KU LEUVEN

ARENBERG DOCTORAL SCHOOL
Faculty of Engineering Science

Length of Stay prediction for Hospital Management

Lyse Naomi Wamba Momo

Supervisors:
Prof. dr. ir. B. De Moor
Em. Prof. dr. F. Rademakers

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor of Engineering Science (PhD): Electrical Engineering

October 2024

Length of Stay prediction for Hospital Management

Lyse Naomi WAMBA MOMO

Examination committee:
Prof. dr. ir. L. Vandewalle, chair
Prof. dr. ir. B. De Moor, supervisor
Em. Prof. dr. F. Rademakers, supervisor
Prof. dr. ir. C. Vens
Prof. dr. ir. A. Bertrand
Prof. dr. L. Houthuys
(VUB)
Prof. dr. L. Peeters
(U Hasselt)

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor of Engineering Science (PhD): Electrical Engineering

October 2024

© 2024 KU Leuven – Faculty of Engineering Science
Uitgegeven in eigen beheer, Lyse Naomi Wamba Momo, Celestijnenlaan 200A box 2402, B-3001 Leuven (Belgium)

Alle rechten voorbehouden. Niets uit deze uitgave mag worden vermenigvuldigd en/of openbaar gemaakt worden door middel van druk, fotokopie, microfilm, elektronisch of op welke andere wijze ook zonder voorafgaande schriftelijke toestemming van de uitgever.

All rights reserved. No part of the publication may be reproduced in any form by print, photoprint, microfilm, electronic or any other means without written permission from the publisher.

Preface

A well known African proverb says, *"It takes a village to raise a child"*. This proverb translates almost perfectly my trajectory, including all that preceded this work and the 5-years doctoral journey. I am thus very grateful for this 'village' and the myriad of help I received which I will not even be able to singularly enumerate. Pardon me in advance if your name is not written down here but know, you are part of this work.

Thank you Prof. Bart, for at least three reasons. First, for the opportunity to do a PhD in your research lab. Being paid to learn is quite an underrated flex in my opinion. Secondly, for your short though very sharp advises and your comprehension when situations may arise and thirdly, for offering me a paid additional year, without which, it would not have been possible for me to finish.

I am deeply grateful to Prof. Frank, my co-supervisor. Your continuous availability to review, re-review, and critique any research work I emailed you was invaluable and motivating. Thank you for consistently attending our weekly meetings and serving as the bridge to reach any staff member of UZ Leuven to facilitate data access processes. One instance is when you convinced Professor Dirk Vlasselaers to be my reference for accessing the Amsterdam University Medical Centre database. I also appreciate your continuous medical insights during the post-analysis of the research findings. Lastly, thank you for the detailed post-analysis of my very first rejection email when our systematic literature review paper did not pass the review process.

My appreciation to the Flanders AI Research (FLAIR) program and its facilitators for establishing this research initiative that brings together academicians and people from the industry. Engaging with researchers of various levels at meetings, workshops, and conferences was an enlightening experience that exposed me to diverse research areas and facilitated valuable networking opportunities.

My heartfelt thanks to the management staff of the University Hospital of

Leuven, especially Wouter Voorspoels and Christiaan Franz. I appreciate your participation and contributions during the weekly meetings, your constant interaction during the data wrangling process, and your valuable inputs that gave me a true appreciation of the management of a real hospital.

I am grateful to every member of the jury and the supervisory committee for their support over the past four years, particularly during the yearly assessments. My appreciation to Prof. Vandewalle for accepting to serve as the chair of the examination committee. My gratitude to, Prof. Houthuys and Prof. Peeters for accepting to serve as examiners of this work. Thank you, Prof. Bertrand for your constant availability at each year's assessment and for your insightful questions, one of which even helped me to anticipate and better respond to the reviewers of my first journal paper. A special appreciation goes to you, Prof. Vens, for your inputs over the years and also the engaging conversations at my posters and demos whenever we met at a conference or a workshop. It was always a pleasure for me to update you and your students on my challenging journey towards data access.

I am sincerely thankful to Prof. Nkemzi, one of my mentors and my supervisor during my first Masters at the University of Buea. Thank you Prof., for always encouraging me and for offering me a laser point, torch and ball pen, a year ago saying (sarcastically) "if you have not started writing your dissertation because you do not have a pen, then take this one and have it done".

One of the advantages of being part of a large research lab like ours is the opportunity to meet, interact, collaborate, laugh, and learn with many people. I am especially grateful to my senior colleagues: Dr. Tina, for warmly welcoming me to the lab, inquiring about my projects during my first year, checking on me, and promising to attend my defense. Your reassurance and reminder that each PhD journey is unique yet will reach a happy conclusion were invaluable. I still cherish the walk you invited me on after the COVID lockdown, on the other side of Leuven station. Dr. Thibaut, for providing a quick and efficient tour of the ESAT building on my first day and for consistently sharing valuable input, just as Dr. Hannes did. Dr. Xi Shi, for accepting to collaborate on the systematic literature review paper. Above all, thank you all for the inspiration. My heartfelt thanks to Willem and Giulia, the first colleagues I collaborated with to design the strategy for requesting data from the University Hospital. I appreciate your support when meeting the IT management at UZ before COVID-19. Willem, thank you for helping me write the ethical document to establish the collaboration with the hospital, allowing me to use your server for my first journal paper experiments, and for all your advice. I am deeply grateful to Dr. Katrien for reviewing my manuscript during the third round of journal review. Thank you for your patience and availability to discuss issues related to the system identification and modeling course. Many thanks to Kostas and

Sartak, my fellow teaching assistants, for creating a supportive and positive work environment. My sincere gratitude also goes to Dr. Mauricio for always being available to discuss research-related matters or not and for his constant support. Thank you to my first-floor colleagues for the wonderful ambiance and laughs: Fernando, Giulia, Yoeri, Kostas, Melanie, Thomas, Zahra, Vincent, Jaldert, Axel, Alexandre, and Yamini. It was always enjoyable to knock on each other's offices, check in, and discuss the progress of our research. Although I didn't share many lunch and coffee breaks with my fourth-floor colleagues, we always had a great time on the dance floor at the departmental (ESAT) parties. It was a well-deserved good time after our collective "research-aholics non-anonymous" sessions. Thank you, Sibren, Lucas, Christof, Bram, and Oliver. Special thanks to Lola for your natural support and our French conversations, to Giulia for the carbonara at your place and our morning Teams calls, and to Zahra for the sweet tea, the runs, the Iranian treats and for reviewing my dissertation. Finally, my sincere appreciation to Vincent, the best collaborator I had throughout my PhD. Your constant availability to review my work, discuss research ideas, collaborate on projects, listen to me on tough days, and chat in Spanish was truly appreciated. The academic trips to Gran Canaria (with Kostas) and Limassol were fun, of course, academically fun.

I extend my deep gratitude to the administrative and technical staff at STADIUS, to Ida, Aldona, Elsy, John and Maarteen for your constant availability to resolve pressing issues I could have.

My PhD journey was quite challenging, but the Black in AI community was instrumental in helping me through it. It all began when I was selected for the BlackAIR summer research program and met incredible Black women like Dr. Lyne Tchapmi, Nyalleng Moorosi, Dr. Flora Tasse, and Prof. Elaine Nsoesie. Special thanks to Flora for encouraging me throughout my doctoral journey, and to Nyalleng for laughing at me when I was lamenting my lack of achievements in my second year. Prof. Elaine, my deepest gratitude goes to you, for saving me in February 2024 when I was in a mental struggle from an 18-month minor review process at the journal. Later, I joined the Black in AI organization and met incredible people like its CEO Gelyn Watkins, Marquita Riggins, and many others. Working together to organize the Black in AI workshop at the NeurIPS conference allowed me to showcase my management skills and experience the vibrant atmosphere of these conferences. I am immensely grateful for the opportunity to create an environment where we can shine. Thank you to everyone I collaborated with on this journey.

Now, a tribute to my family and friends which will be written in French because they are all French speakers. The preface will then be concluded in English.

Un immense merci à mes parents pour votre sacrifice et votre accompagnement

tout au long des années jusqu'à ce jour. Je vous dois tout et j'espère que cette thèse vous rendra fiers. Vous avez toujours été là, dans les plus petites choses, comme venir me chercher à l'agence de voyage à 5h du matin quand je revenais de Bamenda, comme dans les grandes choses, comme financer entièrement mes 2 années de Master en Belgique à plus de 20 000 euros à l'avance. Merci à ma grand-mère, Maman Denise, pour les appels et les prières. Encore une fois, ***meŋ á syakne***!!!

Un merci spécial à ma grande sœur Leslie, pour tes encouragements constants tout au long de la thèse, jusqu'à la dernière minute. Merci à ma grande sœur Flavia pour les cadeaux que tu rapportes du Cameroun sans même demander chaque fois que tu viens en Belgique, et surtout pour les photos/vidéos drôles de mes neveux et de ma nièce. Un merci à ma petite sœur Gisèle et à mes petits frères, Junior et Guy, pour vos encouragements. Merci à ma cousine préférée Bibi pour les appels vidéos avec Nunu, Manon et Ryan, pour arbitrer des problèmes que je ne comprends pas toujours, et pour la bonne nourriture quand je suis au Cameroun. Un merci à ma tante préférée, Maman Christelle, merci pour tout.

Un merci particulier à mes neveux Patxi, Sasha, Kevyn et à mes nièces, Opale et Elya pour les fous rires et les jeux d'enfants qui m'ont permis plus d'une fois de penser à autre chose.

À mes beaux-frères, Charles, pour mon premier 13 km de course pendant la COVID-19, et Ernest, pour les discussions sur la politique camerounaise et les restos en Belgique comme au Cameroun.

À mes ami(e)s devenus frères et sœurs, Darel, Yannick, Dominique, Bisona, Vanessa et Sibelle, merci à chacun(e) de vous pour les encouragements et les fous rires.

À celles rencontrées dans le cadre professionnel, devenues des amies pour la plupart, les femmes de Women in Machine Learning and Data Science, Yaoundé-Chapter, merci à vous toutes pour les encouragements.

Un merci immense à celle que j'appelle quand ça ne va pas, où je vais et où je suis toujours bien accueillie, ma mère Stéphanie. Docteure en mathématiques, tu m'as montré le chemin et m'as aidée sur la route. Merci à toi, ton mari, Papa Eric, et à mes petits frères Dylan et Bradley et ma petite sœur Cierra.

À mes ami(e)s docteurs ou thésards, merci pour le soutien, Dr. Milaure, Dr. Yves, Dr. Geraldin, Willy, William, Damola, Tosin, 1000 mercis. Une mention spéciale au Dr. Marcellin Atemkeng pour ta bienveillance, tes conseils académiques et ta constante disponibilité.

Enfin, merci aussi à mes amis, Nnamdi, Rodin, Stevye, Zeddy, Yannick Tedom et Gautier.

Despite the challenges I faced, I am grateful for this whole PhD journey for the number of things it taught me among which: patience, focus, diligence, organized thinking and maturity. During the Black in AI Summer research program, we had each week on Thursday, a talk from a Black woman, excelling in the field. Each talk was particular and left a mark on me, on how representativity matters. I thought, if they made it, so I can.

In this part of the world, that is in the West, in 2024, it is difficult to imagine that from where I come from, Cameroon-Africa, the education and curiosity of young girls is still threatened by outdated and insensitive 'cultures' and 'traditions'. This thesis is thus dedicated to them, that they get the full right and freedom to pursue their dreams. As the award-winning author Chimamanda Ngozi Adichie rightfully said, "Culture does not make people. People make culture. If it is true that the full humanity of women is not our culture, then we can and must make it our culture".

Abstract

Providing compassionate, accessible, and high-quality care while also containing cost is the core mission of a hospital. For this to be achieved, all healthcare services must be efficiently planned from admission up to discharge. Effective handling of these services is determinant to the hospital throughput and can be quantified via measures like the Length of inpatient Stay (LoS). The LoS or the duration of stay is simply, the length of time a bed is occupied between admission and discharge, within a single admission episode. In an ideal world, hospital managers would know in advance, i.e., as soon as admission or even before, for each planned or urgently admitted patient, based on their admission diagnostic, how much resources should be mobilized. Such information would be valuable to ensure optimal patient flow within the hospital.

The prediction of LoS is a known problem that is actively been studied and reported in the literature. The presence of publicly available datasets has driven research towards the development of sophisticated models to capture the complexities in patient data in order to understand and predict LoS, without necessarily adapting these methods to a practical hospital setting where additional challenges can emerge. These challenges include: data accessibility, real-time data modeling, interpretability and the understanding and use of these models by the ultimate end-users (clinicians, capacity planners and hospital management teams).

This thesis, carried out in collaboration with the University Hospital of Leuven, contributes to the literature by examining the LoS prediction problem from the perspective of a real-world hospital environment.

Firstly, this work introduces a unique application of domain adaptation and discriminative learning for LoS prediction across heterogeneous feature spaces to facilitate data accessibility and handle issues related to data heterogeneity across multiple domains. Domain adaptation enables to leverage shared patient information across different hospital units by extracting and transferring

learned weights from one hospital unit (the source domain) to another (the target domain). By recognizing the singularity of each unit of admission, this work accommodates domain adaptation, i.e., weights fine-tuning across a heterogeneous feature space between the two domains. Discriminative learning then enables the simultaneous training of the two sets of features (coinciding and non-coinciding features), by updating a feature set weights at a slower rate than the others. In a practical hospital setting with d number of medical departments (each defined by its specialty), the heterogeneous feature space across units, domain adaptation and discriminative learning are the three key elements of this work that can enhance its transferrability and use.

Secondly, we address the important problem of explainability in deep learning models by visualizing the latent spaces of sequential models used for predicting hourly remaining Length of Stay (LoS). Through dimensionality reduction techniques, we demonstrate how these models learn patterns in the data over time, revealing that even with just 5-6 hours of early admission data, it is possible to differentiate short-stay patients from those with extended stays who are at high risk of in-hospital mortality. Finally, we illustrate how these prediction systems can be integrated into a clinical decision support system, providing real-time insights that can enhance capacity planning.

Finally, this work evaluates the contribution of different data modalities for the continuous prediction of remaining LoS, by utilizing cross-attention mechanisms to extract the latent correlations among these modalities. Our approach handles missing modalities both at the time and the patient levels. The modalities considered include, timestamped thoracic X-rays, timestamped clinical reports and timestamped physiological parameters (such as laboratory results and vital parameters), along with static admission and demographic information. A time-dependent multimodal fusion is implemented to simulate a hospital environment where information is processed at the charted time.

The contributions made in this thesis aim at taking an initial step towards the real-time implementation and adoption of AI-driven clinical decision support systems within healthcare facilities regarding LoS.

Beknopte samenvatting

Het bieden van toegankelijke en hoogwaardige zorg rekening houdend met kostenefficiëntie vormt de kernmissie van een ziekenhuis. Om dit te bereiken, moeten alle zorgdiensten efficiënt worden gepland, vanaf de opname tot en met het ontslag van een patiënt. Het effectief beheer van deze diensten is bepalend voor de doorstroming in het ziekenhuis en kan worden gekwantificeerd door middel van parameters zoals de ligduur (LoS). De ligduur, of de opnameduur, is simpelweg de tijdsduur dat een bed bezet is tussen opname en ontslag van één patiënt, binnen één enkele opnameperiode. In een ideale situatie zouden ziekenhuismanagers van tevoren, dat wil zeggen direct bij opname of zelfs eerder, voor elke geplande of spoedopname, op basis van de diagnose bij of vóór opname reeds weten hoeveel middelen moeten worden gemobiliseerd voor de doorstroom van de desbetreffende patiënt. Dergelijke informatie zou waardevol zijn om een optimale patiëntstroom binnen het ziekenhuis te waarborgen.

Het voorspellen van de ligduur is een bekend probleem dat actief wordt bestudeerd in de literatuur. De beschikbaarheid van openbaar toegankelijke datasets heeft het onderzoek gestuurd naar de ontwikkeling van geavanceerde modellen om de complexiteit in patiëntgegevens te begrijpen om op basis daarvan de ligduur te voorspellen. Het implementeren en toepassen van deze methoden in een praktische ziekenhuisomgeving zorgt voor extra uitdagingen. Deze omvatten: beschikbaarheid van de data, het modelleren in real-time, de interpretatie van de resultaten die volgen uit deze modellen door de uiteindelijke eindgebruikers (clinici, capaciteitsplanners en ziekenhuismanagementteams).

Dit manuscript, uitgevoerd in samenwerking met het Universitair Ziekenhuis Leuven, draagt bij aan de literatuur door het probleem van de voorspelling van de ligduur te onderzoeken vanuit het perspectief van een werkelijke ziekenhuisomgeving.

In dit werk, wordt, ten eerste, een unieke toepassing geïntroduceerd met betrekking tot het adaptief leren van de desbetreffende modellen in termen

van de data binnen dit domein. Bijkomend wordt gebruik discriminerend leren gebruikt voor de voorspelling van de ligduur. Hierbij wordt gebruik gemaakt van heterogene representaties van de data om de verwerking van de data vergemakkelijken. Daarnaast laat dit ook toe om problemen die volgen uit de heterogene aard van de data aan te pakken. Het afstemmen van de modellen en methoden op het specifieke domein laat toe om gedeelde patiëntinformatie over verschillende ziekenhuisafdelingen te benutten door de gewichten uit de geleerde modellen in de ene ziekenhuisafdeling (brondomein) naar de andere (doeldomein) over te dragen. Door de singulariteit van elke opname-eenheid te herkennen, maakt dit werk domeinaanpassing mogelijk. Meer specifiek wil dit zeggen dat geleerde gewichten uit bepaalde brondomeinen kunnen worden afgestemd op heterogene representaties van de intersectie tussen brondomein en doeldomein. Door vervolgens discriminerend leren toe te passen kunnen de modellen gelijktijdig leren van de resulterende sets aan kenmerken. dwz, de set met features uit de intersectie en de set met unieke features in elk van de domeinen (i.e., overeenkomende en niet-overeenkomende kenmerken). De term discriminatie volgt uit het gegeven dat voor de ene set gewichten sneller, of net trager, worden bijgewerkt gedurende het leerproces. In een praktische ziekenhuisomgeving met een groot aantal medische afdelingen, elk gedefinieerd door zijn specialiteit, zijn de heterogene representaties van de lokale data, de toepassing van het domein adaptief, en discriminerend leren van modellen, de drie kernelementen, die het leren van modellen die toepasbaar zijn in verschillende doeldomeinen, mogelijk maken.

Ten tweede behandelen we de interpretatie en verklaarbaarheid van de deep learning-modellen door de aangeleerde representaties van sequentiële modellen, die worden gebruikt voor het voorspellen van de resterende ligduur per uur, te visualiseren. Door middel van technieken voor dimensiereductie tonen we aan hoe deze modellen in de loop van de tijd patronen in de gegevens leren. Hierbij laten we zien dat ze zelfs met slechts 5-6 uur aan data, patiënten met een kort verblijf kunnen onderscheiden van patiënten met een lang verblijf die daarenboven een hoog risico op sterfte in het ziekenhuis hebben. Ten slotte laten we zien hoe dergelijke voorspellingssystemen kunnen worden geïntegreerd in een klinisch beslissingsondersteuningssysteem, waarbij real-time inzichten worden verkregen die de capaciteitsplanning kunnen verbeteren.

Tot slot beoordeelt dit werk de bijdrage van verschillende gegevensmodaliteiten aan de continue voorspelling van de resterende ligduur door gebruik te maken van cross-attention-mechanismen om de correlaties tussen deze modaliteiten te berekenen uit de data. Onze benadering gaat om met ontbrekende modaliteiten op zowel het tijd- als het patiëntniveau. De overwogen modaliteiten omvatten thorax röntgenfoto's, klinische rapporten en fysiologische parameters (zoals laboratoriumresultaten en vitale parameters), allen geïndexeerd per tijdseenheid,

samen met statische opname- en demografische informatie. Er wordt een tijdsafhankelijke multimodale fusie geïmplementeerd om een ziekenhuisomgeving te simuleren waarin informatie op het geregistreerde tijdstip wordt verwerkt.

De bijdragen in dit proefschrift beogen een eerste stap te zetten richting de realtime implementatie en adoptie van AI-gestuurde klinische beslissingsondersteunende systemen binnen zorginstellingen op vlak van ligduur.

List of Abbreviations

AI	Artificial Intelligence
ANN	Artificial Neural Network
$ANOVA$	Analysis of Variance
$AUMC$	Amsterdam University Medical Center
$BERT$	Bidirectional Encoder Representations from Transformers
$CDSS$	Clinical Decision Support System
CPU	Central Processing Unit
$EHDS$	European Health Data Space
EHR	Electronic Health Record
EU	European Union
FCN	Fully Connected Neural Network
FFN	FeedForward Neural Network
$FLAIR$	Flanders AI Research
GPU	Graphic Processing Unit
GRU	Gated Recurrent Unit
HDA	Health Data Agency
ICU	Intensive Care Unit
IFM	Intermediate Fusion Model
IHM	In-Hospital Mortality

LFM	Late Fusion Model
LLM	Large Language Model
LoS	Length of Stay
$LSTM$	Long Short-Term Memory
MAE	Mean Absolute Error
$MIMIC$	Medical Information Mart for Intensive Care
MLM	Masked Language Modeling
MLP	Multi-Layer Perceptrons
MSE	Mean Squared Error
$MSLE$	Mean Squared Logarithmic Error
RCT	Randomized Clinical Trial
$ReLU$	Rectified Linear Unit
$t-SNE$	t-distributed Stochastic Neighbor Embedding
TCN	Temporal Convolutional Neural Network
$Tukey'sHSD$	Tukey's honest significance test
$UMAP$	Uniform Manifold Approximation and Projection
UZ	Universitair Ziekenhuis
VSC	Vlaams SuperComputer Centrum

Contents

Abstract vii

Beknopte samenvatting ix

List of Abbreviations xiii

Contents xv

List of Figures xix

List of Tables xxv

1 Introduction 1
- 1.1 Hospital Management . . . 2
- 1.2 LoS prediction for hospital management . . . 3
  - 1.2.1 The meaning of LoS prediction . . . 3
  - 1.2.2 Secondary use of Electronic Health Records (EHRs) . . . 3
  - 1.2.3 Literature review on LoS prediction . . . 5
- 1.3 Context of our research . . . 10
- 1.4 Structure of the thesis and our contribution . . . 10
- 1.5 Acknowledgments . . . 14

2 Background 17
- 2.1 The Data . . . 17
- 2.2 The Learning Process . . . 21
- 2.3 Model architectures . . . 22
  - 2.3.1 Linear regression . . . 23
  - 2.3.2 FeedForward neural networks . . . 23
  - 2.3.3 Gated recurrent neural networks . . . 25
  - 2.3.4 Temporal convolutional neural network . . . 27
  - 2.3.5 Attention-based neural network . . . 28

3 Length of Stay Prediction for hospital management using domain adaptation 33
3.1 Introduction . . . 34
3.2 Key Contributions . . . 35
3.3 Methodology . . . 36
3.3.1 Datasets . . . 36
3.3.2 Preprocessing . . . 36
3.3.3 Prediction task . . . 38
3.3.4 Models . . . 38
3.3.5 Domain adaptation . . . 38
3.3.6 Interpretability using Expected gradients: . . . 42
3.3.7 Evaluation metrics: . . . 42
3.4 Results & Discussion . . . 43
3.4.1 Hyperparameter Optimization . . . 43
3.4.2 Model performance . . . 45
3.4.3 Additional Analyses . . . 52
3.5 Conclusion . . . 57
3.6 Additional experiments . . . 58
3.6.1 Naive prediction models . . . 58
3.6.2 Erratum . . . 59

4 Understanding Deep Learning for Length of Stay prediction via Latent space visualization 61
4.1 Introduction . . . 62
4.2 Key Contributions . . . 64
4.3 Methodology . . . 64
4.3.1 Dataset . . . 64
4.3.2 Preprocessing . . . 65
4.3.3 Prediction task . . . 68
4.3.4 Experimental Settings . . . 68
4.3.5 Latent space projection and dashboard construction . . . 69
4.3.6 Evaluation Metrics . . . 70
4.4 Results & Discussion . . . 70
4.4.1 Hyperparameter Optimization . . . 70
4.4.2 Model Performance . . . 70
4.4.3 Latent Space Visualization . . . 71
4.4.4 Feature Importance . . . 79
4.4.5 Additional Analysis . . . 81
4.5 Conclusion . . . 82

5 Multimodal Learning Through Time with Cross-Attentions for Length of Stay Prediction 84
5.1 Introduction . . . 85

5.2 Key Contributions . . . 86
5.3 Methodology . . . 87
5.3.1 Dataset . . . 87
5.3.2 Preprocessing . . . 88
5.3.3 Prediction Task . . . 91
5.3.4 Models . . . 91
5.3.5 Experimental Settings . . . 95
5.3.6 Evaluation Metrics . . . 96
5.4 Results & Discussion . . . 97
5.4.1 Hyperparameter Optimization . . . 97
5.4.2 Model Performance . . . 97
5.4.3 Model Performance per class of LoS . . . 98
5.5 Conclusion . . . 99

6 Conclusion and Future Perspectives 101
6.1 Scientific conclusions and limitations per chapter . . . 101
6.2 Cross-chapter discussion . . . 103
6.3 Practical Takeaways . . . 105
6.4 Future perspectives . . . 106

A Appendix Chapter III 111
A.1 Patient features extracted per ICU unit . . . 111
A.1.1 Data . . . 112
A.2 Implementation details . . . 118
A.2.1 Hyperparameter Search Methodology and Implementation Details . . . 118
A.3 Loss Curves . . . 119
A.4 Multiple comparisons of Means using Tukey test . . . 120

B Appendix Chapter IV 122
B.1 Hyperparameters Optimization . . . 122
B.2 UMAP Visualization of the latent space . . . 122

C Appendix Chapter V 126
C.1 Hyperparamter Optimization . . . 126

D Workdone on UZ Leuven data 129
D.1 Early binary LoS prediction using the MIR Dataset . . . 129
D.1.1 Introduction . . . 129
D.1.2 Data . . . 130
D.1.3 Data pre-processing . . . 130
D.1.4 Feature engineering . . . 131
D.1.5 Topic Modeling . . . 132

D.1.6 Data Imputation 133
D.1.7 Performance metrics 134
D.1.8 Data Modelling 134
D.1.9 Model explanaibility 135
D.1.10 Key Takeaways 137
D.2 FHIR Dataset 138

Bibliography **141**

List of Figures

1.1 Histograms showing the distribution of the ICU LoS extracted from the MIMIC-IV database. 9
1.2 Visual summary of our work showing domain adaptation between groups of patients (in red), the exploration of the latent space of models for patient clustering (in blue) and the fusion of different data modalities generated by patients (in green). 11

2.1 Overview of the different data modalities used in this work shown using a specific ICU patient from the MIMIC-IV database. The first row shows static demographic and admission information collected once at admission. The second row shows time dependent information collected at different hours of the hospitalization. The first two figures of the second row present recordings of some vital signs and laboratory results recorded at different hours. The last two figures of the second row show chest x-rays and clinical reports charted at irregular time intervals. . 18
2.2 Figure highlighting the difference in the frequency of recording of some vital signs and laboratory parameters over time across four ICU patients. Recordings can be continuous or discontinuous over time. 19
2.3 Example of a clinical report for a patient recorded at a specific hour. 20
2.4 Single output perceptron (left) and a single output 3−layer feedforward neural network (right) 24
2.5 LSTM architecture with forget gate [1] 26

2.6 Temporal Convolutional Networks figure from [2]. Original image downloaded from the provided arXiv repository by the authors. (a) A dilated causal convolution with dilation factors $d = 2^0, 2^1, 2^2$ and filter size $k = 3$. The receptive field covers all values from the input sequence. (b) The TCN residual block with an optional 1x1 convolution added to down sample the input layer if this is different from the output layer. 27
2.7 Transformer Encoder architecture connected to a single output multi-layer perceptron and an activation function for the output prediction. PE indicate the positional encoding and RC, the residual connections. 31

3.1 Proposed architecture: Domain adaptation via weight transfer from the source domain to the target domains. I_{n_S}, $I_{n_T}, H_{n_S}, O_S, O_T$ are respectively, the source inputs, target inputs, source hidden unit, source output and target output and $n_S \neq n_T$. Black dots are weights of coinciding features and orange diamonds are random weights for non-coinciding features. 40
3.2 Discriminative fine-tuning applied on CSICU from eICU-CRD framed as a multi-input problem where coinciding and non-coinciding features are split across two layers, *lstm_c* and *lstm_nc* respectively. Different learning rates are then assigned to the two layers. 42
3.3 Training and validation loss curves from model optimization on some eICU-CRD units. 44
3.4 Training and validation curves obtained after model optimization on some MIMIC-IV units. 44
3.5 Distribution of number of epochs (left) and MAE (right) on the Test set with (blue) and without (lime green) weight transfer on eICU-CRD. Weight transfer significantly reduces both training time and prediction error. 46
3.6 Distribution of number of epochs to reach convergence with (blue) and without (lime green) weight transfer on MIMIC-IV with Med-Surg ICU, MICU, SICU and CVICU as potential source domains. Weight transfer significantly reduces training time on the majority of the target domains. 48
3.7 Distribution of MAE on test set with (blue) and without (lime green) weight transfer on MIMIC-IV with Med-Surg ICU, MICU, SICU and CVICU as potential source domains. 49
3.8 Number of epochs and error measures with (blue) and without (lime green) weight transfer on eICU-CRD where hyperparameters are obtained from the source domain. 53

3.9 Number of epochs and MAE when performing full model transfer. Tukey's HSD test is used for multiple comparisons [3]. Individual model (lime green), weight transfer (blue) and full model transfer - including weights and optimizer state (dark green). 53
3.10 Number of epochs and MAE on CSICU from the eICU-CRD dataset when performing discriminative learning. Tukey's HSD test is used for multiple comparisons [3]. Individual model (lime green), weight transfer (blue) and different learning rates (red). 54
3.11 CTICU top predictive features without and with weight transfer 56
3.12 TSICU top predictive features without and with weight transfer 57

4.1 Overview of modeling process. In **a)** laboratory and vital signs extracted data are hourly sampled using the mean to produce a 3D tensor input matrix where the time dimension varies across patients, i.e., $t_1 \neq t_2 \neq t_n$. The processed data is sent through four different neural networks (LSTM, GRU, Transformer and TCN) in **b)** and then through a single-neuron FCN for predictions as seen in **c)**. All models' embeddings obtained at the last layer in b) before being sent to c) are then extracted at each time step in **d)** between hour 1 and hour 336 and projected onto a 2D space in **e)**. 69
4.2 t-SNE 2D projections of LSTM model embeddings at different hours, labeled with LoS. 73
4.3 t-SNE 2D projections of GRU model embeddings at different hours, labeled with LoS. 73
4.4 t-SNE 2D projections of Transformer model embeddings at different hours, labeled with LoS. 74
4.5 t-SNE 2D projections of TCN model embeddings at different hours, labeled with LoS. 74
4.6 t-SNE 2D projections of the TCN with attention model embeddings at different hours, labeled with LoS. 74
4.7 t-SNE 2D projections of TCN model embeddings at different hours, labeled with both LoS and in-hospital mortality. 76
4.8 Cluster of 50 patients identified at hour 5 (left figure) where 94% of the patients will experience in-hospital mortality at different hours of their stays and for which 75% has a LoS $\leq$ 3 days (right figure). 77
4.9 Patient id=6 predictions' over time shown at hours 11, 34, 35 and 47 using the Transformer model. Figure shows how the patient is incorrectly predicted as a long stay at hour 11 to hour 34 and switches side to the short stay cluster at hour 35 till hour 47. . 78
4.10 Heatmap of mechanical ventilation indicators from admission to hour 47 78

4.11 Top 25 most important features on the unseen test patients using LSTM (a) and GRU (b) 79
4.12 Top 25 most important features on the unseen test patients using the Transformer (a) and the TCN network (b) 80

5.1 Intermediate Fusion showing the concatenation (represented with a circle) of modality embeddings extracted from the different encoders. Embeddings from the clinical reports and the images are projected using the $\mathbf{0_v}$to the dimension of the base modality before aggregation by concatentation or the sum. 94
5.2 Late fusion model in which each modality independently predicts LoS, and the final prediction is obtained by averaging the ensemble of these predictions. The individual predictions are projected to the dimension of the base modality using the predictions of the latter. 95
5.3 Intermediate Fusion with cross attentions (represented with dashed squares and dashed lines). The attention outputs are simply added to the representation of the base modality for LoS prediction ($\hat{\boldsymbol{y}}_t$). 96
5.4 MSLE per category of LoS using the best models from EFM, IFM-att(TN) and IFM-att(TN_i). The vertical gray line shows the difference in the prediction error across the different models for each class. EFM here refers to the unimodal model. 99

A.1 Loss curves obtained after training CSICU and SICU in eICU-CRD data with optimized hyperparameters 119
A.2 Loss curves obtained after training MICU and SICU in MIMIC-IV data with optimized hyperparameters as in Section 4.4.1. . . . 120

B.1 UMAP 2D projections of LSTM model embeddings, labelled with LoS. 123
B.2 UMAP 2D projections of GRU model embeddings, labelled with LoS. 123
B.3 UMAP 2D projections of Transformer model embeddings, labelled with LoS. 123
B.4 UMAP 2D projections of TCN model embeddings, labelled with LoS. 125
B.5 UMAP 2D projections of the TCN with attention model embeddings at different hours, labelled with LoS. 125
B.6 UMAP 2D projections of TCN model embeddings, labelled with LoS and IHM. 125

D.1 Binary 7 days classification using XGBoost. 20 most important features from SHAP algorithm with XGBoost model. Features from care module are represented with counts, i.e., the number of times these features were recorded in 24h. 136
D.2 Multi-class classification using XGBoost. 20 most important features from SHAP algorithm with XGBoost model. Features from care module are represented with counts, i.e., the number of times these features were recorded in 24h. 137
D.3 Snapshot of local MongoDB called *los* showing some resources in the database. 139
D.4 Each patient is seen as a unique entry in the patient resource . 140

List of Tables

1.1 Some key statistics of UZ Leuven in 2022 that show patient flow within the hospital and consequently the volume of data generated and archived to ensure a thorough follow-up of all admitted patients. 4

2.1 Datasets used in each chapter with rationale. 21

3.1 The input space was augmented with binary indicators of the same size as the original input space to indicate imputed and newly recorded parameter values. A variable *hour* indicating the time of the day at which a record was done was also added. . . 37
3.2 Baseline characteristics per ICU unit in MIMIC-IV. Total No. features = (No. inputs × 2)+1 37
3.3 Hyperparameters results on eICU-CRD. S: source domain and T: target domain. 43
3.4 Hyperparamters results on MIMIC-IV. S: source domain and T: target domain. 44
3.5 eICU-CRD: Baseline models per unit and “all stays" model. Error estimates on test set with 95% C.I. “all stays" model significantly overestimates the prediction error of Med-Surg ICU, CCU-CTICU, CICU and CSICU. 46
3.6 MIMIC-IV: Baseline models per unit and “all stays" model. Error estimates on test set with 95% C.I. "all stay" model MAE overestimates the MAE of CVICU, CCU and NI by 1.79, 0.57 and 1.15 days, respectively. 47
3.7 CPU time in hours for re-training all models 100 times using the eICU-CRD data. OP_m: Optimized model on each ICU unit, WT_m: Weight transfer model, FT_m: Full model transfer. . . . 50

3.8 CPU time in hours for re-training models 100 times in MIMIC-IV data. OP_m: Optimized model on each ICU unit, WT_m: Weight transfer model. 51
3.9 Metrics obtained upon applying the naive mean and median models prediction on each ICU unit from the eICU-CRD dataset 59
3.10 Metrics obtained upon applying the naive mean and median models prediction on each ICU unit from the MIMIC-IV dataset. 59

4.1 Descriptive statistics of final data cohort 65
4.2 Input Features extracted from the AUMC database 67
4.3 Performance on the test set. The best performance is shown in bold. Mean ± standard deviation of metrics reported are calculated based on 15 iterations for each model. τ is the training time for one complete model run measured in hours. The TCN model with attention (TCN-att) achieves the best performance across all metrics, although it requires more training time. 71
4.4 Performance on test set with associated window length l reported. The best performance is shown in bold. Mean ± standard deviation calculated based on 15 iterations for each model. τ is the training time for one complete model run measured in hours. The TCN-att network reports the best performance with a window length of only 5 hours. 82

5.1 Descriptive statistics of data cohorts. Percentage of stays with the image modality in *CohortTN_i* are shown in parenthesis. . 88
5.2 Input features extracted from MIMIC-IV database. 90
5.3 Results of meta-finetuning the Clinical-T5-Sci model for total remaining LoS prediction only with the first note recorded within 24 hours of ICU admission. All intermediate model layers are kept frozen when no finetuning is performed. 92
5.4 Model performance on the test set of the *CohortTN* and the *CohortTN_i*. The best performance is shown in bold. Mean ± standard deviation calculated based on 3 iterations for each model. 97
5.5 Model performance on the test set of *CohortTN*. IFM-sum: this is the sum of the embeddings from the base modality and the text modality. IFM-sum att: an extension of IFM-sum where attention outputs from the two modalities are also added. IFM-att: only the attention outputs are added to the base modality. The best performance is shown in bold. Mean ± standard deviation calculated based on 3 iterations for each model. 98

A.3 Hyperparameters search space and corresponding scale. No. Hidden units was enlarged to the interval [4 - 512] for Neuro-Stepdown ICU patients in MIMIC-IV and the learning rate to $[1e^{-5} - 1e^{-2}]$ for Medical-Surgical ICU patients in MIMIC-IV . 118
A.4 SICU: Multiple comparison of Means for epochs and Test MAE using Tukey HSD [3], $\alpha = 5\%$. OM: Optimized Model on each sub-ICU unit, WTM: Weight Transfer Model, FTM: Full Transfer Model 120
A.5 CCU-CTICU: Multiple comparison of Means for epochs and Test MAE using Tukey HSD, $\alpha = 5\%$. OM: Optimized Model on each sub-ICU unit, WTM: Weight Transfer Model, FTM: Full Transfer Model 121
A.6 CTICU: Multiple comparison of Means for epochs and Test MAE using Tukey HSD, $\alpha = 5\%$. OM: Optimized Model on each sub-ICU unit, WTM: Weight Transfer Model, FTM: Full Transfer Model 121
A.7 CSICU: Multiple comparison of Means for epochs and Test MAE using Tukey HSD, $\alpha = 5\%$. OM: Optimized Model on each sub-ICU unit, WTM: Weight Transfer Model, Diff_LR: Different Learning Rates Model 121

B.1 Hyperparamters values obtained upon optimization 124

C.1 Hyperparamters values obtained upon optimization on the physiological parameters with the Early Fusion Model (EFM) . 126
C.2 Hyperparamters values obtained upon optimization on the physiological parameters (T)/notes(N) with the Intermediate Fusion Model (IFM) 127
C.3 Hyperparamters values obtained upon optimization on the physiological parameters (T)/notes(N) with the Late Fusion Model (LFM) 127
C.4 Hyperparamters values obtained upon optimization on the physiological parameters (T)/notes(N)/Images(I) with the Intermediate Fusion Model (IFM) with cross-attentions 128

D.1 Distribution of LoS in fractional days 129
D.2 LoS $\leq 14, > 14$ 130
D.3 LoS $\leq 7, > 7$ 131
D.4 $[<= 7, 7 - 14, > 14]$ 131
D.7 shape of ytrain:(44437,) shape of ytest: (19044,) 134
D.8 7 days binary classification using XGBoost. 135
D.9 Three classes multi-classification using XGBoost. 135

Chapter 1

Introduction

This dissertation contributes new methodologies for Length of Stay (LoS) prediction aimed at improving hospital management, with a particular focus on the transferability of these methods to real-world hospital settings. The research presented here is a collection of projects developed through, or inspired by, our collaboration with the University Hospital of Leuven, which required a LoS prediction system for hospital management.

By leveraging publicly available data and adapting a wide array of tools from the current machine learning landscape, we addressed challenges related to data heterogeneity and specificity, the explainability of sequential models, and the integration of multimodal data types.

In the next sections of this introductory chapter we provide a more detailed overview of the importance of the LoS prediction problem and the relevance of our work. We begin by addressing the challenges of hospital management and the metrics used in Belgium to evaluate it in Section 1.1. Subsequently, in Section 1.2, we present a comprehensive review of the LoS prediction problem, explaining what LoS prediction entails and how it can be achieved. In Section 1.3, we establish the context within which our research was conducted, and the contributions made are outlined in Section 1.4. The chapter concludes with acknowledgments of all contributions received, in Section 1.5.

