

LSTM networks were employed to model and predict patient outcomes from time-series medical data.

5.7.1 Significance of LSTM for Time Series Data

LSTM networks are particularly well-suited for time series data due to their ability to capture sequential dependencies and handle long-range temporal relationships. In the context of our research, where we are predicting patient mortality based on time-series medical data, the significance of LSTM arises from several factors:

- **Temporal Dependencies:** Patient health data is inherently sequential, where each observation is influenced by previous events. LSTMs excel at modelling such temporal dependencies, making them ideal for capturing the evolving nature of a patient's condition over time.

- **Feature Learning:** LSTMs can automatically learn and extract relevant features from the input data, reducing the need for manual feature engineering. This is crucial in the medical domain, where relevant features may not be immediately obvious.

5.7.2 Other Model Parameters

In addition to the LSTM architecture, several other model parameters were considered and tested during our research. These parameters include:

- **Hidden Size:** The size of the hidden state in the LSTM, determines the model's capacity to capture complex patterns. It was selected from choices of 64, 128, 256, and 512.
- **Number of Layers:** The number of LSTM layers in the network. A higher number of layers can capture more intricate dependencies but also increases model complexity. We explored choices ranging from 2 to 1024.
- **Learning Rate:** The learning rate used during training, affects the convergence of the model. We sampled from a log-uniform distribution between 0.0001 and 0.3.
- **Batch Sizes:** Different batch sizes for training, validation, and testing data, were chosen from options of 64, 128, 256, and 512. Batch size affects training stability and memory usage.

These parameter variations allowed us to optimize the model's architecture and hyperparameters to achieve the best possible performance in predicting patient mortality based on time-series medical data.

5.8 Performance Metrics

To evaluate the performance of the predictive models for patient mortality was employed a set of well-established performance metrics tailored to the specific objectives of the research. These metrics provided comprehensive insights into the model's effectiveness in predicting patient outcomes.

5.8.1 Threshold Optimization for Binary Classification

In binary classification tasks, such as predicting patient mortality, the threshold represents the decision boundary that separates the two classes. The choice of this threshold can significantly impact the model's performance.

The challenge lies in striking the right balance between sensitivity and specificity, which measure, respectively, the ability to identify positive or negative samples — i.e. they measure the model's ability to minimize false negatives or false positives.

To address this trade-off between sensitivity and specificity, a ROC curve analysis was used. The ROC curve visualizes the model's performance across a range of thresholds. Each point on the curve represents a sensitivity-specificity pair for a given threshold. The threshold value chosen was the one that maximized the geometric mean.

$$geometric\ mean = \sqrt{TP\ rate * (1 - FP\ rate)}$$

5.8.2 AUROC

The AUROC curve metric is widely used for binary classification tasks, such as predicting patient mortality. It evaluates the model's ability to distinguish between the two classes (survival and death) by assessing the trade-off between true positive rate (sensitivity) and false positive rate. In our context, a higher score indicates that the model is better at discriminating between patients who survived and those who did not, which is crucial for timely interventions and resource allocation.

5.8.3 Precision, Recall, and F1-Score

Precision, recall, and F1-score are essential metrics for imbalanced datasets, where one class is significantly less prevalent than the other — which is the case for this study. Precision measures the accuracy of positive predictions, recall assesses the ability to capture all positive cases, and the F1-score combines both to provide a balanced evaluation. In this research, identifying true positives (correctly predicting deaths) while minimizing false positives (incorrectly predicting deaths) is critical for clinical decision-making and resource allocation.

6

Results and Discussion

This chapter presents the culmination of the research efforts, offering valuable insights into the complex realm of predicting patient mortality using LSTM networks. The study explored processing real-world medical data, constructing predictive models, and determining optimal temporal window size for accurate predictions.

The following sections will comprehensively explore the outcomes, implications, and discussions surrounding these key research inquiries.

6.1 Results

6.1.1 Feature Selection: Dimensionality reduction

To address the second research question — how dimensionality reduction and the LSTM model influence prediction — experiments were conducted with both the entire feature set and a reduced set of selected features. The full feature set comprised 664 columns, while the selected feature set contained around 302 features based on their missing value ratios.

