Temporal Phenotypic Clustering of Electronic Health Records



Henrique Aguiar

St. John's College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Trinity Term 2026

This thesis is dedicated to my family: my wonderful sister, Marta, and my incredible parents, Rui and Lídia, without whom I would not be where I am today.

Acknowledgements

Completing this thesis was one of the most challenging and humbling endeavors of my life. The years I spent in pursuit of this degree tested my limits in ways I did not expect, and I am grateful to have had the support of so many extraordinary people along the way. The completion of this doctorate is owed in no small part to them and their support, and I will do my best to convey my appreciation to all of them, although I am confident any attempt will inevitably fall short of what is deserved.

My deepest and most heartfelt thanks go to my supervisors, Dr. Tingting Zhu and Dr. Mauro Santos, without whom this thesis would not have been possible. The work presented in this thesis would have been of much lesser quality without their continuous support, guidance, and mentorship. They were always open to discuss new ideas, even if that meant a discussion of several hours, and they were always available with any challenges, personal and professional, that arose along the way. I am incredibly grateful to be included on your growing list of mentees. I also extend my gratitude to Professor Peter Watkinson, whose critical clinical insights were invaluable in guiding our research and who always kindly supported my research efforts. An additional acknowledgment is also given to Professor David Prytherch and Professor Gary Smith for offering an additional clinical perspective. Finally, a warm thanks goes to the wider Critical Care Research Group team, particularly Rachel Henning and Dr. Robert Hatch, for their expertise and assistance with the numerous roadblocks during this journey.

During my doctoral studies, I was fortunate to be a member of several academic environments and academic communities. Within my academic group, I had the opportunity to work with and learn from many incredible researchers and I am thankful for all the conversations, collaborations, and interactions. Some special names I would like to mention are Dr. Zhiyao Luo, Dr. Munib Mesinovic, Dr. Yu Liu, Dr. Jin Li, and Dr. Ghadeer Ghosheh. Many hotpots, meals, and punting trips were had and I will treasure those memories fondly. I was also fortunate to be affiliated with the Computational Health Informatics Lab and thank its members for the many conversations, shared experiences, and warm atmosphere. During the final years of my Ph.D., I had the opportunity to visit Nanjing Southeast University in China and to work and collaborate with a great

team of researchers, so I extend a special thanks to Professor Chengyu Liu and the rest of the lab.

This thesis was completed within the Health Data Science Center for Doctoral Training (CDT), supported by a scholarship from the Engineering and Physical Sciences Research Council (EPSRC). To both, I am grateful for their financial and administrative support, as well as the opportunity to be part of the broader community that was nurtured — thanks that extend to the University of Oxford and the Department of Engineering Science. Many people were included; however, I would like to make a particular note of Professor Thomas Nichols, Professor Jens Rittscher, the late Professor Nina Hallowell, and Leema Chapman, with whom I interacted commonly. Many other students within the program became not only exceptional fellow researchers, but also close friends, and I extend my special thanks to all of them, but particularly to Dr. Hang Yuan, Dr. Kangning Zhang, and Dr. Yifan Yu. Beyond academia, I was privileged to complete industry placements during this degree, at Huawei, in Paris, and DataSpartan, in London, and these were experiences that broadened my perspective and enriched my doctoral journey in unexpected ways. I am thankful to both companies for the opportunity and to all the people with whom I had the opportunity to work during those placements, who were always kind and supportive.