1.1 Hospital Management

The primary objective of a medical institution is to deliver compassionate and exceptional care to patients, all within a secure and nurturing work environment. Typically, hospitals are organized in medical departments, each offering specialized care and equipped with essential resources such as hospitalization rooms with an assigned number of beds, technical equipment, medical, paramedic, nursing staff, and more [4]. Hospital management is then the effective handling of all hospital activities and operations, the efficient use of all available resources (both human and logistic) to ensure the timely provision of high quality of care [5].

The **quality** and **efficiency** of care are two of the measurable indicators often used to assess the effectiveness of an established health system and consequently the hospital management. In Belgium, the quality of the healthcare system is defined as *"how much more likely healthcare makes it to obtain the desired health results, and how consistent healthcare services are with the current scientific knowledge"* [6]. Regarding the efficiency of a healthcare system, it entails the optimal use of the limited available resources, especially the healthcare workforce for the provision of good quality care [7]. Like other countries in Europe (specifically, EU-27), the Belgian health system is also subjected to the greying phenomenon and the increase in chronic diseases with age [8]. The immediate effect of these is that while the demand for healthcare services is constantly on a rise, the healthcare workforce on the other hand, is reduced [8]. Because hospital management systems have to take these into account, building efficient management systems is crucial. Unexpected events such as the COVID-19 pandemic have also shown that even in scenarios where hospitals had ample resources, they can quickly become overwhelmed. A concrete example is explained by the authors in [9] where they highlight the concrete measures undertaken by hospitals in England during the pandemic surge. In effect, elective surgeries planned during the pre-pandemic phase were cancelled to accommodate for resources which were normally adequate but became limited during the COVID surge phase. The specific resources that became limited included nurses, beds and junior doctors from critical care, and beds from general and acute care.

Concretely, efficient planning of healthcare activities from admission up to discharge signifies that after diagnostic workup, surgery or other forms of treatment are planned and administered, rooms must be allocated, available staff must be adequately managed and resources should then be allocated in proportion to the needs of the patient. One of the factors influencing the needs of a patient is their duration of stay in a hospital bed. The longer this stay, the higher the total demand for resources and needs of the patient. In Belgium, the

metrics utilized to assess the efficiency of the health system can be categorized into two main groups: the adoption and use of newer and up-to standard clinical practices and the evaluation of the Length of Stay (LoS) following a normal delivery [7].

The Length of Stay which is defined as the duration between hospital admission and discharge in a single care episode [10], is a metric often used to compare the level of care coordination across countries within the same geographical area and having comparable levels of technology [11]. Given that a hospital's capacity is fixed and known, if the LoS for each hospitalized patient is known in advance, it can help maintain an optimal patient flow through an optimal capacity planning.

1.2 LoS prediction for hospital management

1.2.1 The meaning of LoS prediction

Anticipating inpatients' LoS upon hospital admission can be achieved by making an 'educated guess' about how long this patient will occupy a hospital bed. The 'educated guess', i.e., the prediction of the LoS for an inpatient p, admitted under a certain care program c, can be computed for example by applying statistical measures (such as the mean, the median, etc.) to historical or retrospective LoS data of those patients previously admitted under the same care program c. The relevance of a prediction lies in the time frame. That is, anticipating the LoS of a patient should be done as soon as the patient is admitted. However, at least two problems arise when applying simple statistical measures on previously recorded data: (i) such predictions do not take into account the specificity of each single patient and, (ii) are not updated during the patient's hospitalization. Because a specific patient could have a confounding factor that can influence their LoS and create a deviation from the group mean or median LoS, an alternative then could be to learn a function f that relates patient parameters or features to their LoS. Such a function can be learned using historical patient data.

1.2.2 Secondary use of Electronic Health Records (EHRs)

The movement or transition of patients from one medical department to the other during their hospital treatment generates volumes of data stored in databases called electronic health records (EHRs) [12]. To make this more concrete in our minds, Table 1.1 shows the data generated by patients admitted to the Universitair Ziekenhuis KU Leuven (UZ Leuven) throughout the year

2022. EHRs are thus a digitized version of patient files enriched on a daily basis with the potential to help medical institutions in making informed and tailored decisions about a patient's care [12].

Table 1.1: Some key statistics of UZ Leuven in 2022 that show patient flow within the hospital and consequently the volume of data generated and archived to ensure a thorough follow-up of all admitted patients.

No. admissions	No. day admissions	No. deliveries
52,879	124,255	2,390
No. consultations	No. emergencies	No. patient transfers
759,073	72,097	263,157
No. function tests	No. lab tests	No. radiology exams
126,295	16 million	401,757

The medical domain is the domain par excellence where academic research has revolutionized the provision of its services through evolving clinical practices, the discovery of new drugs, the development of new technologies, and more. At the heart of this research revolution are Randomized Controlled Trials (RCTs), constituting the back-bone of what is today known as 'evidence-based medicine' [13]. RCTs have been employed in medicine for as long as the 19^{th} century or even before and have led to revolutionary advances in medicine. Since the trials on streptomycin for pulmonary tuberculosis by Sir Austin Hill, RCTs have been increasingly used over the years [13]. In a nutshell, an RCT is a prospective study that works by recruiting 'specific' patients upon obtaining their consent and who are then monitored in either an interventional or a control group throughout a period of time [13]. The success of the RCT then relies on at least two important factors that can simultaneously be seen as limitations. These factors are the availability of a large and representative size of the patient population involved in the study, and the retention of these patients throughout the trial [14].

The digitization of patient data started as hospital discharge databases and registry databases [12] for billing purposes and basic personal health data recording (diagnoses and treatments), respectively. Later on, with incentives from governments, hospitals began to rapidly adopt and expand the data digitization beyond just administrative data, first in the United States from 2007 and then followed by other countries worldwide [15]. The accumulation of clinical data over the years, coupled with the decrease in the cost of computing power made it possible to store or archive vast amounts of retrospective patient data whether at a single center, across multiple centers or nationwide. Accessing such large, heterogeneous, and population-representative databases without requiring patients consents (since all data are retrospective, of course, factoring

the approval of the Ethical Committee) appeared as a means to overcome the previously mentioned limitations of RCTs. Some studies even showed that cost-sensitive and time-sensitive processes like patient recruitment could be facilitated by combining both EHRs and clinical trials [16].

In today's era of 'precision-medicine', the reuse or secondary use of healthcare data for clinical research has enabled a unique opportunity for providing a better understanding and screening of patient populations, supporting hypothesis testing, investigating the feasibility of clinical trials, early detection of diseases, continuous assessment of treatment effects and outcomes and more [16]. Beyond these medical prowess, the reuse of healthcare data has also contributed to the managerial improvement of clinical facilities using for example patient data for monitoring and predicting managerial metrics like the LoS. In Belgium, the Belgian Health Data Agency (HDA) was established through legislation on March 14, 2023, and officially launched on January 17, 2024 with the aim to facilitate the accessibility, interoperability, and the reuse of health data and healthcare-related data for secondary applications in accordance with the European Health Data Space (EHDS) policies.

1.2.3 Literature review on LoS prediction

The importance of LoS as a measure of resource consumption and hospital (or medical unit) performance has attracted the attention of multiple researchers. The literature or work done on this topic can be broadly divided into two categories: (i) the investigation of factors that can explain LoS using simple statistical tools and (ii) the use of prediction models to anticipate or predict inpatients' LoS. These two categories also follow a chronological sequence, with the former being more common in the earlier years before the widespread adoption of prediction tools that is seen today.

Regarding the first category, that is, the investigation of factors influencing inpatients' LoS, the motivation for researchers was firstly, to understand the factors affecting inpatients' LoS and the extent to which these could be generalized. Secondly, whether the clear identification of influential explanatory factors and their inclusion in capacity planning models can improve decision making regarding resource allocation and optimization [17]. For example, if features such as *'time of arrival in the hospital'* or *'week-end vs day of the week'* have a significant influence on the LoS, they could be made actionable. That is to say, patients with a potential risk of prolonged LoS (e.g., elective surgery patients) could then be scheduled at the end of the week to exploit the week-end. Additional motivations for understanding factors influencing LoS included, the search for some common universal factors across multiple patient groups (only

'increased age' was identified) and the identification of features that could be used for constructing a predictive model for LoS [18, 19, 17, 20]. Only simple statistical methods were used here: independent t-tests, analysis of variance (ANOVA) for comparing the mean in LoS across different groups (medical specialty of the main referent, admission unit, etc.) and multiple linear regression or multivariate logistic regression to understand how collectively a number of independent variables grouped into clinical, surgical, contextual, organizational factors affect or explain the LoS [17]. Multivariate logistic regression was used when the original continuous dependent variable was converted into discrete categories such as normal versus prolonged LoS [18, 19, 17, 20]. Although linear and logistic regression can be used for predictions, most authors in this group did not undertake this step. The main goal was to understand which of the regressed features were the most significant.

The second category of researchers in the literature focused on building predictive models for LoS. LoS prediction started with clinicians trying to predict hospital LoS [21] or intensive care unit (ICU) LoS [22] for timely hospital management. In the ICU context, using demographic, administrative, and clinical admission data, practitioners were only able to achieve an overall accuracy of 52% in a three-class multiclassification model for predicting patient stays, categorized as: less than 48 hours, 2-5 days, and more than 5 days. As reported in [22], the manual prediction of LoS even reached a lower accuracy for medical and emergency surgery patients [22] and those with longer expected ICU LoS. Vicente et al. [23] suggested that prediction accuracy increased with seniority in the medical field. Comparing manual predictions to automated predictive models, the authors in [24] showed the superiority of the latter over the former for near-future discharge predictions whilst requiring less time. The increased availability of retrospective patient data has contributed to a rich literature in LoS prediction which we discuss considering five dimensions: (i) the type of prediction models employed, (ii) the structure of the outcome variable, (iii) the patient population involved, (iv) the heterogeneity of the feature space in terms of data modalities, and (v) the temporality in the patient data.

i. **Prediction models:** a myriad of models, from the simplest to the most sophisticated ones, have been used for the LoS prediction problem. These models can be broadly classified under **statistical models**, **traditional machine learning models** and **deep learning models**.

   – **Statistical models:** this category includes models such as (multiple or multivariate) linear regression and logistic regression, which are utilized for modeling quantitative and qualitative outcomes, respectively. In the clinical field, these models are greatly valued for their simplicity in understanding and interpretation. However,

because they express the outcome of interest as a weighted linear combination of the input attributes, they become unsuitable for modeling non-linear problems such as the LoS [25]. Furthermore, these models lack the capability to handle time-dependent data, requiring the creation of manually crafted features from longitudinal patient records. Various authors have used either (multivariate) linear regression [26, 27, 28, 29, 30] or logistic regression [31, 32, 33, 34, 35] either as primary models or as baseline models in their research. Logistic regression with L2 regularization (Ridge regression) was employed in [36, 37].

- **Traditional machine learning models:** In contrast to statistical models where there is an a priori specification of the function relating the inputs to the output attributes, machine learning models learn the function structure from some given training data [38]. This group includes static non-linear models which outperformed linear statistical models in capturing complex patterns in patient records. The most used models within this category included tree-based approaches (e.g., decision trees [34, 37, 27, 28], random forests [31, 32, 33, 37, 39, 40, 41, 42, 43, 44], gradient boosted trees [28], extreme gradient boosted trees [40, 45, 42]), kernel-based approaches (e.g., support vector machines) [45, 34, 42], and Bayesian methods (e.g., Naive Bayes) [46]. A limitation of this category of models, similar to the previously mentioned statistical models, is their reliance on manually engineered features that encapsulate the time dependency in longitudinal records.
- **Advanced machine learning models:** this class of models includes deep learning models and more sophisticated machine learning models such as reinforcement learning-based architectures. Deep learning models are made up of Artificial Neural Networks (ANNs) stacked in **multiple layers** (hence, the term 'deep') to progressively extract high-level features from the raw inputs [25]. An ANN is a block of structures called nodes, interconnected to each other in a hierarchical manner via weighted edges and non-linear functions [47], which gives them the ability to handle non-linearities and more complex interactions in the data than some traditional machine learning models [27]. In comparison with traditional machine learning models, deep learning models perform best with larger datasets, thus requiring more computing power including accelerated Graphic Processing Units (GPUs) processors. Different types of neural network architectures were used by various authors.

  Standard ANNs such as multi-layer perceptrons (MLP) or fully connected network (FCN) or feed forward networks (FFN) that

do not explicitly model the time dimension and rely on manually extracted features were used in [34, 37, 39, 27, 28, 48, 29].

Sequence-to-sequence networks, tailored specifically for handling time-dependent data through the processing of information and prediction at every time step, were also employed and are gaining more prominence in recent research studies on LoS prediction. The most frequent used sequential architectures and their variants include Long Short-Term Memory (LSTM) networks [49, 50, 29, 35, 51, 36, 52, 31], Gated Recurrent Units (GRU) [36, 53, 32, 54], attention-based networks [50, 35] and time-dependent convolutional networks [50].

Finally, reinforcement learning-based architectures, in which an agent is trained within an environment to perform a series of actions and is rewarded when this is successful [38], was employed in [36]. In this study, reinforcement learning was used for training two agents in order to mimic instances where patients seek second opinions from other medical practitioners.

ii. **Outcome variable:** the LoS attribute naturally occurs as a continuous variable often measured in days. However, the majority of the literature has often converted this outcome into class categories predicting the likelihood of a patient's LoS being greater than or less than a defined threshold of d days. Though this categorization greatly limits the utility of LoS prediction systems by hospital management teams, it has always been primarily motivated by the right skewness of the outcome of interest as shown in Figure 1.1. Additional methods employed towards overcoming difficulties posed by the distribution skewness include **truncation** [48, 36, 28, 55], that is, the removal of extended stays to reduce their effect on the overall model accuracy, applying the **logarithmic transform** [44, 42] to 'normalize' the original distribution, and the careful choice of the **loss function** [50, 56] such as the mean-squared logarithmic error (MSLE) loss.

iii. **Patient population:** surveying the literature on the LoS prediction shows that more often than not, researchers focus on specific patient populations. These populations are defined using for example, their **age group** (e.g., geriatric patients, pediatric patients, etc.), their **medical diagnosis** (cardiovascular patients, ICU patients with heart failure, ICU patients with sepsis, diabetic patients, kidney failure, etc.), or their **unit of admission** (ICU patients). Currently, the most investigated patient population is the ICU population, a rare and expensive resource compared to acute ward data. At least three reasons can account for the choice of ICU data. Firstly, the dense amount of information captured

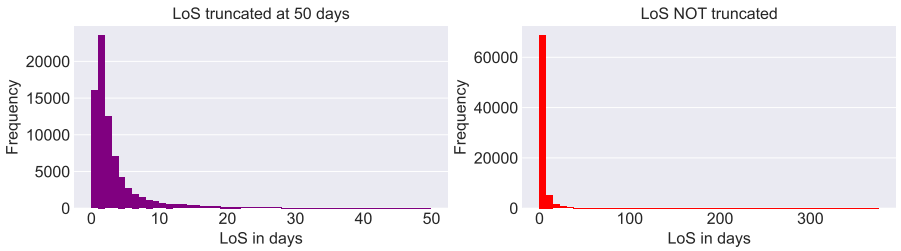


Figure 1.1: Histograms showing the distribution of the ICU LoS extracted from the MIMIC-IV database.

from the continuous monitoring of patients in ICU constitutes a good ground for well-calibrated prediction models [57]. Secondly, most publicly available EHRs for benchmarking clinical prediction models comprise mostly ICU patients [58, 59, 60, 61] and lastly, ICU LoS is also an important management resource for a highly demanded and critical unit such as the ICU unit. Besides these, other reasons evoked for looking at specific patient groups are either the clinical interest for the specific group, or the means to tackle the existing heterogeneity across different patient groups.

iv. **Data Modalities:** the default data type used by all researchers for developing LoS prediction models is **static** information, including both patient demographics and admission information. Over the years, more modalities have been added to this first modality with the goal to improve prediction accuracy [31, 52, 33]. Other modalities used included: **time series** information extracted from patient laboratory examinations, interventions, procedures, and vital signs, followed by **textual** information extracted from clinical notes and finally **visual** information from patients' thoracic x-rays. The incorporation of various data types has also given the opportunity to researchers to explore different combinations of models and fusion strategies to extract the rich information embedded in these data and predict LoS.

v. **Real-time modeling and prediction:** the advent of complex sequence-to-sequence models have given the opportunity to researchers to directly model the time dimension in the longitudinal patient records. Recent studies have employed post-admission data, that is, data available during the patient stay within hospital or ICU premises in order to update LoS prediction models during hospitalization. A first approach was carried out by extracting the first 24 or 48 hours of patient data, re-sampled on an hourly basis and fed to a recurrent architecture such that LoS

predictions are computed and updated hourly during the 24 or 48 hours of stay [56, 31, 32, 33, 40, 27]. In the second and very similar approach, hourly predictions of the remaining time in ICU were computed between admission and death or discharge from ICU [50, 29, 52, 35, 30, 36]

1.3 Context of our research

In the previous sections, we discussed how LoS predictions can be beneficial for hospital management and how data (in different modalities) can be used towards this goal. Our thesis, the LoS prediction problem for hospital management, was initiated in close collaboration with UZ Leuven (the data provider and end user) under the Flanders AI Research Program (FLAIR). Though both parties (UZ Leuven and STADIUS, KU Leuven) were eager to collaborate, a number of bottlenecks were encountered on the way. Finding a means to overcome these challenges ultimately shaped this thesis.

Our biggest challenge was data accessibility which was hindered for over two years by the COVID-19 pandemic and some internal dynamics in the IT department of UZ Leuven. Therefore, all the projects included in this thesis were carried on, on publicly available datasets. However, the projects were designed such that the proposed methodologies could be transferable to our partnering hospital.

1.4 Structure of the thesis and our contribution

The previous sections have highlighted the significance of LoS prediction for hospital management, the attention it has attracted so far in the scientific community and the context surrounding our work. A visual summary of our work is provided in 1.2 and discussed in more details in the next few lines.

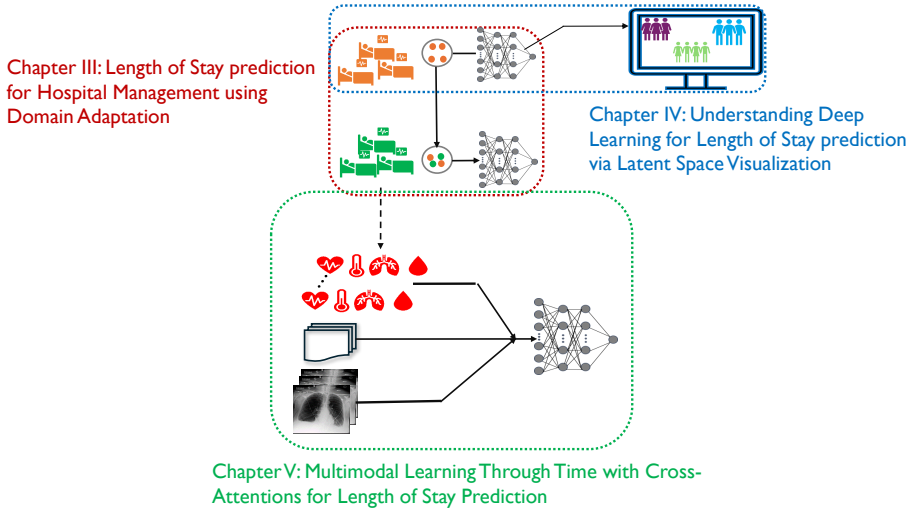


Figure 1.2: Visual summary of our work showing domain adaptation between groups of patients (in red), the exploration of the latent space of models for patient clustering (in blue) and the fusion of different data modalities generated by patients (in green).

Chapter 3: Length of Stay prediction for Hospital Management using Domain Adaptation

Keywords: transferrability, scalability, interpretability

The core idea in this chapter originated from two key interactions with the hospital. The first interaction, with the IT team responsible for data extraction, revealed that processes could be expedited by focusing on a specific patient population instead of the entire hospital population. The second interaction occurred while submitting the requested document to the hospital's ethical committee. The UZ Leuven ethical committee, which meets once a month, reviews submitted documents and deliberates on the approval of research projects involving human subjects. This document thoroughly outlines the planned study, extending beyond the motivation and objectives by specifying the patient population, the features to be used, and, most importantly, detailing the computer infrastructures for data gathering, preprocessing, and modeling. Once again, it became clear to us that requesting patient data one population at a time would expedite the processes.

The idea presented in this chapter is essentially a response to the question:

"Can we apply all the knowledge learned from one patient population to the next when the subsequent population data is received?"

In this chapter, two publicly available datasets are used to mimic such a real hospital environment where patients are separated into populations, here defined by their unit of admission. By emphasizing on their individual specificities resulting in a heterogeneous feature space across populations, domain adaptation is used to share learned knowledge by transferring learned weights either partially or fully from a target domain to a source domain. Compared against the benchmark, the proposed weight transfer model showed statistically significant gains in prediction accuracy (between 1% and 5%) as well as computation time (up to 2 hours saved) for some target domains. The proposed method provides an adaptable clinical decision support system for hospital management that can facilitate processes of data access via the ethical committee, computation infrastructures and time.

Applying our methodology in a clinical setting allows knowledge acquired from a trained LoS prediction model in one hospital department to be transferred to a related department without compromising the unique characteristics of the latter. This approach enhances both prediction performance and training efficiency, while also eliminating the need to centralize all hospital data prior to model training and inference.

This chapter is detailed in the following paper:

> **Momo, Lyse Naomi Wamba**, Nyalleng Moorosi, Elaine O. Nsoesie, Frank Rademakers, and Bart De Moor. **"Length of stay prediction for hospital management using domain adaptation."** *Engineering Applications of Artificial Intelligence 133 (2024): 108088.*

Chapter 4: Understanding Deep Learning for Length of Stay prediction via Latent Space Visualization

Keywords: Real-time setup, interpretability

Given that our thesis required a constant interaction with the hospital personnel, enhancing explainability was an important aspect of our work. We not only aimed to predict remaining length of stay using sophisticated sequential deep learning models but we also needed to investigate intuitive ways to make these prediction systems understandable for the end-users. Additionally, investigate how such prediction systems can be integrated into clinical decision support systems and used in real-time.

Given that our thesis involved continuous interaction with hospital personnel, enhancing model explainability was a key focus of our work. We not only aimed

to predict remaining length of stay using advanced sequential deep learning models but also sought to identify intuitive ways to make these predictions understandable for the end-users. Additionally, we needed to explore how these prediction systems could be integrated into clinical decision support systems for real-time use.

In this work, we used a dataset with a patient population that is closer in demographics to what we would expect from the intensive critical care unit of UZ Leuven. Compared to patients admitted to the intensive care units in the United States, we experienced longer stays with a higher variability in LoS.

State-of-the art sequence-to-sequence deep learning models (Long-Short Memory Network, Gate Recurrent Units, Temporal Convolutional Network, the Transformer model and the Temporal Convolutional Network with attention) were applied on 271 input features extracted from 20,486 ICU stays. These networks demonstrated their superiority over traditional machine learning algorithms and basic mean and median naive models, reducing the prediction error of the latter by almost 2.5 days. Additionally, by exploiting the models' latent spaces using dimensionality reduction techniques and an interactive dashboard, we show how such prediction models can be integrated into Clinical Decision Support Systems and used in a practical hospital setting to understand how the models learn the clinical data over time and how patient predictions can change over time.

Analysis of the models' latent spaces provided valuable insights, showing that even with just 5-6 hours of data, all models could effectively distinguish between short stays ($0 < \text{LoS} \leq 3$ days), medium stays ($3 < \text{LoS} < 7$ days), and long stays ($7 \leq \text{LoS} < +\infty$ days). Moreover, the models were able to identify clusters of patients at risk of in-hospital mortality early, as well as those approaching discharge.

This chapter is detailed in the following paper:

> **Momo, Lyse Naomi Wamba**, Vincent Scheltjens, Wouter Verbeke, Frank Rademakers, and Bart De Moor. **"Understanding Deep Learning for Length of Stay prediction via Latent space visualization." submitted** for publication

Chapter 5: Multimodal Learning Through Time with Cross-Attentions for Length of Stay Prediction

Keywords: Real-time setup, Multimodality

Length of Stay is a complex outcome influenced by multiple factors. In this study, our goal was to assesses the impact of different data modalities for LoS

prediction and moreover, fuse these different modalities of data at their charted time.

The modalities considered include static demographic and admission data, time-varying physiological measurements, clinical text, and imaging data, analyzed across two patient cohorts with missing modalities at both the time and patient levels. By utilizing cross-attention mechanisms to capture the correlations between different modalities at each time point, we observe improvements in prediction performance across all metrics. Incorporating cross-attention between the base modality (a combination of static data and physiological parameters) and clinical reports reduces the mean absolute error (MAE) from 2.97 to 1.86 days. Additionally, including chest X-rays—available for less than 25% of the patients—yields a slight reduction in error for patients with LoS between 4 and 14 days.

Our prediction system which leverages pre-trained clinical large language and chest X-ray models, is designed to function in real-world hospital settings where patient data is recorded at varying times and missing modalities are common.

This chapter is summarized in the following paper:

> **Momo, Lyse Naomi Wamba**, Vincent Scheltjens, Wouter Verbeke, Frank Rademakers, and Bart De Moor. **"Multimodal Learning Through Time with Cross-Attentions for Length of Stay Prediction"** to be submitted for publication.

1.5 Acknowledgments

This research received funding from the Flemish Government under the 'Onderzoeksprogramma Artificiële Intelligentie (IA) Vlannderen' program, from KU Leuven research funds (projects BOF/23/064, C16/15/059, C3/19/053, C24/18/022, C3/20/117, C3I-21-00316), from the European Research Council under the European Union's Horizon 2020 research and innovation programme (ERC Adv. Grant, grant agreement No 885682) and from the Flemish Government Agencies (FWO) under the EOS Project no G0F6718N (SeLMA).

The computational resources and services used in this work were provided by the VSC (Flemish Supercomputer Center), funded by the Research Foundation - Flanders (FWO) and the Flemish Government.

The work presented in Chapter 3 was carried out in the course of the three months Black in AI Summer Research Program 2021. The authors greatly thank Dr. Lyne Tchapmi and Dr. Lauren Gillespie for constant useful feedback.

The work presented in chapter 4 and chapter 5 were carried out at the research group STADIUS.

Generative AI tools were used for improving sentences by extracting different synonyms to words previously used.

Chapter 2

Background

This chapter brushes on the basics of machine learning for those methods and techniques that were employed in this work. It explains the underlying mechanisms behind these methods starting from the dataset at hand. The content of this chapter is based on the book by Goodfellow et al[25].

2.1 The Data

As mentioned in the introductory chapter, the movement of patients from one medical unit to the other generates data right from the patient registration. Sometimes, data generation can start even before patient registration within the hospital premises when this patient's arrival mode is the **emergency** for example. A patient arriving at the hospital with an ambulance already has informative data collected (e.g., vital signs). Once within the hospital premises, depending on whether the patient's admission is elective or urgent, the patient is transferred to an admission unit where first care is administered. Such care includes vital signs check, laboratory tests to be performed and more. Subsequently, transfers can happen to some 'function unit' such as the radiology unit where medical imaging procedures such as X-rays are taken.

All these transitions generate longitudinal records, with no uniformity in the time dimension which is rather, directly linked to the patient's state and the availability of the required resources. In Figure 2.1 we show an overview of the different modalities used in this work (not all modalities were always used). Static information is recorded once at admission whereas, time-varying vital

signs (e.g. systolic arterial blood pressure, heart rate), time-varying laboratory results (e.g., mean corpuscular volume (MCV), partial pressure of oxygen (pO2)), time-varying thoracic scans and clinical reports are collected over time and at different time points.

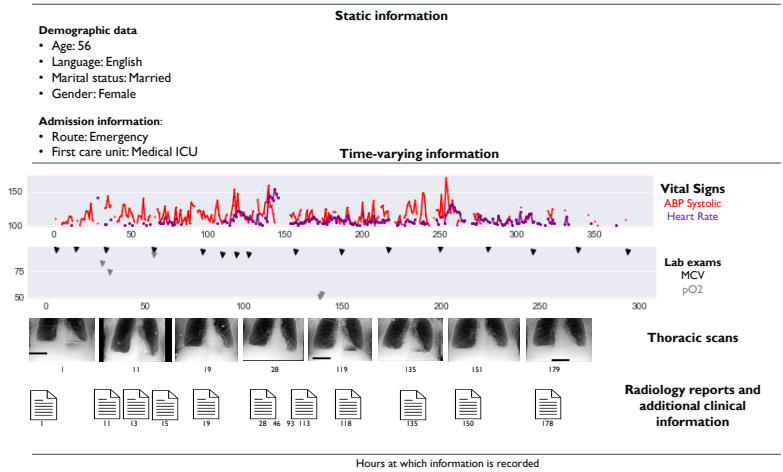


Figure 2.1: Overview of the different data modalities used in this work shown using a specific ICU patient from the MIMIC-IV database. The first row shows static demographic and admission information collected once at admission. The second row shows time dependent information collected at different hours of the hospitalization. The first two figures of the second row present recordings of some vital signs and laboratory results recorded at different hours. The last two figures of the second row show chest x-rays and clinical reports charted at irregular time intervals.

Using four ICU patients extracted from the MIMIC-IV database [60, 61, 62], Figure 2.2 highlights the density in the data recording across these patients. These four patients with different observed LoS (ranging from half a day to 16 days) were respectively admitted to the Medical ICU (MICU), Trauma ICU (TICU), Coronary Care ICU (CCU) and MICU (reading from top to bottom and from left to right).

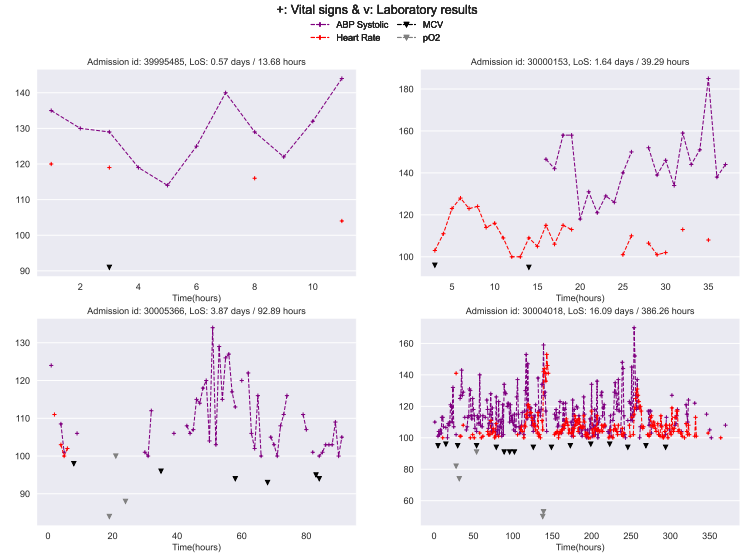


Figure 2.2: Figure highlighting the difference in the frequency of recording of some vital signs and laboratory parameters over time across four ICU patients. Recordings can be continuous or discontinuous over time.

Figure 2.3 shows an example of a clinical report written 3 hours after ICU admission for an ICU patient.

```
TYPE OF EXAMINATION:  Chest AP portable single view.

INDICATION:  Status post left subclavian line placement attempt.
Evaluate for possible pneumothorax.

FINDINGS:  AP single view of the chest has been obtained with patient
in supine position.  Available for comparison is a preceding AP
single view chest of the boarded patient obtained five hours earlier
during the same day.

The patient remains intubated, the ETT terminating in the trachea some
4 cm above the level of the carina.  Unchanged position of previously
described NG tube curving up in the fundus of the stomach.
There is no pneumothorax or mediastinal widening.  Pulmonary
vasculature remains normal and no acute infiltrates are present.

IMPRESSION:  No evidence of pneumothorax following unsuccessful left
subclavian line placement attempt.HEAD CT WITHOUT INTRAVENOUS CONTRAST

INDICATION:  ___ male with bilateral subdural hematomas.

COMPARISON:  ___, 10:28 a.m.

TECHNIQUE:  Non-contrast head CT.

FINDINGS:  There is slight increase in extent of a subdural hematoma
along the left cerebral convexity.  The appearance of the brain is
otherwise unchanged, including a large right subdural hematoma along
the cerebral convexity, subdural hematoma extending along the falx,
the left subdural hematoma extends to the cerebellar tentorium,
as before.  The degree of right-sided sulcal effacement is similar.
There is no shift of normally midline structures.
The basilar cisterns remain preserved.  Again seen is a left frontal
subgaleal hematoma.  Imaged osseous structures are stable in
appearance. With the exception of mucosal thickening in the maxillary
sinuses bilaterally, imaged paranasal sinuses are clear.

IMPRESSION:
1.  Slight increase in size of left subdural hematoma along the
cerebral convexity, otherwise unchanged appearance of the brain,
with bilateral subdural hematomas and hematoma at the falx.
2.  Stable extent of right sulcal effacement, no shift of normally
midline structures. No new acute intracranial hemorrhage.
```

Figure 2.3: Example of a clinical report for a patient recorded at a specific hour.

Table 2.1 provides an overview of the datasets used in each chapter, along with the rationale for their selection.

Table 2.1: Datasets used in each chapter with rationale.

Chapter	Dataset	Rationale
III	eICU-CRD & MIMIC-IV	Large heterogeneous EHR and easily accessible. Each patient can be mapped to their unit of admission, providing the ideal setup for applying domain adaptation, where different units were used either as a source domain or a target domain.
IV	Amsterdam UMC	Availability of the outcome of interest, with the dataset being largely unexplored in the literature, and its demographics being similar to those of the UZ Leuven ICU.
V	MIMIC-IV	EHR dataset with multiple data modalities linkable my unique patient identifiers.

2.2 The Learning Process

As mentioned in the introductory chapter, the prediction of LoS involves learning a function f that best relates the patient attributes to their LoS. This process happens in two steps: (i) training and (ii) inference. In the first step, the function f is learned and in the second step, the learned function f is then applied to 'new and unseen' patient stays or instances to make predictions. To briefly explain these processes, we formally define the problem at hand starting with the data.

For a single instance, the input data is defined as follows;
-Static demographic and admission information features $\mathsf{X^s} \in \mathbb{R}^{h^s}$, where h^s is the total number of features.
-Time-dependent multivariate physiological parameters $\mathsf{X^p} \in \mathbb{R}^{t^p \times h^p}$, where t^p is the maximum number of recordings of physiological parameters and h^p is the number of parameters.
-Time-dependent images $\mathsf{X^i} \in \mathbb{R}^{t^i \times c \times he \times w}$, where t^i is the total number of images collected over time, c is the number of image channels, he and w are respectively the height and width of each image.
-Time-dependent clinical notes $\mathsf{X^n} \in \mathbb{R}^{t^n \times s \times h^n}$, where t^n is the total number of notes charted over time, s is the maximum sequence length of a single note and

h^n is the embedding dimension of each note.
-The observed LoS attributes $y \in \mathbb{R}$.

Given the data, we seek to find the function $f : \mathbb{R}^h \longrightarrow \mathbb{R}$ that best approximates y, such that the LoS predictions $\hat{y} = f(\mathbf{X}; \theta)$, where the different data modalities k are grouped into the input feature matrix $\mathbf{X}$ and h is the total number of input features obtained after concatenating different input modalities.

The training procedure then involves finding the parameters θ by minimizing the prediction error (i.e., the error between the observed and predicted LoS) measured using a specified function denoted by $\ell(\theta)$. Throughout this work, $\ell(\theta)$ is specified using the mean squared logarithmic error (MSLE) function defined as:

$$\ell(\theta) = \frac{1}{n} \sum_i \left[\log(y_i + \mathbf{1_v}) - \log(f(\mathbf{x}_i; \theta) + \mathbf{1_v})\right]^2,$$

where n is the batch size, that is, the number the number of samples or instances to be processed in a single iteration within a complete round of training (i.e., the epoch). This loss function was chosen due to its capability to handle the skewness in the data as shown in [50].

The inference stage then starts when training is completed and the resulting function $f(\mathbf{X}; \theta)$ is used with the found parameters θ. At inference the trained model is simply applied to the 'new and unseen' data to get predicted LoS values.

2.3 Model architectures

The introductory chapter highlighted the importance of the LoS metric as a managerial tool for hospitals and therefore the need to maintain this attribute in its original continuous form. Throughout this work, the main task is the prediction of inpatients LoS in fractional days.
Given patients' retrospective data with observed LoS labels, supervised learning was then performed in the regression setting. The different model architectures used throughout this work either as baseline models or main models are explained below.

2.3.1 Linear regression

Regarding supervised learning and under the regression setting, the linear regression model is one of the simplest model in the literature [25]. This model expresses the outcome as a weighted linear combination of the input features. Mathematically, we write:

$$f(\mathbf{X}; \theta, \mathbf{b}) = \theta^T \mathbf{X} + \mathbf{b}, \tag{2.1}$$

where b is a bias term. Methods such as the least squares method can then be used to find the optimal parameters θ that minimize the prediction error $\ell(\theta)$ [25]. Equation 2.1 shows that the linear regression model does not explicitly model the time dimension in the patients' multivariate time series. Therefore, it is custom practice to manually extract features that implicitly describe patterns such as the **trend** for example in the data. In doing so, this often creates a new feature space $\mathbf{X} \in \mathbb{R}^{n \times l}$ where $l > h$. For example, suppose 100 raw time series features are extracted from the original database and five summary statistics (e.g. the mean, the standard deviation, the minimum value, the maximum value, the difference between the first and last recorded values at each prediction time point) are extracted, this will then result in 500 input features solely from the time series features. Because the number of instances is fixed to n, the increased feature space may create a problem termed, the curse of dimensionality. The learning algorithm then struggles to understand the myriad of complex interactions in the feature space than can potentially differentiate a 2 days stay from a 10 days stay or even more complex, a 2 days stay from a 3.5 days stay.

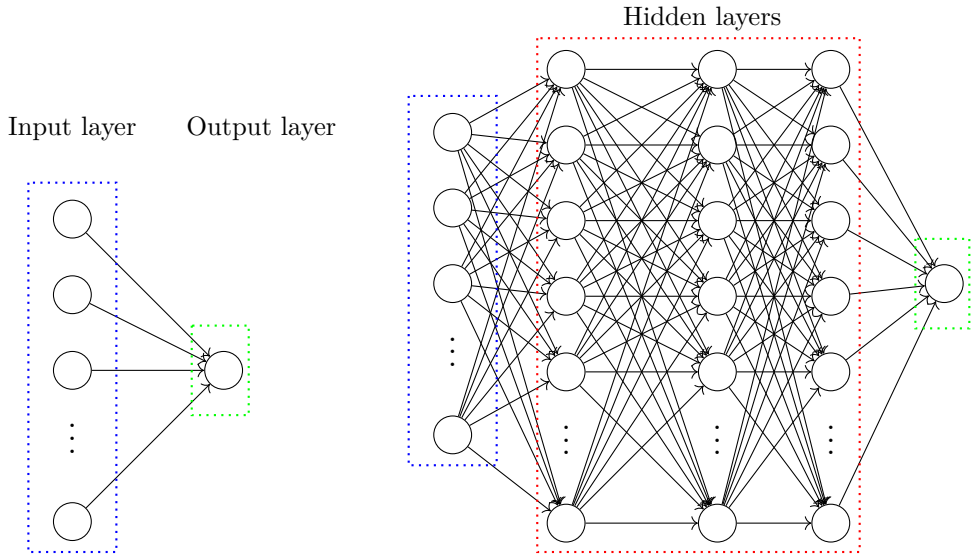
2.3.2 FeedForward neural networks

Neural networks or artificial neural networks (ANNs) are the building blocks of the deep learning paradigm. Deep learning algorithms were developed to overcome some of the limitations of traditional machine learning algorithms including the curse of dimensionality mentioned previously [25]. The former algorithms thus have the potential to learn complex functions f in high-dimensional spaces, for complex AI problems including text translation, object detection and more, though at a higher computational cost, compared to the latter.

Feedforward neural networks (FFNs) are simply multilayer perceptrons. A single perceptron or a single artificial neuron as shown in Figure 2.4a, is a weighted connection of input features to the predicted output $\hat{\mathbf{y}}$ via a (non) linear function g.

$$\hat{\mathbf{y}} = g\left(\theta^T \mathbf{X} + \mathbf{b}\right) \tag{2.2}$$

The important difference between Equation 2.1 and Equation 2.2 is the function g, often called the **activation function** that models the non-linearity in the data [25].