With the full feature set, three trials were performed, in the same conditions, each resulting in different training times and test performance stated in table 6.1.

Trial	Training duration (hours)	AUROC	Sensitivity	specificity	Precision	F1-score
1	11.94	0.8244	0.7309	0.7671	0.1766	0.2845
2	10.93	0.8030	0.6866	0.7840	0.1707	0.2734
3	25.06	0.8543	0.7436	0.8400	0.2429	0.3662

Table 6.1: Training Times and AUROC scores for models using the whole sample.

For the selected feature set, we also conducted three trials with the results in table 6.2.

These results indicated that the selected feature set achieved competitive performance with significantly reduced training times, suggesting that feature selection based on missing value ratios can effectively streamline the model without sacrificing predictive performance.

Trial	Training duration (hours)	AUROC	Sensitivity	specificity	Precision	F1-score
1	9.81	0.8375	0.7346	0.8103	0.2123	0.3294
2	8.27	0.8164	0.6693	0.8077	0.1766	0.2795
3	7.80	0.7941	0.6427	0.8136	0.1800	0.2812

Table 6.2: Training Times and AUROC scores for models using dimension reduced sample.

6.1.2 Model Hyperparameter Tuning

To explore the optimization of the model, the reduced feature data set was used for hyperparameter tuning. Experiments were made with various model hyperparameters:

- Hidden Size — number of features in the hidden state was chosen between 64, 128 and 256;
- Number of Layers – number of recurrent layers of stacked LSTM networks was chosen between 2, 4, 8 and 16;
- Learning Rate — chosen randomly from an interval between 0.0001 and 0.9;
- Batch Size — the number of samples used in training the model was 64, 128 or 256

The combination of values for the different hyperparameters were chosen randomly for each trial. These trials were performed on the reduced feature data set.

The table 6.3 summarizes the outcomes of these experiments.

	hidden size	num_layers	learning rate	batch size	iteration	time (seconds)	loss
1	128	16	0.000256	128	10	8746.29	0.3846
2	64	4	0.000139	128	1	889.48	0.6719
3	64	4	0.000110	64	4	3399.69	0.5399
4	256	2	0.001543	128	10	10288.8	0.2436
5	256	4	0.182834	256	10	9100.23	0.1743
6	128	2	0.000498	256	1	914.02	0.6598
7	128	8	0.000696	128	2	2121.5	0.4827
8	256	8	0.002063	128	8	8372.6	0.2475
9	64	4	0.003927	128	8	7168.33	0.2355
10	64	16	0.212850	64	8	6972.99	0.2368
11	256	4	0.017269	128	10	8794.06	0.2119
12	64	16	0.001503	256	1	863.88	0.5631
13	64	16	0.000389	128	1	896.14	0.5863
14	128	2	0.006847	128	10	8684.3	0.1822
15	256	2	0.087035	256	10	10463.6	0.1657

Table 6.3: Results on Hyperparameter Tuning.

These findings underscored the influence of the hyperparameters to the model and the importance of hyperparameter tuning for optimal performance. The results provided insights into which combinations of hyperparameters yielded the best outcomes, such as lower loss and improved predictive ability.

6.1.3 Prediction Window Size

To address the third research question — What is the optimal prediction window duration for LSTM-based models when applied to medical data analysis? — a study was conducted to compare different-sized prediction windows.

The metrics for each window size are in table 6.4.

Window Size (hours)	Training duration (hours)	AUROC	Sensitivity	specificity	Precision	F1-score
24	8.63	0.8160	0.6822	0.8105	0.1896	0.2967
36	5.95	0.8302	0.6972	0.8274	0.3045	0.4238
48	3.09	0.8520	0.7675	0.7896	0.3404	0.4716

Table 6.4: Training Times and metric scores for different prediction window sizes.

These findings indicated that different prediction windows had an impact on training time, however, the impact was not as noticeable on the predictive performance. Thus, the choice of window size should align with the specific requirements and constraints of the clinical application.

In the subsequent sections, we delve deeper into the implications of these findings, discussing their relevance and shedding light on the broader context of predicting patient mortality using LSTM neural networks.