On a more personal note, I was extremely fortunate to be surrounded by a wide and warm circle of friends throughout this journey. To Dr. Benjamin Burman and Dr. Ruby-Anne Birin — you came into my life later than most, but I will never forget the countless Friday night dinners, the walks, and the adventures we had together. It was a great pleasure to attend your wedding and I look forward to treasuring our friendship in the future. To Michael Leadbetter and Jannus Estoesta, thank you for the laughter, warmth, your welcoming nature, and, in particular, your extensive patience in dealing with me! Many others deserve a mention, and I am grateful to Dr. Hang Yuan, Dr. Kangning Zhang, Dr. Yifan Yu, Dr. Giorgia Gaeta, Dr. Li Pi, Dr. Andreea Oncescu, Emil Nichita, Dr. Cong Lu, Dr. Anna Olerinyova, Shaked Keidar and Dr. Hadas Sloin, Xavier Tierney, Iulian Sclifos, and Bruno Nunes, for the shared experiences and support. To my Oxford Judo friends, thank you for the countless hours of training and the camaraderie. Finally, a particular thank you to my fellow Junior Deans at St. John's College, Dr. Ruby-Anne Birin, Dr. Razia Chowdry, Dr. Diala Al Masri, and Dr. Manal Bougazzoul, for making the role as rewarding as it was. This was a challenging role, but it was made so much easier thanks to the tightness and closeness we developed. The role and the opportunity to transition to St. John's were great additions to my doctoral life, for which I am grateful.

Lastly and most of all, I would like to thank my wonderful family. To my grandmothers Maria Armanda and Lídia, thank you for your unfailing love and

warmth, even though the precise nature of this research, or even what a doctorate entails, remains somewhat elusive. To my grandfathers Silvério and Manuel, who are no longer with us, I carry your spirits with me, particularly in the hardest times, and I have no doubt you would have been proud. To my uncle and aunt, Silvério and Carla Neves, thank you for your kindness, for always welcoming me so warmly and for the additional joy brought into the family through the addition of Lady, Kiko, and Bond. To my sister, Marta Aguiar, thank you for a lifetime of patience, honesty, and unwavering support, as well as your ongoing efforts to teach me how to be a functioning, respectable adult. And, to conclude, to my parents, Dr. Rui Aguiar and Lídia Aguiar, words are insufficient to express what is warranted. You have sacrificed more than I will ever understand in order to ensure that I could achieve my goals and realize my ambitions, often at considerable cost. I have been, am, and will always be deeply grateful for your love.

Abstract

Temporal phenotypic clustering (TPC) constitutes a machine learning paradigm for identifying clinically meaningful patient subgroups using electronic health record (EHR) data. TPC jointly leverages different sources of information, including longitudinal physiological trajectories and others, such as information related to the occurrence of adverse events. Despite growing interest in the application of machine learning to clinical data, current TPC methods fail to fully integrate these sources during cluster formation and have demonstrated limited clinical validation in real-world cohorts. This thesis makes several contributions to the TPC literature. First, TPC is formally defined within an EHR context, and a unified experimental framework is introduced, covering data considerations and the evaluation of different data sources. Second, a new model is proposed to address settings characterized by outcome distributions with severe class imbalance. Third, an interpretability-aware method is introduced, incorporating an attention-derived layer that quantifies the contributions of individual features and time points to the assignment of patients to phenotypes. Fourth, a new approach for dynamic phenotyping is presented, which naturally extends the static TPC framework to support patient trajectory forecasting. Methods are evaluated across different clinical settings, including acute care and emergency department contexts. Overall, the models proposed in this thesis advance the state of the art in TPC and demonstrate the feasibility of deploying such methods to support clinical decision-making and phenotype discovery in real-world EHR settings.

Contents

List of Figures xv

List of Tables xvii

Mathematical Notation xix

Glossary xxi

1 Introduction 1
- 1.1 Motivation . . . 2
- 1.2 The Computational Problem . . . 4
- 1.3 Limitations of Existing Work . . . 7
- 1.4 Datasets and Research Scope . . . 9
- 1.5 Thesis Layout . . . 11
  - 1.5.1 Publications . . . 13

2 Related Work 15
- 2.1 Introduction . . . 15
- 2.2 The Clustering Problem . . . 16
  - 2.2.1 Soft and Hard Clustering . . . 18
  - 2.2.2 The Clustering Objective . . . 21
  - 2.2.3 General Families of Clustering Algorithms . . . 23
- 2.3 Handling Temporal Data . . . 25
  - 2.3.1 Univariate Data . . . 27
    - 2.3.1.1 Similarity-Based Techniques . . . 27
    - 2.3.1.2 Representation-Based Techniques . . . 29
  - 2.3.2 Multivariate Data . . . 31
- 2.4 Applications in the Healthcare Domain . . . 35
  - 2.4.1 Deep Learning for Clustering Temporal Data . . . 36
    - 2.4.1.1 Data-Driven Adaptations . . . 36
    - 2.4.1.2 Clinically-Driven Adaptations . . . 37
  - 2.4.2 Temporal Phenotypic Clustering with EHR data . . . 39
- 2.5 Conclusion . . . 42