(a) Single output perceptron (b) 3−layer FFN

Figure 2.4: Single output perceptron (left) and a single output 3−layer feedforward neural network (right)

By introducing multiple layers of neurons between the inputs (input layer) and the output (output layer), we get a Feedforward network (FFN) as shown in Figure 2.4b. FFNs are so-called because the different layers are arranged sequentially. In essence, the inputs at each layer L comes from the preceding layer $L-1$. The output at each layer is thus given by:

$$f(\mathbf{h^L}; \theta, \mathbf{b}) = g(\boldsymbol{\theta}^L \boldsymbol{h}^{L-1} + \boldsymbol{b}^L), \tag{2.3}$$

where, $\boldsymbol{h}^0 = \boldsymbol{X}$ and $\boldsymbol{h}^L = \hat{\boldsymbol{y}}$.

The intermediate layers between the input and output layers are called the **hidden layers** and the number of neurons at each hidden layer are called **hidden units**. An FFN with $h \in \mathbb{N}$ hidden layers is called an h−layer FFN. Similarly to Equation 2.1, Equation 2.3 do not explicitly model the time dimension in the patient records, also requiring hand-engineered features.

2.3.3 Gated recurrent neural networks

Recurrent Neural Networks (RNNs) are a type of deep learning algorithms that can handle temporal or sequential data, processing information and predicting output at each time step [63, 64, 65, 25].

Throughout our work, (highly) dense patient ICU data was resampled hourly and used for modeling. Because patients have different duration of stays, the multivariate time series sequences used for modeling were therefore of varying lengths (across the different minibatches). RNNs then became suitable for our case because they allow **parameter sharing** on the time dimension [25], i.e., we do not need to define a new network any time the length of the sequences changes. This is possible because contrarily to FFNs where information flows in one direction, RNNs use feedback connections such that, at each time step, the network receives (part of) its output information from the previous time steps as inputs [25].

Among the different RNNs that exist, gated RNNs which include the Long Short-Term Memory and Gated Recurrent Unit, and which have shown good performances in the field of Natural Language Processing [25] were used in our work. These gated RNNs typically contain gating mechanisms that control information flow and help decide at each discrete time step t, how much of the information furthest in the past to forget and most recent to keep. These two gated RNNs were developed with the aim of overcoming issues associated with gradients vanishing or exploding faced by optimization algorithms [25, 63, 64, 65], thereby affecting model training [25].

1. **Long Short-Term Memory (LSTM):** The LSTM network [66] can be summarized using two main components: the LSTM cell or memory cell state $\boldsymbol{c}_t$ and the control gates - the forget ($\boldsymbol{f}_t$), input ($\boldsymbol{i}_t$) and output gates ($\boldsymbol{o}_t$). The gates operate on the memory cell state to output timely predictions $\boldsymbol{h}_t$. The LSTM network used here [1] is given in Figure 2.5 and is defined as follows,

$$\begin{aligned} \tilde{\boldsymbol{c}}_t &= g\left(\boldsymbol{\theta}_{\tilde{c}h} h_{t-1} + \boldsymbol{\theta}_{\tilde{c}x} \boldsymbol{x}_t + \boldsymbol{b}_{\tilde{c}}\right) \\ \boldsymbol{c}_t &= \boldsymbol{f_t} \odot \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \odot \tilde{\boldsymbol{c}}_t \\ \boldsymbol{h}_t &= \boldsymbol{o}_t \odot g(\boldsymbol{c}_t), \end{aligned} \tag{2.4}$$

   where $\boldsymbol{h}_t$ is the hidden state at each discrete time t , $\boldsymbol{\theta}_{ab}$ are weight parameters from layers a to b and $\odot$ is the element-wise product.

The gate functions $\boldsymbol{f}_t, \boldsymbol{o}_t$ and $\boldsymbol{i}_t$ control the amount of information flow via a sigmoid (logistic) activation function and are defined as,

$$\boldsymbol{f}_t = \sigma\left(\boldsymbol{\theta}_{fh}\boldsymbol{h}_{t-1} + \boldsymbol{\theta}_{fx}\boldsymbol{x}_t + \boldsymbol{b}_f\right)$$

$$\boldsymbol{i}_t = \sigma\left(\boldsymbol{\theta}_{ih}\boldsymbol{h}_{t-1} + \boldsymbol{\theta}_{ix}\boldsymbol{x}_t + \boldsymbol{b}_i\right)$$

$$\boldsymbol{o}_t = \sigma\left(\boldsymbol{\theta}_{oh}\boldsymbol{h}_{t-1} + \boldsymbol{\theta}_{ox}\boldsymbol{x}_t + \boldsymbol{b}_o\right).$$

When updating the cell and hidden states in (2.4), $\boldsymbol{i}_t$ determines which new information is to be stored in the cell state, $\boldsymbol{f}_t$ determines which information is to be disregarded and $\boldsymbol{o}_t$ determines the output from the cell state.

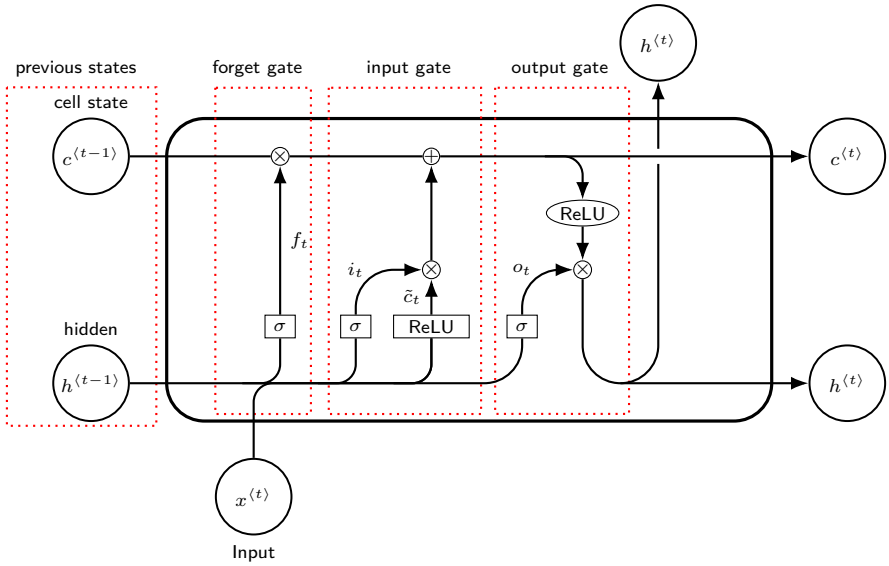


Figure 2.5: LSTM architecture with forget gate [1]

2. **Gated Recurrent Unit (GRU):** The GRU network [67] is very similar to the LSTM network though with reduced complexity in terms of computation and implementation since they have only two gates, the reset $\boldsymbol{r_t}$ and update $\boldsymbol{z_t}$ gates that operate on the hidden states $\boldsymbol{h_t}$. While $\boldsymbol{z_t}$ controls how much of the past information $\boldsymbol{h_{t-1}}$ contributes to $\boldsymbol{h_t}$, $\boldsymbol{r_t}$, controls how much of $\boldsymbol{h_{t-1}}$ is to be ignored when considering the future.

The underlying equations of the GRU model are defined as,

$$\tilde{\boldsymbol{h}}_t = \tanh\left(\boldsymbol{\theta}_x \boldsymbol{x}_t + \mathbf{r}_t \odot \left(\boldsymbol{\theta}_h \boldsymbol{h}_{t-1}\right) + \boldsymbol{b}_h\right)$$

$$\boldsymbol{h}_t = \boldsymbol{z}_t \boldsymbol{h}_{t-1} + \left(\mathbf{1} - \boldsymbol{z}_t\right) \tilde{\boldsymbol{h}}_t,$$

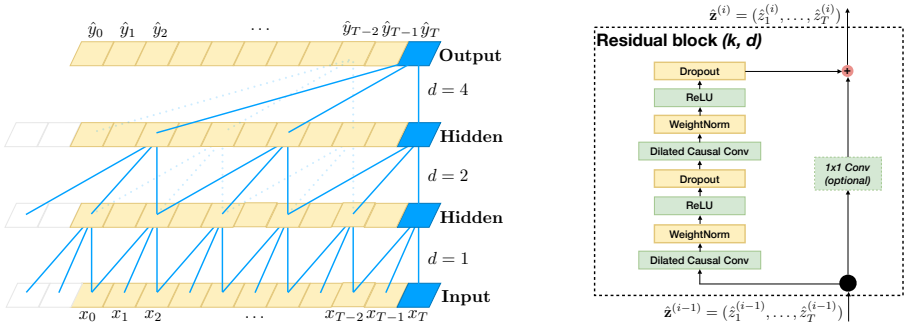
where the gates that control information flow through the GRU, $\boldsymbol{r}_t$ and $\boldsymbol{z}_t$ are given by,

$$\boldsymbol{r}_t = \sigma\left(\boldsymbol{\theta}_{rx} \boldsymbol{x}_t + \boldsymbol{\theta}_{rh} \boldsymbol{h}_{t-1} + \boldsymbol{b}_r\right)$$

$$\boldsymbol{z}_t = \sigma\left(\boldsymbol{\theta}_{zx} \boldsymbol{x}_t + \boldsymbol{\theta}_{zh} \boldsymbol{h}_{t_1} + \boldsymbol{b}_z\right)$$

2.3.4 Temporal convolutional neural network

The recently introduced Temporal Convolution Network (TCN) by Bai et al. [2] shown in Figure 2.6 was also used in our work to learn the temporal dynamics in the patient data. At the core of this network are the 1-D Convolutional Neural Network. Convolutional Neural Networks (ConvNets or CNNs) [68] inspired by the human vision system, have been and are still very popular for image-driven tasks such as image or video segmentation, object recognition and more [69, 70]. The ability of CNNs to handle array-like inputs of one, two and three dimensions [71] has made them suitable for modeling sequential data like signals [72] or time series or text data [64] reaching state-of-art performances.



(a) (b)

Figure 2.6: Temporal Convolutional Networks figure from [2]. Original image downloaded from the provided arXiv repository by the authors. (a) A dilated causal convolution with dilation factors $d = 2^0, 2^1, 2^2$ and filter size $k = 3$. The receptive field covers all values from the input sequence. (b) The TCN residual block with an optional 1x1 convolution added to down sample the input layer if this is different from the output layer.

Similarly to the Gated RNNs explained previously, CNNs also posses properties that render them suitable with our data. Two of these properties are **parameter sharing** and **sparse connectivity** [25]. The latter property displayed in Figure 2.6a translates to the fact the outputs (at each layer) are not directly dependent on **each individual input unit** but on k number of inputs (called the receptive field for that output unit), k, determined by the specified kernel size. The resulting benefits of this sparseness include computational speed-ups and lower memory requirements to store the weight parameters.

The TCN architecture is made up of three main components: a 1-D CNN, dilated causal convolutions and residual connections [73]. The 1-D CNN convolves (referring to the convolution operation $*$) over the input sequence with a filter of size k. Left zero padding (shown on Figure 2.6a with white parallelograms) with size $k-1$ is carried out to ensure that both input and output layers at each network depth are of the same size.

In order to efficiently learn long-range dependencies, the causal convolutions are dilated using different dilation factors d (incremented with the network depth) to increase the network receptive field. We see this in Figure 2.6a, where at each network layer i, the dilation factor $d = 2^i$. Mathematically, the dilated convolution operation F on an element at time t of the time series sequence is defined as:

$$F(t) = \sum_{i=0}^{k-1} f(i) \cdot \boldsymbol{X}_{t-d \cdot i},$$

where, $f : \{0, ..., k-1\} \rightarrow \mathbb{R}$ is the filter, $\boldsymbol{X} \in \mathbb{R}^n$ is a 1-D input sequence and $\cdot$ is the dot product. Residual connections (also shown in Figure 2.6b on the right branch) are then inserted at the end of each layer to learn the layer inputs transformation function. These connections were introduced as a means to handle vanishing gradients when networks are very deep [73]. The final output h is given by:

$$\boldsymbol{h} = g(\boldsymbol{X} + \mathcal{F}(\boldsymbol{X})),$$

where $\mathcal{F}(\boldsymbol{X})$ is the output of the sequence of operations performed on the input X and g, a non-linear function.

2.3.5 Attention-based neural network

The attention-based transformer network introduced by Vaswani et al. [74] that captures dependencies in the data by weighting different parts of the input sequence was also used. These dependencies or correlations are computed by

taking the dot product of the inputs. This operation is termed self-attention. From the original architecture, only the encoder block shown in Figure 2.7 was used. This encoder is simply made up of an input embedding layer (via a 1D convolution), a multi-head attention layer, layer normalizations and a feed-forward layer. Unlike the previously discussed gated architectures, this model lacks recurrence, allowing information to flow in only one direction. An advantage of this is that the network depth does not depend on the input length. Because the LoS prediction is a time-dependent problem, positional encodings (PE) denoted p from the original proposed model are added to the input embeddings to encode the time. In summary, first, input data was compressed using a 1D-convolution network, to which deterministic positional encodings were added and then attention scores were calculated using h_e number of heads obtained from optimization. Three main operations were carried on in this work: input embedding using a 1D-convolutional network in (2.5), multiple attention scores calculation in (2.6) and deterministic positional encoding calculation in (2.7). That is,

$$\tilde{\boldsymbol{e}}(n, C_{out}) = \sum_{k=0}^{h-1} \langle \boldsymbol{\theta}_{\tilde{e}x}(C_{out}, k), \boldsymbol{X}(n, k) \rangle + \boldsymbol{b}_{C_{out}}, \tag{2.5}$$

where k is the kernel size used to slide over the input sequence, h, the number of input channels (corresponds to the input feature size), C_{out}, the number of output channels, and n is the batch size.

The multi-head attention layer which consists of concatenating the results of multiple self-attention calculations, was computed using three components: the query(Q), the key(K) and the values(V). Each self-attention α^{h_e} was computed as follows:

$$\boldsymbol{\alpha}^{h_e}(Q, K) = \text{softmax}\left(\frac{\langle Q^{h_e}(e), K^{h_e}(e) \rangle}{\sqrt{d_k}}\right), \tag{2.6}$$

where, $Q^{h_e}(\boldsymbol{x}) = \theta_{qh_e}\boldsymbol{x} + \boldsymbol{b_q}$, $K^{h_e}(\boldsymbol{x}) = \theta_{kh_e}\boldsymbol{x} + \boldsymbol{b_k}$, $V^{h_e}(\boldsymbol{x}) = \theta_{vh_e}\boldsymbol{x} + \boldsymbol{b_v}$ $\boldsymbol{e} = \tilde{\boldsymbol{e}} + \boldsymbol{p}$ is the sum of input embeddings $\tilde{\boldsymbol{e}}$ and positional encodings $\boldsymbol{p}$ and d_k is the dimension of the embedded inputs (i.e. C_{out}).

As noticed in the definition of the key, query and value affine functions, the weight parameters vector does not dependent on the input sequence $\boldsymbol{x_i}$ which therefore enable **parameter sharing** across different input sequences therefore helping to learn very long sequences with varying lengths [75]. The hidden

outputs (predictions) $\boldsymbol{h}$ are then calculated as;

$$\tilde{\boldsymbol{u}} = \sum_{a=1}^{h_e} \theta_{\tilde{u}} \alpha^a V^a(e)$$

$$\boldsymbol{u} = LayerNorm(\boldsymbol{e} + \tilde{\boldsymbol{u}}; \gamma_1, \beta_1)$$

$$\tilde{\boldsymbol{h}} = ReLU(\boldsymbol{u}\boldsymbol{\theta}_1 + \boldsymbol{b}_1)\boldsymbol{\theta}_2 + \boldsymbol{b}_2$$

$$\boldsymbol{h} = LayerNorm(\boldsymbol{u} + \tilde{\boldsymbol{h}}; \gamma_2, \beta_2),$$

where $\boldsymbol{e} = \tilde{\boldsymbol{e}} + \boldsymbol{p}, \gamma, \beta$ are learnable affine transform parameters, $\boldsymbol{\theta}$ are weight parameters, and h_e the number of attention heads. Deterministic positional encodings from [74] are given below,

$$p_{pos,2i} = \sin\left(\frac{pos}{1000^{2i/d_k}}\right), \; p_{pos,2i+1} = \cos\left(\frac{pos}{1000^{2i/d_k}}\right). \tag{2.7}$$

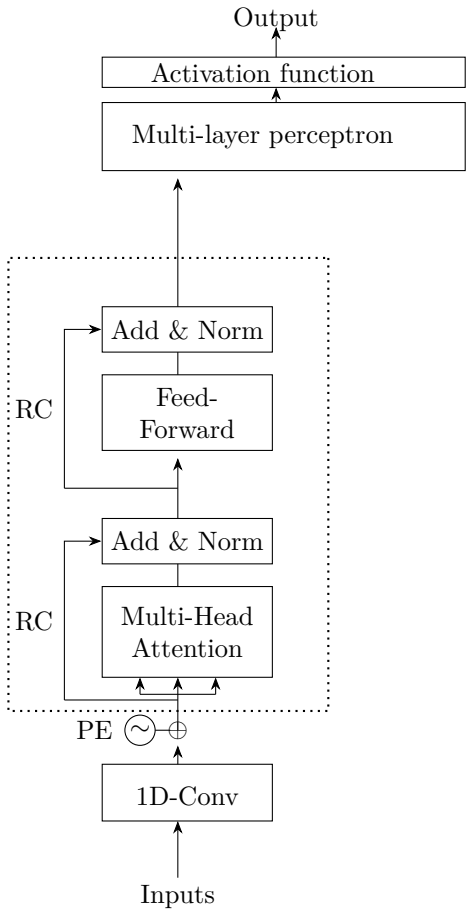


Figure 2.7: Transformer Encoder architecture connected to a single output multi-layer perceptron and an activation function for the output prediction. PE indicate the positional encoding and RC, the residual connections.

Chapter 3

Length of Stay Prediction for hospital management using domain adaptation

This chapter is published and reprinted with permission from:

Momo, Lyse Naomi Wamba, Nyalleng Moorosi, Elaine O. Nsoesie, Frank Rademakers, and Bart De Moor. **"Length of stay prediction for hospital management using domain adaptation."** *Engineering Applications of Artificial Intelligence 133 (2024): 108088.*

Authors Contributions: L.N.W.M: Conceptualization, Methodology, Data curation, Model implementation, Writing - original draft & editing. **N.M**: Methodology supervision. **E.O.N**: Methodology supervision & draft review. **F.R**: Supervision, draft review & editing. **B.DM**: Review.

3.1 Introduction

Predictions of LoS or remaining time in ICU can enable hospital management systems to efficiently allocate both human and logistic resources [50, 76, 77], reassure family members and more importantly, improve patient care and satisfaction [26, 31]. These predictions, for each individual patient within the hospital at time t would enable timely capacity planning and management. With the large volumes of data being generated by patients during their hospital stay, accurate, robust and generalizable LoS prediction models can be developed. Robustness is accounted for by the heterogeneous nature of the patient population within an entire hospital

However, the heterogeneity in the patient population as reflected in the data, for example, the differences in the length and density of data points between ICU and non-ICU patients as mentioned in [78], or in variations in recording frequencies of the same parameters across different units [79], different age or disease groups, poses a number of challenges. One of these challenges is the risk of affecting the overall model accuracy especially when the underlying distribution in each sub-population is different. In order to handle this diversity in patient populations, several authors have restricted their work on specific patient sub-populations delineated by age groups [80, 81], diagnostics [82] or medical units [83, 84]. In the work presented in [79], the authors highlighted this diversity by identifying that PCO2 was measured more frequently in cardiac ICUs while Troponin T was measured more frequently in coronary ICUs. More specifically, regarding patient heterogeneity and clinical outcome prediction, [85] first clustered ICU patients to obtain more homogeneous groups (clusters) onto which a multitask model was learned for mortality prediction. Multitask learning, much related to transfer learning [86] was used to simultaneously train and share learned parameters for mortality risk predictions for all clusters. Similarly, by considering four ICU sub-populations as different domains (either source or target), domain adaptation was applied in [79] for fine-tuning pre-trained weights from a CNN-LSTM-FCN network for mortality risk predictions.

Domain adaptation is a subclass of transfer learning in which a learned model from a source domain is transferred to a target domain for the prediction of the same task, while accommodating to the difference in data distributions across domains [86]. In the present work, domain adaptation is also applied for LoS prediction. In both [79] and [85], the authors acknowledged the specificities in patient populations for which individual models were constructed using a mechanism (domain adaptation and multitask learning, respectively) that also allowed to simultaneously learn the existing similarities in these different populations with a net result in prediction accuracy gains as well as computation time. In our work, the specificity of each patient population

is emphasized by not restricting the input spaces across all populations to be identical. More specifically, by setting a threshold on the frequency of recording patient parameters, most recorded features in each population were kept for modeling. Domain adaptation was then applied to transfer pre-trained weights from a source to a target domain, where these two domains could have non-identical input spaces. In doing so, the transfer of pre-trained weights only happened for coinciding features between the two domains while for non-coinciding features, random weights were used. As a means to understand this simultaneous training of both pre-trained and random weights, an analysis was carried out by employing discriminative learning [87] where different learning rates were assigned to the two sets of features (coinciding and non-coinciding).

In the present work, we employ domain adaptation by transferring knowledge learned from one medical unit or population (source domain) to others (target domains). Benefits of this approach when transposed to a real hospital environment, are multifold because it can help overcome technical challenges related to data modeling and accuracy as well as managerial challenges related to data access and ethical approvals from a hospital. On the technical side, the storage, manipulation, curation, pre-processing and modeling of all patients' data of an entire hospital might stand as a big computing resource challenge in terms of memory capacity and computation time. The latter might even be more problematic in a real time LoS prediction setup. On the managerial side, obtaining ethical approval for accessing a portion of the hospital data or data from one unit can be easier and, more importantly, faster than requesting an approval for all patients in the entire hospital. Moreover, given that most hospitals (e.g., in Europe) have been incentivized to work in a very decentralized manner [88], i.e., allowing decision making at the local hospital unit levels, launching projects like LoS predictions at a unit level by the unit head can be more effective and faster than initiatives from the hospital top management.

3.2 Key Contributions

This work attempts at simulating a real hospital setting, where, LoS predictions are calculated by emphasizing on each unit's singularity, whilst sharing the underlying similarities among patients. The contributions are summarized as follows;

- Using the first 24 hours of post ICU admission data, LoS predictions for ICU patients are computed by applying domain adaptation from a source to different target domains without enforcing a homogeneous feature space across these two domains.

- By using a second dataset with four potential source domains, we investigate and gain insights into the choice of a suitable source domain for weights initialization on the target domains.
- Additional analyses are performed to understand the relevance and gains of domain adaptation. These include, the effect of weight transfer on feature importance on the target domains, the use of the source domain hyperparameters on all target domains, the transfer of the full model, and discriminative learning to understand how simultaneously training pre-trained and random weights affect model accuracy and computation time.

3.3 Methodology

3.3.1 Datasets

The multi-centre eICU collaborative research database (eICU-CRD) [58] and uni-centre MIMIC-IV [89, 62] database were used. The eICU-CRD database contains detailed information of over 200,000 admissions from 139,367 unique patients admitted to 208 hospitals and 335 ICU units within the USA between 2014 and 2015. The MIMIC-IV is an EHR database containing information from 69,619 ICU stays from over 50,000 patients admitted to Beth Israel Deaconess Medical Center in the USA between 2008 and 2019.
The baseline characteristics of the selected data cohorts are given in Table 3.1 and Table 3.2 for eICU-CRD units and MIMIC-IV units respectively.

3.3.2 Preprocessing

By using the pipeline by [50], we selected and pre-processed the first 24 hours of vital signs, and laboratory data recorded post ICU admission. In eICU-CRD, data from 91,277 unique patients corresponding to 110,079 stays admitted to eight ICU units was extracted. These units are the following: Medical Surgical (Med-Surg ICU), Coronary care - cardio-thoracic ICU (CCU-CTICU), Medical ICU (MICU), Neurological ICU (NICU), Cardiac ICU (CICU), Surgical ICU (SICU), Cardio-thoracic ICU (CTICU) and Cardio-surgical ICU (CSICU). Similarly, data from 44,245 unique patients and 60,492 ICU stays admitted to nine ICU units was extracted from MIMIC-IV. In MIMIC-IV, these units are: Medical ICU (MICU), Medical Surgical (Med-Surg ICU), Cardiac Vascular ICU (CVICU), Surgical ICU (SICU), Trauma ICU (TICU), Coronary care ICU (CCU), Neuro surgical ICU (Neuro SICU), Neuro intermediate (NI), Neuro

Table 3.1: Baseline characteristics per ICU unit in eICU-CRD. Total No. features = (No. inputs × 2)+1 [2]

ICU unit	No. ICU stays	No. inputs	Mean(LoS) in days	Std(LoS) in days	Median (Age)	Gender (% Male)
Med-Surg ICU	58,335	25	7.55	6.88	66	52.9
MICU	10,128	24	9.24	8.65	65	52.1
CCU-CTICU	9,950	27	7.24	6.78	67	59.0
NICU	8,777	25	8.26	8.04	63	51.9
CICU	7,744	25	7.87	7.61	65	50.4
SICU	7,684	26	9.45	8.47	65	57.5
CTICU	4,286	33	8.79	7.58	66	62.5
CSICU	3,175	35	7.32	6.39	69	59.8
All stays	110,079	26	7.93	7.36	65	54.3

Table 3.2: Baseline characteristics per ICU unit in MIMIC-IV. Total No. features = (No. inputs × 2)+1

ICU unit	No. ICU stays	No. inputs	Mean(LoS) in days	Std(LoS) in days	Median (Age)	Gender (% Male)
MICU	12,378	29	9.88	8.99	65	54.4
CVICU	11,070	38	8.05	7.03	69	67.4
Med-Surg ICU	10,152	23	9.93	9.41	66	51.3
SICU	9,262	28	10.82	9.94	64	53.5
TSICU	6,992	35	7.87	7.61	64	57.6
CCU	6,881	24	8.36	7.72	72	57.1
Neuro SICU	1,646	28	11.71	10.75	67	50.6
NI	1,491	20	7.15	7.67	68	50.6
NS	620	20	7.53	7.46	68	52.6
All stays	60,492	33	9.69	9.02	67	56.5

stepdown (NS). After hourly sampling the extracted data using the mean, features were only kept for modeling if at least two unique recordings over 24 hours were present for at least 30% of the patients.[1] Missing values were then imputed by first forward filling and then backward filling the most recent value for each patient, as in [27, 53] since we are only working with the first 24 hours of data.

[1] See Appendix A.1 for details on feature extraction.

[2] The input space was augmented with binary indicators of the same size as the original input space to indicate imputed and newly recorded parameter values following [90]. A variable *hour* indicating the time of the day at which a record was done was also added.

3.3.3 Prediction task

The prediction task here is the time lapse between ICU admission and hospital discharge using the first 24 hours of data into ICU. Selected admissions in both datasets are such that they have spent at least 24 hours in ICU to prevent data leakage. Datasets were split into train, validation and test sets following the 70:15:15 ratio.

3.3.4 Models

Long Short-Term memory (LSTM) network

This work utilises the LSTM neural network described in the previous chapter. For our implementation, the open source deep learning library Keras [91], with Tensorflow [92] as back-end was used.
In Keras, the weights $\boldsymbol{W}_{ab}$ are distributed in the kernel, the recurrent kernel and the bias term of the LSTM layer. The kernel stores all weights multiplied by the inputs x_t, i.e., $\boldsymbol{W}_x = \{W_{\tilde{c}x}, W_{ix}, W_{fx}, W_{ox}\}$, the recurrent kernel stores those multiplied by the hidden state h_{t-1}, i.e., $\boldsymbol{W}_h = \{W_{\tilde{c}h}, W_{ih}, W_{fh}, W_{oh}\}$ and the bias stores all bias terms, i.e., $\boldsymbol{b} = \{b_{\tilde{c}}, b_i, b_f, b_o\}$. Therefore, the kernel weight matrix is of dimension $(n_D \times 4H)$, where n_D is the input dimension of the domain D (source or target) and H is the number of hidden units of the LSTM layer. The recurrent weight matrix is of dimension $(H \times 4H)$ and the bias is a vector of dimension $(4H \times 1)$.
From equations (2.4) and (2.5), it follows that the model generates a prediction at each time step t given by h_t. In this work, we opted for the many-to-one LSTM architecture, rather than the many-to-many structure, in order to get only the prediction at the final time step $(t = 24h)$.

3.3.5 Domain adaptation

Domain adaptation via weight transfer

The literature has shown that the use of pre-trained weights can sometimes be more beneficial than using random initial weights. This advantage is observed not only across different prediction tasks using the same data [93] but also from one related dataset to another [94, 95]. The net benefit of fine-tuning these pre-trained weights on the target domain is often reduced computation time for hyperparameters optimization and model training [96].
Given the input space $\bar{\boldsymbol{X}}$ extracted from the database, the model inputs $\boldsymbol{X}$ are

obtained by augmenting this input matrix with a matrix of binary indicators $\tilde{\boldsymbol{X}}$ such that, $\boldsymbol{X} = \text{concat}(\bar{\boldsymbol{X}}; \tilde{\boldsymbol{X}})$ where,

$$\tilde{\boldsymbol{X}} = \begin{cases} 0, \text{if } \bar{\boldsymbol{X}} \text{is recorded} \\ 1, \text{if } \bar{\boldsymbol{X}} \text{is imputed,} \end{cases} \tag{3.1}$$

following [90].
Following the feature extraction process explained in Section 3.3.1, the input matrix $\boldsymbol{X}$ can differ both in the number of features and/or the features themselves[3]. Thus, given a source domain (S) and a target domain (T), we have

$$\boldsymbol{X}_S \subset \boldsymbol{X}_T \tag{3.2a}$$

$$\boldsymbol{X}_S \not\subset \boldsymbol{X}_T \text{ or } \boldsymbol{X}_T \not\subset \boldsymbol{X}_S \tag{3.2b}$$

$$\boldsymbol{X}_S = \boldsymbol{X}_T \text{ or } \boldsymbol{X}_T \subset \boldsymbol{X}_S. \tag{3.2c}$$

In the case of (3.2a) and (3.2b), only partial weight transfer from the source to the target can occur for coinciding features between the two domains. In case (3.2c), where all inputs in T are found in S, total weight transfer occurs. The source domain S is the original domain used to train the model, while the target domain T refers to the new domain where the pre-trained model is fine-tuned. The proposed architecture is given in Figure 3.1.

[3]Further details can be found in Tables 3.1 and 3.2 and Appendix A.1.

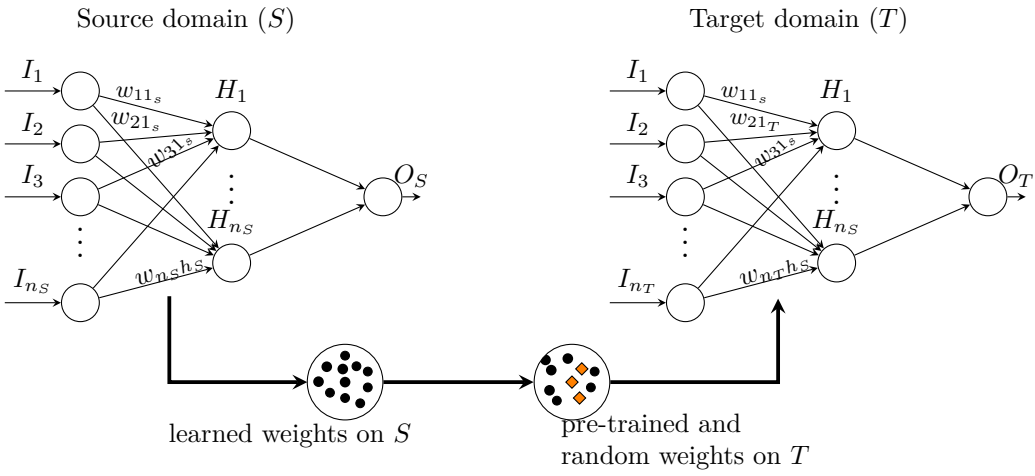


Figure 3.1: Proposed architecture: Domain adaptation via weight transfer from the source domain to the target domains. $I_{n_S}, I_{n_T}, H_{n_S}, O_S, O_T$ are respectively, the source inputs, target inputs, source hidden unit, source output and target output and $n_S \neq n_T$. Black dots are weights of coinciding features and orange diamonds are random weights for non-coinciding features.

The pseudo algorithm used for assigning pre-trained weights (either partially or totally) from the source to the target is given in Algorithm 1. After initializing the LSTM model for each target T, all initial recurrent kernel weights and bias weights are obtained from the trained source model S, i.e., $\boldsymbol{W}_{hT} = \boldsymbol{W}_{hS}$ and $\boldsymbol{b}_T = \boldsymbol{b}_S$. The target domain kernel weights $\boldsymbol{W}_{xT}$ are obtained either totally or partially from S as they depend on the input space $\boldsymbol{X}$.

Domain Adaptation via weight transfer and discriminative learning

During TL, all hyperparameters from S except the batch size are transferred, i.e., the learning rate, the dropout rate and the number of hidden units from S are all transferred to T. Our intuition here is that during training, coinciding features between T and S, which receive fully trained weights, are overfitting whilst the random weights of non-coinciding features are still learning. To study this, discriminative fine-tuning was applied, as illustrated in Figure 3.2. This approach involves assigning distinct learning rates (lr) to the two groups of

Algorithm 1 Weight Transfer from Source domain S to Target domain T

Inputs: Given:
Target data of size $m_T \times 24 \times n_T$: $\boldsymbol{X}_T = \{x_{1t}, ..., x_{n_{T_t}}\}$, $t = 1, ..., 24$, m_T is the number of patients in T and n_T is number of input features in T
Source data of size $m_S \times 24 \times n_S$: $\boldsymbol{X}_S = \{x_{1t}, ..., x_{n_{S_t}}\}$, $t = 1, ..., 24$, m_S is the number of patients in S and n_S is number of input features in S
$\boldsymbol{W}_{xS}$= load source model kernel weights of size $N_S \times 4H$
Output: Target kernel weight matrix $\boldsymbol{W}_{xT}$ of size $N_T \times 4H$
Initialization of weight matrix $\boldsymbol{W}_{xT}$ using Glorot uniform distribution [97]
for $x \in \boldsymbol{X}_T$ **do**
target_index = $\boldsymbol{X}_T[x]$
if $x \in \boldsymbol{X}_S$ **then**
$\boldsymbol{W}_{xT}$[target_index] = $\boldsymbol{W}_{xS}$[source_index]
else
$\boldsymbol{W}_{xT}[x] = \boldsymbol{W}_{xT}[x]$
end if
end for

inputs (layers) and optimizing them using a multi-optimizer [98, 87]:

$$lr_T = \begin{cases} lr_S & \text{if} \quad \boldsymbol{x}_T \notin \boldsymbol{X}_S \\ \alpha lr_S & \text{if} \quad \boldsymbol{x}_T \in \boldsymbol{X}_S \end{cases} \tag{3.3}$$

The learning rate from S is transferred to the non-coinciding features (features in T not in S) while it is reduced by α for coinciding features.

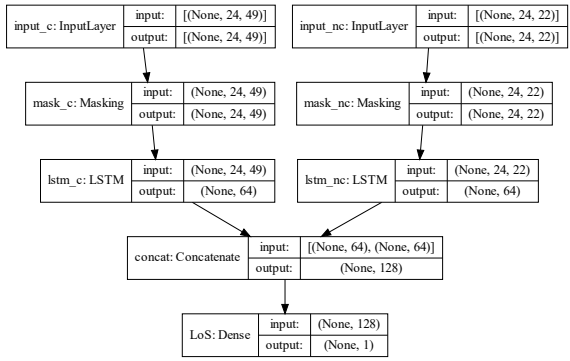


Figure 3.2: Discriminative fine-tuning applied on CSICU from eICU-CRD framed as a multi-input problem where coinciding and non-coinciding features are split across two layers, *lstm_c* and *lstm_nc* respectively. Different learning rates are then assigned to the two layers.

3.3.6 Interpretability using Expected gradients:

The Expected Gradients (EG) method [99] was used in order to unlock the black-box nature of our LSTM models and appreciate the effect of weight transfer on the overall feature importance. The gradient explainer of the SHAP software package[4] was used. Ample details on the method can be found in [99].

3.3.7 Evaluation metrics:

Model training was carried out using the mean squared log error to take into account the skewness in the outcome. The model performance was reported using the Mean Absolute Error (MAE), the Mean Absolute Percentage Error (MAPE), and the Mean Squared Error (MSE) defined below:

$$MAE = \frac{1}{n_D}\sum_{i=1}^{n_D}|y_i - \hat{y}_i|, MAPE = \frac{1}{n_D}\sum_{i=1}^{n_D}\left|\frac{y_i - \hat{y}_i}{y_i}\right|, MSE = \frac{1}{n_D}\sum_{i=1}^{n_D}(y_i - \hat{y}_i)^2,$$

where, n_D is the number of stays in each domain, y_i is the observed LoS, and $\hat{y}_i$ is the predicted LoS.

[4] `https://github.com/slundberg/shap`

3.4 Results & Discussion

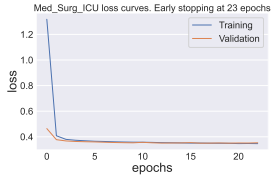
This section presents and discusses all numerical results both with and without weight transfer. It also includes additional analyses, such as cases where hyperparameter optimization was not performed on target domains, full model transfer and discriminative learning.

3.4.1 Hyperparameter Optimization

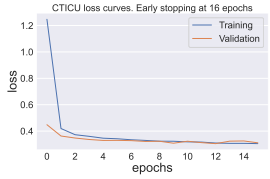
Training and model optimization followed a systematic procedure (see Appendix A.2.1) by carefully monitoring both the generalization error (error on the validation set) and the loss curves in Figures 3.3 and 3.4. The batch size, the dropout, the number of hidden layers (No. layers), the number of hidden units (No. units) and the learning rate (lr) were all model hyperparameters obtained via Bayesian optimization [100, 101] and are shown in Tables 3.3 and 3.4. The Adam optimizer was used.

Table 3.3: Hyperparameters results on eICU-CRD. S: source domain and T: target domain.

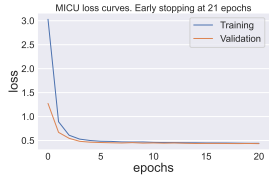
ICU population	batch size	No. layers	No. units	lr	dropout	domain type
Med-Surg ICU	512	1	64	1e-3	0.1	S
MICU	128	1	32	8.99e-4	0.1	T
CCU-CTICU	128	1	16	1.129e-3	0.2	T
NICU	128	1	16	1e-3	0.2	T
CICU	32	1	64	1e-3	0.2	T
SICU	64	1	32	8.99e-4	0.2	T
CTICU	64	1	16	1e-3	0.2	T
CSICU	64	1	64	1e-4	0	T
All stays	512	1	64	0	1.129e-3	



(a) Med-Surg ICU



(b) CTICU



(c) MICU

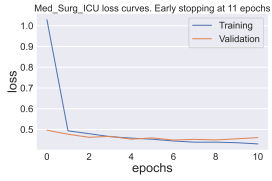
Figure 3.3: Training and validation loss curves from model optimization on some eICU-CRD units.

Table 3.4: Hyperparamters results on MIMIC-IV. S: source domain and T: target domain.

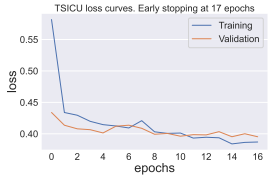
ICU population	batch size	No. layers	No. units	lr	dropout	domain type
MICU	128	1	32	7.142e-4	0.3	S or T
CVICU	128	1	64	5.46e-3	0.3	S or T
Med-Surg ICU	128	1	64	8.99e-4	0.2	S or T
SICU	128	1	64	4.833e-4	0.2	S or T
TSICU	64	1	64	1e-2	0.3	T
CCU	64	1	16	1e-2	0.1	T
Neuro SICU	16	1	8	6.952e-4	0.2	T
NI	16	1	16	4.833e-4	0.2	T
NS	8	1	16	1.624e-4	0.1	T
All stays	512	1	64	0.1	1.438e-3	



(a) All patients data



(b) Med-Surg ICU



(c) TSICU

Figure 3.4: Training and validation curves obtained after model optimization on some MIMIC-IV units.