6.2 Discussion

This discussion chapter delves into the interpretation of the main findings from the results section. It will have the findings interpretations in the context of our research questions, elucidating their significance and implications for predicting patient mortality.

6.2.1 Processing and Structuring Medical Data for LSTM Models

The study addressed the first research question by demonstrating that comprehensive feature selection can significantly reduce the dimensionality of medical data without compromising predictive accuracy. This has practical implications for handling large-scale medical datasets, allowing for more efficient model training and inference.

6.2.2 Feature Selection: Dimensionality Reduction

The experiments involved training the predictive model with all available features and then with a subset of selected features based on their missing value ratios. The results indicated that while the model trained with all features achieved slightly higher AUROC values, the training times became longer.

This observation raised a pertinent consideration for practical applications where computational resources may be limited. In these scenarios, the adoption of the dimensionality-reduced model may be preferable. It allows for expedited model training and application, making it a pragmatic solution for resource-constrained settings.

Nevertheless, it is crucial to delve into the clinical implications of variable discardment. Variables characterized by high missing values may seem redundant, but they could represent critical medical measurements or procedures, that are infrequent or even rare, that offer valuable insights into a patient's overall condition and survival prediction.

The decision to retain or discard such variables should be taken in accordance with medical knowledge and clinical expertise. It merits a more profound clinical discussion to study whether the omitted information, despite its sporadic occurrence, holds the potential to significantly enhance the performance of survival prediction.

6.2.3 Model Hyperparameter Tuning

Regarding the second research question, the investigation into model hyperparameters explored the selection of different combinations and their influence on the model's forecasting performance by comparing the errors in its predictions.

- Hidden Size — models with 64 layers in the hidden state had the highest loss (average of 0.4618), while models with 256 layers had the lowest loss (average of 0.2086);

- Number of Layers – on average, models with 16 recurrent layers of LSTM networks performed the worst (mean loss of 0.4427) and the ones with 2 layers had better outcomes (mean loss of 0.3128);

- Learning Rate — models with higher learning rates had better performances, although the highest tested was a relatively low value of 0.212850. Also, the models which had the best performances in the first training iterations had the lowest learning rates — these models also had relatively high losses;

- Batch Size — the average loss for each batch size of 64, 128 and 256 was, respectively, 0.3884, 0.3607 and 0.3905. However, the randomly chosen trial hyperparameters heavily preferred the 256 batch size, which ended up over-represented.

Larger Hidden Sizes and more layers may increase model complexity but could lead to overfitting if not properly regularized. Which appears to have happened in the cases with the highest number of network layers. However, it wasn't verified in the case of the Hidden Size — models with 256 layers in their hidden states performed better. This may be explained by the high volume of input data.

Higher Learning Rates performed better, nevertheless, the values tested did not cover a sufficient range to reach any insightful conclusion on the optimal learning rate.

Overall, with the hyperparameters tested, the optimal model seems to have high hidden state sizes, 2 layers of stacked LSTM networks and a learning rate higher than 0.05.

6.2.4 Prediction Window Size

The selection of an appropriate prediction window size was a critical aspect of this study, as it directly influenced the practical utility of the predictive models in a clinical context. While it is common to assume that larger prediction windows may yield improved predictive performance by leveraging more data, this study's results indicated that the choice of window size did not have a great impact on the model's performance and, thus, should be guided by the clinical reality and its necessities.

In this investigation of various prediction window sizes, it became evident that predictive performance remained consistently reliable across a range of window sizes. This robustness highlighted the adaptability of the model to different temporal scales.

Additionally, the clinical relevance of smaller prediction windows cannot be overstated, particularly in the context of an ICU. Patients in ICUs often exhibit rapidly changing medical conditions that demand continuous monitoring and timely interventions. By adopting smaller prediction windows, evaluations can be conducted more frequently, with predictions made at shorter intervals.

This frequent status assessment, sometimes even twice a day, offers healthcare professionals a more granular understanding of a patient's evolving condition. It empowers them to promptly identify subtle changes or deteriorations, enabling timely interventions and personalized treatment adjustments.