3 Experimental Pipeline **43**
3.1 Introduction . . . 43
3.2 Data . . . 43
3.2.1 HAVEN . . . 44
3.2.2 MIMIC-IV-ED . . . 48
3.3 A Unified Approach for Preprocessing . . . 51
3.3.1 Outlier Removal . . . 51
3.3.2 Removal of Multiple Admissions . . . 52
3.3.3 Time Windowing . . . 53
3.3.4 Missingness Proportion Thresholding . . . 54
3.3.5 Missing Value Imputation . . . 54
3.3.6 Feature Normalization . . . 55
3.4 Evaluation Procedure . . . 57
3.4.1 Temporal Aspect . . . 57
3.4.2 Outcome Aspect . . . 61
3.4.2.1 MCS Classification . . . 63
3.4.3 Combined Evaluation . . . 66
3.5 Conclusion . . . 68

4 Temporal Phenotypic Clustering with Imbalanced Data **71**
4.1 Introduction . . . 71
4.2 Benchmark Methods . . . 72
4.2.1 Clustering and TPC SOTA Models . . . 73
4.2.1.1 TSKM . . . 73
4.2.1.2 SOM-VAE . . . 78
4.2.1.3 AC-TPC . . . 81
4.2.2 Proxy Task Benchmarks . . . 86
4.2.2.1 SVM . . . 87
4.2.2.2 XGBoost . . . 87
4.2.2.3 NEWS2 . . . 88
4.3 Improving Multiclass Outcome Modeling . . . 89
4.4 Implementation . . . 92
4.4.1 Model Training . . . 92
4.4.2 Hyperparameter Selection . . . 93
4.4.3 Selection of K . . . 94
4.5 Results . . . 95
4.5.1 Temporal Aspect . . . 95
4.5.2 Outcome Aspect . . . 96
4.6 Conclusion . . . 99

5 Interpretable Deep Clustering **101**
5.1 Introduction . . . 101
5.2 Background . . . 102
5.3 CAMELOT . . . 104
5.3.1 Model architecture . . . 105
5.3.2 Encoder Architecture . . . 107
5.3.2.1 Custom Attention Mechanism . . . 107
5.3.2.2 Encoder Pass . . . 109
5.3.2.3 Design Rationale . . . 110
5.3.3 Model Training . . . 110
5.3.3.1 Main Training Phase . . . 110
5.3.3.2 Initialization . . . 113
5.3.3.3 Cluster Collapse . . . 113
5.3.4 Attention Map Visualization . . . 115
5.4 Implementation . . . 118
5.4.1 Experimental Setup . . . 118
5.4.2 Ablation Studies . . . 120
5.5 Results . . . 121
5.5.1 Temporal Aspect . . . 121
5.5.2 Outcome Aspect . . . 122
5.5.3 Attention Maps . . . 125
5.5.4 Ablation Studies . . . 127
5.6 Conclusion . . . 130

6 Dynamic Phenotyping **131**
6.1 Introduction . . . 131
6.2 Background . . . 132
6.2.1 Dynamic Clustering . . . 132
6.2.2 Dirichlet Distribution . . . 135
6.3 DeepGC . . . 139
6.3.1 Model Architecture . . . 139
6.3.2 Probabilistic Modeling . . . 142
6.3.3 Model Training . . . 146
6.3.4 Cluster Assignment and Trajectory Generation . . . 149
6.4 Implementation . . . 150
6.4.1 Experimental Setup . . . 151
6.4.2 Post Hoc Analysis . . . 153
6.4.2.1 Predictive Horizon Analysis . . . 153
6.4.2.2 Utility of the Generative Component . . . 154