3.4.2 Model performance

The performance of the weight transfer model was compared to that of the baseline model, where individual models were constructed for each unit using the hyperparameters obtained from optimization and listed in Tables 3.3 and 3.4. Also, for comparisons with the common practice in the literature, a model was run where all units are considered as a single homogeneous entity and their data were learned together in a single "all stay" model. For the weight transfer model, all hyperparameters except the batch size from the source are transferred to the targets. All models were run 100 times by shuffling the data to obtain a distribution of the prediction error on the test set. 95% confidence intervals (C.I) were estimated by calculating the 2.5th and 97.5th percentiles of the prediction error on the test set. We conducted pairwise t-tests on population means between the "all stay" model and each population model. The significance levels are denoted by stars in Tables 3.5 and 3.6: * for $p < 0.05$, ** for $p < 0.01$. The red crosses on boxplot figures indicate instances where the model stagnates, i.e., it does not learn and stops after a few epochs due to the early stopping condition (that prevents overfitting).

eICU-CRD data

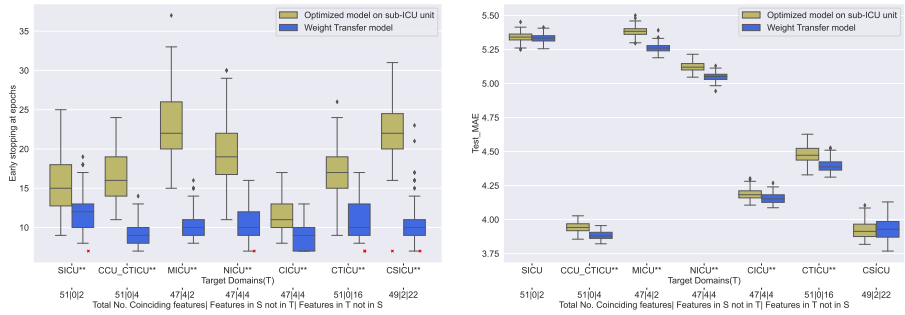
Table 3.5 shows that feeding the model with all ICU stays data for LoS predictions irrespective of their admission units can sometimes overestimate the error in a unit. Looking at the most interpretable error measure, i.e., the MAE (reported in fractional days), we see that the error of the “all stays" model significantly overestimates that of Med-Surg ICU, CCU-CTICU, CICU and CSICU by 3.9, 6.35, 1 and 8.69 hours respectively.[5]

[5]Please disregard the above lines of text and the caption of Table 3.5 as these models were tested on two different datasets and therefore can not be compared directly. Read the Erratum in 3.6.2

Table 3.5: eICU-CRD: Baseline models per unit and “all stays" model. Error estimates on test set with 95% C.I. “all stays" model significantly overestimates the prediction error of Med-Surg ICU, CCU-CTICU, CICU and CSICU.

ICU unit	MAE	MAPE	MSE
Med-Surg ICU	4.03(4.02-4.06)**	0.69(0.66-0.71)**	45.29(44.86-46.28)**
MICU	5.34(5.32-5.46)**	0.77(0.77-0.84)**	80.49(77.61-82.26)**
CCU-CTICU	3.93(3.88-4.02)**	0.65(0.65-0.72)**	49.66(47.15-50.04)**
NICU	5.13(5.06-5.20)**	0.85(0.80-0.87)**	68.93(67.05-70.81)**
CICU	4.15(4.12-4.27)**	0.73(0.70-0.83)**	48.88(46.79-50.06)**
SICU	5.29(5.27-5.30)**	0.69(0.65-0.73)**	78.46(77.52-83.13)**
CTICU	4.51(4.38-4.59)**	0.56(0.52-0.59)**	59.76(55.88-60.97)**
CSICU	3.83(3.82-4.08)**	0.69(0.67-0.80)**	41.46(41.41-47.27)**
All stays	4.20(4.19-4.22)	0.68(0.66-0.70)	50.33(49.97-51.44)

Figure 3.5 shows the effect of using pre-trained weights from S as initial weights for coinciding features and assigning all hyperparameters (except the batch size) from S to T. In Figure 3.5a, we see that model convergence always occurs significantly faster even when a good number of the features in T are not in S, as it is the case for CSICU and CTICU. Figure 3.5b shows that for most of the units, specifically, CCU-CTICU, MICU, NICU, CICU and CTICU, weight transfer significantly improves prediction accuracy.



(a) Number of epochs to reach convergence (b) Mean absolute error in days on test set

Figure 3.5: Distribution of number of epochs (left) and MAE (right) on the Test set with (blue) and without (lime green) weight transfer on eICU-CRD. Weight transfer significantly reduces both training time and prediction error.

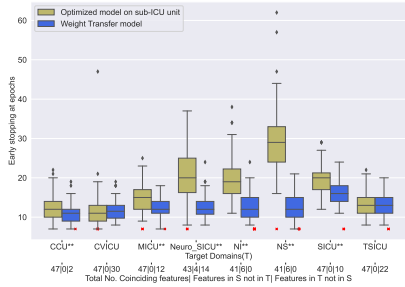
MIMIC-IV data

The observation drawn from Table 3.5, holds for MIMIC-IV data in Table 3.6, where the "all stay" model does not always perform significantly better than unit models as seen for CVICU, CCU and NI. For these units, the "all stay" model MAE overestimates the MAE by 1.79, 0.57 and 1.15 days, respectively.[6].

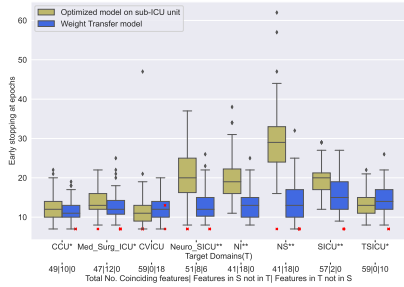
Table 3.6: MIMIC-IV: Baseline models per unit and “all stays" model. Error estimates on test set with 95% C.I. "all stay" model MAE overestimates the MAE of CVICU, CCU and NI by 1.79, 0.57 and 1.15 days, respectively.

ICU unit	MAE	MAPE	MSE
Med-Surg ICU	5.68(5.57-5.72)**	0.84(0.75-0.86)**	91.45(88.74-94.89)**
MICU	5.16(5.09-5.27)	0.71(0.69-0.80)**	68.36(65.06-69.38)**
CVICU	3.32(3.23-5.93)**	0.40(0.38-0.77)**	42.72(40.80-78.07)**
SICU	5.84(5.80-5.95)**	0.68(0.68-0.79)**	92.87(87.65-93.37)**
TSICU	6.48(6.38-6.54)	0.79(0.70-0.82)	102.77(98.94-108.41)
CCU	4.55(4.48-4.62)**	0.72(0.66-0.76)**	62.63(61.21-64.79)**
Neuro SICU	7.23(7.19-8.19)**	0.91(0.84-1.06)**	141.09(132.41-220.36)**
NI	3.96(3.96-4.25)**	0.62(0.64-0.77)**	50.73(48.91-55.68)**
NS	5.58(5.00-5.71)**	0.87(0.69-0.91)**	94.24(83.99-102.35)**
All stays	5.11(5.05-5.14)	0.67(0.61-0.68)	73.82(73.03-76.95)

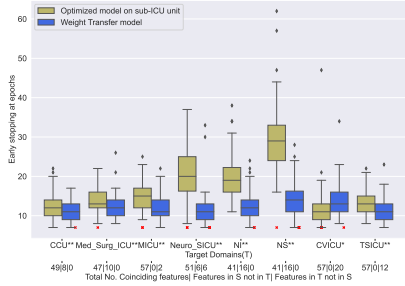
[6]Please disregard the above lines of text and the caption of Table 3.6 as these models were tested on two different datasets and therefore can not be compared directly. Read the Erratum in 3.6.2



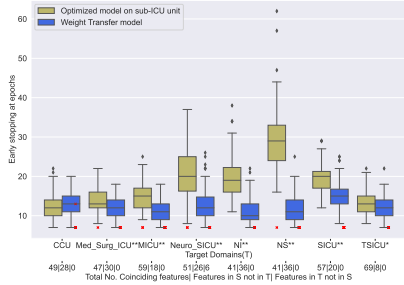
(a) Source domain (S): Med-Surg ICU



(b) Source domain (S): MICU



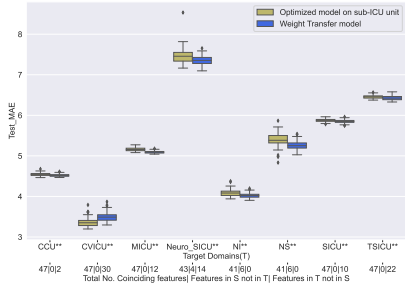
(c) Source domain (S): SICU



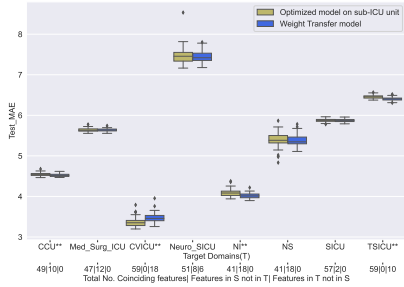
(d) Source domain (S): CVICU

Figure 3.6: Distribution of number of epochs to reach convergence with (blue) and without (lime green) weight transfer on MIMIC-IV with Med-Surg ICU, MICU, SICU and CVICU as potential source domains. Weight transfer significantly reduces training time on the majority of the target domains.

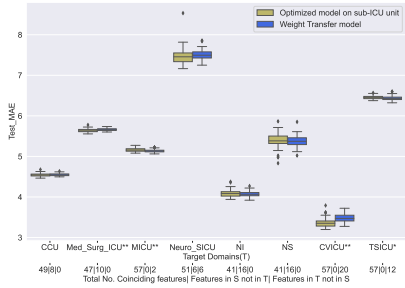
Looking at the four potential source domains on Figures 3.6 and 3.7, low populated ICUs like Neuro-SICU, NI and NS, experience the highest gains with all source domains. As target domain, Med-Surg ICU gives the overall best improvement either in computation time, prediction accuracy or even both (see Figures 3.6a and 3.7a). As target domain, CVICU is negatively impacted by weight transfer (Figures 3.6d and 3.7d).



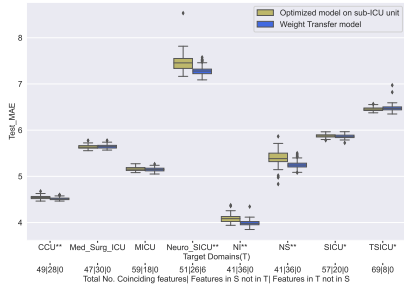
(a) Source domain (S): Med-Surg ICU



(b) Source domain (S): MICU



(c) Source domain (S): SICU



(d) Source domain (S): CVICU

Figure 3.7: Distribution of MAE on test set with (blue) and without (lime green) weight transfer on MIMIC-IV with Med-Surg ICU, MICU, SICU and CVICU as potential source domains.

The results from Tables 3.5 and 3.6 show that considering the ICU populations as a single homogeneous entity not only reduces model utility by hospital management systems because each patient is assigned to a bed belonging to a specific unit, but the resulting model from merging all the patients does not always return an optimal error estimate for individual groups of patients. For example, CVICU patients in MIMIC-IV have a prediction error of 3.32 days, which is 1.79 days less than the former model. Given a pair (c, nc) of c coinciding features and nc non-coinciding features between a source and a target domain, Figures 3.5a and 3.5b show significant benefits of weight transfer both in terms of convergence time as well as prediction error in patient sub-populations such as CCU-CTICU (51,4), MICU (47,2), NICU(47,4) and CTICU (51,16). However, we notice that with 49 coinciding and 22 non-coinciding features, the prediction error increases for CSICU patients. In MIMIC-IV, the main gain, as observed in Figure 3.6, is that narrower boxplots are obtained after weight transfer. For some targets, e.g., CVICU, a significant drop in prediction accuracy is observed in Figures 3.6a, 3.6b, 3.6c and 3.6d.

As mentioned earlier, one of the foremost benefits of transfer learning or domain adaptation is the gain in computation time. During domain adaptation, all hyperparameters except the batch size were transferred from the source to the target domains which means that no hyperparameter optimization was performed on the weight transfer models. We measured the CPU time for the systematic Bayesian optimization procedure (explained in Appendix A.2.1) using a python timer. This process on average took 1 hour and 45 minutes. Table 3.7 shows the execution times for some ICU units. OP_m: Optimized model on each ICU unit, WT_m: Weight transfer model, FT_m: Full model transfer.

Table 3.7: CPU time in hours for re-training all models 100 times using the eICU-CRD data. OP_m: Optimized model on each ICU unit, WT_m: Weight transfer model, FT_m: Full model transfer.

ICU units	100 $\times OP_m$	100 $\times WT_m$	100 $\times FT_m$
NICU	2.24	1.40	
MICU	2.99	1.95	
CSICU	1.75	1.07	
SICU	2.87	2.39	1.86

Given our resources (see headline of Appendix A.2) and the size of each ICU unit, a minimum of 1 hour was saved on units like MICU and CSICU, taking into account the time needed for hyperparameter optimization when constructing an independent model for each unit. Regarding the MIMIC-IV data, Table 3.8 shows that over 2 hours were saved for the target domain NS. As discussed before, weight transfer had a negative impact on CVICU when used as a target domain. This is reflected in Table 3.8 where this target unit required a longer computation time and could be explained by the fact that longer training times were required by the optimization algorithm to search for more optimal model weights.

Table 3.8: CPU time in hours for re-training models 100 times in MIMIC-IV data. OP_m: Optimized model on each ICU unit, WT_m: Weight transfer model.

$100 \times OP_m$ on T_k	$100 \times WT_m$ with S_i			
NS	Med-Surg ICU	MICU	SICU	CVICU
3.63	1.66	1.78	1.85	1.59
Neuro-SICU	Med-Surg ICU	MICU	SICU	CVICU
3.44	2.32	2.26	2.21	2.60
NI	Med-Surg ICU	MICU	SICU	CVICU
2.75	2.11	2.23	2.28	2.29
CVICU	Med-Surg ICU	MICU	SICU	CVICU
1.75	1.99	2.18	2.10	/

After discussing the impact of weight transfer on prediction error and computation time, we now discuss instances where domain adaptation works (even when only partial information is transferred) and instances where it does not work. Looking at the results for the eICU-CRD data (Figure 3.5), where the Med-Surg ICU unit is the only source domain, statistical significant improvements both in terms of early convergence and prediction error on all target populations except CSICU were observed. For CSICU, the error distribution appeared to increase after partial weight transfer which could be due to the fact that this unit has the highest number of non-coinciding features (22) that in turn receive random weights. As later shown in Figure 3.10, by assigning different learning rates to the two groups of features (coinciding and non-coinciding features), the error distribution becomes narrower (Figure 3.10b). In an attempt to investigate the effect of the choice of the source domain, the MIMIC-IV data was used with four potential source domains. Figures 3.6 and 3.7 show that contrary to eICU-CRD, where the source domain Med-Surg ICU occupies over 50% of the data, this same domain returns the overall best performance when its weights are used to initiate training in the rest of the domains. The Med-Surg ICU that contains both medical and surgical patients, representing a diverse patient population, has a greater impact on the other populations when used as source domain. In terms of similarity between populations, Figure 3.6c shows that the source domain SICU has the greatest

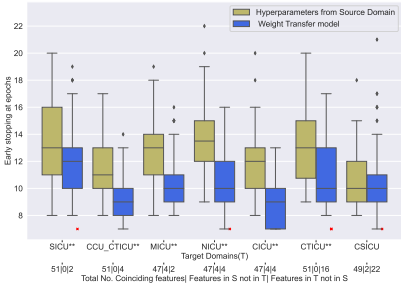
impact on the target TSICU. However, instances when domain adaptation works or not could not all be explained. For example, in Figure 3.6d, CVICU has a slightly negative though non-significant impact on CCU. Though CVICU is a surgical unit type, and CCU a medical one, SICU, which is also surgical, appears to have a significant positive impact on CCU (Figure 3.6c) with exactly the same 49 coinciding features. Thus pre-trained weights of CVICU used for total weight transfer do not improve learning as much as those of SICU.

3.4.3 Additional Analyses

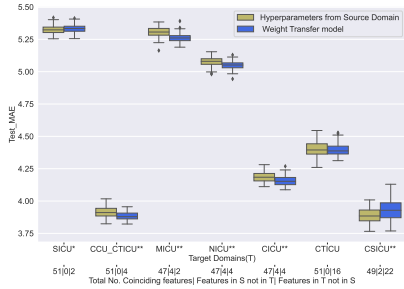
To further understand the benefits of domain adaptation, a number of additional analyses were conducted using the eICU-CRD dataset. In the first analysis, the hyperparameters found as a result of optimization on the source domain were used as the optimal hyperparameters on the target domains when training the individual unit models. In the second analysis, non-coinciding features between the source and the target domains were removed and a complete model transfer (transferring both the pre-trained weights and the model optimizer state) was carried out. In the third analysis, different learning rates were used with the input feature groups during training as explained in section 3.3.5 and Eq(3.3) with $\alpha = 10^{-1}$ and finally in the last analysis, we look at the effects of domain adaptation on model interpretability.

No hyperparameter optimization on target units for individual unit models

The intuition here was to understand whether the gains from performing weight transfer could come from poor hyperparameter optimization on each target unit and not the pre-trained weights. On Figure 3.8a, the individual unit models (lime green box plots) are obtained using the source domain hyperparameters while the blue box plots show the results obtained by transferring pre-trained weights (either totally or partially) from the source domain to the target domains.
Figures 3.8a and 3.8b show that pre-trained weights do speed up model convergence and improve prediction accuracy even when both models (with and without weight transfer) are trained with the same hyperparameters, showing the benefits of domain adaptation.



(a) Number of epochs to reach convergence

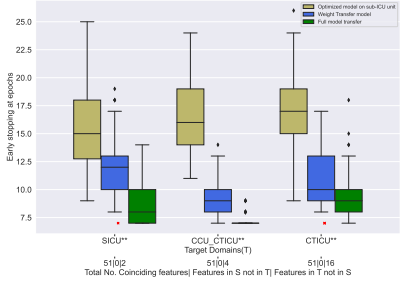


(b) Mean Absolute Error on test set

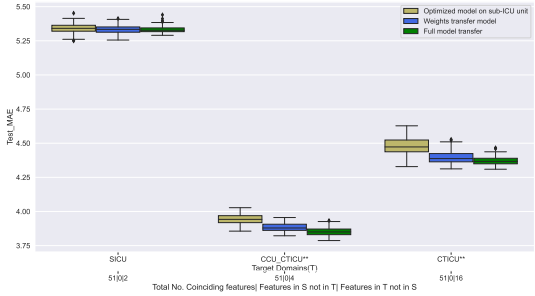
Figure 3.8: Number of epochs and error measures with (blue) and without (lime green) weight transfer on eICU-CRD where hyperparameters are obtained from the source domain.

Full Model Transfer

In this second analysis, non-coinciding features are removed from the target domains to obtain a homogeneous feature space between the source and the targets. In this way, not only the weights are transferred but also the model optimizer state. Only SICU, CCU-CTICU and CTICU qualify for this exercise as their input space is a subset of the source, Med-Surg ICU.



(a) Number of epochs to reach convergence



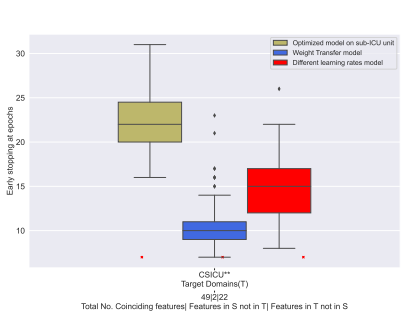
(b) Mean Absolute Error on test set

Figure 3.9: Number of epochs and MAE when performing full model transfer. Tukey's HSD test is used for multiple comparisons [3]. Individual model (lime green), weight transfer (blue) and full model transfer - including weights and optimizer state (dark green).

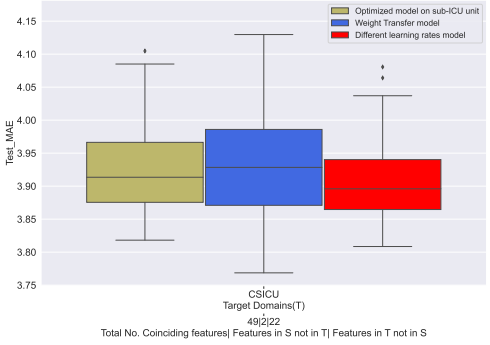
Figure 3.9 shows that if we had restricted the input space across all domains to be identical and continued training the source domain model on the target domains input data, thereby transferring not only the weights but also the optimizer state, model convergence would have occurred earlier than when transferring only the weights. It also leads to an improvement in prediction accuracy. However, this would not be possible for all units because not all recorded features in the source domain are found in all targets.

Weight Transfer with different learning rates

In this analysis, non-coinciding and coinciding features are trained simultaneously using different learning rates as explained in Section 3.3.5 and Eq (3.3). As explained before, the idea is to adjust for the knowledge contained within the pre-trained weights by reducing their learning rate such that the weights attached to these features are updated slower than the random weights, which contain no prior information.



(a) Number of epochs to reach convergence



(b) Mean Absolute Error on test set

Figure 3.10: Number of epochs and MAE on CSICU from the eICU-CRD dataset when performing discriminative learning. Tukey's HSD test is used for multiple comparisons [3]. Individual model (lime green), weight transfer (blue) and different learning rates (red).

On Figure 3.10, one observes that using discriminative learning rates produces a finer confidence interval for the prediction error (Figure 3.10b) while requiring more training time compared to weight transfer but still less than optimizing a full model (Figure 3.10a).

Model Interpretability

As final analysis, we observe the effect of domain adaptation on feature importance by extracting the 25 most important features on the same ICU unit, before and after transferring pre-trained weights. Given a trained model and the 3D training data set (number of patients $\times$ number of time steps $\times$ number of inputs), 3D feature contributions for model prediction on the test set are obtained for each patient, at each time step and for each input. Overall feature importance is then plotted by averaging over the time dimension and subsequently the patient dimension.
Patient features which can globally be split under charted parameters and lab results are observed to be present without and with weight transfer, though the order is not conserved. Figures 3.11 and 3.12 show that overall, the most important features for early prediction of LoS, that can be grouped under vital signs and lab results are not affected by weight transfer. However, we observe a change in the order of the features after weight transfer.

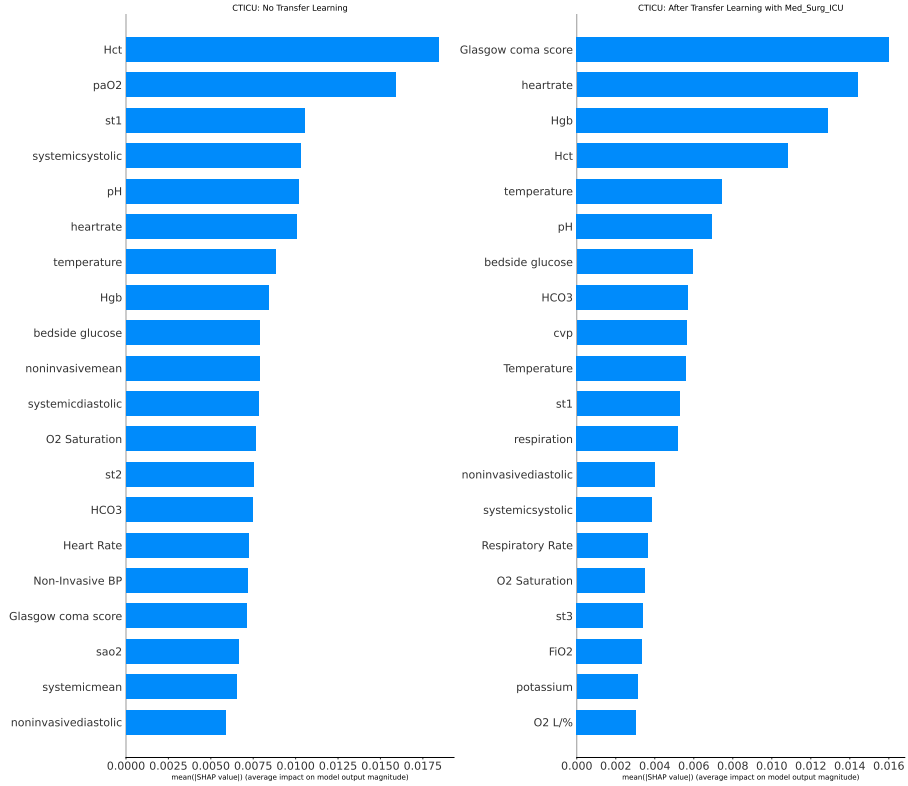


Figure 3.11: CTICU top predictive features without and with weight transfer

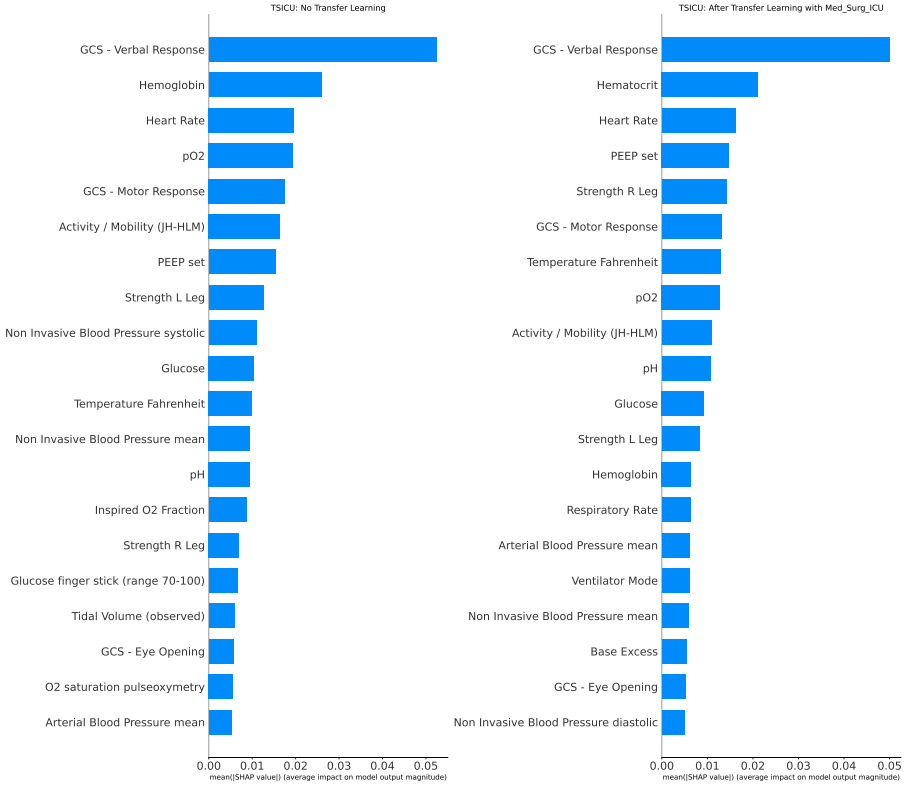


Figure 3.12: TSICU top predictive features without and with weight transfer

3.5 Conclusion

In this work, domain adaptation is exploited to reuse knowledge learned from a source unto target domains by transferring learned weights from the trained source model. By not enforcing a homogeneous feature space across all units, we allow both shared and unit-specific information to be disseminated in the targets by fine-tuning both pre-trained (from the source) and random (unit-specific) weights. This resulted in statistically significant improvements in computation time as well as prediction accuracy for most of the targets. However, we noticed that weight transfer was not always beneficial, especially when the target had a high number of non-coinciding features that receive random weights. By implementing discriminative learning , an improvement in the prediction error was noticed. In terms of feature importance, it appeared that

the proposed approach maintains the overall importance by keeping the majority of important features though it displaces the order. An additional insight drawn from this work is that, significant improvements in both prediction accuracy and computation time are observed when the source domain consists of a diverse population.

As mentioned in the context section, this project arose as a potential solution to solve or counter the first bottlenecks encountered during the data acquisition process with the university hospital of Leuven (UZ Leuven). This first step in this data acquisition process is the submission of an ethical document to the ethical committee of the hospital. This document contains explicit mention among other things of which patient population one requires, which exact patient features will be used for modeling, how these features be anonymized and by who and how and where will the data be managed (stored, accessed, pre-processed and modeled). Taking all the above in consideration with the IT team of the hospital, it then became obvious that requesting data at the medical unit level (one unit at a time) was a more feasible approach than at the hospital level to speed up processes of data acquisition and data management. By acknowledging the singularities in each ICU unit, that is, keeping only the most recorded features per unit, we believe that this work can be transposed to a real hospital environment where measures like the LILRANK [102] can be used to group or cluster medical departments based on their mode of functioning and other factors. The proposed approach here will then be applied at low cost of data management (since the centralization of data at the hospital level is not needed) within these clusters, where the source domain can be chosen as a diverse population and its weights used to initiate training on the targets. Moreover, this will be beneficial for those units with limited data as domain adaptation will allow a pre-trained larger model to be fine-tuned on available data instead of training a model from scratch.

3.6 Additional experiments

3.6.1 Naive prediction models

Naive mean and median models are included for early LoS prediction in each ICU unit. These predictions were derived as the mean and median LoS from the training set for each individual group of patients.

Table 3.9: Metrics obtained upon applying the naive mean and median models prediction on each ICU unit from the eICU-CRD dataset

ICU unit	MAE		MAPE		MSE	
	Mean	Median	Mean	Median	Mean	Median
Med-Surg ICU	5.08	4.32	1.27	0.82	47.223	47.46
MICU	6.63	5.63	1.46	0.91	81.9	83.51
CCU_CTICU	5.09	4.21	1.30	0.84	51.40	50.97
NICU	5.91	5.26	1.39	0.88	67.58	72.62
CICU	5.52	4.47	1.46	0.88	53.02	51.82
SICU	6.69	5.65	1.29	0.83	81.52	83.01
CTICU	6.28	4.90	1.21	0.76	64.88	59.79
CSICU	5.02	4.14	1.31	0.86	44.39	43.10
All stays	5.47	4.58	1.32	0.85	55.17	55.04

Table 3.10: Metrics obtained upon applying the naive mean and median models prediction on each ICU unit from the MIMIC-IV dataset.

ICU unit	MAE		MAPE		MSE	
	Mean	Median	Mean	Median	Mean	Median
Med-Surg ICU	8.09	6.02	1.82	0.95	102.46	96.09
MICU	7.08	5.39	1.54	0.84	76.68	70.25
CVICU	5.95	3.95	1.08	0.57	56.89	46.57
SICU	8.45	6.48	1.79	0.99	109.55	100.25
TSICU	8.79	7.11	1.64	0.97	115.96	113.83
CCU	6.11	4.84	1.39	0.83	68.29	66.43
Neuro SICU	9.21	7.89	1.79	1.09	144.80	151.24
NI	5.37	4.44	1.53	0.86	53.83	54.73
NS	6.26	5.56	1.44	0.97	89.07	93.43
All stays	7.51	5.70	1.53	0.83	89.17	84.43

3.6.2 Erratum

In Section 3.4.2, under the sub-sections **eICU-CRD data** and **MIMIC-IC data**, we stated that the "All patients" model MAE overestimates the error in some ICU units. Upon further review, we retract this statement, as the two models were tested on different test populations.

We urge readers to disregard these affirmations. We apologize for any inconvenience caused by this error. Importantly, the primary findings of our study regarding the benefits of domain adaptation remain unaffected.

Chapter 4

Understanding Deep Learning for Length of Stay prediction via Latent space visualization

This chapter is submitted as:

> **Momo, Lyse Naomi Wamba**, Vincent Scheltjens, Wouter Verbeke, Frank Rademakers, and Bart De Moor. **"Understanding Deep Learning for Length of Stay prediction via Latent space visualization." submitted** for publication.

Authors Contributions: L.N.W.M: Conceptualization, Methodology, Data curation, Model implementation, Writing - original draft & editing. **V.S**: Methodology, Data curation, Baseline model implementation & draft revision. **W.V**: Methodology supervision & draft review. **F.R**: Supervision, draft revision & editing. **B.DM**: Review.

4.1 Introduction

As mentioned previously, inpatient LoS is an important economic and managerial metric used to evaluate performances, resource utilization and care quality improvement both at the hospital level as well as for specific units such as, the intensive care unit (ICU) [103, 104, 105]. LoS, defined as the length of time (often measured in days) of an inpatient between admission and discharge in a single care episode [10], is also a prognostic tool that gives an indication of the patient recovery trajectory. For critically ill patients admitted to the ICU, a unit highly solicited for care [50], continuous monitoring of their LoS is important with respect to both clinical and managerial aspects. On the managerial side, LoS monitoring will help in the efficient allocation of resources such as beds and other logistics to incoming patients. Regarding the clinical aspects, it will help in the early identification of risk of deterioration and/or death [36, 29, 31, 54, 56, 50]. Using historical patient data stored in electronic health records (EHRs), LoS prediction models can be developed and embedded into clinical decision support systems (CDSS) [106]. Such systems can help in the early detection of deterioration in inpatients' states, monitor patients' recovery trajectories, reduce the risk of occurrence of adverse events, and more [51, 54, 36, 52, 29, 31, 56, 50].

Modeling LoS is a complex task as has been highlighted by a number of authors [107, 108]. The first complexity arises from the interplay between the multiple factors that affect LoS which are dependent both on the individual patient specificity as well as the hospital or unit mode of functioning. The second complexity comes from the data-related challenges which include the sequential nature of the data and the skewness of the outcome. The time dimension naturally occurring when collecting clinical data during patient hospitalization and that needs to be taken into account during modeling in order to update LoS predictions, has made deep learning models, specifically, sequence-to-sequence models, more suitable for modeling as compared to static machine learning models. The skweness in the outcome distribution has often led a number of researchers to change the outcome structure from continuous to discrete classes, predicting the probability of remaining LoS under or above a pre-defined threshold, e.g., < 5 days, ≥ 7 days, etc., as in [52, 31, 109]. The third challenge stems from the use of deep learning models, which are built by successively combining non-linear functions [110]. These models, often referred to as black-box models due to their lack of transparency, have contributed to slowdown the integration of these prediction system into CDSS and their adoption by hospital management systems [36, 111, 112]. While LoS prediction models in CDSS can only be used for managerial purposes (not for patient discharge decisions) due to legal and ethical constraints, they remain valuable

for efficient capacity planning. For this purpose, more granular predictions are preferable. Additionally, developing explainability tools is crucial to understand both the features driving predictions and the internal mechanics of the models and the inference process. This transparency can help incentivize medical institutions to adopt such AI-driven models.

The Length of Stay (LoS) prediction problem has attracted significant attention from researchers, as evidenced by numerous studies in the literature [50, 56, 109, 36, 29, 31, 54]. The intricate nature of LoS prediction, coupled with the substantial volume of patient-generated data during hospital stays, creates an ideal scenario for applying data-intensive deep learning models. Previous studies have employed these deep learning models for studying the complex interactions within clinical data for LoS and other clinical outcomes [36, 50, 29, 27, 51, 52], often benchmarking them against static machine learning models. The latter typically required manually engineered features and generally reported lower performance in comparison to their deep learning counterparts [29, 27, 49]. When addressing model interpretability, researchers often focus on ranking input features based on their contribution to the model's predictions. This method explains how the presence of a particular attribute in a patient's data impacts the predicted outcome. However, the end-users of CDSS might be interested in more granular information such as, the inner workings of the models at each time step, how patients' predictions may change over time and which attribute influences such a change. Such information is very important to plan future admissions. For deep learning models, gradients-based explanatory models such as the integrated gradients method [113] are often used. However, these methods are not very easy to understand by clinicians and hospital planners. This work aims to bridge this gap in model interpretability by introducing a simple and comprehensive approach to enhance model explainability.

In this study, we present a CDSS for remaining ICU LoS prediction for patients admitted to the Amsterdam University Medical Center (AUMC). The Amsterdam University Medical Center (AUMC) [114] is the first freely available European critical care database containing data from a 32-bed mixed surgical-medical academic ICU and a 12-bed medium care unit. In contrast to the existing US-based ICU databases where patients' stays are typically shorter, here we experience longer stays and a higher variability in the LoS of almost 10 days as reported by the standard deviation. Previous research on AUMC has established a benchmark for readmission and/or death prediction within 7 days of ICU discharge, using static machine learning models amongst which logistic regression and gradient boosted trees [114]. In this work, we establish a benchmark for remaining LoS prediction using deep learning models and we build an interactive dashboard to intuitively understand the prediction models over time, using the models' latent spaces.

4.2 Key Contributions

By assigning a unique LoS label at each hour of the stay between admission and discharge or death, continuous remaining ICU LoS predictions are computed using different algorithms. The contributions are summarized as follows;

- We predict hourly remaining LoS for critically ill AUMC patients and compare the predictive performance of sequence-to-sequence models to that of static machine learning algorithms and naive baselines.
- We develop an interactive dashboard to visualize all sequential deep learning models' transformation of the inputs, i.e., models' embeddings, on unseen test observations over time. This not only showed the models' capabilities of clustering patients over time with respect to their true LoS but also revealed neighborhoods of patients at risk of experiencing in-hospital mortality with only 5 hours of post ICU admission data.
- We perform an additional analysis inspired by the visualization in which model inputs are constructed using a sliding window l, with the optimal window value obtained by hyperparameter optimization and show that this approach is both time and resource efficient.
- Finally, we provide a comprehensive code for data extraction and preprocessing of the AUMC data adapted from the pipeline by [50] to manipulate the raw AUMC data of over 200GB on any computer.

4.3 Methodology

4.3.1 Dataset

The Amsterdam University Medical Center (AUMC) [59] is the first freely available European critical database containing detailed data points of 23,106 adult ICU stays from 20,109 patients admitted to the Amsterdam University Medical Center between 2003 and 2016. Based on the preprocessing that will be described in 4.3.2, the final data cohort of 20,481 stays from 19,360 patients was extracted from AUMC. The high level descriptive statistics of the data are given in Table 4.1.

Table 4.1: Descriptive statistics of final data cohort

No. ICU stays	20481
Train	14329
Test	3066
Validation	3086
Basic demographic statistics	
Age class (mode)	60-69
Gender (Male)	64%
In-hospital mortality	31.57%
LoS statistics in days	
mean	4.60
median	1.08
std	9.61
skewness	6.04

4.3.2 Preprocessing

The original AUMC database contains nearly 1.0 billion clinical observations and 5.0 million medication records.

During preprocessing, 115 input features (laboratory results and vital signs) from 20,481 stays pertaining to 19,360 unique adult patients with a LoS > 5 hours were carefully extracted from the original database following a systematic procedure. Only laboratory features present in at least 25% of stays and vital signs present in at least 35% of stays were retained for subsequent modeling and are presented in Table 4.2. After data extraction, each feature was hourly aggregated using the mean. Imputation of missing information was carried out similarly as in [50] by first forward filling the last recorded value per feature and per patient. Next, the original feature space was augmented with a matrix of an identical size and filled with decay curves. These curves indicate the time points at which data was imputed and the time from which imputation was done [50, 53, 90]. All demographic information available in the database was extracted and after one-hot encoding, the final input space resulted into 271 features consisting of time varying features (from lab results and vital signs) and repeated demographic information over the hours.

Further data preprocessing was needed for the static machine learning models. Combining strategies from previous works [29, 115, 49], different statistical measures were extracted from each of the original 104 original time series (without the decay curves). These statistics include the range (maximum-

minimum), the difference between the last and first values, the standard deviation, the frequency of measurement and the mean.

Table 4.2: Input Features extracted from the AUMC database

Static	Lab	Non-Lab	Computed feature
admission count	ALAT	ABP diastolisch	Time in ICU
admission year group	APTT	ABP gemiddeld	
age group	ASAT	ABP systolisch	
gender	Act.HCO3	Adem Frequentie (Set)	
height group	Alb.Chem	Ademfreq.	
location	Alk.Fosf.	Ademfrequentie Monitor	
origin	Amylase	Backup druk (Set)	
specialty	Anion-Gap	Barometer druk	
urgency	B.E	CO2 min prod	
weight group	Bezinking	CO2 tidal productie	
	Bilirubine	CVD	
	CK	Cdyn	
	CO-Hb	Eind exp. druk	
	CRP	Eind insp. cyclus (Set)	
	Ca-ion (7.4)	End tidal CO2 concentratie	
	Calcium totaal	EtCO2 (%)	
	Chloor	Exp. minuut volume	
	Cl(onv.ISE)	Exp. tidal volume	
	Eryś	Hartfrequentie	
	Fosfaat	Insp. Minuut volume	
	Glucose	Insp. Rise time (Set, %) nieuw	
	Hb	Insp. Rise time (Set, sec) nieuw	
	Hb(v.Bgs)	Insp. Time (sec) nieuw	
	Ht	Insp. tidal volume	
	Ht(v.Bgs)	MVe Spontaan nieuw	
	K (onv.ISE)	Maaghevel	
	Kalium	Mean luchtweg druk	
	Kreatinine	Niet invasieve bloeddruk diastolisch	
	LD	Niet invasieve bloeddruk gemiddeld	
	Lactaat	Niet invasieve bloeddruk systolisch	
	Leuco's	O2 concentratie	
	MB-Massa	O2 concentratie (Set)	
	Magnesium	O2 l/min	
	Meettemp	Ontlasting	
	Met-Hb	PAP diastolisch	
	Na (onv.ISE)	PAP gemiddeld	
	Natrium	PAP systolisch	
	O2-Content	PC boven PEEP (Set)	
	O2-Saturatie	PEEP (Set)	
	OxyHb-Fr.	P Bovenste (Set)	
	PO2	PS boven PEEP (Set)	
	Prothrombinetijd	Piek druk	
	Prothrombinetijd	SBI (Zwakke ademhalingsindex)	
	Thrombo's	SIMV ademcyclus (Set)	
	TroponineT	ST segment analyse afleiding	
	Ureum	ST segment analyse afleiding I	
	UrineCAD	ST segment analyse afleiding II	
	WOBv	Saturatie (Monitor)	
	Y-GT (bloed)	Temp Axillair	
	eGFR(MDRD)	Thoraxdrain1 Productie	
	pCO2	Thoraxdrain1 Stand	
	pH	Ti (sec) nieuw	

4.3.3 Prediction task

Hourly remaining ICU LoS between admission and discharge or death is predicted using data recorded up to 14 days (336 hours) post ICU admission. Formally, the prediction task is to approximate the true remaining LoS, y_{t+1} by $\hat{y}_{t+1}$ from learning the input transformation $f(\boldsymbol{X}_t)$ via the function f. The target y_{t+1} is obtained from subtracting each hour $t \in \{1, 2, ...\}$ from the observed LoS (in hours).