Therefore, the preference for smaller prediction windows is not solely based on their performance equivalence to larger windows but also on their alignment with the clinical need for continuous, real-time monitoring and early intervention in the management of critically ill patients. This consideration underscores the importance of tailoring predictive models to the specific requirements and dynamics of healthcare settings.

6.3 Predicting Patient Mortality

This section undertakes a comprehensive analysis of the performance of our LSTM models in predicting patient death. It evaluates the implications of various metrics, such as AUROC, sensitivity, specificity, and precision. Furthermore, it assesses whether the model meets clinical or practical criteria for application in real-world scenarios.

6.3.1 AUROC

The LSTM model exhibited robust predictive performance, as evidenced by high AUROC values during testing. Across different experiments, the AUROC values consistently exceeded 0.80, with some models reaching values as high as 0.85. This indicates that the models possess the ability to distinguish between patients at risk of death and those not at risk with a high degree of accuracy.

6.3.2 Sensitivity and Specificity

Sensitivity and specificity are vital metrics when evaluating the performance of a predictive model in healthcare. Sensitivity measures the model's ability to correctly identify patients who are at risk of death (true positives), while specificity assesses the model's capacity to correctly identify patients not at risk (true negatives).

The models achieved balanced sensitivity and specificity, indicating their effectiveness in identifying both positive and negative cases. In particular, the model demonstrated sensitivity values of approximately 0.70, ensuring that it can capture a substantial portion of patients at risk. Simultaneously, specificity values exceeding 0.80 affirmed the model's capability to identify patients not at risk correctly.

6.3.3 Precision

The low precision values between 0.15 and 0.35 suggest that the model has a relatively high rate of false positives, indicating it sometimes incorrectly identifies that patients are at risk of mortality. This is common for models as they trade high sensitivity scores for lower precision.

In a clinical context, this trade-off is valuable since high sensitivity is crucial in ensuring that patients at risk are correctly identified. However, lower precision indicates a higher rate of false alarms which may lead to unnecessary medical interventions. Maximizing precision while not compromising sensitivity should be considered when applying a similar model in order to enhance the management of clinical resources.

7

Conclusion

In conclusion, this research has tried to yield insights into the prediction of patient mortality using LSTM neural networks applied to real-world medical data. Our study was anchored in addressing three aspects: Processing and structuring medical data for LSTM models, feature selection and choosing optimal prediction windows.

The research commenced by examining how to effectively process and structure complex real-world medical data from the MIMIC dataset for LSTM model utilization. It addressed issues related to data variability, missing values, and categorical features. The proposed data preprocessing and structuring methodology could serve as a robust foundation for future research endeavors in the domain of healthcare analytics.

The subsequent objective aimed to discern the impact of dimensionality reduction and the LSTM model on prediction. Experiments were conducted with both the entire feature set and a reduced set of selected features, demonstrating that comprehensive feature selection can significantly reduce dimensionality and training time while maintaining similar performances. This guarantees that a potential introduction of the model in the clinic could be deployed even in situations with computational restraints.

Finally, the study explored prediction window size for LSTM-based models in the context of medical data analysis. The study compared different-sized prediction windows and, revealed that varying prediction window sizes did not significantly affect the model's performance. Consequently, the choice of prediction window should be meticulously tailored to the specific clinical requirements, rather than being determined solely by a one-size-fits-all approach. For model deployment in fast-paced and high-stakes environments like the LSTM a model with a smaller prediction window might be the best choice since it offers patient status evaluation more frequently.

In clinical settings, the early identification of patients at risk of death is paramount for enabling timely interventions and potentially enhancing patient outcomes. A model's robust performance aligns with the exacting clinical prerequisites for such predictions.

In summary, this study has endeavoured to provide not only valuable insights into patient mortality prediction but also a methodological framework that can be leveraged in future healthcare analytics research. The journey has illuminated the significance of

dimensionality reduction, variable selection, and the personalized adaptation of prediction windows, all of which contribute to the broader understanding of real-world medical data and its transformative potential in clinical practice. The ultimate goal is that these contributions serve as catalysts for further advancements and refinements in the field of healthcare analytics and patient care.