6.5 Results . . . 154
6.5.1 Temporal Aspect . . . 155
6.5.2 Outcome Aspect . . . 156
6.5.3 Post Hoc Analysis . . . 160
6.5.3.1 Predictive Horizon Analysis . . . 160
6.5.3.2 Generative Utility . . . 162
6.6 Conclusion . . . 164

7 Conclusion **167**
7.1 Summary of Contributions . . . 168
7.2 Limitations . . . 172
7.2.1 Definition of the Computational Setting . . . 172
7.2.2 Methodological Assumptions . . . 173
7.2.3 Data Scope and Clinical Translation . . . 175
7.3 Future Directions . . . 175
7.4 Concluding Remarks . . . 178

References **181**

A Hyperparameter Grids **211**
A.1 Experimental Setup for Chapter 4 . . . 211
A.2 Experimental Setup for Chapter 5 . . . 213
A.3 Experimental Setup for Chapter 6 . . . 215

B Supplementary Results Chapter 5 **217**
B.1 HAVEN-T2RF . . . 217
B.1.1 Empirical Cluster Outcome Counts . . . 217
B.1.2 Descriptive Statistics of Learned Cluster Phenotypes . . . 217
B.1.3 Mean Trajectory Visualizations . . . 220
B.2 MIMIC-IV-ED . . . 221

C Supplementary Results Chapter 6 **223**
C.1 Proof of the DeepGC ELBO . . . 223
C.2 Cluster Transition Structure in HAVEN-T2RF . . . 226
C.3 DeepGC results in MIMIC-IV-ED . . . 226

List of Figures

1.1 Schematic visualization of the thesis structure 11

2.1 Top-down structure of Chapter 2 17
2.2 Temporal phenotypic clustering framework 40

3.1 Overview of data processing for HAVEN 48
3.2 Processing steps for TPC 52
3.3 Mean vital-sign trajectories in HAVEN by outcome class 56

4.1 Sketch of TSKM clustering and centroid estimation 74
4.2 Sketch of SOM-VAE Architecture 78
4.3 Sketch of AC-TPC Architecture 82
4.4 Mean respiratory rate trajectories per cluster 97
4.5 Cluster outcome characterizations (HAVEN) 99

5.1 Architecture of the CAMELOT model 105
5.2 Overview of the AttentionRNNEncoder layer 108
5.3 Cluster outcome distributions of CAMELOT and AC-TPC 124
5.4 CAMELOT feature-time relevance maps 127

6.1 Architecture of the DeepGC model 140
6.2 Overview of the Cluster-VRNN network 143
6.3 DeepGC cluster assignment trajectories (HAVEN-T2RF) 159
6.4 Generated trajectories from DeepGC (HAVEN-T2RF) 163

B.1 Mean RR and SpO_2 trajectories of CAMELOT clusters 220
B.2 CAMELOT feature-time relevance maps (MIMIC-IV-ED) 222

C.1 Sankey diagram of DeepGC cluster transitions (HAVEN-T2RF) . . 226
C.2 DeepGC cluster assignment trajectories (MIMIC-IV-ED) 227
C.3 Sankey diagram of DeepGC cluster transitions (MIMIC-IV-ED) . . 228

List of Tables

3.1 Laboratory test variables included in HAVEN 46
3.2 Comparative summary of HAVEN and MIMIC-IV-ED 51

4.1 Temporal aspect performance of AC-TPC$_w$ 95
4.2 Outcome aspect performance of AC-TPC$_w$ 98

5.1 Temporal aspect performance of CAMELOT 121
5.2 Outcome aspect performance of CAMELOT 123
5.3 CAMELOT ablation results 128

6.1 Temporal aspect performance of DeepGC 155
6.2 Outcome aspect performance of DeepGC 157
6.3 Forward generation horizon sensitivity 161
6.4 Outcome performance under a restricted input window 162

A.1 Hyperparameter grids for models in Chapter 4 212
A.2 Hyperparameter grids for models in Chapter 5 214
A.3 Hyperparameter grids for models in Chapter 6 215