The input sequences that are given to the models are constructed by initially using the first 5 hours of data to predict the remaining LoS at hour 6. Subsequently, one hour is added at a time to the sequence to predict the future remaining LoS. In the remainder of this work we will refer to this approach as the stacking approach.

4.3.4 Experimental Settings

Naive models (mean and median training LoS), traditional machine learning models (linear regression and a 2-layer feed-forward neural network as done in previous works [29, 30]) and sequential models were all used for modeling. These sequence-to-sequence models are: Long-Short Term Memory network (LSTM), Gated Recurring Units (GRU), the Transformer network, the Temporal Convolutional Neural networks (TCN), and the Temporal Convolution Neural networks with attention (TCN-att). The LSTM, GRU, Transformer and the TCN models were previously explained in Chapter 2. The TCN-att model is obtained by applying a self-attention mechanism over the different channel views obtained from the TCN model.

For the mean and median models, predictions were obtained by extracting the mean and median LoS from the training set, which were later used for predictions on the unseen test observations. The overall modeling process is summarized schematically in Figure 4.1.

All models were trained using the AdamW optimizer [116] and the mean square log error (MSLE) as loss function. All models were implemented with Pytorch and trained on the same Skylake architecture using a single NVIDIA P100 GPU with 16GB of GDDR for each iteration of training. Computational resources were made available by the VSC (Vlaams Supercomputer Centrum).

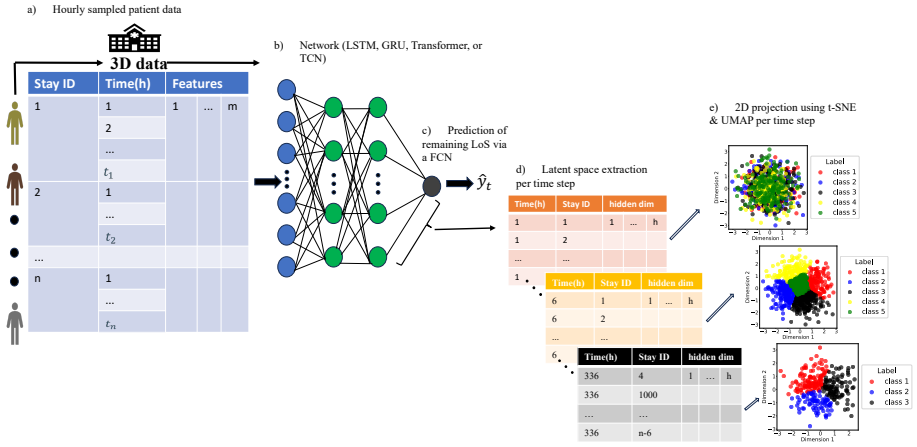


Figure 4.1: Overview of modeling process. In **a)** laboratory and vital signs extracted data are hourly sampled using the mean to produce a 3D tensor input matrix where the time dimension varies across patients, i.e., $t_1 \neq t_2 \neq t_n$. The processed data is sent through four different neural networks (LSTM, GRU, Transformer and TCN) in **b)** and then through a single-neuron FCN for predictions as seen in **c)**. All models' embeddings obtained at the last layer in b) before being sent to c) are then extracted at each time step in **d)** between hour 1 and hour 336 and projected onto a 2D space in **e)**.

4.3.5 Latent space projection and dashboard construction

In order to understand how the models learn the clinical information embedded in the patient data, we extracted the latent spaces of all models and applied dimensionality reduction techniques. These reduced representations were then visualized interactively on a dashboard. The t-Distributed Stochastic Neighbor Embeddings (t-SNE) [117] and the Uniform Manifold Approximation and Projection (UMAP) [118] were used to project the high dimensional embeddings spaces to a two-dimensional space. Though UMAP has proven to be superior to t-SNE in terms of scalability, speed and its ability to preserve the global structure in the data (that is, the inter-cluster distance between "semantically" closer structures), both methods are very similar in that they aim to minimize the divergence between the original high-dimensional data representation and the targeted lower-dimensional representation. The latter onto which the data points are projected. The dashboard was constructed using both the Dash and Plotly python frameworks [119, 120].

4.3.6 Evaluation Metrics

The model performance was reported using the Mean Absolute Error (MAE), the Mean Squared Log Error (MSLE), and the Mean Squared Error (MSE) defined as follows:

$$MAE = \frac{1}{n}\sum_{t=1}^{n}|\boldsymbol{y}_t - \hat{\boldsymbol{y}}_t|$$

$$MSLE = \frac{1}{n}\sum_{t=1}^{n}(\log(\boldsymbol{y}_t + 1) - \log(\hat{\boldsymbol{y}}_t + 1))^2$$

$$MSE = \frac{1}{n}\sum_{t=1}^{n}(\boldsymbol{y}_t - \hat{\boldsymbol{y}}_t)^2$$

4.4 Results & Discussion

This section presents and discusses all numerical results obtained including all insights obtained from using the dashboard.

4.4.1 Hyperparameter Optimization

Hyperparameter optimization was carried out using both the grid search approach and population based training [121]. The obtained hyperparameters for each model are shown in Appendix B.1 and Table B.1. Given the relatively small dataset size (14,330 patients) and a high LoS variability of 9.61 days, the batch size appeared an important hyperparameter during optimization. A large batch size led to poor performance, as the models struggled to learn patterns amidst the substantial variability in LoS.

4.4.2 Model Performance

Upon stacking patient inputs one hour at a time for continuous remaining LoS prediction, each model performance across different metrics, and associated training time and number of model parameters are reported in Table 4.3.

Table 4.3 shows that the TCN model with attention (TCN-att) reports the best average performance on all metrics with a MAE of 4.70 days.

Table 4.3: Performance on the test set. The best performance is shown in bold. Mean ± standard deviation of metrics reported are calculated based on 15 iterations for each model. τ is the training time for one complete model run measured in hours. The TCN model with attention (TCN-att) achieves the best performance across all metrics, although it requires more training time.

Model	MAE	MSE	MSLE	τ	No. params
Mean	8.28	284.3	1.24		
Median	9.01	333.09	2.26		
LR	6.95±0.01	235.08±2.23	0.79±0.00		
FFN	6.94±0.02	233.01±2.20	0.77±0.01		
LSTM	6.59±0.05	227.62±6.81	0.68±0.01	6.17	337539
GRU	6.61±0.12	228.10±4.94	0.68±0.01	**6.04**	114691
Transformer	6.62±0.07	230.93±6.20	0.68±0.02	7.56	28129
TCN	6.61±0.07	225.54±4.06	0.69±0.02	6.42	**22647**
TCN-att	**4.70±0.12**	**175.09±5.87**	**0.27±0.02**	8.22	186708

4.4.3 Latent Space Visualization

During modeling, the input data was passed through different layers of each model in parallel and the final representations at the last layer were sent to a FCN for predictions. The adequate number of neurons, i.e., the hidden dimension of each model was determined by hyperparameter optimization as 128 for LSTM, 64 for GRU, 32 for Transformer, 11 for TCN, and 64 for TCN-att. The trained models in the stacking input approach were applied on the unseen test data at inference. These enriched data representations, i.e., model embeddings were then extracted (before prediction) and projected onto a 2D space using both t-SNE and UMAP. In essence, for each model and at each discrete time step, the embeddings were extracted to create a new dataframe, in total producing 336 dataframes (corresponding to each hour of data), each populated with embeddings of those patients not yet discharged at hour t (as shown in Figure 4.1d) and then projected. The 2D projections, with each point on the scatter plot representing a test patient, were then labeled using the corresponding true LoS converted to ten classes. This exercise was carried out in an attempt to understand how each model learned the stacked input data at each time step thereby contributing to enhance model explainability.

Model understanding of clinical information

The t-SNE projections of all models' embeddings at specific time points (hours) are shown in Figure 4.2, Figure 4.3, Figure 4.4, Figure 4.5, and Figure 4.6 for LSTM, GRU, the Transformer model, the TCN, and the TCN-att respectively. These visualizations are seen on the interactive dashboard demo.

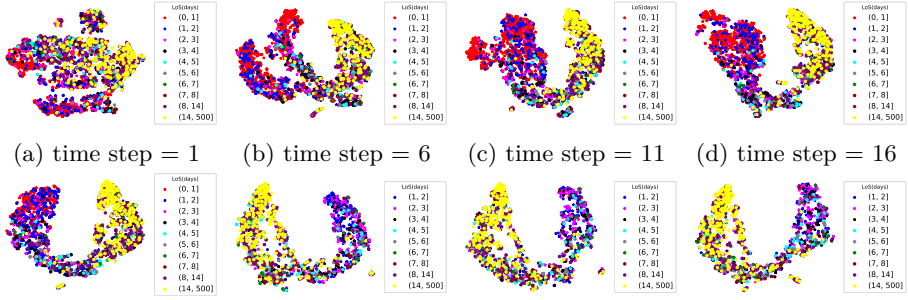
At time step 1 (hour 1), none of the five models have yet identified any patterns in the data. However, by hour 6, a clear distinction emerges between short stays ($0 < \text{LoS} \leq 3$ days, represented by red, blue, and magenta stars) and medium to long stays (LoS > 3 days) in Figures 4.2(b), 4.3(b), 4.4(b), and 4.5(b). Notably, the TCN-att model in Figure 4.6(b) demonstrates a more distinct and precise separation between the different LoS classes at hour 6, highlighting its superiority over the other four models.

Subsequently at hour 16, a third cluster emerges for the LSTM, GRU, and the TCN networks, where medium stays ($3 < \text{LoS} < 7$ days represented in black, cyan and grey circles) are separated further from both the short stays (star-shaped in red, blue and magenta) and long stays ($7 \leq \text{LoS} < +\infty$ days represented as circled-shaped in yellow, purple, dark-red and green). These three clusters are visible in (Figure 4.2), (Figure 4.3) and (Figure 4.5), and this separation amplifies even further at hour 21. For the Transformer network, the medium stays ($3 < \text{LoS} < 7$ days represented in black, cyan and grey circles) form a separate cluster from the short and long stays only at hour 21, and just like TCN, the embeddings at this hour form multiple clusters compared to the three clusters seen in the recurrent architectures. At later time steps (hours 26-31), the separation persists as patients with the shortest length of stay (1 day or 24 hours) are discharged, and the remaining patients with similar LoS values cluster more tightly together. The TCN-att model on the other hand continuously improves its ability to separate the different classes over time, with only occasional confusion at the boundaries where classes with similar LoS values overlap.

Intuitively, all models progressively learn the clinical patterns that distinguish patients with varying lengths of stay, with the TCN-att model showing superior ability in this regard.

As mentioned in the introduction, LoS prediction, i.e., anticipation on bed occupancy rate is important especially for hospital planning. Timely LoS predictions can enable effective planning of surgeries, resource allocation and more in ICU as well as in the entire hospital. In a true hospital setting, when patients are admitted, it is important to be able to separate the day patients from the overnights and within the overnights, short from medium and long stays. This will enable to plan incoming patients (when the short stays have

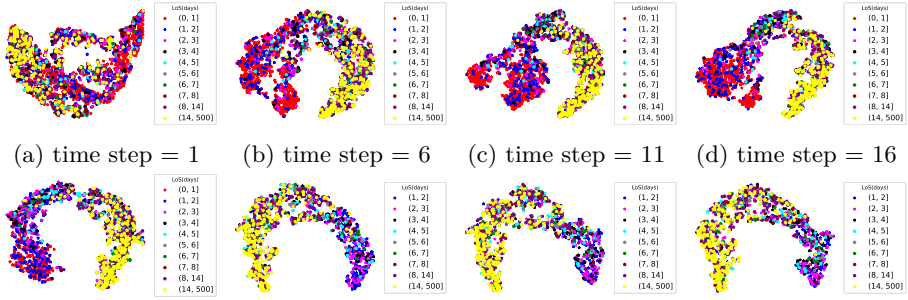
been identified), and allocate more adequate resources to the identified long stays. The latent space visualization exercise then helps in this regard as it gives the intuition that early on admission data (only first 5-6 hours of post ICU admission data) can enable stays clustering in ICU.



(a) time step = 1 (b) time step = 6 (c) time step = 11 (d) time step = 16

(e) time step = 21 (f) time step = 26 (g) time step = 31 (h) time step = 36

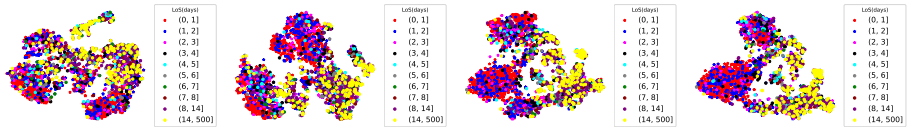
Figure 4.2: t-SNE 2D projections of LSTM model embeddings at different hours, labeled with LoS.



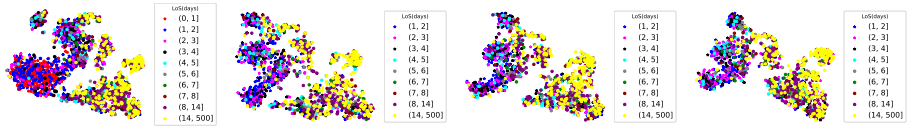
(a) time step = 1 (b) time step = 6 (c) time step = 11 (d) time step = 16

(e) time step = 21 (f) time step = 26 (g) time step = 31 (h) time step = 36

Figure 4.3: t-SNE 2D projections of GRU model embeddings at different hours, labeled with LoS.

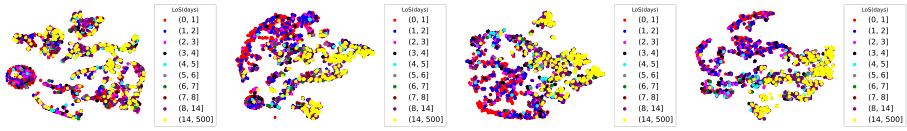


(a) time step = 1 (b) time step = 6 (c) time step = 11 (d) time step = 16

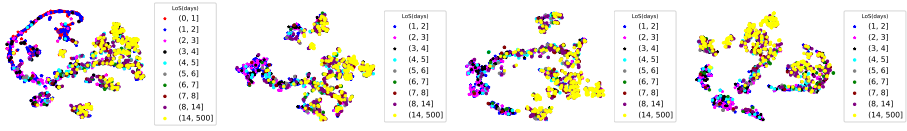


(e) time step = 21 (f) time step = 26 (g) time step = 31 (h) time step = 36

Figure 4.4: t-SNE 2D projections of Transformer model embeddings at different hours, labeled with LoS.

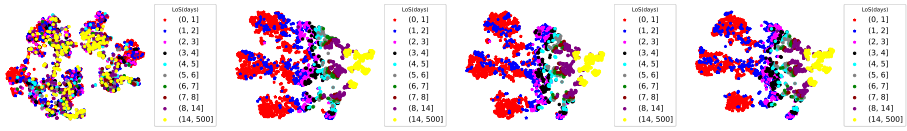


(a) time step = 1 (b) time step = 6 (c) time step = 11 (d) time step = 16

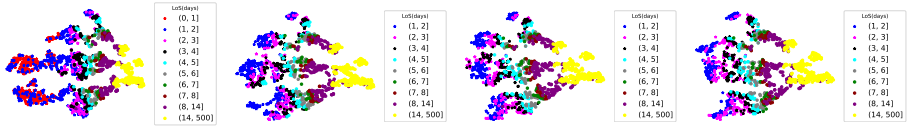


(e) time step = 21 (f) time step = 26 (g) time step = 31 (h) time step = 36

Figure 4.5: t-SNE 2D projections of TCN model embeddings at different hours, labeled with LoS.



(a) time step = 1 (b) time step = 6 (c) time step = 11 (d) time step = 16



(e) time step = 21 (f) time step = 26 (g) time step = 31 (h) time step = 36

Figure 4.6: t-SNE 2D projections of the TCN with attention model embeddings at different hours, labeled with LoS.

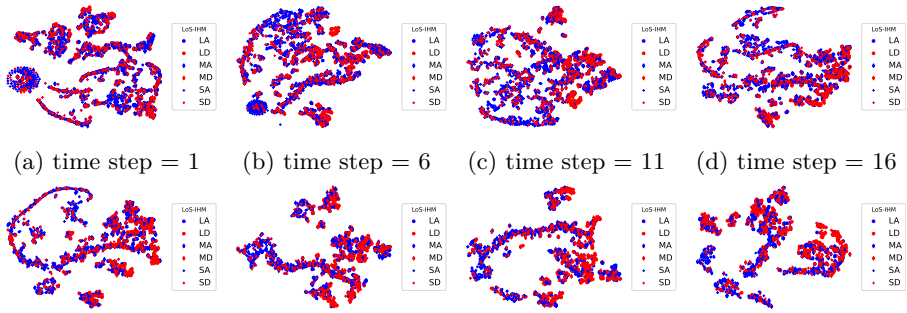
Model identification of patients at risk of in-hospital mortality (IHM): in-hospital mortality used as labels

The extracted and projected model embeddings per time step were also labeled using in-hospital mortality, a feature not used during modeling, neither as inputs nor output. For improved readability, the labels were split into 6 classes, i.e., SA, SD, MA, MD, LA, LD denoting short (S), medium (M) and long (L) stays for which the patients are either alive (A) or dead (D) at the time of discharge. More specifically, the intervals for the different LoS are constructed such that, S: $0 < \text{LoS} \leq 3$ days, M: $3 < \text{LoS} \leq 7$ days and L: $7 < \text{LoS} < +\infty$ days.
Figure 4.7 shows the t-SNE projections of TCN embeddings, labeled with both LoS and in-hospital mortality. On Figure 4.7, survivors are colored in blue and patients that eventually die appear in red.

From Figures 4.5 and 4.7, we observe that the long stays region (region in which yellow is most dominant on Figure 4.5) coincides with the dead patients region (red-dominant region on Figure 4.7) which is inline with the known correlation that exists between long stays and increased risk of mortality [114]. Given that all models were trained to predict solely the Length of Stay (LoS) without considering in-hospital mortality as either an input or output feature, the agreement between Figure 4.5 and Figure 4.7 (as explained above) shows that the trained models[1] effectively learned the clinical patterns in the patient parameters that can predict more than one clinical outcome. Thus, the models did not learn random noise.

In a clinical setting, this implies that patients likely to experience long stays and those at high risk of in-hospital mortality can be identified and prioritized as early as 5-6 hours after ICU admission. Risk stratification in the ICU is crucial, and mortality prediction has been extensively studied in the literature. Typically, mortality predictions are made using data from the first 24 hours post-ICU admission, similar to the APACHE score (Acute Physiology and Chronic Health Evaluation II), which serves as an early warning tool to identify at-risk patients and allows clinicians to adjust the care protocol accordingly [122]. However, our findings suggest that risk assessment can begin much sooner, within just 5-6 hours.

[1]This pattern observed on the TCN model was also observed on the LSTM, the GRU and the Transformer models.



(a) time step = 1 (b) time step = 6 (c) time step = 11 (d) time step = 16

(e) time step = 21 (f) time step = 26 (g) time step = 31 (h) time step = 36

Figure 4.7: t-SNE 2D projections of TCN model embeddings at different hours, labeled with both LoS and in-hospital mortality.

Early identification of a cluster of patients at high risk of in-hospital mortality (IHM)

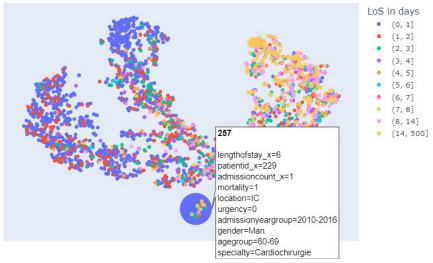
Using the interactive dashboard and the LSTM model, we monitored patient id 257 from hour 1 to hour 5 (just before dying at hour 6). At hour 5, when performing the last prediction of this patient, we identified that the neighborhood of patient id 257 (see Figure 4.8(a)), a neighborhood of exactly 50 patients including id 257, had an observed mean LoS of 5.4 days (129.64 hours) and median LoS of 0.81 days (19.5 hours) with a 94% mortality rate as shown in Figure 4.8(b) (this is visible at minute 02:36 of the dashboard demo) [2].

By monitoring the predictions of patient id 257, that experiences IHM at hour 6, we observed that the neighborhood of this patient, constituted of 50 patients with over the majority having a LoS of less than 3 days (see Figure 4.8) eventually experience IHM. In a hospital, using prospective data, it means that, if a patient dies early on, few hours after admission, and appears in a cluster with other patients, these patients should then be monitored very closely as they may present a high risk of IHM as well.

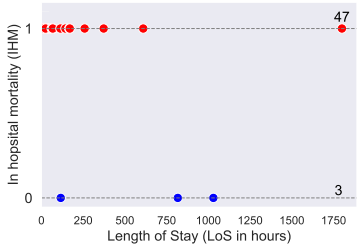
Identification of a near-to-discharge patient

Using the interactive dashboard and the Transformer model, we discovered that patient id 6, a short stay patient with observed LoS of 48 hours, was initially predicted as a long stay from hour 1 to hour 34 (as seen on Figures 4.9(a) and

[2]The same observation is made with the TCN and the Transformer models which also produce clusters of patients.



(a) LSTM embeddings at time step = 5 with id=257 highlighted



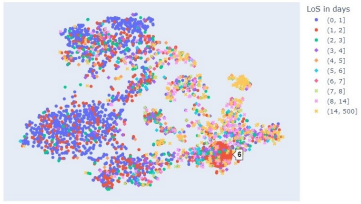
(b) LoS vs IHM plot of all 50 patients in the neighborhood of id=257.

Figure 4.8: Cluster of 50 patients identified at hour 5 (left figure) where 94% of the patients will experience in-hospital mortality at different hours of their stays and for which 75% has a LoS ≤ 3 days (right figure).

(b)). Subsequently, from hour 35 till hour 47 (just before discharge at hour 48), this patient was predicted correctly as a short stay. This is also seen on the dashboard demo at minute 04:00.

Investigating this with the original data, it appears that, patient id 6 was admitted to ICU after a neuro-surgery and placed in a coma from admission (i.e., hour 1) to hour 31 (as seen on Figure 4.10 where mechanical ventilation was administered).

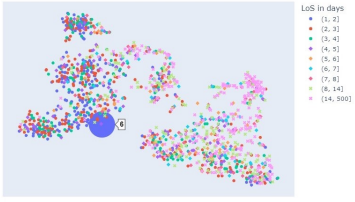
For capacity planners, it is very important to be able to identify patients that can be discharge in the near future so that upon confirmation with the unit head, subsequent plannings can be done on the previously occupied bed. Patient id 6 is initially predicted as a long stay patient from hour 1 to hour 34 (Figure 4.9) when the patient is maintained in a state of coma and is under mechanical ventilation from hour 1 to hour 31 (see Figure 4.10). When the patient is allowed to breath by himself from hour 32, the model then adjust its prediction at hour 35 and predicts him as a short stay till discharge. The model could not predict that the patient was doing better and will eventually not need mechanical ventilation because this is indicated by a subsided cerebral oedema and a drop in the intracranial pressure, which information, was not available to the model as none of the 104 time series information extracted can be used as proxy for determining these.



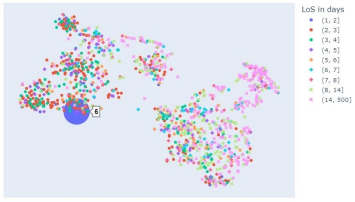
(a) time step = 11



(b) time step = 34



(c) time step = 35



(d) time step = 47

Figure 4.9: Patient id=6 predictions' over time shown at hours 11, 34, 35 and 47 using the Transformer model. Figure shows how the patient is incorrectly predicted as a long stay at hour 11 to hour 34 and switches side to the short stay cluster at hour 35 till hour 47.

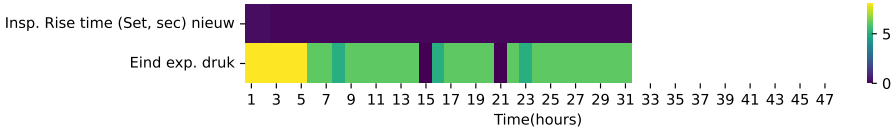
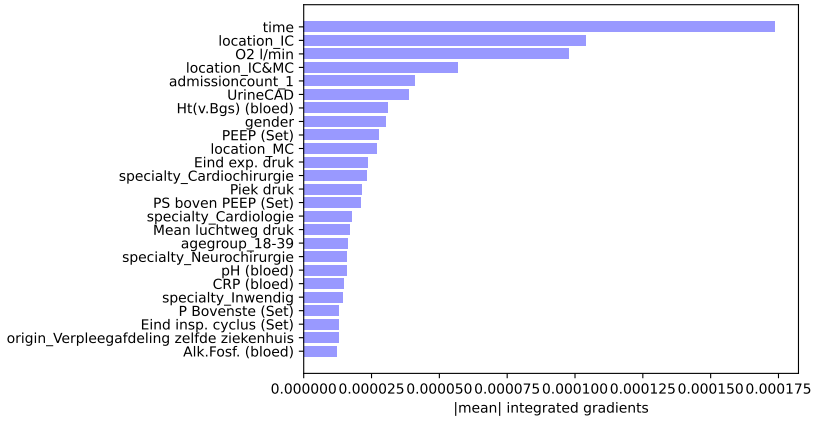


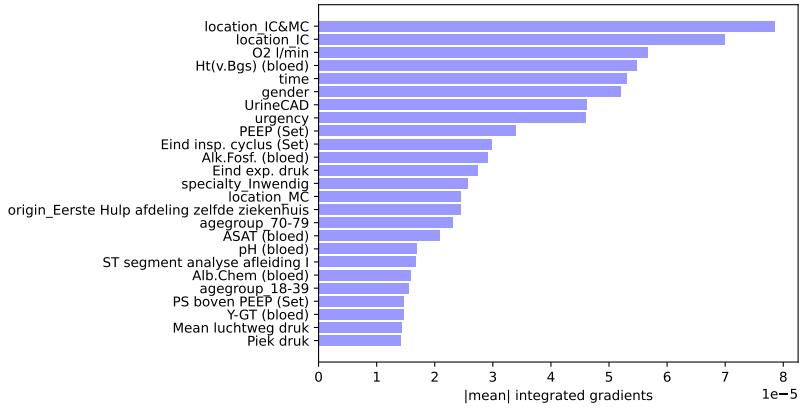
Figure 4.10: Heatmap of mechanical ventilation indicators from admission to hour 47

4.4.4 Feature Importance

In order to further enhance interpretability, beyond the inner working of the models, features driving the LoS predictions were computed using the integrated gradient methods on the unseen test data and their importance ranked.



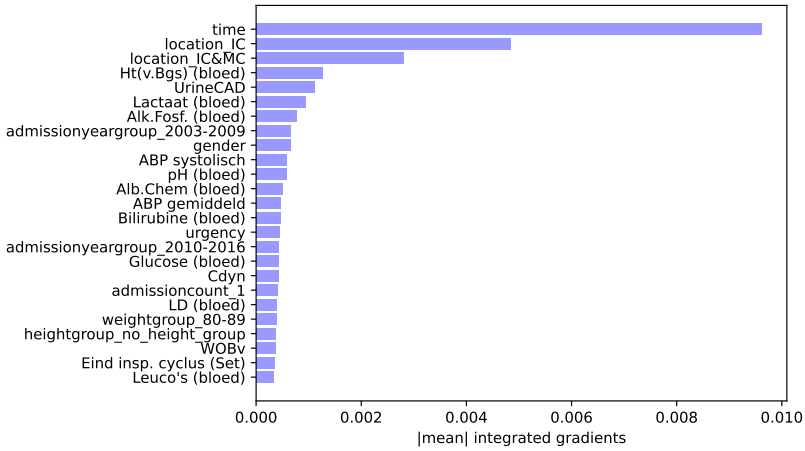
(a) LSTM



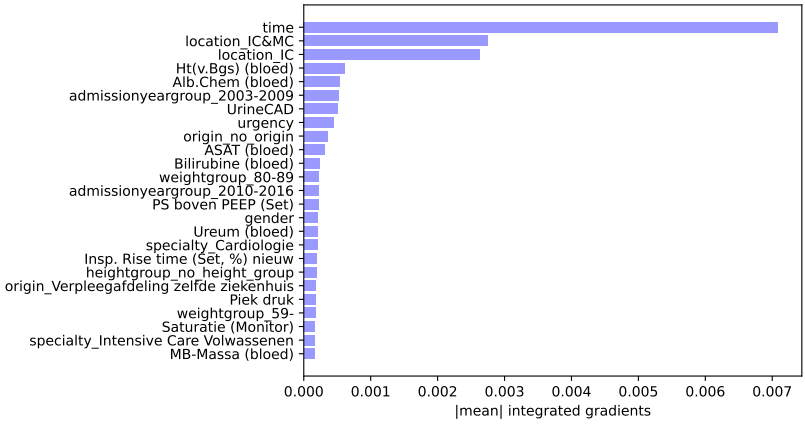
(b) GRU

Figure 4.11: Top 25 most important features on the unseen test patients using LSTM (a) and GRU (b)

Figure 4.11 and Figure 4.12 show the 25 most important features towards remaining LoS prediction per model. From these figures, the following is observed:



(a) Transformer



(b) TCN

Figure 4.12: Top 25 most important features on the unseen test patients using the Transformer (a) and the TCN network (b)

- Similarly to the embeddings plots where LSTM and GRU show a similar clustering, which is different from the Transformer model and the TCN network, here, the top discriminating features show a similar pattern. The LSTM and GRU models focus on the unit of admission of the patient (*location_IC, location_IC&MC, location_MC*), ventilation-related features (*O_2l/min, PEEP(Set), Eind insp.cyclus (Set), Eind exp.druk*), low-blood level indication (*Ht(v.Bgs)*) the gender (*male*), and

a potential bedridden condition (UrineCAD which means catheterurine). These features which could indicate circulatory problems and the need for ventilation (that is, patients need to be intubated) are more likely to extend the stay.

- The Transformer and the TCN models in Figure 4.12 indicate the time variable as being most important, followed by the admission location, low-blood level (*Ht(v.Bgs)*), UrineCAD, emergency admission, circulatory problems related features (*Lactate, Alkanine Fosfatase, Albumin*). The time factor indicates that, the more time spent in ICU, the higher the risk of staying even longer.

4.4.5 Additional Analysis

The results from the visualization showed that the models are able to learn or gain knowledge from early data inputs (5 hours of data only), updating their predictions with time. Thus, instead of stacking input sequences (one hour at a time) as often done in the literature, model input sequences were also constructed using a sliding window. By doing this, we also assess the relevance of using a long history length, i.e., early admission laboratory results and vital signs collected at admission to predict remaining LoS at hour 200 (i.e., 8.3 days later). The length of the window l was added to the list of hyperparameters. Intuitively, by testing different values of l, the models experiment with different history lengths, with a minimum of $l = 2$, i.e., using the most recent 2 hours of available data to predict remaining ICU at the next hour.
Table 4.4 reports model performance of the best performing model in this analysis and the associated value of the window l. Model performance with arbitrary and high values of the window l are also reported.

The results indicate that each model either shows slight improvements in performance or delivers comparable performance to its stacking input counterparts, while requiring nearly 2 hours less training time. Additionally, Table 4.4 demonstrates that short window lengths, such as the 2-hour window used for the Transformer network, can be effectively employed for model training, yielding performance similar to that achieved by stacking multiple hours of input data. This observation falls in line with gathered medical expertise (after discussion with experts) that most recently recorded time-varying patient information of laboratory tests and vital signs are most valuable for taking the decision to discharge a patient. In addition, it helps to circumvent memory issues encountered when training with very long input sequences (e.g. stacking 300+ hours of patient parameters per patient in the batch with the stacking approach).

Table 4.4: Performance on test set with associated window length l reported. The best performance is shown in bold. Mean ± standard deviation calculated based on 15 iterations for each model. τ is the training time for one complete model run measured in hours. The TCN-att network reports the best performance with a window length of only 5 hours.

Model	l	MAE	MSE	MSLE	τ
LSTM	4	6.57±0.06	226.65±4.95	0.66±0.01	4.20
GRU	3	6.44 ±0.04	221.60±3.67	0.66±0.01	**4.04**
Transformer	2	6.59±0.06	231.95±4.71	0.68±0.01	5.38
TCN	6	6.64±0.06	226.37±5.08	0.68±0.01	4.55
TCN-att	5	**4.70±0.12**	**179.50±7.42**	**0.28±0.02**	6.56
TCN-att	12	5.08±0.13	185.77±7.47	0.31± 0.02	5.21
TCN-att	24	5.44±0.69	200.86±32.33	0.31±0.83	5.37
TCN-att	48	6.82±1.79	260.38±67.17	0.86±1.50	3.12
TCN-att	72	6.67±0.19	256.72±11.81	0.42±0.03	2.94

4.5 Conclusion

In this work, we predicted hourly remaining Length of Stay (LoS) for critically ill patients admitted to Amsterdam UMC. Our results demonstrated that deep learning models outperform traditional machine learning methods and naive mean and median models, reducing the prediction error by nearly 2.5 days (from a MAE of 6.94 days to a MAE of 4.70 days). Of the five sequential deep learning models evaluated, the Temporal Convolutional Network with attention (TCN-att) returned the best performance using both the stacking input approach and the sliding window approach to input construction. The sliding window approach, in particular, delivered a comparable performance while being computationally efficient, requiring fewer training resources and less time. Next, in order to enhance model explainability and provide a clearer understanding of the models' inner workings, we applied dimensionality reduction techniques to the latent spaces of all models and visualized the results interactively using a custom-developed dashboard. This analysis revealed that, after just 5-6 hours of data, all models were able to distinguish short stays ($0 < \text{LoS} \leq 3$ days) from medium ($3 < \text{LoS} < 7$ days) and long stays ($7 \leq \text{LoS} < +\infty$ days). This separation became more pronounced over time (between hours 16-21), with medium stays also being clearly differentiated from long stays. The TCN-att model demonstrated a distinct separation between extreme LoS classes very early on (5-6 hours of data), although some overlap remained among patients belonging to neighhouring LoS classes. Moreover, by labelling these clusters not

only with LoS but also in-hospital mortality, the model showed that they are able to capture the clinical information that associates long stays with increased risk of mortality. Other valuable insights drawn from visualizing the models embeddings included the early identification of a cluster of patients with a high risk of in-hospital mortality and the identification of a patient close to discharge. These insights are valuable both for capacity planning and for patient care.

Chapter 5

Multimodal Learning Through Time with Cross-Attentions for Length of Stay Prediction

This chapter is submitted as:

> **Momo, Lyse Naomi Wamba**, Vincent Scheltjens, Wouter Verbeke, Frank Rademakers, and Bart De Moor. **"Multimodal Learning Through Time with Cross-Attentions for Length of Stay Prediction" to be submitted for publication.**

Authors Contributions: L.N.W.M: Conceptualization, Methodology, Data curation, Model implementation, Writing - original draft & editing. **V.S**: Methodology & draft revision. **W.V**: Methodology supervision & draft review. **F.R**: Supervision, draft revision & editing. **B.DM**: Review.

5.1 Introduction

The length of a patient's hospital stay is closely linked to the diagnosis of their condition. Diagnosing a patient involves conducting a range of tests, such as laboratory assessments, as well as providing various care procedures such as monitoring vital signs, administering medications and more. When the practitioner considers it necessary, additional functional tests, such as radiography scans, may be requested for a specific patient. These different assessment procedures are conducted at varying time points, depending on the patient's recovery progress and the availability of resources. The resulting electronic health record (EHR) of a patient then contains records of different parameters across different data modalities, with timestamps that do not necessarily align.

The integration of diverse data modalities for outcome prediction using deep learning architectures is an actively researched topic, both in the medical field and beyond. In the medical field, multimodal learning has been extensively utilized by numerous researchers for predicting clinical outcomes. Most studies have focused on combining structured data from physiological patient parameters with unstructured data from clinical reports [123, 124, 52, 125, 126, 127, 31]. Researchers demonstrated that using laboratory results and vital signs as the base modality for predicting binary LoS greater than 3 or 7 days, prediction performance was enhanced by incorporating information from unstructured clinical reports. These researchers performed binary LoS classification, along with other clinical outcomes like mortality, using the first 24 or 48 hours of post ICU admission data, meaning that they had just one prediction time point. Recently, the authors in [128] conducted a similar research using three modalities, incorporating thoracic radiography scans along with physiological parameters and clinical notes.

With the exception of [125], previous studies utilized patient cohorts where all patients had data from each modality, which limits the transferability of these prediction systems to real-world hospital environments. Similar to our work, the authors in [125] examined two data cohorts: one with patients who had both physiological parameters and clinical notes, and another where some patients were missing clinical notes. For handling the missing modality, they introduced a modality-specific token [125] in the spirit of the classify [CLS] token used in BERT language models. Our study also considers two data cohorts: the first combines physiological parameters with clinical notes, while the second builds on the first by adding chest X-rays for patients who have them. However, imputation for missing modality information at the time level is performed with a zero-vector ($\mathbf{0_v}$). This approach indicates to the model that there is no information available for this modality at that time point, which

more accurately reflects the absence of data compared to imputation with a non-zero value, which would incorrectly suggest that data was indeed recorded.

The fusion of multimodal data has received increasing attention in the literature, with researchers exploring the most effective methods for combining different data sources. In the context of predicting LoS, many studies have combined modality-based information through concatenation or addition, employing various fusion mechanisms [31, 52, 123, 128]. In the intermediate fusion mechanism (referred to as such in [128] and as early fusion in [125]), modality-based representations generated by separate and independent model encoders are fused and subsequently learned by a predictor network. In the late fusion approach (also termed as such in [128]), each modality is used independently for predictions, and the final prediction is obtained by averaging or applying another aggregate measure. However, previous studies have suggested that simple aggregation of data modalities is not always effective and may fail to capture the key information and correlations across different data types [129]. To address this, an iterative attentional feature fusion method was introduced to improve the extraction of relevant features [129]. Additionally, when examining the role of clinical text in predicting clinical outcomes, [130] demonstrated that performance gains, particularly for outcomes like readmission, could be achieved by carefully selecting important sections of the text data rather than using the entire text. In [125], the authors proposed a Transformer-based architecture combined with contrastive learning to better capture meaningful correlations between patient modalities for binary LoS classification at 3-day and 7-day thresholds.

In this study, we predict the remaining ICU length of stay at each hour between admission and discharge or death using the MIMIC-IV, MIMIC-IV notes, and MIMIC-IV CXR datasets [60, 61]. Since each data modality is recorded over time, we address missing data modalities at two levels. The first level involves the time dimension, where a patient may have all three data types, but they are recorded at irregular intervals (e.g., the first note or chest X-ray may be recorded only 10 hours after ICU admission). The second level concerns the integration of chest X-rays, which are included only for patients who have them (representing less than 25% of the cohort).

5.2 Key Contributions

The key contributions of this work can be summarized as follows:

- We predict continuous remaining ICU LoS between admission and discharge using different data modalities: static demographic and admission

information, time-varying numerical physiological measurements, time-varying textual information and time-varying imaging information. The first two modalities are considered the base modality.

- We address missing modalities at two levels: (i) at the time level, where a patient may lack a modality-based recording at a specific time point and (ii) by using an unpaired dataset in which less than 25% of the patients have the image modality data. Additionally, unlike the authors in [128, 31], we do not balance our dataset by excluding patients with a LoS < 3 days.
- Finally, we demonstrate that employing cross-attention for modality fusion allows the models to leverage important cross-correlations between modalities. This approach results in a reduction in mean absolute error (MAE) from 2.97 days to 1.86 days with the bi-modal attention model of the base modality and the clinical notes. The tri-modal attention model that incorporates the chest x-rays shows a slight error reduction for patients with a LoS between 4 and 14 days.