Future Work

While this research has reached some conclusions in addressing these questions, several avenues for future research in this field remain unexplored.

Future studies could explore the applicability of transfer learning from other MDC codes or even other medical datasets to measure these strategies' generalization capabilities. Moreover, future work should involve the clinical validation of these findings. Collaboration with hospitals to test these strategies in real-world data from new patients and clinics will be crucial to evaluate the potential integration into clinical environments.

Having a structure for the medical data can be a starting point for future work for feature engineering that can create new insights into a patient's status at any point in their hospital stay, but also extract new informative temporal features from the time series.

Other future studies might want to delve into the model's interpretability. Understanding which variables are weighing in on the model's decisions is very important for algorithms like this to be introduced into healthcare environments so that healthcare professionals understand how the predictions are based on.

Finally, an essential trajectory for future research involves the seamless integration of continuous data streams in real time. This approach would empower the model to provide dynamic and ongoing assessments of patient risk, reflecting the evolving nature of clinical conditions. As such, it represents a pivotal avenue for further exploration in the field of predictive modeling in healthcare.

As data collection methods, modelling techniques, and medical knowledge continue to evolve, the field of predicting patient death remains dynamic and promising. Future research endeavors in this area hold the potential to make significant contributions to enhancing patient care, optimizing healthcare resource allocation, and ultimately saving lives.

Bibliography

[1] J. M. Lourenço, *The NOVAthesis LaTeX Template User's Manual*, NOVA University Lisbon, 2021. [Online]. Available: https://github.com/joaomlourenco/novathesis/raw/main/template.pdf (cit. on p. ii).

[2] H. Atasoy, B. N. Greenwood, and J. S. McCullough, "The digitization of patient care: A review of the effects of electronic health records on health care quality and utilization", *Annual review of public health*, vol. 40, pp. 487–500, 2019 (cit. on p. 1).

[3] K.-H. Yu, A. L. Beam, and I. S. Kohane, "Artificial intelligence in healthcare", *Nature Biomedical Engineering*, vol. 2, pp. 719–731, 10 2018-10, ISSN: 2157-846X. DOI: 10.1038/s41551-018-0305-z. [Online]. Available: https://www.nature.com/articles/s41551-018-0305-z (cit. on pp. 1, 4).

[4] A. Bharadwaj, O. A. El Sawy, P. A. Pavlou, and N. v. Venkatraman, "Digital business strategy: Toward a next generation of insights", *MIS quarterly*, pp. 471–482, 2013 (cit. on p. 1).

[5] R. A. Miller, "Medical diagnostic decision support systems–past, present, and future: A threaded bibliography and brief commentary", *Journal of the American Medical Informatics Association*, vol. 1, pp. 8–27, 1 1994-01, ISSN: 1067-5027. DOI: 10.1136/jamia.1994.95236141. [Online]. Available: https://academic.oup.com/jamia/article-lookup/doi/10.1136/jamia.1994.95236141 (cit. on p. 1).

[6] E. S. Berner, *Clinical decision support systems*. Springer, 2007, vol. 233 (cit. on p. 1).

[7] L. Liu, J. Shen, M. Zhang, Z. Wang, and J. Tang, "Learning the joint representation of heterogeneous temporal events for clinical endpoint prediction", *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, 1 2018-04, ISSN: 2374-3468. DOI: 10.1609/aaai.v32i1.11307. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/11307 (cit. on pp. 1, 12).

[8] Q. Zhong, J. W. Mueller, and J.-L. Wang, "Deep extended hazard models for survival analysis", *Advances in Neural Information Processing Systems*, vol. 34, pp. 15 111–15 124, 2021 (cit. on p. 1).

[9] D. Neil, M. Pfeiffer, and S.-C. Liu, "Phased lstm: Accelerating recurrent network training for long or event-based sequences", *Advances in neural information processing systems*, vol. 29, 2016 (cit. on pp. 1, 12).

[10] A. E. Johnson *et al.*, "Mimic-iii, a freely accessible critical care database", *Scientific data*, vol. 3, no. 1, pp. 1–9, 2016 (cit. on pp. 2, 14).