B.1 Empirical outcome counts of CAMELOT clusters 218
B.2 Description of temporal aspect features in CAMELOT clusters . . . 219
B.3 Empirical outcome counts of CAMELOT clusters (MIMIC-IV-ED) 221

C.1 Empirical outcome counts of DeepGC clusters (MIMIC-IV-ED) . . 227

Mathematical Notation

x, y	scalars
$\mathbf{x}, \mathbf{x}^j, \mathbf{y}$	vectors or column-vectors if applicable
$\mathbf{X}, \mathbf{Y}$	matrices
x_i, x_i^j	$i-$th entry of vector $\mathbf{x}, \mathbf{x}^j$
$\mathbf{x}_{\leq i}, \mathbf{x}_{<i}$	subset of vector $\mathbf{x}$ up to entry i (inclusive, exclusive)
$X_{i,j}$	i, j entry of matrix $\mathbf{X}$
$x_{a,i}$	i-th entry of vector $\mathbf{x}_a$
$\mathbf{X}_{i,:}, \mathbf{X}_{:,j}$	i-th row, j-th column vectors of matrix $\mathbf{X}$
$\|\mathbf{x}\|_p$	L_p norm of $\mathbf{x}$. If unspecified, defaults to $p = 2$.
$\mathbb{R}$	Set of real numbers
$\mathbb{U}$	Unit interval $[0, 1]$
$\mathbb{B}$	Binary set $\{0, 1\}$
$[n]$	Set $\{1, \cdots, n\}$
$\#S$	Cardinality of set S
Δ_{K-1}	$K-1$-th dimensional simplex
$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$	Multivariate normal distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$.
$\text{Dir}(\boldsymbol{\alpha})$	Dirichlet distribution with parameter $\boldsymbol{\alpha}$
$\mathcal{U}[a, b]$	Uniform distribution in the interval $[a, b]$
$\Gamma_{\text{d}}(\alpha, \beta)$	Gamma distribution with shape α and rate β
$D_{\text{KL}}(P\|Q)$	Kullback-Leibler divergence between distributions P and Q
$\text{Softmax}(\mathbf{x})$	Softmax function applied to vector $\mathbf{x}$
$\sigma(\mathbf{x})$	Sigmoid function applied element-wise to vector $\mathbf{x}$.
$\text{ReLU}(\mathbf{x})$	Rectified Linear Unit function applied element-wise to vector $\mathbf{x}$.
$\mathcal{L}_{\text{BinCE}}$	Binary Cross-Entropy Loss Function
$\mathcal{L}_{\text{CatCE}}$	Categorical Cross-Entropy Loss Function
$\mathbf{e}_c$	Standard basis vector with 1 in position c
$a \coloneqq b$	a is defined as b.

Glossary

HR	Heart Rate
RR	Respiratory Rate
DBP	Diastolic Blood Pressure
SBP	Systolic Blood Pressure
TEMP	Temperature
FiO_2	Fraction of Inspired Oxygen
AVPU	Alert, Verbal, Pain, and Unresponsive Scale
SpO_2	Saturation of Peripheral Oxygen
ED	Emergency Department
ICU	Intensive Care Unit
TPC	Temporal Phenotypic Clustering
MCS	Multiclass Classification Setting
DL	Deep Learning
NN	Neural Network
MLP	Multi-Layer Perceptron
RNN	Recurrent Neural Network
LSTM	Long Short-Term Memory

There are neither beginnings nor endings
to the turning of the Wheel of Time.
But it was a beginning.

— Robert Jordan, The Eye of the World

1 Introduction

The growing availability of Electronic Health Record (EHR) data in healthcare systems worldwide provides new opportunities for the application of computational methods to longstanding challenges in the clinical domain. Indeed, EHR systems contain high-dimensional longitudinal data, capturing patients' different physiological presentations and responses to care of each patient, a setting in which computational methods and, in particular, machine learning models, are particularly well-suited. Nevertheless, despite advances in modeling approaches and the increased availability of EHR data, obtaining meaningful clinical insight is not guaranteed. Clinical data are heterogeneous and recorded at irregular and clinically-driven intervals. Furthermore, the data structure is derived from workflow and documentation practices, rather than research design, which introduces artifacts that make the data less robust to analyze. Therefore, in order for computational techniques to assist in improving healthcare delivery, careful experimental design is needed across all stages, e.g., data collection, data processing, model design, evaluation, and post hoc analyses.