5.3 Methodology

5.3.1 Dataset

The Medical Information Mart for Intensive Care (MIMIC-IV) [60, 61] is a publicly available uni-center dataset of patients admitted to the Beth Israel Deaconess Medical Center in the USA between 2008 and 2019. The MIMIC-IV database contains different modalities of patient data among which textual information (MIMIC-IV Note), images (MIMIC-IV CXR), highly sampled ECG waveforms (MIMIC-IV ECG), hospital and ICU level patient information such as labs, vital signs, electronic medication and more. Three main modalities of data were used in this work: (i) T: time series information from laboratory results and vital signs coupled with static demographic and admission information, (ii) N: free-text clinical notes, and (iii) I: thoracic scans.

Starting with the first modality (T) as the baseline, we extracted two data cohorts. The first cohort includes ICU patients with the first two modalities and is denoted as *CohortTN*. The second cohort is an extension of the first, incorporating images for patients who have these available and is denoted as *CohortTN_i*. This second cohort that includes the exact patients in *CohortTN* thus allows for missing radiography images. The data characteristics of these two cohorts are summarized in Table 5.1 below.

Table 5.1: Descriptive statistics of data cohorts. Percentage of stays with the image modality in *CohortTN_i* are shown in parenthesis.

	CohortTN	*CohortTN_i*
No. ICU stays		
Train	46490	46490(23.49%)
Test	9790	9790(22.50%)
Validation	10060	10060(23.90%)
Basic demographic statistics		
Gender (% Male)	56.07	56.07
In-hospital mortality (%)	11.68	11.68
LoS statistics in days		
mean	3.69	3.69
median	2.06	2.06
std	5.09	5.09
skewness	5.04	5.04
Statistics on Notes (train)		
total No. documents	43753366	43753366
min word count	3	3
max word count	3765	3765
average document length (words)	216	216
Statistics on Images (train - for 23.49% with images)		
total No. images	/	37605
min No. images per patient	/	1
max No. images per patient	/	36
mean No. images per patient	/	4

5.3.2 Preprocessing

Using the data extraction pipeline proposed in [50], we extracted 15 demographic and admission features along with 99 time-varying laboratory measurements and vital signs from the original database, as detailed in Table 5.2. This raw data was resampled hourly using the mean and subsequently normalized using the 5^{th} and 95^{th} percentiles. Missing information was imputed by forward filling values per patient and subsequently with zero (that is, imputation with the median value). To aid the models to deal with missing information, the hourly sampled dataset was augmented with a matrix of identical size containing decay curves that indicate the last hour from which data was missing [50, 56, 53]. After one-hot encoding of all extracted static features, a final dataset of 196 features was obtained.

The original MIMIC-CXR and MIMIC-Notes datasets are indexed by the patient admission identifier and not the ICU admission identifier. Using the time of admission into ICU and the charted times of the notes and chest X-rays, we mapped both data modalities to the corresponding ICU stay. These two modalities were also aggregated hourly. All images collected within the same hour were kept for modeling. All text reports collected within the same hour of hospitalization were combined. Imputation was not carried out on these two data modalities to ensure a regular time interval as in the physiological parameters dataset. However, a time-aware vector was added in both modalities final datasets, to indicate the lapse of time between two consecutive recordings.

The width and the height of all images were resized to 224 and standardized using a mean = 0.5 and standard deviation = 0.229 following the densenet121 architecture preprocessing.

Tokenization (that is, the process to convert the textual data to numerical values) was carried out using the trained tokenizer from the Clinical-T5-Sci model [131] (this model will be introduced in Section 5.3.4). This tokenizer uses the SentencePiece model [132] which forms its vocabulary by iteratively extracting meaningful subwords [131].

Table 5.2: Input features extracted from MIMIC-IV database.

Demographic and admission information		
admission location	age	ethnicity
eyes	first care unit	gender
height	hour	insurance
language	marital status	motor
verbal	weight	

Laboratory parameters			
Alanine Aminotransferase (ALT)	Alkaline Phosphatase	Anion Gap	Asparate Aminotransferase (AST)
Base Excess	Bicarbonate	Bilirubin, Total	Calcium, Total
Calculated Total CO2	Chloride	Creatinine	Free Calcium
Glucose	H	Hematocrit	Hematocrit, Calculated
Hemoglobin	I	INR(PT)	L
Lactate	MCH	MCHC	MCV
Magnesium	Oxygen Saturation	PT	PTT
Phosphate	Platelet Count	Potassium	Potassium, Whole Blood
RDW	RDW-SD	Red Blood Cells	Sodium
Sodium, Whole Blood	Temperature	Urea Nitrogen	White Blood Cells
pCO2	pH	pO2	

Vital signs features		
Activity / Mobility (JH-HLM)	Apnea Interval	Arterial Blood Pressure Alarm - High
Arterial Blood Pressure Alarm - Low	Arterial Blood Pressure diastolic	Arterial Blood Pressure mean
Arterial Blood Pressure systolic	Current Dyspnea Assessment	Daily Weight
Fspn High	GCS - Eye Opening	GCS - Motor Response
GCS - Verbal Response	Glucose finger stick (range 70-100)	Heart Rate
Heart Rate Alarm - Low	Heart rate Alarm - High	Hemoglobin
Inspired O2 Fraction	Magnesium	Mean Airway Pressure
Minute Volume	Minute Volume Alarm - High	Minute Volume Alarm - Low
Non Invasive Blood Pressure diastolic	Non Invasive Blood Pressure mean	Non Invasive Blood Pressure systolic
Non-Invasive Blood Pressure Alarm - High	Non-Invasive Blood Pressure Alarm - Low	O2 Flow
O2 Saturation Pulseoxymetry Alarm - Low	O2 saturation pulseoxymetry	PEEP set
PSV Level	Pain Level	Pain Level Response
Paw High	Peak Insp. Pressure	Phosphorous
Plateau Pressure	Platelet Count	Resp Alarm - High
Resp Alarm - Low	Respiratory Rate	Respiratory Rate (Set)
Respiratory Rate (Total)	Respiratory Rate (spontaneous)	Richmond-RAS Scale
Strength L Arm	Strength L Leg	Strength R Arm
Strength R Leg	Temperature Fahrenheit	Tidal Volume (observed)
Tidal Volume (set)	Vti High	

5.3.3 Prediction Task

We predict remaining ICU LoS between admission and discharge or death by assigning a unique value to each hour of the stay using data collected up until 14 days (335) hours post ICU admission. At each hour, this unique value is obtained by subtracting this hour from the observed LoS.

Starting with the first five hours of observed data, remaining LoS predictions at the sixth hour are computed and so on. A moving window of size 5, shifted by one hour is used to construct the input model sequences.

5.3.4 Models

This section describes the models used both in the unimodal setup as well in the multimodal setups. The overall model is constructed sequentially by using blocks of encoders. Predictions were performed using a series of feed-forward networks which we term as the predictor network.

Time Series Encoder

The time series encoder is a block of codes used to learn the sequential information within the data. This encoder consists of stacked layers of Temporal Convolutional Networks (TCN) explained in Chapter 2. This encoder was used to learn the sequential patterns in the unimodal time series physiological parameters data modality consisting of laboratory results, vital signs, and repeated static information. The static information which is only collected at admission is repeated at each hour.

Text Encoder

The text encoder consists of a text feature extractor model and a time series model. The former is used to learn the semantic information contained in the clinical reports and the latter, to learn the evolution over time of these reports.

Clinical-T5-Sci, initialized from the SciFive model was trained for additional epochs for MLM. In order to fine-tune this LLM for our specific task, remaining ICU LoS, we adopted the cost efficient meta-finetuning technique [124] in which the model is first fine-tuned only on a single note per patient (only the first note collected within the first 24 hours of admission from the training set) for total remaining LoS prediction. Subsequently, the fine-tuned model was used to

obtain model embeddings after freezing all intermediate layers. In this way, we increase the knowledge bag of the model by providing it with information helpful for predicting LoS. Table 5.3 shows the prediction errors obtained on the notes with and without meta-finetuning. During meta-finetuning and embeddings

Table 5.3: Results of meta-finetuning the Clinical-T5-Sci model for total remaining LoS prediction only with the first note recorded within 24 hours of ICU admission. All intermediate model layers are kept frozen when no finetuning is performed.

Model	MSE	MSLE
Meta-finetuning	25.71	0.38
All layers frozen	27.65	0.43

generation, only the encoder part of the Clinical-T5-Sci model was used. The classify token [CLS] was extracted to flatten the 3-dimensional output of the LLM encoder model into a 2-dimensional representation. This representation obtained for each patient and at each hour (where the note is available) was then used to form input sequences onto which the time dimension was learned using stacked TCN layers (the same time series encoder was used to ensure a fair comparison of the different model architectures). The resulting embeddings from the last TCN layer were then averaged and fused with the other modalities or sent to the predictor network for prediction depending on the fusion mechanism.

Image Encoder

Similarly to the text encoder, the image encoder is made up of an image feature extractor and stacked TCN layers to learn the time dimension in the extracted image features.

The image feature extractor also leverages knowledge from a pre-trained model to generate the embeddings from chest radiography images. We use the pre-trained model built using the DenseNet121 architecture [133] and pre-trained on seven X-rays datasets [134]. These datasets are: the NIH chest X-ray 14 [135], the PadChest dataset [136], the CheXpert dataset [137], MIMIC-IV CXR [61], OpenAI [138], Google [139], and the RSNA Pneumonia dataset. Given that the MIMIC-IV CXR dataset was used for training this model for predicting various lung diseases, which we believe are correlated to the LoS, we did not fine-tune the model but merely used it for feature extraction. Maximum pooling was used to flatten the 3-dimensional resulting embeddings into 2 dimensions.

Predictor Network

The predictor network used for generating LoS predictions ($\hat{\boldsymbol{y}}_t$) is composed of two feedforward neural networks as defined in Equation 2.3, with a Tanh activation and a dropout layer in between, and concludes with a ReLU activation.

$$\hat{\boldsymbol{h}}_t = \text{Dropout}\left(Tanh\left(\boldsymbol{\theta}^t \boldsymbol{h}_t + \boldsymbol{b}_t\right)\right)$$

$$\hat{\boldsymbol{y}}_t = ReLU\left(\hat{\boldsymbol{\theta}}^t \hat{\boldsymbol{h}}_t + \hat{\boldsymbol{b}}_t\right),$$

where $\boldsymbol{\theta}$ and $\hat{\boldsymbol{\theta}}$ are weight parameters, $\boldsymbol{b}_t$ and $\hat{\boldsymbol{b}}_t$ are biased terms and $\boldsymbol{h}_t, \hat{\boldsymbol{h}}_t$ are resulting model embeddings. The Dropout mechanism is a regularization technique used in deep learning architectures to prevent overfitting (excessive learning) where some model neurons are set to zero following some random distribution [140].

Fusion Mechanisms

Different fusion approaches were used to combine the information from each data modality in order to predict the remaining LoS.

- **Intermediate Fusion Model (IFM)**: in this approach, each data modality is learned with a separate encoder. The resulting embeddings are then fused together by concatenation (IFM-concat) or summing (IFM-sum as in [128]) and passed through the predictor network. Because modalities can be missing either at the patient level (for the *CohortTN_i*) or at the hour level (when a note or an image is not recorded until a certain hour for a patient in both cohorts), modality fusion was only done at the hour at which information became available. For example, suppose a patient did not have a note or an image recorded before hour 10, the $\mathbf{0_v}$ was used at each missing hour (from hour 1 to hour 9). Subsequently, embeddings from hour 10 are filled-forward up until the next hour where a new note appears. Using the $\mathbf{0_v}$ at these time points is somehow close to the dropout regularization method (but not the same of course) where we want to tell the model there is no information about this modality at this time point. Again the idea here is to mimic a real-hospital situation where patient information arrive at different time points and these are passed into a "prediction system" to predict LoS using the available information. The IFM architecture is shown in Figure 5.1.
- **Late Fusion Model (LFM)**: for late fusion, each modality is learned separately from feature extraction with a dedicated encoder to LoS

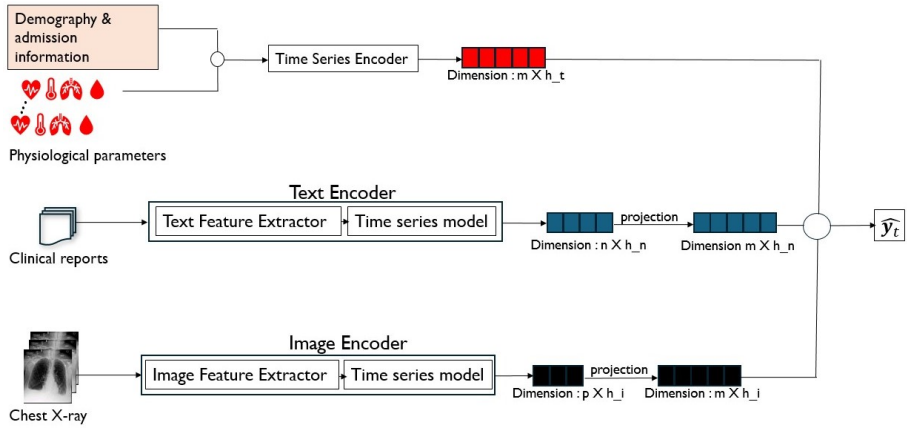


Figure 5.1: Intermediate Fusion showing the concatenation (represented with a circle) of modality embeddings extracted from the different encoders. Embeddings from the clinical reports and the images are projected using the $\mathbf{0_v}$to the dimension of the base modality before aggregation by concatentation or the sum.

prediction using a dedicated predictor network. The final prediction is then obtained by averaging the predictions from all modalities. Because the physiological parameters (laboratory measurements and vital signs) is the base modality, the predictions from this are used whenever an additional modality is missing as shown in Figure 5.2. Translating this to a clinic means that predictions using this fusion approach are dependent on the available modalities at the prediction time.

- **Intermediate Fusion Model with Attention (IFM-att)**: this fusion approach is very similar to the IFM but instead of merely concatenating the resulting embeddings from each separate modality, a cross-attention [74] is computed between the different modalities. The attention outputs calculated as given in Equation (5.1), are then added to the embedded physiological parameters data (labs, vital signs and static information, which constitute our base modality) before being passed through the predictor network.

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V, \tag{5.1}$$

where the query Q and the key K are the text and the base modalities respectively and the value V is the base modality for *CohortTN*. In the second cohort, *CohortTN_i*, two additional cross-attentions are computed

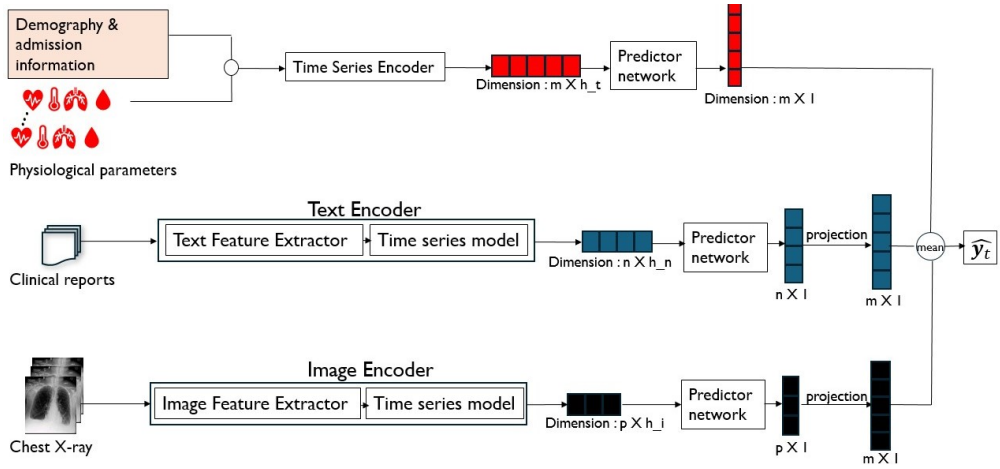


Figure 5.2: Late fusion model in which each modality independently predicts LoS, and the final prediction is obtained by averaging the ensemble of these predictions. The individual predictions are projected to the dimension of the base modality using the predictions of the latter.

between the three data modalities. The first cross-attention is applied between the image and text modalities, where K and V are the text and Q is the image modality. The second cross-attention is computed between the image modality and the base modality (the combination of physiological parameters and static information), with K and V as the base modality, and Q as the images. The purpose of using the image or text modalities as the query is to assess their relevance in relation to the base modality, which has already demonstrated strong performance for LoS predictions. Multiple cross-attentions are computed using Equation 5.1 and the results of these concatenated via the multi-head attention mechanism. The attention mechanism operates by focusing on the key information in the data. By applying a cross-attention between the different modalities, we help the model to focus on the important information that relates the different modalities. This architecture is shown in Figure 5.3.

5.3.5 Experimental Settings

The unimodal model, which consists of a TCN network trained on physiological parameters and static information, along with the multimodal IFM (IFM-concat & IFM-sum) and LFM models, were used as baseline models. The

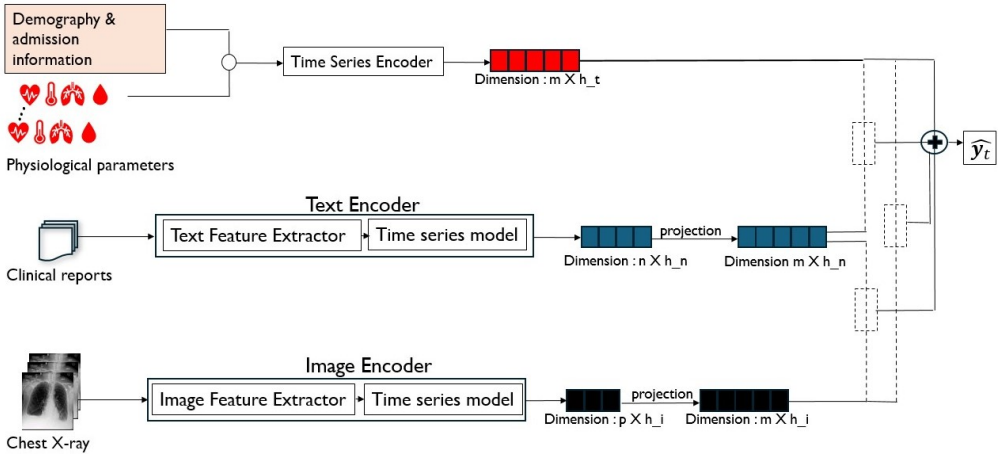


Figure 5.3: Intermediate Fusion with cross attentions (represented with dashed squares and dashed lines). The attention outputs are simply added to the representation of the base modality for LoS prediction ($\hat{\boldsymbol{y}}_t$).

performance of these models was then compared to the intermediate fusion model with cross-attention mechanisms. All models were trained using the AdamW optimizer [116] and the mean square log error as loss function. All models were implemented with Pytorch and on a single GPU. Computational resources were made available by the Vlaams Supercomputer Centrum.

5.3.6 Evaluation Metrics

The model performance was reported usingthe Mean Absolute Error (MAE), the Mean Squared Log Error (MSLE), and the Mean Squared Error (MSE) defined as follows:

$$MAE = \frac{1}{n}\sum_{t=1}^{n} |\boldsymbol{y}_t - \hat{\boldsymbol{y}}_t|$$

$$MSLE = \frac{1}{n}\sum_{t=1}^{n} (\log(\boldsymbol{y}_t + 1) - \log(\hat{\boldsymbol{y}}_t + 1))^2$$

$$MSE = \frac{1}{n}\sum_{t=1}^{n} (\boldsymbol{y}_t - \hat{\boldsymbol{y}}_t)^2$$

5.4 Results & Discussion

This section presents and discusses all results obtained on the baselines and the multimodal cross-attention models.

5.4.1 Hyperparameter Optimization

Hyperparameter optimization was carried out using the random search method by sampling 20 different values from the specified search space. The results of the hyperparameter optimization are presented in Appendix C.

5.4.2 Model Performance

The model performance for both the unimodal and multimodal models is presented in Table 5.4. IFM with attention (IFM-att) is only reported for *CohortTN_i* given that it returned the best performance with the first cohort. The results in Table 5.4 show that simply aggregating (by concatenation or

Table 5.4: Model performance on the test set of the *CohortTN* and the *CohortTN_i*. The best performance is shown in bold. Mean ± standard deviation calculated based on 3 iterations for each model.

Model	MAE	MSE	MSLE
Unimodal			
	2.97 ±0.01	37.00 ±0.51	0.40 ±0.00
Multimodal(*CohortTN*)			
IFM-concat	2.97 ±0.01	36.84 ±0.57	0.39 ±0.00
IFM-sum	2.97±0.02	37.65±0.66	0.39 ± 0.00
LFM	3.00 ±0.01	38.10 ±0.36	0.40 ±0.00
IFM-att	**1.86** ±0.04	**22.47** ±0.55	**0.17**±0.01
Multimodal(*CohortTN_i*)			
IFM-att	2.02 ± 0.10	23.40 ± 1.31	0.19 ±0.01

summation) the embeddings learned from clinical notes with those learned from the base modality using the intermediate fusion method yields comparable results to the unimodal model. Furthermore, the late fusion model, which combines modality-based predictions by summing them, indicates that clinical reports alone are not robust predictors of LoS, as highlighted in the literature [125]. However, incorporating cross-correlations between the base modality and

clinical text using a 4-multihead attention mechanism significantly improves predictions on the *CohortTN*, enhancing performance across all metrics. The mean absolute error (MAE) improves by more than one day (1.11 days). When cross-attention between chest X-rays and physiological parameters, as well as chest X-rays and clinical reports, is integrated for the *CohortTN_i*, the performance decreases compared to the bi-modal cohort, but it still improves prediction accuracy by 95% compared to the unimodal model. The reduced performance from the bi-modal attention to tri-modal attention models may be attributed to the small batch size of 8 used and the fact that the image modality is present in less than 25% of the patients.

We performed another analysis on the cross-attention fusion model, where, instead of only adding the attention outputs, we also include the text embeddings through summation. This is denoted as IFM-sum att. The results are shown in Table 5.5.

Table 5.5: Model performance on the test set of *CohortTN*. IFM-sum: this is the sum of the embeddings from the base modality and the text modality. IFM-sum att: an extension of IFM-sum where attention outputs from the two modalities are also added. IFM-att: only the attention outputs are added to the base modality. The best performance is shown in bold. Mean ± standard deviation calculated based on 3 iterations for each model.

Model	MAE	MSE	MSLE
Multimodal(*CohortTN*)			
IFM-sum	2.97±0.02	37.65±0.66	0.39 ± 0.00
IFM-sum-att	1.94±0.05	23.53 ± 1.22	0.18 ± 0.01
IFM-att	**1.86** ±0.04	**22.47** ±0.55	**0.17**±0.01

This approach resulted in lower performance compared to IFM-att though not statistically significant as measured by an independent t-test (we however note that our small sample size of 3 samples per group may not be adequate to detect mean differences). This outcome may suggest that the features extracted from the notes are noisier, and the cross-attention mechanism helps filter out this noise to provide more relevant features for prediction.

5.4.3 Model Performance per class of LoS

By categorizing the LoS into ten classes using various thresholds, we aim is to evaluate the performance of the bi-modal and tri-modal models across different patient groups: those with short, medium, and extended

stays. Following previous literature [52], the LoS is divided into ten categories: $[(0,1],(1,2],(2,3],\ldots,(6,7],(7,8],(8,14],(14,+\infty]]$. These bins represent patients who stay in the ICU for up to 1 day, 2 days, 3 days, and so on, up to 1 week, over 1 week, up to 2 weeks, and more than 2 weeks.

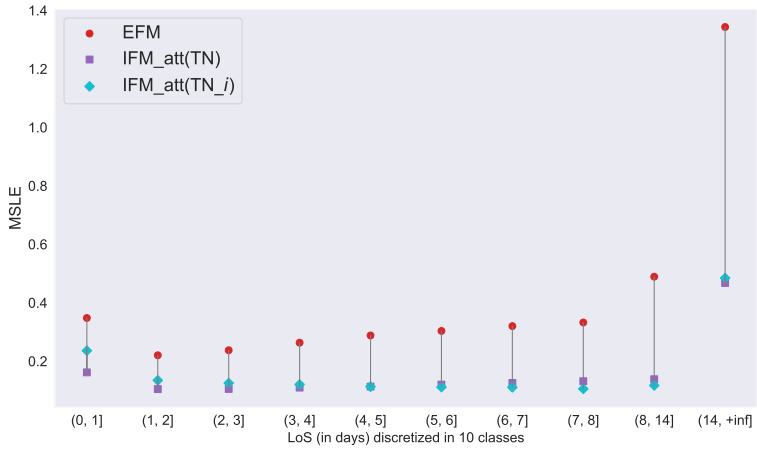


Figure 5.4: MSLE per category of LoS using the best models from EFM, IFM-att(TN) and IFM-att(TN_i). The vertical gray line shows the difference in the prediction error across the different models for each class. EFM here refers to the unimodal model.

Figure 5.4 illustrates that all models—the unimodal, the bi-modal with attention, and the tri-modal with attention—struggle with predicting LoS for patients at the extremities, specifically those with LoS ≤ 1 day and LoS > 8 days, where they have the highest error. However, despite the tri-modal model with attention slightly under performing compared to the bi-modal model with attention, it performs comparably or better for patients with LoS between 4 and 14 days.

5.5 Conclusion

This work assesses the contribution of different data modalities for continuous remaining ICU LoS prediction. By constructing a prediction system adapted to a real-hospital environment where patient information are charted at irregular time intervals and may or may not be charted for some patients (missing modality), we show that improvements in prediction performance can be achieved using cross-attention mechanisms that extract the important cross correlations within

the patients' data modalities. Upon adding the cross attentions between the base modality and the clinical reports, the mean absolute error (MAE) reduces from 2.97 days to 1.86 days. Though the image-based cross attentions do not further reduce the overall MAE, they show a little improvement or comparable performance as the bi-modal attention model on patients with a LoS between 4 and 14 days.

Chapter 6

Conclusion and Future Perspectives

This thesis primarily investigates the prediction of patients' Length of Stay under different settings. The research presented in the different chapters draws inspiration from our collaboration with the University Hospital of Leuven, where each project was designed with the intention of transferring the proposed methods to a real hospital environment. This chapter will begin by discussing the scientific contributions and limitations of our research in Section 6.1, followed by a cross-chapter discussion that highlights how methods from different chapters are applicable to others in Section 6.2. Next, in Section 6.3, it highlights the practical relevance of some aspects of our work in a clinic and finally in Section 6.4 it offers future perspectives for enhancing hospital management systems with a comprehensive decision support system.

6.1 Scientific conclusions and limitations per chapter

The recent advancements in AI have underscored its crucial role across diverse research domains, particularly in the medical field, leading to several groundbreaking developments. Nevertheless, a gap remains between AI's potential applications in healthcare and their practical implementation in clinical settings. Unlike much of the existing literature that primarily focuses on the former, this dissertation addresses the Length of Stay (LoS) prediction problem

from a pragmatic viewpoint—focusing on strategies for translating AI into practical, clinical use for LoS prediction.

Chapter 3

This chapter proposes a method to apply domain adaptation across heterogeneous feature spaces in order to take into account the singularity and specificity of patients in different target domains. In doing so, we obtained significant performance gains both in terms of prediction accuracy and computation time, especially in target domains with very few patients. Beyond applying the proposed method to different medical departments, domain definitions can also be based on distinct patient populations, such as pediatric versus geriatric, or on patients admitted for various cancer types.

Performing further investigations confirmed the benefits of domain adaptation as well as insights into the choice of a suitable source domain as a heterogeneous domain. However, it will be interesting to further investigate firstly, some of the instances where weight transfer did not work. Secondly, when performing discriminative learning to stabilize training between the two sets of coinciding and non-coinciding features across a target domain, a fixed factor of $\alpha = 10^{-1}$ was used to reduce the learning rate for coinciding features, whereas α can be treated as a hyperparameter and its value determined by the model performance. Lastly, though the proposed methodology wants to mimic a real hospital with multiple departments where data and a trained model from one department can boost performance in another, it was not tested in a true hospital setting in order to quantify the real benefits.

Chapter 4

In Chapter 4, our objectives were twofold: (i) to understand the learning dynamics of sequence-to-sequence models for hourly remaining Length of Stay (LoS) predictions by extracting and visualizing their latent spaces through an interactive dashboard, and (ii) to explore how these prediction models can be integrated into clinical decision support systems for hospital use. The strength of this work lies in its contributions to model transparency and explainability, extending beyond traditional error metrics to emphasize practical utility.

One of the notable limitations of this work is the prediction error that appears quite high but we think this can be attributed to the combination of the small dataset and the high variability in LoS (almost 10 days). An improvement could be the generation of synthetic patient time series data to augment the current

available data. Another possible improvement is the development of a new model architecture in a meta-learning setting such that different populations in the data are handled both separately and together. One of the issues we struggled with when visualizing the patient information in the interactive dashboard was finding or developing effective ways to visualize the high dimensional time series data over time.

Chapter 5

In chapter 5, we aimed at building a LoS prediction system that simultaneously handles or fuses different patient modalities while considering the time at which these were generated. We obtained improvements in prediction accuracy by fusing the different data modalities using cross-attentions between the different modalities to effectively extract the relevant information across them.

However, as it stands, this work still has a number of limitations. Firstly, our multimodal architectures were not trained end-to-end (since text and image embeddings were obtained separately) and therefore, it is very difficult to trace back from the predictions to the original raw features to compute feature importance. Secondly, the prediction accuracy could probably have been improved with an end-to-end model where both the text extractor and image extractor models are fine-tuned. Thirdly, more sophisticated methods could be designed to handle missing modalities. Finally, given that some of the textual information are radiology reports charted at the same time as the chest x-rays, an alignment of these modalities at these specific time points could be of interest.

6.2 Cross-chapter discussion

Data origin

Throughout the chapters, this dissertation employs various methodologies across three distinct datasets, representing patient populations primarily from the United States and the Netherlands. Working with these datasets highlighted that there may exists potential organizational factors and practices that influence Length of Stay. Where available, such factors could be incorporated into learning models to enhance LoS predictions.

In Chapter 3, titled Length of Stay Prediction for Hospital Management, we demonstrated that applying domain adaptation from one source domain

to a target domain with differing feature spaces can significantly improve both prediction accuracy and computation time. This suggests that, even when organizational factors vary between domains (such as different medical units or hospitals), the proposed domain adaptation approach, combined with discriminative learning, remains effective for transferring knowledge across domains.

Multimodal Fusion

In Chapter 5, titled Multimodal Learning through Time with Cross-Attentions for Length of Stay Prediction, different data modalities are fused by calculating and extracting correlations among them, resulting in improved prediction accuracy. The fusion architecture, which utilizes separate model encoders to merge patient information at specific charted times, makes this approach adaptable for the models in Chapters 3 and 4. Other data modalities can be integrated seamlessly to improve the prediction accuracy in these chapter. Given the relatively high error measures obtained in both previous chapters, fusing additional data modalities in this way can help reduce the prediction errors.

Moving window approach to input sequence construction

In Chapter 4, we gained insights into how sequential models gradually learn patient data by comparing the moving window approach to input sequence construction with the stacking approach. Based on this, Chapter 5 adopted the moving window approach, replacing the hour-by-hour stacking method that created longer training sequences. This method, which is also clinically relevant as practitioners typically review the most recent laboratory results for discharge decisions, allowed us to efficiently process multimodal fused data—with 768 and 1024 input features for text and image modalities, respectively—on a single NVIDIA P100 GPU. This overall design makes our prediction system efficient and well-suited for continuous use in practical clinical settings.

6.3 Practical Takeaways

Exploitation of available resources

The field of artificial intelligence has experienced a significant surge with the rise of Large Language Models (LLMs). Recently, researchers have begun leveraging these models in the clinical domain, demonstrating that clinical LLMs outperform traditional methods like Doc2Vec and Word2Vec in tasks such as summarizing, generating clinical text and more. Chapter 5 explores one such model, the Clinical T5-Sci, which is pre-trained on vast amounts of clinical text. This model can then be applied to a variety of downstream clinical applications. Utilizing resources like these accelerates research and facilitates real-time AI applications, as the models are fine-tuned rather than built from scratch, ensuring both efficiency and effectiveness.

Another tool utilized was the pre-trained DenseNet121 model, developed by [134], to extract embeddings from the image modality. Models like these, pre-trained on diverse datasets, eliminate the need to train from scratch—a process that can be time-consuming and often yields limited benefits.

Interactive visualization

Designing and utilizing a dashboard to explore the latent spaces of sequential models at each prediction hour provides a practical approach for leveraging these systems in capacity planning. This approach not only offers valuable insights, as discussed in Chapter 4, but can be extended to generate discharge prediction curves for individual patients. For example, a general downward trend in a patient's discharge prediction curve could prompt capacity planners to consult medical practitioners and confirm potential discharges, and organize patient flow accordingly.

Collaboration with experts

Our collaboration with capacity planners from UZ Leuven provided valuable insights that shaped the project design, ensuring it was tailored to the needs of end users. Additionally, it informed data pre-processing strategies; for instance, urgent night admissions are often not flagged as urgent (emergency department reception is closed), despite the mode of arrival or the unit of admission being important predictive features. This dissertation highlights the importance of

close collaboration with end users, particularly capacity planners, to enhance the utility and effectiveness of such prediction systems.

Efficiency

All the projects presented in this dissertation were executed on a single GPU, utilizing either an NVIDIA P100 with 16 GiB GDDR or a V100 with 32 GiB GDDR (Graphics Double Data Rate). which gives an incentive for applying these in real-time. This makes them highly practical for real-time applications, as the computational resources needed are accessible and scalable. Such efficiency underscores the feasibility of implementing these solutions in real-world scenarios, enabling broader adoption in clinical and operational settings.

6.4 Future perspectives

Predicting patients' LoS is primarily a managerial problem, as ethical considerations prevent the use of these models for deciding about patients' discharge. Thus, the prediction systems should be designed to aid hospital management, specifically in capacity planning. Despite the increasing number of research contributions in this area, we believe that further research directions can be pursued towards the development of prediction systems that comprehensively support hospital management systems.

- **Data**
  - Synthetic Data Generation: Accessing hospital data is often challenging due to the stringent precautions required when handling highly sensitive information, such as medical records. To comply with regulatory frameworks like the General Data Protection Regulation (GDPR), hospitals must anonymize such data to ensure that re-identification of patients is impossible. However, the cost of implementing these anonymization processes can be significant. An emerging alternative actively explored in the literature is data synthesis. In this approach, hospitals can share non-sensitive summary metrics (e.g., mean, skewness, frequency of measurements) about their data. These metrics are then used to generate synthetic datasets within a secure environment. The resulting synthetic data, which does not require anonymization, can then be shared and utilized to train multiple models for various tasks. However, a key challenge lies in synthesizing such data while preserving the intricate

correlations between different data modalities for each patient, as well as addressing the irregularity in the time dimension within each modality. For example, a patient's demographic data and thoracic scans may influence their physiological parameters, which in turn shape their generated clinical reports. Effective data synthesis must capture these relationships across modalities while also incorporating the temporal aspect, enabling the generation of realistic time series data. Moreover, the synthesized data should meet critical evaluation metrics of **resemblance**, ensuring a strong resemblance to the original data, **utility** for downstream tasks, and **privacy**, i.e., robust protection of patient as mentioned in [141].

- Federated learning: Given the hospitals' reluctance to share or centrally aggregate their data, models can instead be trained in a federated learning scheme. In this approach, each hospital's data remains on its own server, where individual models are trained locally. The knowledge from these local models—contained in the weight parameters and/or gradients—can subsequently be aggregated on a central server and strategically redistributed to each hospital's model. This process allows each local model to benefit from the collective knowledge without requiring all hospitals to centralize their data. Federated learning has shown competitive performances compared to central training in the literature and different strategies are continuously developed to improve learning in these scenarios. In one of our recent studies [142], we show that certain clients (hospitals) can be pre-excluded from the federation if they are not representative of the global patient population. By conducting federated learning with pre-excluded clients, we ensure a privacy guarantee for these clients, as their data remains completely inaccessible. Additionally, federated learning can be integrated with the domain adaptation approach proposed in Chapter III, allowing for variability in the input space across different hospitals. This approach would even enable the inclusion as input features of organizational factors, that may influence the LoS in different hospitals, for training silo-based models.
- Data accessibility initiatives: The establishment of the Belgian Health Data Agency (HDA) by law in March 2024 marks a significant step forward in facilitating access to hospital data in Belgium. The HDA, through its HDA Academy, offers data literacy courses targeting all stakeholders, including data generators (patients), data providers (hospitals), and data analysts. This initiative creates an ideal ecosystem to advance research involving the secondary use of healthcare data. The Belgian HDA serves as a crucial bridge between

data providers and data users. It simplifies data access by supporting anonymization processes while ensuring that data users uphold their responsibilities. Projects such as the LoS prediction, initiated at hospital levels and not unit-levels can then be encouraged by the Belgian HDA that recognizes the importance of secondary use of healthcare data for improving the health experience of Belgians.

  - Data availability at the hospital level: data is the fundamental component for developing prediction systems. Although the research community diligently works to make clinical information accessible, the available data remains insufficient. For instance, most publicly available datasets with LoS labels come from Intensive Care Units (ICUs). However, hospitals do not only have ICU units, but also, acute care wards that accommodate less critical patients and serve as the discharge destination for ICU patients. The public availability of ICU data has consequently led to the development of prediction models specifically adapted to the high density of information in ICU. Therefore, it is essential to encourage the release of data from acute care wards to facilitate the development of models suitable for broader hospital use.
  - Information on in-hospital movements: the prediction of the duration of bed occupancy can be supplemented with the prediction of patient movement within the hospital. Predicting which is the next service a patient will require and for how long, can provide a broader view and enable to anticipate on resource utilization at any time t.

- **Contextualization**: In this study, we used datasets from different continents, including the MIMIC-IV and eICU-CRD datasets from the United States and the Amsterdam University Medical Center dataset from Europe. Analyzing these diverse datasets confirmed existing literature that organizational factors influence LoS. The shorter stays observed in the American datasets compared to the longer stays in the European dataset suggest that additional facilities in the US help reduce ICU LoS. Therefore, hospitals and researchers can brainstorm on ways to quantify and incorporate these organizational factors such that they are exploited by prediction systems.

- **Algorithms**:

  - Graph representation learning: healthcare professionals rely on the abundant medical literature and past experience to make critical life-saving decisions daily. This approach can be mimicked using graph representation learning and modeling, where the past experience will then be coded by constructing graphs such that patients are connected

to each other based on specific similarity measures. This learning paradigm can offer more flexibility to incorporate domain knowledge such as when constructing the similarity matrix. In a recent paper by Li. et al 2022 [143], the authors believe that graph representation learning will transform the study of complex systems which makes their application suitable for LoS modeling as this attribute can be explained by multiple factors including their interactions.

Imputing missing information in healthcare datasets is also an important topic because missing information almost always conveys valuable information and sometimes, this missing information needs no imputation. Such an instance is imputing the blood exam for pregnancy, the Human chorionic gonadotropin (hCG) for male observations in the dataset. You. et al. [144] showed that graph learning could be exploited in such scenario where by constructing a bipartite graph of patients and attributes nodes, missing information can be imputed as edge predictions, thereby leveraging valuable information from their neighbors.

Finally, Walke. et al [145] in their recent paper argue that if graph representation learning and modeling of patient data is a success, clinical information can even be stored as graphical databases instead of relational databases which is still the common practice. Storing the data in graph databases will then facilitate their retrieval and use for downstream tasks as the storage will already contain some of the relationships that exits in the data.

- Meta-learning: the healthcare domain shows numerous singularities. Patients are very specific even when presenting similar symptoms or medical histories. The longitudinal nature of medical data introduces temporal effects, where outcomes may vary over time due to factors like the introduction of new techniques or technologies, leading to shifts in data distributions. Chapter III showed that such shifts can appear in different ICU populations. However, developing individualized models is often impractical due to limited sample sizes. Therefore, we believe that more efforts can be directed towards the construction of meta-learners that are robust to distribution shifts in the data.

- **Post-Analysis**: Because LoS is mainly relevant for hospital management, the aggregate metrics used now in the literature do not give all information about the discharge of a specific patient. Therefore, it might be necessary to find means to communicate the findings on a per patient case to enable capacity planning.

Appendix A

Appendix Chapter III

A.1 Patient features extracted per ICU unit

1. By using the pipeline by [50], SQL queries were used to extract data tables (*vitalperiodic*, *vitalaperiodic*, *respiratorycharting*, *lab*, *nursecharting* from eICU-CRD and *labevents*, *chartevents* from MIMIC-IV) by imposing thresholds of presence of each feature in at least 25% 13% or 12.5% of all adult patients.