[11] M. Tayefi *et al.*, "Challenges and opportunities beyond structured data in analysis of electronic health records", *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 13, no. 6, e1549, 2021 (cit. on p. 3).

[12] K. P. Murphy, *Machine learning: a probabilistic perspective*. MIT press, 2012 (cit. on pp. 4, 5).

[13] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org (cit. on pp. 4–6, 8, 9).

[14] A. Geron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*, 2nd. O'Reilly Media, Inc., 2019, ISBN: 1492032646 (cit. on pp. 4, 9).

[15] L. Liu *et al.*, "Learning hierarchical representations of electronic health records for clinical outcome prediction.", *AMIA ... Annual Symposium proceedings. AMIA Symposium*, vol. 2019, pp. 597–606, 2019, Segment a patient's event sequence into sub-sequences, ISSN: 1942-597X. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/32308854%20http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC7153073 (cit. on pp. 5, 12).

[16] G. Lee, K. Nho, B. Kang, K.-A. Sohn, and D. Kim, "Predicting alzheimer's disease progression using multi-modal deep learning approach", *Scientific reports*, vol. 9, no. 1, p. 1952, 2019 (cit. on pp. 5, 11).

[17] M. Hossin and M. N. Sulaiman, "A review on evaluation metrics for data classification evaluations", *International journal of data mining & knowledge management process*, vol. 5, no. 2, p. 1, 2015 (cit. on p. 5).

[18] V. Bewick, L. Cheek, and J. Ball, "Statistics review 13: Receiver operating characteristic curves", *Critical care*, vol. 8, no. 6, pp. 1–5, 2004 (cit. on pp. 6, 7).

[19] M. P. LaValley, "Logistic regression", *Circulation*, vol. 117, no. 18, pp. 2395–2399, 2008 (cit. on p. 6).

[20] G. I. Webb, E. Keogh, and R. Miikkulainen, "Naıve bayes.", *Encyclopedia of machine learning*, vol. 15, pp. 713–714, 2010 (cit. on p. 6).

[21] J. Zou, Y. Han, and S.-S. So, "Overview of artificial neural networks", in 2008, pp. 14–22. DOI: 10.1007/978-1-60327-101-1_2. [Online]. Available: http://link.springer.com/10.1007/978-1-60327-101-1_2 (cit. on pp. 6, 8).

[22] C. M. Bishop, "Neural networks and their applications", *Review of Scientific Instruments*, vol. 65, pp. 1803–1832, 6 1994-06, ISSN: 0034-6748. DOI: 10.1063/1.1144830. [Online]. Available: http://aip.scitation.org/doi/10.1063/1.1144830 (cit. on p. 7).

[23] P. J. Drew and J. R. Monson, "Artificial neural networks", *Surgery*, vol. 127, no. 1, pp. 3–11, 2000 (cit. on p. 7).

[24] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A survey of convolutional neural networks: Analysis, applications, and prospects", *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 12, pp. 6999–7019, 2022. DOI: 10.1109/TNNLS.2021.3084827 (cit. on p. 8).

[25] L. Medsker, D. L. Jain, and B. R. L. N. Y. Washington, "Recurrent neural networks e dited by design and applications", 2001 (cit. on pp. 8, 9).

[26] A. Rehmer and A. Kroll, "On the vanishing and exploding gradient problem in gated recurrent units", *IFAC-PapersOnLine*, vol. 53, pp. 1243–1248, 2 2020, ISSN: 24058963. DOI: 10.1016/j.ifacol.2020.12.1342. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2405896320317481 (cit. on p. 9).

[27] S. Hochreiter and J. Schmidhuber, "Long short-term memory", *Neural Computation*, vol. 9, pp. 1735–1780, 8 1997-11, ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735. [Online]. Available: https://direct.mit.edu/neco/article/9/8/1735-1780/6109 (cit. on pp. 9, 10).

[28] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with lstm", *Neural computation*, vol. 12, no. 10, pp. 2451–2471, 2000 (cit. on p. 10).