This thesis is concerned with one particular challenge, *Temporal Phenotypic Clustering* (TPC), which uses the collected longitudinal physiological data, as well as future adverse events. The goal of TPC is the discovery and characterization of subgroups of the patient cohort that exhibit similar behavior within

groups and dissimilar between groups, and this chapter introduces this topic as follows. Section 1.1 establishes the clinical context and why TPC is not only a natural problem to consider, but also has the potential to improve the delivery of healthcare. Section 1.2 provides a high-level overview of TPC within a computational framework, translating between clinical ideas and mathematical concepts. Section 1.3 discusses the current progress in the TPC task and the existing limitations that remain to be addressed, some of which are addressed by the research in this thesis, which presents and situates the contributions of the thesis to the scientific literature. The datasets, clinical settings, and scope of this research are discussed in Section 1.4. Finally, Section 1.5 offers a high-level description of the thesis and the contribution of each chapter.

1.1 Motivation

In clinical contexts, the notion of *phenotypes* arises naturally in a wide range of settings [1, 2], disease areas [3–6], and clinical disciplines [7, 8]. In general, a phenotype denotes a group of patients who share similar patterns across different clinically relevant properties, such as physiological characteristics, disease mechanisms, and responses to treatment. A formal definition is deferred to Section 2.4.2. A well-known example is found in Chronic Obstructive Pulmonary Disease (COPD), which exhibits a "frequent exacerbator" clinical phenotype that is characterized by repeated exacerbations, a higher burden of symptoms, and worse long-term outcomes, which require specific management strategies and, often, targeted therapies [9]. Although phenotypes do not always represent categories that are formally defined, they function as a natural organizing principle in clinical practice, guiding expectations and informing the considerations relevant to determining the most appropriate intervention or treatment plan for each patient.

Specifically, phenotypes are often longitudinal in nature [10, 11], so differences between subgroups are not always apparent from a snapshot assessment of a patient at a single time point [12]. Patients who share a similar diagnosis

at the beginning of a hospital admission may follow markedly different trajectories over the course of hours or days: from those who stabilize rapidly and are discharged without complications to those who deteriorate progressively and, in the most severe cases, experience life-threatening adverse events. Phenotypic differences are therefore manifested in the temporal properties of patient trajectories, including the rate and shape of physiological evolution, as well as the variability of measurements over time, rather than in their values at any single observation [13, 14]. Phenotype discovery should accordingly account for this temporal dimension, despite the additional challenge that it introduces [2, 15].

The presence of clinical heterogeneity hinders the identification of phenotypes in practice [2, 16]. Heterogeneity manifests itself in several ways, including baseline patient characteristics, disease progression patterns, and responses to treatment. A common consequence is that clinical protocols establish a course of action based on population-level averages, applying uniform treatment pathways to patient groups that may, in practice, comprise several physiologically distinct subpopulations [17, 18]. Such protocols do not account for the heterogeneity present within these groups and, in doing so, are unable to adapt care delivery to the needs of individual patients. The clinical consequences can be significant: patients whose deterioration patterns deviate from the assumed population average are at elevated risk of delayed intervention, inappropriate treatment allocation, and in-hospital mortality [19].

The identification of clinically meaningful phenotypes at scale poses severe challenges that current tools and practices are not well-suited to address. Manual clinical assessment, while essential and representing the gold standard for individual patient evaluation, is inherently limited in its capacity to integrate the volume and complexity of data generated during a hospital admission, and is further constrained by the number of patients it can meaningfully impact. Several existing approaches identify phenotypes by using rule-based clinical tools, providing an aggregate summary of patient information, such as early warning scores [20]. Such approaches discard the temporal structure and, therefore,

are unable to capture the trajectory-level differences that define temporal phenotypes [21]. Although clinically useful as aggregate risk summaries, they are insufficiently expressive for the purposes of phenotype identification and characterization at scale. More broadly, the identification of phenotypes requires the development of models capable of operating in large and heterogeneous cohorts; however, existing approaches are typically validated in narrow settings, limited to specific diseases, data sources, or patient populations, and generally do not generalize easily [2].