2. Next, using the *unit type* column from *patient* table in eICU-CRD and *first_careunit* column from *icustays* table in MIMIC-IV, stay ids were collected for each ICU unit type.

3. Extracted tables in 1. were then filtered on stays ids from 2. of each ICU unit for both datasets.

4. Again by modifying the pipeline by [50], data curation and pre-processing was done per ICU unit type for both datasets one at a time.

5. In the pre-processing, all stays data tables for each unit, were merged, re-sampled hourly using the mean and scaled. From this, only features with at least 2 unique values for at least 30% of the patients were retained for modeling. For each dataset and for each ICU unit type, final extracted features are grouped under the table they appear in the original database.

6. Steps 2., 3., 4. and 5. were also performed on the initial tables extracted in 1. without filtering per *unit type* which constitutes the *all stays* data.

A.1.1 Data

The list of features extracted from the eICU-CRD and MIMIC-IV databases are given in Table A.1.1 and Table A.1.1 respectively.

Patients	Source Tables				
	vitalperiodic	***vitalaperiodic***	***respiratory***	***lab***	***nursecharting***
All Stays	cvp, heartrate, st1, st2, st3, respiration, temperature, sao2	noninvasivediastolic, noninvasivemean, noninvasivesystolic, systemicdiastolic, systemicmean, systemicsystolic		FiO2, bedside glucose	Glasgow coma score, Heart Rate, O2 L/%, O2 Saturation, Pain Score/Goal, Invasive BP, Non-Invasive BP, Respiratory Rate, Sedation Scale/Score/-Goal, Temperature
Med Surg ICU	cvp, heartrate, st1, st2, st3, respiration, temperature, sao2	noninvasivediastolic, noninvasivemean, noninvasivesystolic, systemicdiastolic, systemicmean, systemicsystolic		FiO2, bedside glucose	Glasgow coma score, Heart Rate, O2 L/%, O2 Saturation, Pain Score/Goal, Invasive BP, Non-Invasive BP, Respiratory Rate, Temperature
SICU	cvp, heartrate, st1, st2, st3, respiration, temperature, sao2	noninvasivediastolic, noninvasivemean, noninvasivesystolic, systemicdiastolic, systemicmean, systemicsystolic		FiO2, bedside glucose	Glasgow coma score, Heart Rate, O2 L/%, O2 Saturation, Pain Score/Goal, Invasive BP, Non-Invasive BP, Respiratory Rate, Sedation Scale/Score/-Goal, Temperature
NICU	heartrate, st1, st2, st3, respiration, sao2	noninvasivediastolic, noninvasivemean, noninvasivesystolic, systemicdiastolic, systemicmean, systemicsystolic	Total RR	FiO2, bedside glucose	Glasgow coma score, Heart Rate, O2 L/%, O2 Saturation, Pain Score/Goal, Invasive BP, Non-Invasive BP, Respiratory Rate, Sedation Scale/Score/-Goal, Temperature
MICU	heartrate, st1, st2, st3, respiration, sao2	noninvasivediastolic, noninvasivemean, noninvasivesystolic, systemicdiastolic, systemicmean, systemicsystolic		FiO2, bedside glucose	Glasgow coma score, Heart Rate, O2 L/%, O2 Saturation, Pain Score/Goal, Invasive BP, Non-Invasive BP, Respiratory Rate, Sedation Scale/Score/-Goal, Temperature

Continues on next page

Table A.1 – *Continued from previous page*

Patients	Source Tables				
	vitalperiodic	***vitalaperiodic***	***respiratory***	***lab***	***nursecharting***
CTICU	cvp, heartrate, st1, st2, st3, respiration, temperature, sao2	noninvasivediastolic, noninvasivemean, noninvasivesystolic, systemicdiastolic, systemicmean, systemicsystolic		FiO2, bedside glucose, HCO3, Hct, Hgb, pH, paCO2, paO2, potassium	Glasgow coma score, Heart Rate, O2 L/%, O2 Saturation, Pain Score/Goal, Invasive BP, Non-Invasive BP, Respiratory Rate, Temperature, Bedside Glucose
CICU	cvp, heartrate, st1, st2, st3, respiration, sao2	noninvasivediastolic, noninvasivemean, noninvasivesystolic, systemicdiastolic, systemicmean, systemicsystolic	PEEP	FiO2, bedside glucose	Glasgow coma score, Heart Rate, O2 L/%, O2 Saturation, Pain Score/Goal, Non-Invasive BP, Respiratory Rate, Sedation Scale/Score/Goal, Temperature
CCU-CTICU	cvp, heartrate, st1, st2, st3, respiration, temperature, sao2	noninvasivediastolic, noninvasivemean, noninvasivesystolic, systemicdiastolic, systemicmean, systemicsystolic		FiO2, bedside glucose, potassium	Glasgow coma score, Heart Rate, O2 L/%, O2 Saturation, Pain Score/Goal, Invasive BP, Non-Invasive BP, Respiratory Rate, Sedation Scale/Score/-Goal, Temperature
CSICU	cvp, heartrate, st1, st2, st3, respiration, sao2	noninvasivediastolic, noninvasivemean, noninvasivesystolic, systemicdiastolic, systemicmean, systemicsystolic	Total RR, PEEP, Peak Insp. Pressure, TV/kg IBW, Mean Airway Pressure, Tidal Volume (set), Vent Rate, Exhaled MV, LPM O2	FiO2, bedside glucose, potassium	Glasgow coma score, Heart Rate, O2 L/%, O2 Saturation, Pain Score/Goal, Invasive BP, Non-Invasive BP, Respiratory Rate, Temperature, Bedside Glucose

Patients	Source Tables	
	labevents	***chartevents***
All Stays	Base Excess, Calculated Total CO2, Glucose, Hematocrit, Hemoglobin, pCO2, pH, pO2	Activity / Mobility (JH-HLM), Arterial Blood Pressure diastolic, Arterial Blood Pressure mean, Arterial Blood Pressure systolic, GCS - Eye Opening, GCS - Motor Response, GCS - Verbal Response, Glucose finger stick (range 70-100), Heart Rate, Inspired O2 Fraction, Mean Airway Pressure, Minute Volume, Non Invasive Blood Pressure diastolic, Non Invasive Blood Pressure mean, Non Invasive Blood Pressure systolic, O2 saturation pulseoxymetry, PEEP set, Respiratory Rate, Richmond-RAS Scale, Strength L Arm, Strength L Leg, Strength R Arm, Strength R Leg, Temperature Fahrenheit, Tidal Volume (observed)
MICU	Glucose, Hematocrit, Hemoglobin, pH	Activity / Mobility (JH-HLM), Arterial Blood Pressure diastolic, Arterial Blood Pressure mean, Arterial Blood Pressure systolic, GCS - Eye Opening, GCS - Motor Response, GCS - Verbal Response, Glucose finger stick (range 70-100), Heart Rate, Inspired O2 Fraction, Mean Airway Pressure, Minute Volume, Non Invasive Blood Pressure diastolic, Non Invasive Blood Pressure mean, Non Invasive Blood Pressure systolic, O2 saturation pulseoxymetry, PEEP set, Respiratory Rate, Richmond-RAS Scale, Strength L Arm, Strength L Leg, Strength R Arm, Strength R Leg, Temperature Fahrenheit, Tidal Volume (observed)
CVICU	Base Excess, Calculated Total CO2, Free Calcium, Glucose, Hematocrit, Hematocrit, Calculated, Hemoglobin, Lactate, Potassium, Whole Blood, pCO2, pH, pO2	Activity / Mobility (JH-HLM), Arterial Blood Pressure diastolic, Arterial Blood Pressure mean, Arterial Blood Pressure systolic, GCS - Eye Opening, GCS - Motor Response, GCS - Verbal Response, Glucose finger stick (range 70-100), Heart Rate, Inspired O2 Fraction, Mean Airway Pressure, Minute Volume, Non Invasive Blood Pressure diastolic, Non Invasive Blood Pressure mean, Non Invasive Blood Pressure systolic, O2 saturation pulseoxymetry, PEEP set, Respiratory Rate, Richmond-RAS Scale, Strength L Arm, Strength L Leg, Strength R Arm, Strength R Leg, Temperature Fahrenheit, Tidal Volume (observed), Ventilator Mode

Continues on next page

Table A.2 – *Continued from previous page*

Patients	Source Tables ***labevents***	***chartevents***
Med-Surg ICU	Glucose, Hematocrit,	Activity / Mobility (JH-HLM), Arterial Blood Pressure diastolic, Arterial Blood Pressure mean, Arterial Blood Pressure systolic, GCS - Eye Opening, GCS - Motor Response, GCS - Verbal Response, Glucose finger stick (range 70-100), Heart Rate, Inspired O2 Fraction, Non Invasive Blood Pressure diastolic, Non Invasive Blood Pressure mean, Non Invasive Blood Pressure systolic, O2 saturation pulseoxymetry, Respiratory Rate, Richmond-RAS Scale, Strength L Arm, Strength L Leg, Strength R Arm, Strength R Leg, Temperature Fahrenheit
SICU	Glucose, Hematocrit, Hemoglobin	Activity / Mobility (JH-HLM), Arterial Blood Pressure diastolic, Arterial Blood Pressure mean, Arterial Blood Pressure systolic, GCS - Eye Opening, GCS - Motor Response, GCS - Verbal Response, Glucose finger stick (range 70-100), Heart Rate, Inspired O2 Fraction, Mean Airway Pressure, Minute Volume, Non Invasive Blood Pressure diastolic, Non Invasive Blood Pressure mean, Non Invasive Blood Pressure systolic, O2 saturation pulseoxymetry, PEEP set, Respiratory Rate, Richmond-RAS Scale, Strength L Arm, Strength L Leg, Strength R Arm, Strength R Leg, Temperature Fahrenheit, Tidal Volume (observed)
TSICU	Base Excess, Calculated Total CO2, Glucose, Hematocrit, Hemoglobin, pCO2, pH, pO2	Activity / Mobility (JH-HLM), Arterial Blood Pressure diastolic, Arterial Blood Pressure mean, Arterial Blood Pressure systolic, GCS - Eye Opening, GCS - Motor Response, GCS - Verbal Response, Glucose finger stick (range 70-100), Heart Rate, Inspired O2 Fraction, Mean Airway Pressure, Minute Volume, Non Invasive Blood Pressure diastolic, Non Invasive Blood Pressure mean, Non Invasive Blood Pressure systolic, O2 saturation pulseoxymetry, PEEP set, Respiratory Rate, Richmond-RAS Scale, Strength L Arm, Strength L Leg, Strength R Arm, Strength R Leg, Temperature Fahrenheit, Tidal Volume (observed), Ventilator Mode

Continues on next page

Table A.2 – *Continued from previous page*

Patients	Source Tables	
	labevents	***chartevents***
CCU	Glucose, Hematocrit, Hemoglobin	Activity / Mobility (JH-HLM), Arterial Blood Pressure diastolic, Arterial Blood Pressure mean, Arterial Blood Pressure systolic, GCS - Eye Opening, GCS - Motor Response, GCS - Verbal Response, Glucose finger stick (range 70-100), Heart Rate, Inspired O2 Fraction, Non Invasive Blood Pressure diastolic, Non Invasive Blood Pressure mean, Non Invasive Blood Pressure systolic, O2 saturation pulseoxymetry, Respiratory Rate, Richmond-RAS Scale, Strength L Arm, Strength L Leg, Strength R Arm, Strength R Leg, Temperature Fahrenheit
Neuro-SICU		H, I, L, Activity / Mobility (JH-HLM), Arterial Blood Pressure diastolic, Arterial Blood Pressure mean, Arterial Blood Pressure systolic, GCS - Eye Opening, GCS - Motor Response, GCS - Verbal Response, Glucose finger stick (range 70-100), Heart Rate, Inspired O2 Fraction, Mean Airway Pressure, Minute Volume, Non Invasive Blood Pressure diastolic, Non Invasive Blood Pressure mean, Non Invasive Blood Pressure systolic, O2 saturation pulseoxymetry, PEEP set, Respiratory Rate, Richmond-RAS Scale, Strength L Arm, Strength L Leg, Strength R Arm, Strength R Leg, Temperature Fahrenheit, Tidal Volume (observed)
NI		Activity / Mobility (JH-HLM), Arterial Blood Pressure diastolic, Arterial Blood Pressure mean, Arterial Blood Pressure systolic, GCS - Eye Opening, GCS - Motor Response, GCS - Verbal Response, Glucose finger stick (range 70-100), Heart Rate, Non Invasive Blood Pressure diastolic, Non Invasive Blood Pressure mean, Non Invasive Blood Pressure systolic, O2 saturation pulseoxymetry, Respiratory Rate, Richmond-RAS Scale, Strength L Arm, Strength L Leg, Strength R Arm, Strength R Leg, Temperature Fahrenheit
NS		Activity / Mobility (JH-HLM), Arterial Blood Pressure diastolic, Arterial Blood Pressure mean, Arterial Blood Pressure systolic, GCS - Eye Opening, GCS - Motor Response, GCS - Verbal Response, Glucose finger stick (range 70-100), Heart Rate, Non Invasive Blood Pressure diastolic, Non Invasive Blood Pressure mean, Non Invasive Blood Pressure systolic, O2 saturation pulseoxymetry, Respiratory Rate, Richmond-RAS Scale, Strength L Arm, Strength L Leg, Strength R Arm, Strength R Leg, Temperature Fahrenheit

A.2 Implementation details

All algorithms were written in Python language. All experiments were performed on a 128GB RAM computer with Intel(R) Core(TM) i9-10900X CPU processor and NVIDIA GeForce RTX 2080 Ti GPU.

A.2.1 Hyperparameter Search Methodology and Implementation Details

The hyperparameter search space for all ICU units is given in the table below. An asterisk * is indicated for those in which the space was enlarged for particular units.

Table A.3: Hyperparameters search space and corresponding scale. No. Hidden units was enlarged to the interval [4 - 512] for Neuro-Stepdown ICU patients in MIMIC-IV and the learning rate to $[1e^{-5} - 1e^{-2}]$ for Medical-Surgical ICU patients in MIMIC-IV

Hyperparameter	Search Space Interval	Scale
No.Hidden Layers	[1 - 2]	Linear
No.Hidden units*	[8 - 512]	$\log_2$
Learning rate*	$[1e^{-4} - 1e^{-2}]$	$\log_{10}$
Dropout	[0.1 - 0.5]	Linear
Batch Size	[4 - 512]	$\log_2$

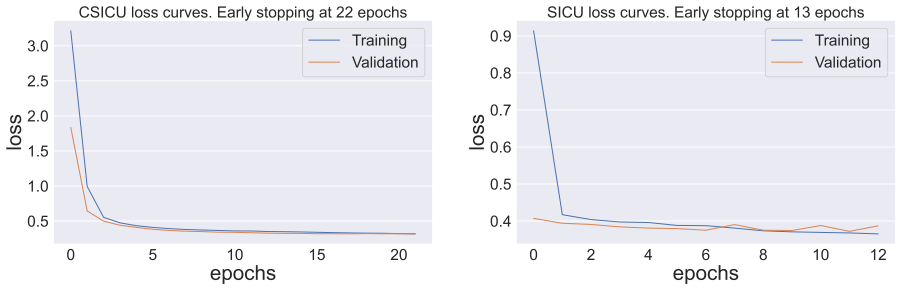
Hyperparameter search was performed in a systematic manner for each of the patient populations using the Bayesian Optimizer from KerasTuner [100, 101]. Essentially, Bayesian Optimization makes use of both a prior function and an acquisition function. The former is used as a surrogate to obtain estimates of the objective function and the latter, that measures the evaluation of the objective function at a new point and proposes next candidate points within the search space [146]. When using the Bayesian optimization, we minimize the validation loss, that is the, mean squared logarithmic error on the validation dataset. As explained before, the candidate hyperparameters values used at each trial are proposed depending on the performance of the previously chosen values. We noticed that orienting the search direction seriously affects the number of hidden units. Therefore, the hidden units space was split into two resulting in a three-step procedure as follows;

1. Firstly, the search space involving all hyperparameters except the batch size and No. hidden units in the interval [8 - 64] was used.
2. Secondly, the previous step was repeated with No. hidden units in the interval [64 - 512].
3. In the third step, the refined space following model performance was used and hyperparameter search was repeated.
4. The best hyperparameters from step three were then used to fit the model using early stopping to prevent overfitting. By monitoring the validation loss over six epochs, early stopping occurs if this doesn't further decrease by at least 0.5%.

In steps 1 and 2, ten trials were done with two executions per trial. In step 3, ten trials were done with three executions per trial. By performing multiple executions per trial ensures that the reported hyperparameter values return the lowest and most stable average error over all trials.

A.3 Loss Curves

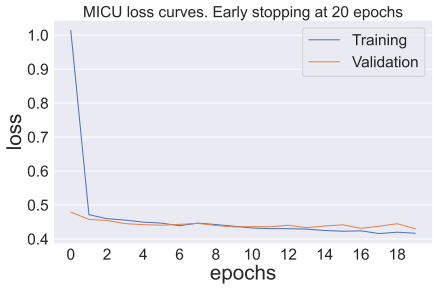
Additional loss curves from Section 4.4.1



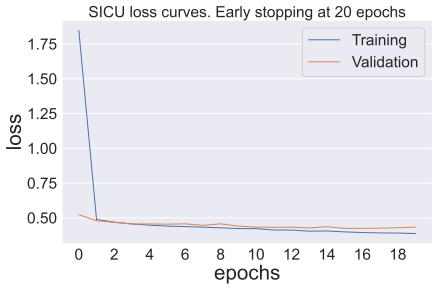
(a) CSICU patients

(b) SICU patients

Figure A.1: Loss curves obtained after training CSICU and SICU in eICU-CRD data with optimized hyperparameters



(a) MICU patients



(b) SICU patients

Figure A.2: Loss curves obtained after training MICU and SICU in MIMIC-IV data with optimized hyperparameters as in Section 4.4.1.

A.4 Multiple comparisons of Means using Tukey test

Complementary results for Figure 3.9 and 3.10.

Table A.4: SICU: Multiple comparison of Means for epochs and Test MAE using Tukey HSD [3], $\alpha = 5\%$. OM: Optimized Model on each sub-ICU unit, WTM: Weight Transfer Model, FTM: Full Transfer Model

	group 1	group 2	mean diff	p-adj	lower	upper	reject H_0
Epochs	OM	WTM	-3.6331	0.001	-4.5746	-2.6917	True
	FTM	OM	6.3301	0.001	5.3886	7.2716	True
	FTM	WTM	-2.697	0.001	1.7531	3.6408	True
Test MAE	OM	WTM	-0.0071	0.2555	-0.0177	0.0035	False
	FTM	OM	0.0095	0.0891	-0.0011	0.02	False
	FTM	WTM	0.0024	0.8418	-0.0082	0.013	False

Table A.5: CCU-CTICU: Multiple comparison of Means for epochs and Test MAE using Tukey HSD, $\alpha = 5\%$. OM: Optimized Model on each sub-ICU unit, WTM: Weight Transfer Model, FTM: Full Transfer Model

	group 1	group 2	mean diff	p-adj	lower	upper	reject H_0
Epochs	OM	WTM	-7.07	0.001	-7.7585	-6.3815	True
	FTM	OM	9.04	0.001	8.3515	9.7285	True
	FTM	WTM	1.97	0.001	1.2815	2.6585	True
Test MAE	OM	WTM	-0.0591	0.001	-0.0702	-0.048	True
	FTM	OM	0.0922	0.001	0.0812	0.1033	True
	FTM	WTM	0.0331	0.001	0.022	0.0442	True

Table A.6: CTICU: Multiple comparison of Means for epochs and Test MAE using Tukey HSD, $\alpha = 5\%$. OM: Optimized Model on each sub-ICU unit, WTM: Weight Transfer Model, FTM: Full Transfer Model

	group 1	group 2	mean diff	p-adj	lower	upper	reject H_0
Epochs	OM	WTM	-6.2298	0.001	-7.0824	-5.3772	True
	FTM	OM	7.8012	0.001	6.9487	8.6538	True
	FTM	WTM	1.5714	0.001	0.7146	2.42283	True
Test MAE	OM	WTM	-0.0829	0.001	-0.0992	-0.0666	True
	FTM	OM	0.1079	0.001	0.00916	0.1242	True
	FTM	WTM	0.0249	0.0011	0.0086	0.0413	True

Table A.7: CSICU: Multiple comparison of Means for epochs and Test MAE using Tukey HSD, $\alpha = 5\%$. OM: Optimized Model on each sub-ICU unit, WTM: Weight Transfer Model, Diff_LR: Different Learning Rates Model

	group 1	group 2	mean diff	p-adj	lower	upper	reject H_0
Epochs	Diff_LR	OM	7.7071	0.001	6.6032	8.811	True
	Diff_LR	WTM	-4.0707	0.001	-5.1746	-2.9668	True
	OM	WTM	-11.7778	0.001	-12.8817	-10.6739	True
Test MAE	Diff_LR	OM	0.0173	0.2146	-0.0069	0.0415	False
	Diff_LR	WTM	0.0214	0.0961	-0.0029	0.0456	True
	OM	WTM	0.0041	0.9	-0.0201	0.0283	False

Appendix B

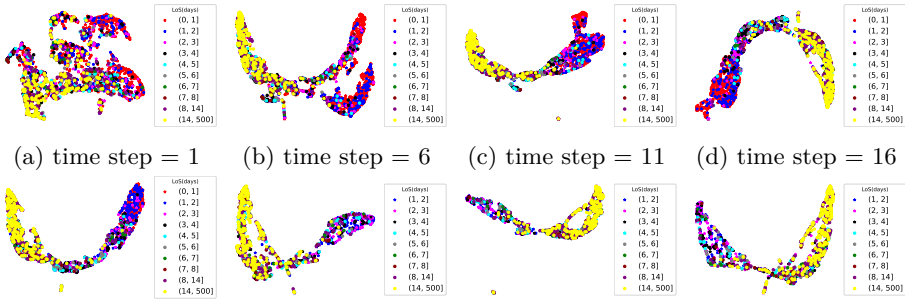
Appendix Chapter IV

B.1 Hyperparameters Optimization

For all four sequential deep learning models, the batch size hyperparameter refers to the number of unique stay ids randomly loaded at each epoch during training, validation and inference. When data for this unique number of stay ids is loaded, within the data loader, multiple sequences are then created (with an average number of sequences around 200) to predict remaining ICU LoS.

B.2 UMAP Visualization of the latent space

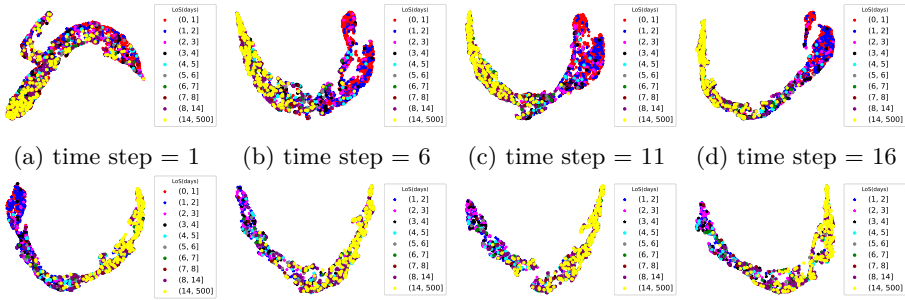
The UMPA visualization of all models' latent spaces are given in Figures B.1, B.2, B.3, B.4, and B.5.



(a) time step = 1 (b) time step = 6 (c) time step = 11 (d) time step = 16

(e) time step = 21 (f) time step = 26 (g) time step = 31 (h) time step = 36

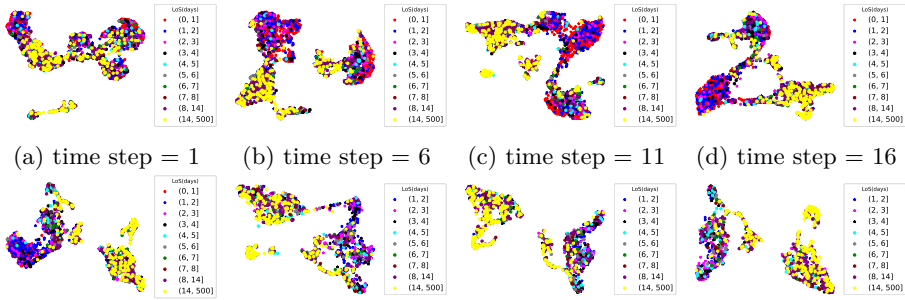
Figure B.1: UMAP 2D projections of LSTM model embeddings, labelled with LoS.



(a) time step = 1 (b) time step = 6 (c) time step = 11 (d) time step = 16

(e) time step = 21 (f) time step = 26 (g) time step = 31 (h) time step = 36

Figure B.2: UMAP 2D projections of GRU model embeddings, labelled with LoS.



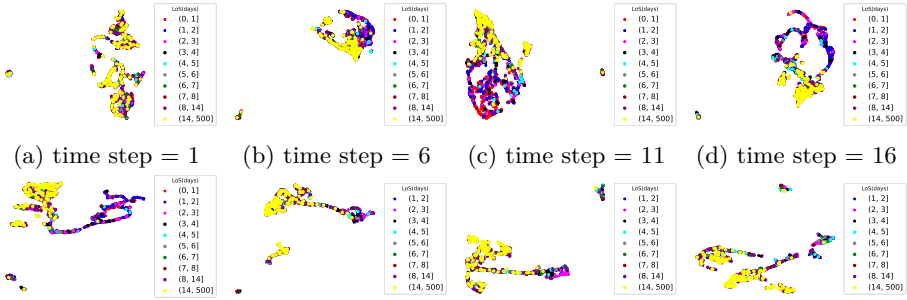
(a) time step = 1 (b) time step = 6 (c) time step = 11 (d) time step = 16

(e) time step = 21 (f) time step = 26 (g) time step = 31 (h) time step = 36

Figure B.3: UMAP 2D projections of Transformer model embeddings, labelled with LoS.

Table B.1: Hyperparamters values obtained upon optimization

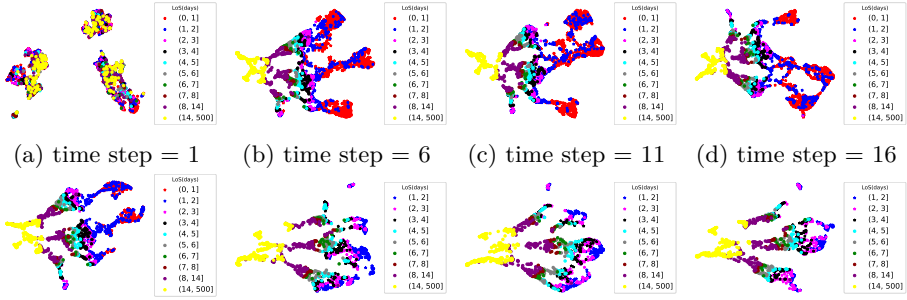
Models	Hyperparameters											
	No.layers	No.units	bsize	lr	dropout	wdecay	l	feed-forward dim	input embedding	No. heads(H)	No.channels	ksize
LR			128	$8.36e^{-4}$		$3.98e^{-4}$						
FFN	2	256	128	$4.28e^{-5}$	0.15	$5.48e^{-5}$						
LSTM	2	128	4*	$9.46e^{-4}$	0.25	$6.21e^{-3}$	4					
GRU	3	64	4*	$5.64e^{-4}$	0.15	$1.67e^{-3}$	3					
Transformer	2	32	4*	$2.94e^{-}4$	0.05	$8.12e^{-}3$	2	32	1D Conv	2		
TCN	8		4*	$3.02e^{-3}$	0.05	$5.06e^{-4}$	6				11	4
TCN-att	3		4*	$9.4e^{-5}$	0.05	$9.88e^{-4}$	5			4	64	4



(a) time step = 1 (b) time step = 6 (c) time step = 11 (d) time step = 16

(e) time step = 21 (f) time step = 26 (g) time step = 31 (h) time step = 36

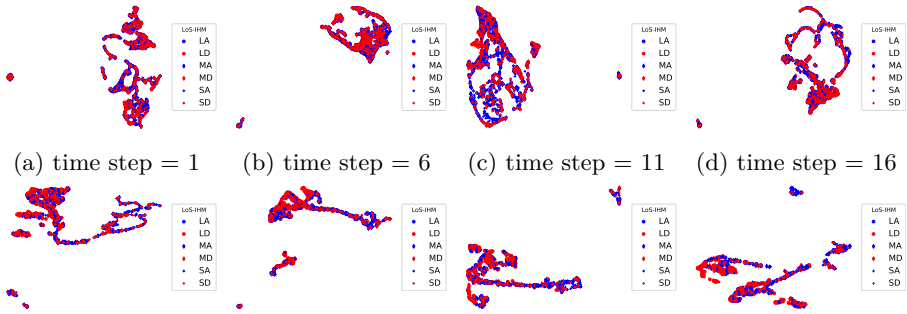
Figure B.4: UMAP 2D projections of TCN model embeddings, labelled with LoS.



(a) time step = 1 (b) time step = 6 (c) time step = 11 (d) time step = 16

(e) time step = 21 (f) time step = 26 (g) time step = 31 (h) time step = 36

Figure B.5: UMAP 2D projections of the TCN with attention model embeddings at different hours, labelled with LoS.



(a) time step = 1 (b) time step = 6 (c) time step = 11 (d) time step = 16

(e) time step = 21 (f) time step = 26 (g) time step = 31 (h) time step = 36

Figure B.6: UMAP 2D projections of TCN model embeddings, labelled with LoS and IHM.

Appendix C

Appendix Chapter V

C.1 Hyperparamter Optimization

The hyperparamters obtained from model optimization are listed in Tables C.2 - C.4 below. *flatten emb* is the method used to convert the 3D output of the time series encoder to a 2D output to be ingested by the predictor network (PN). This can either the mean of all embeddings (mean) or the last embeddings (last). *feed-forward size* is the feed-forward dimension used by the predictor network. The number of attention heads used for computing the cross-attention is denoted as h.

Table C.1: Hyperparamters values obtained upon optimization on the physiological parameters with the Early Fusion Model (EFM)

Hyperparameters	values
No. channels	11
No. layers	3
Dropout (TCN)	0.0
feed-forward size	64
kernel size	5
learning rate	$2.8e^{-4}$
Dropout (PN)	0.15
Batch size	8
Flatten emb	last

Table C.2: Hyperparamters values obtained upon optimization on the physiological parameters (T)/notes(N) with the Intermediate Fusion Model (IFM)

Hyperparameters	values (T/N)
No. channels	11/12
No. layers	3/3
Dropout (TCN)	0.0/0.15
feed-forward size	128
kernel size	5/3
learning rate	$2.8e^{-4}$
Dropout (PN)	0.15
Batch size	8
Flatten emb	last/mean

Table C.3: Hyperparamters values obtained upon optimization on the physiological parameters (T)/notes(N) with the Late Fusion Model (LFM)

Hyperparameters	values (T/N)
No. channels	11/12
No. layers	3/3
Dropout (TCN)	0.1/0.1
feed-forward size	64/64
kernel size	5/3
learning rate	$2.8e^{-4}$
Dropout (PN)	0.15/0.15
Batch size	8
Flatten emb	last/mean

Table C.4: Hyperparamters values obtained upon optimization on the physiological parameters (T)/notes(N)/Images(I) with the Intermediate Fusion Model (IFM) with cross-attentions

Hyperparameters	values (T/N)
No. channels	11/12/10
No. layers	3/3/3
Dropout (TCN)	0.1/0.15/0.1
feed-forward size	64
kernel size	5/3/3
learning rate	$2.8e^{-4}$
Dropout (PN)	0.15
Batch size	8
No. heads (h)	4/4/4
Flatten emb	last/mean/mean

Appendix D

Workdone on UZ Leuven data

D.1 Early binary LoS prediction using the MIR Dataset

D.1.1 Introduction

The first dataset provided by UZ Leuven consisted of retrospective data from 63,481 unique admissions, covering more than 30,000 individual patients. These patients were admitted to cardiology (432, 433, 435), vascular surgery, and cardiac surgery (436, 437) departments between 2011 and 2019. The distribution of their Length of Stay (LoS) is presented in Table D.1.

Table D.1: Distribution of LoS in fractional days

Statistics	LoS
count	63481
mean	7.416408
std	13.516181
min	0.145834
25%	1.491670
50%	2.981257
75%	8.315296
max	386.617533

The **research question** was to predict the Length of Stay (LoS) for short-stay patients admitted to UZ Leuven, regardless of factors such as admission unit, condition, or route. Fine-grained, continuous LoS prediction was performed in a regression format using the first 24 hours of observed data. Short-stay patients were then separated from long-stay patients for further analysis.

D.1.2 Data

The extracted data tables from the MIR data warehouse included:

1. Admission diagnoses and surgery: diagnoses and surgeries at admission though available after discharge.
2. Admission information (gender, language, type of admission, hoofDiagnose, hoofdZorgprogrammaNr, opname DRG, opname, SoI, opname RoM, DRG versie, hoofdIngreep, zorgpgrammaNr, zorgprogrammaOmschrijving, etc.)
3. Planned surgeries: surgeries planned with name and codes of surgeries.
4. Medical activity: test requested and performed on each patient
5. Presence information: presence of patients in each ward, supervisor attending, etc.
6. Care activity: information on all cares given to a patient.

D.1.3 Data pre-processing

1. Outcome was split into two binary outcome types: $[<= 14, > 14]$ days, $[<= 7, > 7]$ days and a three outcome multi-classification: $[<= 7, 7-14, > 14]$ with the following distributions;

Table D.2: LoS $\leq 14, > 14$

Classes	pr_LoS
0	55193
1	8288

Table D.3: LoS $\leq 7, > 7$

Classes	pr_LoS
0	43713
1	19768

Table D.4: $[<= 7, 7 - 14, > 14]$

Classes	pr_LoS
0	43713
1	11480
2	8288

2. All tables: Data recorded only up to 24h after admission was extracted and used.

3. Presence Info: Duplicates removed from presence info based on priority.

4. Medical activity: Drop rows with NANs on activity number and activity description. This removed 53211 lines

D.1.4 Feature engineering

Table	Extracted features	Methodology
Admission	hospital visit count	incremental value per readmission
	hour	hour of the day admission occurred
	period of the day	admission occurred in: $[7 - 22]$hr or not or between $[00 - 08, 08 - 16, 16 - 00]$
	weekday	day of the week on which patient was admitted
	month	month of the year admission occurred
	year	year admission occurred
	sin_hour, cos_hour	sinusoidal representations of hour of the day
	year_season	season of the year admission occurred
	nber_days_subs_adm	number of days since previous admission
	avg_prev_LoS	average previous los
	prev_los	previous los

continues on next page

Table D.5 – *Feature engineering*

Table	Extracted features	Methodology
	is_30_days_readm	is 30 days readmission
	is_90_days_readm	is 90 days readmission
	is_first_adm	is first admission
	opname_SoI_prev	previous discharge SoI
	opname_RoM_prev	previous discharge RoM
Presence Info	upgrade to ICU	upgrade from HOSPITALISATIE to ICU
	upgrade to ED	upgrade from HOSPITALISATIE to ED
	alt_pat_ind	indicator of alternative patients (patients not on the right ward)
	mod_triage_code	modified triage code to include: urgent day & night admit
	specific wards count	count of the number of times HOSPITALISATIE, OPERATIEKWATIER, HOSPITALISATIE ITE, DIALYSE, RADIOTHERAPIE were visited in 24h
	sup_count	count the number of supervisors that attended to the patient in 24h
Planned surgery	typeIngreep_counts	counts of number of planned surgeries
Medical activity	act_counts	counts of number of requested tests
	med_act_desc	concatenation of all descriptions of all activities performed
Discharge diagnoses	weighted_prevLoS DRG_SoI	number of deviations prev LoS is away from the median LoS of the group DRG_SoI combination
Care activity	pnr_count	1. Care counts (frequency of measurement of numeric care) extracted by imposing a threshold of presence in at least 35% of patients.

D.1.5 Topic Modeling

This process involves identifying and extracting latent topics that represent the underlying structure of a corpus (a collection of documents). Essentially, topic modeling aims to uncover the primary topics present in each document by analyzing a set of sentences (documents) composed of groups of words.

In our work, topic modeling was applied to describe medical activities. Here, the description of a single patient serves as a document comprising the words used. Given a corpus C of dimension $N \times D$ we aim to decompose this two matrices:

$$C = T \times W$$

where T is a document-by-topic matrix with each document (admission) represented by a sequence of topics and W, is a topic-by-word matrix in which each topic constitutes a series of words. Three matrix decomposition algorithms were used:

1. Latent Semantic Analysis (LSA) via Truncated Singular Value Decomposition (TSVD).
2. Latent Dirichlet Analysis (LDA)
3. Non-Negative Matrix Factorization (NMF)

with the best results obtained from NMF.

D.1.6 Data Imputation

Imputation was performed by filling missing values with known values. Feature type is N (numerical) or C (categorical).

Features	Type	Imputation value	Used in the model
nber_days_subs_adm	N	0	F
avg_prev_LoS	N	0	F
prev_LoS	N	0	F
taal	C	other	T
Is_30_days_readm	N	0	F
Is_90_days_readm	N	0	F
is_first_adm	N	0	F
opname_SoI_prev	C	No_SoI_available	F
opname_RoM_prev	C	No_RoM_available	F
typeIngreep_count	N	0	T
avg_prev_LoS	N	0	F
pnr_count	N	0	T
laasteTriageCode	C	0	T
isolatieType	C	not_isolated	T
omschrijving	text	geen_operatie_gepland	F

continues on next page

Table D.6 – *Feature Imputation*

Features	Type	Imputation value	Used in the model
activiteitOmschrijving	text	geen_activiteit	T

D.1.7 Performance metrics

Content here follows from [147].

1. Recall or Sensitivity: $\frac{TP}{TP+FN}$
   Measures the ability of a classifier to identify positive labels within the dataset.
2. Precision: $\frac{TP}{TP+FP}$
   Measures the number of positive predictions that actually belong to the class.
3. F-score: $\frac{2 \times precision \times recall}{precision+recall}$
   Gives a balance between recall and precision.

D.1.8 Data Modelling

The data was divided into training and testing sets, and model training was conducted using the XGBoost (Extreme Gradient Boosting) algorithm. Hyperparameter optimization was performed using random search cross-validation with 5 folds.

- **14 days binary classification**

Table D.7: shape of ytrain:(44437,) shape of ytest: (19044,)

	precision	recall	f1-score	support
short stay	0.90	0.97	0.94	16597
extreme stay	0.59	0.31	0.40	2449
accuracy			0.88	19044
macro avg	0.75	0.64	0.67	19044
weighted avg	0.86	0.88	0.87	19044

- **7 days binary classification**

Table D.8: 7 days binary classification using XGBoost.

	precision	recall	f1-score	support
short stay	0.89	0.86	0.88	12988
extreme stay	0.72	0.77	0.75	6056
accuracy			0.83	190444
macro avg	0.81	0.82	0.81	19044
weighted avg	0.84	0.83	0.83	19044

- **3 class multiclassification**

Table D.9: Three classes multi-classification using XGBoost.

	precision	recall	f1-score	support
short stay	0.86	0.91	0.88	12988
medium stay	0.52	0.44	0.48	3700
long stay	0.45	0.42	0.43	2356
accuracy			0.76	19044
macro avg	0.61	0.59	0.60	19044
weighted avg	0.74	0.76	0.75	19044

D.1.9 Model explanaibility

The feature importance plots highlight that variables such as *typeingreep_count* (representing the number of surgeries or interventions within 24 hours) and patient age play a significant role in LoS classification. Additionally, topics derived from Non-Negative Matrix Factorization (NMF) emerged as important, although these topics were not mapped back to their original medical activity descriptions. Finally, the frequency of administered care activities (e.g., *MZemotonderst*, *VPzuurstofsat*, *PHneuszorg*, etc.) proved to be highly influential in the early classification of LoS.