[29] E. Hossain *et al.*, "Natural language processing in electronic health records in relation to healthcare decision-making: A systematic review", *Computers in Biology and Medicine*, vol. 155, p. 106 649, 2023-03, ISSN: 00104825. DOI: 10.1016/j.compbiomed.2023.106649. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0010482523001142 (cit. on p. 11).

[30] J. Koutník, K. Greff, F. Gomez, and J. Schmidhuber, "A clockwork rnn", 2014-02. [Online]. Available: http://arxiv.org/abs/1402.3511 (cit. on p. 11).

[31] P. Tang and X. Zhang, "Features fusion framework for multimodal irregular time-series events", 2022-09. [Online]. Available: http://arxiv.org/abs/2209.01728 (cit. on p. 11).

[32] Y. Si *et al.*, *Deep representation learning of patient data from electronic health records (ehr): A systematic review*, 2021-03. DOI: 10.1016/j.jbi.2020.103671 (cit. on p. 11).

[33] E. Choi, M. T. Bahadori, J. A. Kulas, A. Schuetz, W. F. Stewart, and J. Sun, "Retain: An interpretable predictive model for healthcare using reverse time attention mechanism", 2016-08. [Online]. Available: http://arxiv.org/abs/1608.05745 (cit. on p. 11).

[34] B. C. Kwon *et al.*, "Retainvis: Visual analytics with interpretable and interactive recurrent neural networks on electronic medical records", *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 1, pp. 299–309, 2019. DOI: 10.1109/TVCG.2018.2865027 (cit. on p. 11).

[35] D. Lu, K. Popuri, G. W. Ding, R. Balachandar, and M. F. Beg, "Multimodal and multiscale deep neural networks for the early diagnosis of alzheimer's disease using structural mr and fdg-pet images", *Scientific reports*, vol. 8, no. 1, p. 5697, 2018 (cit. on p. 11).

[36] B. Lei *et al.*, "Deep and joint learning of longitudinal data for alzheimer's disease prediction", *Pattern Recognition*, vol. 102, p. 107 247, 2020 (cit. on p. 11).

[37] M. M. Ghazi *et al.*, "Training recurrent neural networks robust to incomplete data: Application to alzheimer's disease progression modeling", *Medical image analysis*, vol. 53, pp. 39–46, 2019 (cit. on pp. 11, 12).

[38] G. Lee, B. Kang, K. Nho, K.-A. Sohn, and D. Kim, "Mildint: Deep learning-based multimodal longitudinal data integration framework", *Frontiers in genetics*, vol. 10, p. 617, 2019 (cit. on p. 11).

[39] C. Lee, J. Yoon, and M. V. D. Schaar, "Dynamic-deephit: A deep learning approach for dynamic survival analysis with competing risks based on longitudinal data", *IEEE Transactions on Biomedical Engineering*, vol. 67, pp. 122–133, 1 2020-01, ISSN: 15582531. DOI: 10.1109/TBME.2019.2909027 (cit. on p. 12).

[40] X. Cong, P. Z. Hadjipantelis, and J. L. Wang, "Semi-parametric joint modeling of survival and longitudinal data: The r package jsm", *Journal of Statistical Software*, vol. 93, pp. 1–29, 2020, ISSN: 15487660. DOI: 10.18637/jss.v093.i02 (cit. on p. 12).

[41] S. Liu, X. Wang, Y. Xiang, H. Xu, H. Wang, and B. Tang, "Multi-channel fusion lstm for medical event prediction using ehrs.", *Journal of biomedical informatics*, vol. 127, p. 104 011, 2022-03, ISSN: 1532-0480. DOI: 10.1016/j.jbi.2022.104011. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/35176451 (cit. on p. 13).

[42] Y. An, Y. Liu, X. Chen, and Y. Sheng, "Tertian: Clinical endpoint prediction in icu via time-aware transformer-based hierarchical attention network", *Computational Intelligence and Neuroscience*, vol. 2022, pp. 1–13, 2022-12, ISSN: 1687-5265. DOI: 10.1155/2022/4207940 (cit. on p. 13).

[43] National Library of Medicine, *Rxnorm*, https://www.nlm.nih.gov/research/umls/rxnorm/, Accessed on July 19, 2023, 2023 (cit. on p. 21).