To address the aforementioned gaps, EHR data provide, in principle, an extensive and underexploited resource. EHR systems routinely capture the most informative longitudinal physiological measurements for the characterization of phenotypes, including vital-sign trajectories and laboratory test results, as well as records of adverse clinical events and outcomes. Together, the longitudinal physiological record and the record of downstream adverse outcomes constitute data that can be used to identify phenotypic subgroups that are clinically coherent and relevant to outcomes. The growing scale and accessibility of EHR data, combined with advances in computational methods, therefore create an opportunity to develop approaches capable of extracting such phenotypic structure from routinely collected data, an opportunity that this thesis directly pursues.

1.2 The Computational Problem

The clinical rationale behind the identification of phenotypes motivates the use of computational methods capable of operating at the scale of large patient cohorts and also capable of integrating multiple sources of clinical information without requiring pre-specified labels for the subgroup structure to be discovered. Within machine learning frameworks, clustering constitutes the natural paradigm for this purpose. Unlike supervised learning, clustering does not require labeled examples of target categories or values; instead, it focuses on the identification of groups of observations according to patterns of similarity [22, 23]. This matches the phenotype discovery setting: the subgroup structure of a patient population

is not known in advance, and the objective is the discovery and characterization of phenotypes (subgroups) from the data. In healthcare settings, clustering methods have been widely applied for this purpose and provide a foundational framework for methods used to identify phenotypes [16, 24].

Nevertheless, the direct application of standard clustering methods to longitudinal EHR data faces fundamental challenges that limit their general utility in clinical settings. First, patient trajectories are represented as sequences of observations indexed by time, rather than fixed-dimensional feature vectors (which standard models expect), and a significant amount of clinically meaningful structure is present in the temporal dynamics rather than in the observation values at a single time point [25], so that standard clustering methods are not equipped to handle this structure directly. Second, EHR data include various sources of clinical information of fundamentally different types and nature, including physiological time series of observations, outcome records, adverse events, and patient demographics. Each source carries a distinct clinical meaning and may not be treated interchangeably in the input dimensions without the loss of important clinical information. Third, any clinical deployment of clustering methodologies requires the interpretability of cluster assignments and clusters themselves. Although the meaning of interpretability is varied, it is not sufficient to assign a patient to a phenotype without providing clinicians with suitable explanations for the assignment and what the phenotype represents [26]. These three challenges motivate the development of specialized methods for TPC in healthcare settings.

This thesis adopts a particular TPC framework as the formal computational setting within which these challenges are addressed by the models in subsequent chapters. Two distinct data sources (or *components*, which are introduced in more detail in Section 2.4.2) are considered as part of the phenotype discovery process. The first includes longitudinal records of physiological variables, such as measurements of vital signs, laboratory test results, and other clinical observations collected over the course of a hospital admission. The second source relates to

outcome information, which concerns the occurrence or absence of adverse clinical events, such as unplanned escalations to intensive care units or in-hospital mortality. A key property of the TPC framework is that the two sources are not symmetrically available: physiological measurements are observed throughout the admission, so they are available at the time of inference, whereas outcome records are determined from future events that have not yet taken place, and therefore are not available at the time of inference. This asymmetry is a critical component of what distinguishes TPC from other alternatives such as multiview or multitask learning formulations, where all sources of information are assumed to be simultaneously available, and, more generally, are symmetrically interchangeable with respect to model design (particularly with respect to inference time availability) [27]. Consequently, this property imposes specific requirements on model design in order to fully leverage all available data.