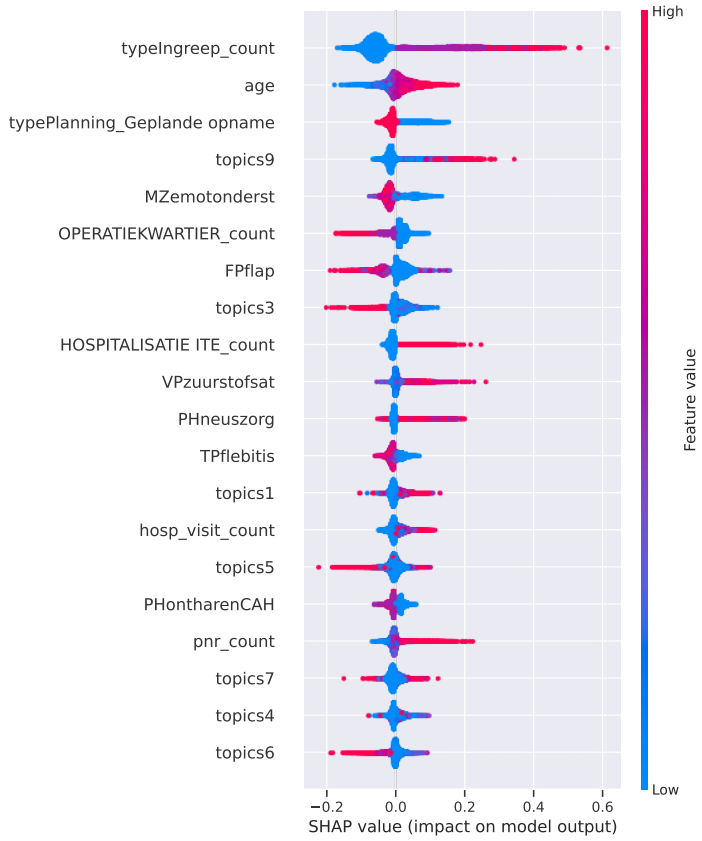


Figure D.1: Binary 7 days classification using XGBoost. 20 most important features from SHAP algorithm with XGBoost model. Features from care module are represented with counts, i.e., the number of times these features were recorded in 24h.

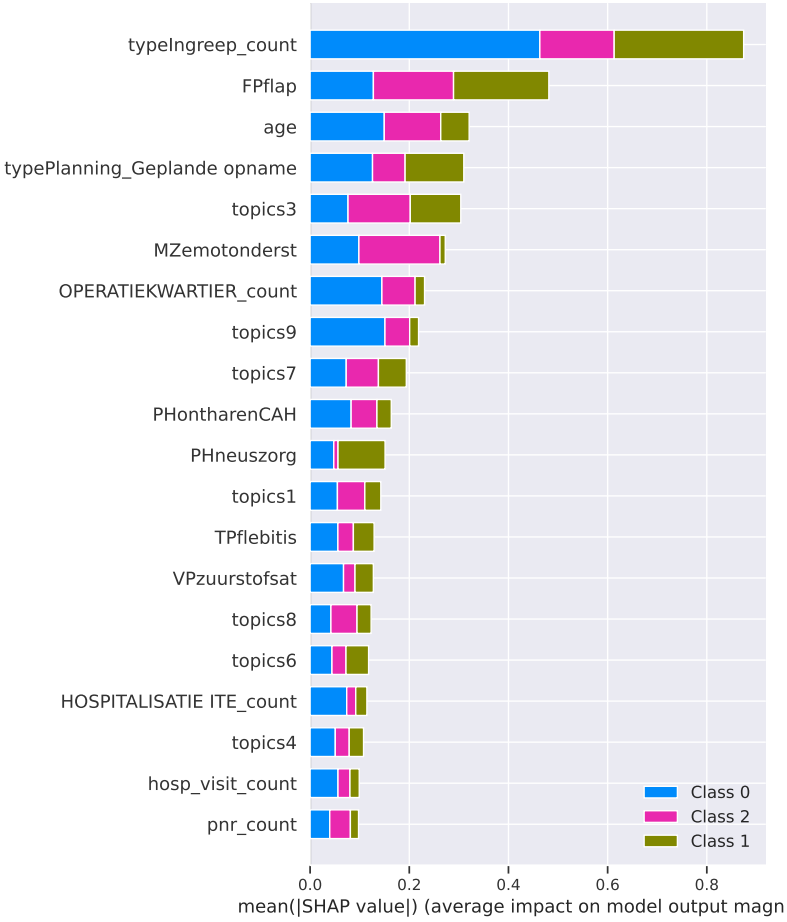


Figure D.2: Multi-class classification using XGBoost. 20 most important features from SHAP algorithm with XGBoost model. Features from care module are represented with counts, i.e., the number of times these features were recorded in 24h.

D.1.10 Key Takeaways

The key takeaways from our preliminary work with the UZ data are as follows:

1. The care administered to patients is a rich source of information for predicting Length of Stay (LoS).

2. One of the most critical factors in any dataset is the frequency with which a patient receives specific care, treatment, or undergoes particular surgeries.

3. Unstructured notes describing medical activities also provide valuable insights for LoS prediction.

4. Feature engineering plays a crucial role, whether in extracting meaningful information (e.g., the day of the week a patient was admitted) or in correcting existing features (e.g., triage codes for urgent night admissions or tracking patients moved from an acute care ward to a critical care ward, such as in the case of `HOSPITALISATIE_ITE`, and flagging alternative patients who are not hospitalized in their admission wards).

D.2 FHIR Dataset

The data received from UZ Leuven's IT department was in FHIR (Fast Healthcare Interoperability Resources) format. This dataset included the following tables:

1. **NxhStay** with 1,354,876 entries.
2. **NxhPresence** with 2,471,621 entries.
3. **NxhContact** with 2,603,320 entries.
4. **NxhClinicalImpression** with 2,617,601 entries.
5. **NxhCareProcedure** with 58,144,530 entries.
6. **NxhProcedure** with 58,720 entries.
7. **NxhCondition** with 78,725 entries.
8. **NxhAllergyIntollerance** with 16,377 entries.
9. **NxhLabObservation** with 13,989,102 entries.

After receiving the link to our FHIR repository on the UZ Leuven server, we connected to the server using POSTMAN and downloaded the various data resources. To gain a clearer view of the data tables, we installed a local MongoDB database, as shown in Figure D.3. Figure D.4 provides a detailed view of the patient resource (from **NxhStay**), which contains essential admission and demographic information for each patient.

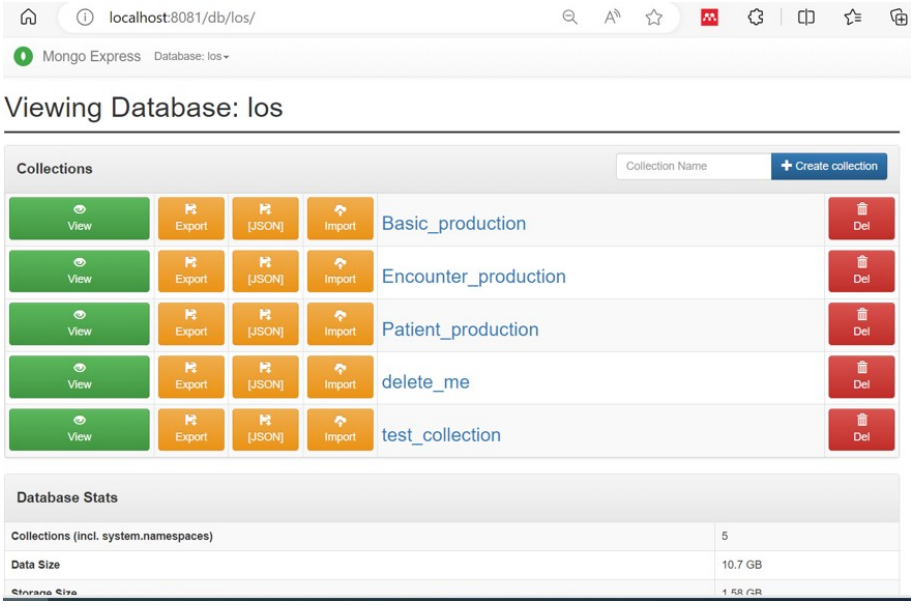


Figure D.3: Snapshot of local MongoDB called *los* showing some resources in the database.

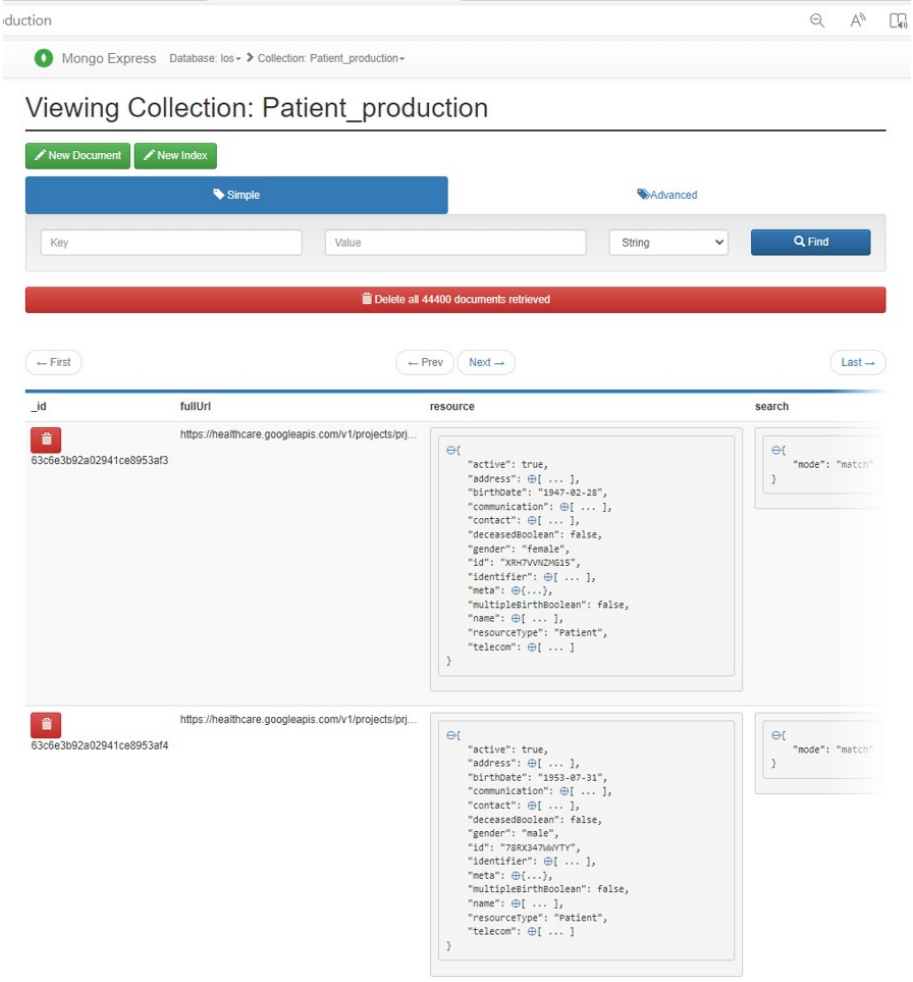


Figure D.4: Each patient is seen as a unique entry in the patient resource

Bibliography

[1] Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. Learning to forget: Continual prediction with lstm. *Neural computation*, 12(10):2451–2471, 2000.

[2] Shaojie Bai, J Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018.

[3] Hervé Abdi and Lynne J Williams. Tukey's honestly significant difference (hsd) test. *Encyclopedia of research design*, 3(1):1–5, 2010.

[4] Health and Food Chain Safety Environment. General hospitals, 2016.

[5] John Hopkins Bloomberg School of Public Health, Asia Pacific Center for Hospital Management and Leadership Research. Hospital management, 2022.

[6] For a healthy Belgium. Quality of care for a healty belgium, 2024.

[7] For a healthy Belgium. Efficiency of healthcare, 2024.

[8] Eurostat. Ageing europe - statistics on population developments - statistics explained, 2019.

[9] Ruth McCabe, Nora Schmit, Paula Christen, Josh C D'Aeth, Alessandra Løchen, Dheeya Rizmie, Shevanthi Nayagam, Marisa Miraldo, Paul Aylin, Alex Bottle, et al. Adapting hospital capacity to meet changing demands during the covid-19 pandemic. *BMC medicine*, 18(1):1–12, 2020.

[10] Segen's Medical Dictionary. length of stay, 2011.

[11] OECD iLibrary. Health at a glance: Europe 2020 : State of health in the eu cycle, 2020.

[12] MIT Critical Data. *Secondary analysis of electronic health records*. Springer Nature, 2016.

[13] Harald O. Stolberg, Geoffrey Norman, and Isabelle Trop. Randomized controlled trials. *American Journal of Roentgenology*, 183(6):1539–1544, 2004. PMID: 15547188.

[14] Nicole Gray Weiskopf. *Enabling the reuse of electronic health record data through data quality assessment and transparency*. Columbia University, 2015.

[15] Jodell E Linder, Lisa Bastarache, Jacob J Hughey, and Josh F Peterson. The role of electronic health records in advancing genomic medicine. *Annual review of genomics and human genetics*, 22:219–238, 2021.

[16] Georges De Moor, Mats Sundgren, Dipak Kalra, Andreas Schmidt, Martin Dugas, Brecht Claerhout, Töresin Karakoyun, Christian Ohmann, Pierre-Yves Lastic, Nadir Ammour, et al. Using electronic health records for clinical research: the case of the ehr4cr project. *Journal of biomedical informatics*, 53:162–173, 2015.

[17] Ahmed Almashrafi, Mustafa Elmontsri, and Paul Aylin. Systematic review of factors influencing length of stay in icu after adult cardiac surgery. *BMC health services research*, 16(1):1–12, 2016.

[18] Yingxin Liu, Mike Phillips, and Jim Codde. Factors influencing patients' length of stay. *Australian Health Review*, 24(2):63–70, 2001.

[19] Philip Yoon, Ivan Steiner, and Gilles Reinhardt. Analysis of factors influencing length of stay in the emergency department. *Canadian Journal of Emergency Medicine*, 5(3):155–161, 2003.

[20] Igor Tona Peres, Silvio Hamacher, Fernando Luiz Cyrino Oliveira, Antônio Márcio Tavares Thomé, and Fernando Augusto Bozza. What factors predict length of stay in the intensive care unit? systematic review and meta-analysis. *Journal of Critical Care*, 60:183–194, 2020.

[21] Gregory Mak, William D Grant, James C McKenzie, and John B McCabe. Physicians' ability to predict hospital length of stay for patients admitted to the hospital from the emergency department. *Emergency medicine international*, 2012, 2012.

[22] Antonio Paulo Nassar Jr and Pedro Caruso. Icu physicians are unable to accurately predict length of stay at admission: a prospective study. *International Journal for Quality in Health Care*, 28(1):99–103, 2016.

[23] Fábio Gusmão Vicente, Frederico Polito Lomar, Christian Mélot, and Jean-Louis Vincent. Can the experienced icu physician predict icu length of stay and outcome better than less experienced colleagues? *Intensive care medicine*, 30(4):655–659, 2004.

[24] Sean Barnes, Eric Hamrock, Matthew Toerper, Sauleh Siddiqui, and Scott Levin. Real-time prediction of inpatient length of stay for discharge prioritization. *Journal of the American Medical Informatics Association*, 23(e1):e2–e10, 2016.

[25] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[26] Ayman Alahmar, Emad Mohammed, and Rachid Benlamri. Application of data mining techniques to predict the length of stay of hospitalized patients with diabetes. In *2018 4th International Conference on Big Data Innovations and Applications (Innovate-Data)*, pages 38–43. IEEE, 2018.

[27] Sanjay Purushotham, Chuizheng Meng, Zhengping Che, and Yan Liu. Benchmarking deep learning models on large healthcare datasets. *Journal of biomedical informatics*, 83:112–134, 2018.

[28] Mahmoud Elbattah and Owen Molloy. Using machine learning to predict length of stay and discharge destination for hip-fracture patients. In *Proceedings of SAI Intelligent Systems Conference*, pages 207–217. Springer, 2016.

[29] Seyedmostafa Sheikhalishahi, Vevake Balaraman, and Venet Osmani. Benchmarking machine learning models on multi-centre eicu critical care dataset. *PloS one*, 15(7):e0235424, 2020.

[30] Hrayr Harutyunyan, Hrant Khachatrian, David C Kale, Greg Ver Steeg, and Aram Galstyan. Multitask learning and benchmarking with clinical time series data. *Scientific data*, 6(1):1–18, 2019.

[31] Dongdong Zhang, Changchang Yin, Jucheng Zeng, Xiaohui Yuan, and Ping Zhang. Combining structured and unstructured data for predictive models: a deep learning approach. *BMC medical informatics and decision making*, 20(1):1–11, 2020.

[32] Shirly Wang, Matthew BA McDermott, Geeticka Chauhan, Marzyeh Ghassemi, Michael C Hughes, and Tristan Naumann. Mimic-extract: A data extraction, preprocessing, and representation pipeline for mimic-iii. In *Proceedings of the ACM Conference on Health, Inference, and Learning*, pages 222–235, 2020.

[33] Gary E Weissman, Rebecca A Hubbard, Lyle H Ungar, Michael O Harhay, Casey S Greene, Blanca E Himes, and Scott D Halpern. Inclusion of unstructured clinical text improves early prediction of death or prolonged icu stay. *Critical care medicine*, 46(7):1125, 2018.

[34] Belal Alsinglawi, Fady Alnajjar, Omar Mubin, Mauricio Novoa, Ola Karajeh, and Omar Darwish. Benchmarking predictive models in electronic health records: Sepsis length of stay prediction. In *International Conference on Advanced Information Networking and Applications*, pages 258–267. Springer, 2020.

[35] Huan Song, Deepta Rajan, Jayaraman Thiagarajan, and Andreas Spanias. Attend and diagnose: Clinical time series analysis using attention models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[36] Junyi Gao, Cao Xiao, Lucas M Glass, and Jimeng Sun. Dr. agent: Clinical predictive model via mimicked second opinions. *Journal of the American Medical Informatics Association*, 27(7):1084–1091, 2020.

[37] Ali Jalali, Hannah Lonsdale, Nhue Do, Jacquelin Peck, Monesha Gupta, Shelby Kutty, Sharon R Ghazarian, Jeffrey P Jacobs, Mohamed Rehman, and Luis M Ahumada. Deep learning for improved risk prediction in surgical outcomes. *Scientific reports*, 10(1):1–13, 2020.

[38] Christopher M Bishop. Pattern recognition and machine learning. *Springer google schola*, 2:5–43, 2006.

[39] Tahmina Zebin, Shahadate Rezvy, and Thierry J Chaussalet. A deep learning approach for length of stay prediction in clinical settings from medical records. In *2019 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, pages 1–5. IEEE, 2019.

[40] Ardeshir Mansouri, Mohammadreza Noei, and Mohammad Saniee Abadeh. Predicting hospital length of stay of neonates admitted to the nicu using data mining techniques. In *2020 10th International Conference on Computer and Knowledge Engineering (ICCKE)*, pages 629–635. IEEE, 2020.

[41] Baran Balkan and Vignesh Subbian. Evaluating icu clinical severity scoring systems and machine learning applications: Apache iv/iva case study. In *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 4073–4076. IEEE, 2018.

[42] Lama Alturki, Khawlah Aloraini, Amina Aldughayshim, and Saleh Albahli. Predictors of readmissions and length of stay for diabetes related patients. In *2019 IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA)*, pages 1–8. IEEE, 2019.

[43] Julia Elrod, Christoph Mohr, Ruben Wolff, Michael Boettcher, Konrad Reinshagen, Pia Bartels, German Burn Registry, and Ingo Koenigs. Using artificial intelligence to obtain more evidence? prediction of length of hospitalization in pediatric burn patients. *Frontiers in Pediatrics*, 8, 2020.

[44] Belal Alsinglawi, Fady Alnajjar, Omar Mubin, Mauricio Novoa, Mohammed Alorjani, Ola Karajeh, and Omar Darwish. Predicting length of stay for cardiovascular hospitalizations in the intensive care unit: Machine learning approach. In *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pages 5442–5445. IEEE, 2020.

[45] Roberto Williams Batista and Reinaldo Sanchez-Arias. A methodology for estimating hospital intensive care unit length of stay using novel machine learning tools. In *2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 827–832. IEEE, 2020.

[46] Jaret M Karnuta, Jessica L Churchill, Heather S Haeberle, Benedict U Nwachukwu, Samuel A Taylor, Eric T Ricchetti, and Prem N Ramkumar. The value of artificial neural networks for predicting length of stay, discharge disposition, and inpatient costs after anatomic and reverse shoulder arthroplasty. *Journal of Shoulder and Elbow Surgery*, 29(11):2385–2394, 2020.

[47] Ameet V Joshi. *Machine learning and artificial intelligence*. Springer, 2020.

[48] Thanos Gentimis, Alnaser Ala'J, Alex Durante, Kyle Cook, and Robert Steele. Predicting hospital length of stay using neural networks on mimic iii data. In *2017 IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence and Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech)*, pages 1194–1201. IEEE, 2017.

[49] Hrayr Harutyunyan, Hrant Khachatrian, David C. Kale, Greg Ver Steeg, and Aram Galstyan. Multitask learning and benchmarking with clinical time series data. *Scientific Data*, 6(1):96, 2019.

[50] Emma Rocheteau, Pietro Liò, and Stephanie Hyland. Temporal pointwise convolutional networks for length of stay prediction in the intensive care

unit. In *Proceedings of the Conference on Health, Inference, and Learning*, pages 58–68, 2021.

[51] Yanbo Xu, Siddharth Biswal, Shriprasad R Deshpande, Kevin O Maher, and Jimeng Sun. Raim: Recurrent attentive and intensive model of multimodal patient monitoring data. In *Proceedings of the 24th ACM SIGKDD international conference on Knowledge Discovery & Data Mining*, pages 2565–2573, 2018.

[52] Swaraj Khadanga, Karan Aggarwal, Shafiq Joty, and Jaideep Srivastava. Using clinical notes with time series data for icu management. *arXiv preprint arXiv:1909.09702*, 2019.

[53] Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. Recurrent neural networks for multivariate time series with missing values. *Scientific reports*, 8(1):1–12, 2018.

[54] Gaspard Harerimana, Jong Wook Kim, and Beakcheol Jang. A deep attention model to forecast the length of stay and the in-hospital mortality right on admission from icd codes and demographic data. *Journal of Biomedical Informatics*, 118:103778, 2021.

[55] Tao Wang, Haiqing Zhang, Antoine Duclos, Cécile Payet, and Daiwei Li. Prediction of hospital length of stay to achieve flexible healthcare in the field of internet of vehicles. *Transactions on emerging telecommunications technologies*, page e3854, 2019.

[56] Catherine Tong, Emma Rocheteau, Petar Veličković, Nicholas Lane, and Pietro Liò. Predicting patient outcomes with graph representation learning. In *International Workshop on Health Intelligence*, pages 281–293. Springer, 2021.

[57] Sujin Kim, Woojae Kim, and Rae Woong Park. A comparison of intensive care unit mortality prediction models through the use of data mining techniques. *Healthcare informatics research*, 17(4):232, 2011.

[58] Tom J Pollard, Alistair EW Johnson, Jesse D Raffa, Leo A Celi, Roger G Mark, and Omar Badawi. The eicu collaborative research database, a freely available multi-center database for critical care research. *Scientific data*, 5(1):1–13, 2018.

[59] Patrick J Thoral, Jan M Peppink, Ronald H Driessen, Eric JG Sijbrands, Erwin JO Kompanje, Lewis Kaplan, Heatherlee Bailey, Jozef Kesecioglu, Maurizio Cecconi, Matthew Churpek, et al. Sharing icu patient data responsibly under the society of critical care medicine/european society of intensive care medicine joint data science collaboration: The amsterdam

university medical centers database (amsterdamumcdb) example. *Critical care medicine*, 2021.

[60] Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-Wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3(1):1–9, 2016.

[61] Alistair Johnson, Lucas Bulgarelli, Tom Pollard, Steven Horng, Leo Anthony Celi, and R Mark IV. Mimic-iv (version 1.0). *PhysioNet*, 2020.

[62] Ary L Goldberger, Luis AN Amaral, Leon Glass, Jeffrey M Hausdorff, Plamen Ch Ivanov, Roger G Mark, Joseph E Mietus, George B Moody, Chung-Kang Peng, and H Eugene Stanley. Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. *circulation*, 101(23):e215–e220, 2000.

[63] Zachary C Lipton, John Berkowitz, and Charles Elkan. A critical review of recurrent neural networks for sequence learning. *arXiv preprint arXiv:1506.00019*, 2015.

[64] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery*, 33(4):917–963, 2019.

[65] Hansika Hewamalage, Christoph Bergmeir, and Kasun Bandara. Recurrent neural networks for time series forecasting: Current status and future directions. *International Journal of Forecasting*, 37(1):388–427, 2021.

[66] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[67] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078, 2014.

[68] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[69] Neena Aloysius and M Geetha. A review on deep convolutional neural networks. In *2017 International Conference on Communication and Signal Processing (ICCSP)*, pages 0588–0592. IEEE, 2017.

[70] Keiron O'Shea and Ryan Nash. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*, 2015.

[71] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[72] Serkan Kiranyaz, Turker Ince, Osama Abdeljaber, Onur Avci, and Moncef Gabbouj. 1-d convolutional neural networks for signal processing applications. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8360–8364. IEEE, 2019.

[73] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[74] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, pages 5998–6008, 2017.

[75] Simon JD Prince. *Understanding Deep Learning*. MIT press, 2023.

[76] Hamed M Zolbanin, Behrooz Davazdahemami, Dursun Delen, and Amir Hassan Zadeh. Data analytics for the sustainable use of resources in hospitals: Predicting the length of stay for patients with chronic diseases. *Information & Management*, page 103282, 2020.

[77] Jingyi Wu, Guilan Kong, Yu Lin, Hong Chu, Chao Yang, Ying Shi, Haibo Wang, and Luxia Zhang. Development of a scoring tool for predicting prolonged length of hospital stay in peritoneal dialysis patients through data mining. *Annals of Translational Medicine*, 8(21), 2020.

[78] Alvin Rajkomar, Eyal Oren, Kai Chen, Andrew M Dai, Nissan Hajaj, Michaela Hardt, Peter J Liu, Xiaobing Liu, Jake Marcus, Mimi Sun, et al. Scalable and accurate deep learning with electronic health records. *NPJ Digital Medicine*, 1(1):1–10, 2018.

[79] Tiago Alves, Alberto Laender, Adriano Veloso, and Nivio Ziviani. Dynamic prediction of icu mortality risk using domain adaptation. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 1328–1336. IEEE, 2018.

[80] CP Launay, H Rivière, A Kabeshova, and O Beauchet. Predicting prolonged length of hospital stay in older emergency department users: use of a novel analysis method, the artificial neural network. *European journal of internal medicine*, 26(7):478–482, 2015.

[81] Enrique Casalino, Mathias Wargon, Anne Peroziello, Christophe Choquet, Christophe Leroy, Sebastien Beaune, Laurent Pereira, Julien Bernard, and Jean-Claude Buzzi. Predictive factors for longer length of stay in an emergency department: a prospective multicentre study evaluating the impact of age, patient's clinical acuity and complexity, and care pathways. *Emergency Medicine Journal*, 31(5):361–368, 2014.

[82] Whitney E Muhlestein, Dallin S Akagi, Jason M Davies, and Lola B Chambless. Predicting inpatient length of stay after brain tumor surgery: Developing machine learning ensembles to improve predictive performance. *Neurosurgery*, 85(3):384–393, 2019.

[83] Ru Ding, Melissa L McCarthy, Jennifer Lee, Jeffrey S Desmond, Scott L Zeger, and Dominik Aronsky. Predicting emergency department length of stay using quantile regression. In *2009 International Conference on Management and Service Science*, pages 1–4. IEEE, 2009.

[84] Pei-Fang Jennifer Tsai, Po-Chia Chen, Yen-You Chen, Hao-Yuan Song, Hsiu-Mei Lin, Fu-Man Lin, and Qiou-Pieng Huang. Length of hospital stay prediction at the admission stage for cardiology patients using artificial neural network. *Journal of healthcare engineering*, 2016, 2016.

[85] Harini Suresh, Jen J Gong, and John V Guttag. Learning tasks for multitask learning: Heterogenous patient populations in the icu. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 802–810, 2018.

[86] Abolfazl Farahani, Sahar Voghoei, Khaled Rasheed, and Hamid R Arabnia. A brief review of domain adaptation. *Advances in data science and information engineering*, pages 877–894, 2021.

[87] Xiaojie Jin, Yunpeng Chen, Jian Dong, Jiashi Feng, and Shuicheng Yan. Collaborative layer-wise discriminative learning in deep neural networks. In *European Conference on Computer Vision*, pages 733–749. Springer, 2016.

[88] Bernd Rechel, Antonio Duran, and Richard Saltman. What is the experience of decentralized hospital governance in europe. 2018.

[89] Alistair Johnson, Lucas Bulgarelli, Tom Pollard, Leo Anthony Celi, Roger Mark, and Steven Horng. Mimic-iv-ed (version 1.0), 2021.

[90] Zachary C Lipton, David C Kale, Randall Wetzel, et al. Modeling missing data in clinical time series with rnns. *Machine Learning for Healthcare*, 56, 2016.

[91] Francois Chollet et al. Keras, 2015.

[92] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

[93] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *Advances in neural information processing systems*, 27, 2014.

[94] Edward Choi, Mohammad Taha Bahadori, Andy Schuetz, Walter F. Stewart, and Jimeng Sun. Doctor ai: Predicting clinical events via recurrent neural networks. In Finale Doshi-Velez, Jim Fackler, David Kale, Byron Wallace, and Jenna Wiens, editors, *Proceedings of the 1st Machine Learning for Healthcare Conference*, volume 56 of *Proceedings of Machine Learning Research*, pages 301–318, Northeastern University, Boston, MA, USA, 18–19 Aug 2016. PMLR.

[95] Ji Young Lee, Franck Dernoncourt, and Peter Szolovits. Transfer learning for named-entity recognition with neural networks. *arXiv preprint arXiv:1705.06273*, 2017.

[96] Priyanka Gupta, Pankaj Malhotra, Jyoti Narwariya, Lovekesh Vig, and Gautam Shroff. Transfer learning for clinical time series analysis using deep neural networks. *Journal of Healthcare Informatics Research*, 4(2):112–137, 2020.

[97] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010.

[98] Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*, 2018.

[99] Gabriel Erion, Joseph D Janizek, Pascal Sturmfels, Scott M Lundberg, and Su-In Lee. Improving performance of deep learning models with axiomatic attribution priors and expected gradients. *Nature machine intelligence*, 3(7):620–631, 2021.

[100] Tom O'Malley, Elie Bursztein, James Long, François Chollet, Haifeng Jin, Luca Invernizzi, et al. Keras tuner. *Github.[(accessed on 31 January 2021)]*, 2019.

[101] Tom O'Malley, Elie Bursztein, James Long, François Chollet, Haifeng Jin, Luca Invernizzi, et al. Kerastuner. `https://github.com/keras-team/keras-tuner`, 2019.

[102] Paul Lillrank, P Johan Groop, and Tomi J Malmström. Demand and supply–based operating modes—a framework for analyzing health care service production. *The Milbank Quarterly*, 88(4):595–615, 2010.

[103] John Rapoport, Daniel Teres, Yonggang Zhao, and Stanley Lemeshow. Length of stay data as a guide to hospital economic performance for icu patients. *Medical care*, pages 386–397, 2003.

[104] Ashish K Khanna, Sonia O Labeau, Kathryn McCartney, Stijn I Blot, Mieke Deschepper, DecubICUs study Team, European Society of Intensive Care Medicine, et al. International variation in length of stay in intensive care units and the impact of patient-to-nurse ratios. *Intensive and Critical Care Nursing*, page 103265, 2022.

[105] Lahn D Straney, Andrew A Udy, Aidan Burrell, Christoph Bergmeir, Sue Huckson, D James Cooper, and David V Pilcher. Modelling risk-adjusted variation in length of stay among australian and new zealand icus. *PloS one*, 12(5):e0176570, 2017.

[106] Reed T Sutton, David Pincock, Daniel C Baumgart, Daniel C Sadowski, Richard N Fedorak, and Karen I Kroeker. An overview of clinical decision support systems: benefits, risks, and strategies for success. *NPJ digital medicine*, 3(1):17, 2020.

[107] Kieran Stone, Reyer Zwiggelaar, Phil Jones, and Neil Mac Parthaláin. A systematic review of the prediction of hospital length of stay: Towards a unified framework. *PLOS Digital Health*, 1(4):e0000017, 2022.

[108] Aditya Achanta, Ask Nordestgaard, Napaporn Kongkaewpaisan, Kelsey Han, April Mendoza, Noelle Saillant, Martin Rosenthal, Peter Fagenholz, George Velmahos, and Haytham MA Kaafarani. Most of the variation in length of stay in emergency general surgery is not related to clinical factors of patient care. *Journal of Trauma and Acute Care Surgery*, 87(2):408–412, 2019.

[109] Ashwani Kumar and Hamideh Anjomshoa. A two-stage model to predict surgical patients' lengths of stay from an electronic patient database. *IEEE journal of biomedical and health informatics*, 23(2):848–856, 2018.

[110] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

[111] Amil Dravid, Florian Schiffers, Boqing Gong, and Aggelos K Katsaggelos. medxgan: Visual explanations for medical classifiers through a generative latent space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2936–2945, 2022.

[112] David Chardin, Cyprien Gille, Thierry Pourcher, Olivier Humbert, and Michel Barlaud. Learning a confidence score and the latent space of a new supervised autoencoder for diagnosis and prognosis in clinical metabolomic studies. *BMC bioinformatics*, 23(1):361, 2022.

[113] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International conference on machine learning*, pages 3319–3328. PMLR, 2017.

[114] Patrick J Thoral, Mattia Fornasa, Daan P de Bruin, Michele Tonutti, Hidde Hovenkamp, Ronald H Driessen, Armand RJ Girbes, Mark Hoogendoorn, and Paul WG Elbers. Explainable machine learning on amsterdamumcdb for icu discharge decision support: uniting intensivists and data scientists. *Critical care explorations*, 3(9), 2021.

[115] Hugo Yèche, Rita Kuznetsova, Marc Zimmermann, Matthias Hüser, Xinrui Lyu, Martin Faltys, and Gunnar Rätsch. Hirid-icu-benchmark–a comprehensive machine learning benchmark on high-resolution icu data. *arXiv preprint arXiv:2111.08536*, 2021.

[116] Ilya Loshchilov and Frank Hutter. Fixing weight decay regularization in adam. *CoRR*, abs/1711.05101, 2017.

[117] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[118] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

[119] Shammamah Hossain. Visualization of Bioinformatics Data with Dash Bio. In Chris Calloway, David Lippa, Dillon Niederhut, and David Shupe, editors, *Proceedings of the 18th Python in Science Conference*, pages 126 – 133, 2019.

[120] Elias Dabbas. *Interactive Dashboards and Data Apps with Plotly and Dash: Harness the power of a fully fledged frontend web framework in Python–no JavaScript required.* Packt Publishing Ltd, 2021.

[121] Max Jaderberg, Valentin Dalibard, Simon Osindero, Wojciech M Czarnecki, Jeff Donahue, Ali Razavi, Oriol Vinyals, Tim Green, Iain

Dunning, Karen Simonyan, et al. Population based training of neural networks. *arXiv preprint arXiv:1711.09846*, 2017.

[122] Hassan Mumtaz, Muhammad K Ejaz, Muhammad Tayyab, Laiba I Vohra, Shova Sapkota, Mohammad Hasan, and Muhammad Saqib. Apache scoring as an indicator of mortality rate in icu patients: a cohort study. *Annals of Medicine and Surgery*, 85(3):416–421, 2023.

[123] Batuhan Bardak and Mehmet Tan. Improving clinical outcome predictions using convolution over medical entities with multimodal learning. *Artificial Intelligence in Medicine*, 117:102112, 2021.

[124] Kexin Huang, Abhishek Singh, Sitong Chen, Edward T Moseley, Chih-Ying Deng, Naomi George, and Charlotta Lindvall. Clinical xlnet: Modeling sequential clinical notes and predicting prolonged mechanical ventilation. *arXiv preprint arXiv:1912.11975*, 2019.

[125] Jinghui Liu, Daniel Capurro, Anthony Nguyen, and Karin Verspoor. Attention-based multimodal fusion with contrast for robust clinical prediction in the face of missing modalities. *Journal of Biomedical Informatics*, 145:104466, 2023.

[126] Jiyoun Song, Mollie Hobensack, Kathryn H Bowles, Margaret V McDonald, Kenrick Cato, Sarah Collins Rossetti, Sena Chae, Erin Kennedy, Yolanda Barrón, Sridevi Sridharan, et al. Clinical notes: an untapped opportunity for improving risk prediction for hospitalization and emergency department visit during home health care. *Journal of biomedical informatics*, 128:104039, 2022.

[127] Bo Yang and Lijun Wu. How to leverage multimodal ehr data for better medical predictions? *arXiv preprint arXiv:2110.15763*, 2021.

[128] Yuanlong Wang, Changchang Yin, and Ping Zhang. Multimodal risk prediction with physiological signals, medical images and clinical notes. *Heliyon*, 10(5), 2024.

[129] Yimian Dai, Fabian Gieseke, Stefan Oehmcke, Yiquan Wu, and Kobus Barnard. Attentional feature fusion. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 3560–3569, 2021.

[130] Chao-Chun Hsu, Shantanu Karnwal, Sendhil Mullainathan, Ziad Obermeyer, and Chenhao Tan. Characterizing the value of information in medical notes. *arXiv preprint arXiv:2010.03574*, 2020.

[131] Eric Lehman and Alistair Johnson. Clinical-t5: Large language models built using mimic clinical text. *PhysioNet*, 2023.

[132] Taku Kudo and John Richardson. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*, 2018.

[133] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[134] Joseph Paul Cohen, Mohammad Hashir, Rupert Brooks, and Hadrien Bertrand. On the limits of cross-domain generalization in automated x-ray prediction. In *Medical Imaging with Deep Learning*, pages 136–155. PMLR, 2020.

[135] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M Summers. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2097–2106, 2017.

[136] Aurelia Bustos, Antonio Pertusa, Jose-Maria Salinas, and Maria De La Iglesia-Vaya. Padchest: A large chest x-ray image dataset with multi-label annotated reports. *Medical image analysis*, 66:101797, 2020.

[137] J Irvin, P Rajpurkar, M Ko, Y Yu, S Ciurea-Ilcus, C Chute, H Marklund, B Haghgoo, R Ball, and K Shpanskaya. A large chest radiograph dataset with uncertainty labels and expert comparison. In *Proc AAAI Conf Artif Intell*, volume 33.

[138] Dina Demner-Fushman, Marc D Kohli, Marc B Rosenman, Sonya E Shooshan, Laritza Rodriguez, Sameer Antani, George R Thoma, and Clement J McDonald. Preparing a collection of radiology examinations for distribution and retrieval. *Journal of the American Medical Informatics Association*, 23(2):304–310, 2016.

[139] Anna Majkowska, Sid Mittal, David F Steiner, Joshua J Reicher, Scott Mayer McKinney, Gavin E Duggan, Krish Eswaran, Po-Hsuan Cameron Chen, Yun Liu, Sreenivasa Raju Kalidindi, et al. Chest radiograph interpretation with deep learning models: assessment with radiologist-adjudicated reference standards and population-adjusted evaluation. *Radiology*, 294(2):421–431, 2020.

[140] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[141] Imanol Isasa, Mikel Hernandez, Gorka Epelde, Francisco Londoño, Andoni Beristain, Xabat Larrea, Ane Alberdi, Panagiotis Bamidis, and Evdokimos Konstantinidis. Comparative assessment of synthetic time series generation approaches in healthcare: leveraging patient metadata for accurate data synthesis. *BMC Medical Informatics and Decision Making*, 24(1):27, 2024.

[142] Vincent Scheltjens, Lyse Naomi Wamba Momo, Wouter Verbeke, and Bart De Moor. Client recruitment for federated learning in icu length of stay prediction. In *2023 IEEE 19th International Conference on e-Science (e-Science)*, pages 1–9, 2023.

[143] Michelle M Li, Kexin Huang, and Marinka Zitnik. Graph representation learning in biomedicine and healthcare. *Nature Biomedical Engineering*, 6(12):1353–1369, 2022.

[144] Jiaxuan You, Xiaobai Ma, Yi Ding, Mykel J Kochenderfer, and Jure Leskovec. Handling missing data with graph representation learning. *Advances in Neural Information Processing Systems*, 33:19075–19087, 2020.

[145] Daniel Walke, Daniel Micheel, Kay Schallert, Thilo Muth, David Broneske, Gunter Saake, and Robert Heyer. The importance of graph databases and graph learning for clinical applications. *Database*, 2024:baad045, 2024.

[146] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. *Advances in neural information processing systems*, 25, 2012.

[147] Marina Sokolova and Guy Lapalme. A systematic analysis of performance measures for classification tasks. *Information processing & management*, 45(4):427–437, 2009.

FACULTY OF ENGINEERING SCIENCE
DEPARTMENT OF ELECTRICAL ENGINEERING
XXXXX
Celestijnenlaan 200A box 2402
B-3001 Leuven
first.last@cs.kuleuven.be
http://www.XXXXX.cs.kuleuven.be