A natural question is whether the two sources must be modeled jointly or whether it is sufficient to cluster patients using only their physiological trajectory data alone and, at a later stage, evaluate the resulting clusters against information pertaining to the outcome data separately. This approach has been adopted in a substantial portion of the existing literature [24]; however, it carries an important limitation: clusters that are coherent with respect to trajectory patterns need not correspond to clusters that are also coherent with respect to their outcome profiles [28]. Joint modeling leverages both outcome information and physiological information to inform cluster formation, encouraging the discovery of phenotypes that are simultaneously coherent in both sources — a property that is essential for the clinical utility of the resulting phenotypes [24, 29].

TPC is also different from purely supervised outcome prediction, though they share some commonalities. In a supervised learning framework, models are trained to predict adverse outcomes from physiological trajectories, optimizing predictive accuracy but not identifying the underlying subgroup structure of the patient population. The clustering objective, on the contrary, pools information

between clinically similar patients and produces phenotypes that can be analyzed, described, and acted upon by clinical staff, a property that a prediction model does not provide. Furthermore, outcome information serves as a supervisory signal to guide the discovery of phenotypic structure, rather than as a prediction target. This distinction motivates the methodological approach of this thesis and the formal definition of the TPC problem as a specific problem within the clustering paradigm.

1.3 Limitations of Existing Work

The clinical and computational arguments of Sections 1.1 and 1.2 establish TPC as a well-motivated paradigm for phenotype discovery from temporal EHR data. Recent work has demonstrated the feasibility of applying outcome-based clustering methods to clinical EHR data, recovering meaningful patient subgroups in particular experimental settings [13, 24, 29]. Nevertheless, several specific limitations remain unaddressed in the current literature, which restrict the generalizability of such TPC approaches and their application to clinical settings. The discussion here is focused on the gaps most directly relevant to the contributions of this thesis; a more in-depth overview is provided in Chapter 2.

A fundamental limitation of existing research in TPC is the absence of a formalized computational setting. Different authors adopt inconsistent notions of the TPC problem, varying the characterization and definition of the data components, and employing different processing schemas [2, 30]. Specifically, there is no standardized evaluation framework for determining the quality of learned phenotypes in all different components in a joint manner. The absence of a shared formal setting carries two critical consequences: experimental reproducibility is limited, and the transfer or comparison of different models across distinct clinical settings involves a highly complex process that does not scale as the number of settings increases. These issues motivate the development of a unified TPC

framework that includes data standardization and a consistent evaluation procedure that can be applied consistently across datasets, clinical settings, and the components considered.

Although validated within their respective experimental frameworks, most of the models proposed in the literature have not yet been validated in clinically realistic settings. Most existing methods are developed and validated under controlled experimental conditions that do not fully reflect the complexity and heterogeneity of real-world clinical cohorts [2, 24, 31]. This provides limited confidence in their scalability to clinical practice. Several practical challenges that arise in such settings have therefore gone largely unaddressed. A prominent example includes the treatment of the outcome class distributions when they exhibit severe imbalance (where one or more classes occur at a significantly lower frequency than the other(s)), a regime that is ubiquitous in healthcare settings [32]. Existing TPC methods were not designed with this regime as primary focus and typically do not explicitly address the severe imbalance in the presence of multiple categories (at least three) within the outcome component, limiting their utility for patients with the highest clinical risk. In this thesis, these limitations are empirically examined in more detail using large-scale EHR cohorts with realistic, imbalanced multiclass outcome structures.

Separately, the interpretability of the learned cluster assignments in the current methods remains limited. Here, interpretability denotes the understanding of what motivates the assignment of patients to clusters and what each cluster phenotype represents. Where interpretability is addressed, it is typically done by applying post hoc explanation methods after model training, which have known theoretical and practical limitations and do not improve the model's capacity to learn clinically meaningful phenotypes [26, 33, 34]. Moreover, no existing TPC method, to the best of the author's knowledge, provides interpretability across both the temporal and physiological variable dimensions simultaneously, a capability that would be useful for understanding the mapping of patients to different phenotypes. Without such interpretability components, particularly in

the context of deep learning models, clinical adoption is unlikely to succeed as clinicians are unable to reconcile the computational assignments with their own constructed assessment of the patient and recommended course of action.

A fourth limitation of the existing body of research pertains to the assignment of a single static phenotype to each patient admission, encoding no information about how cluster membership evolves over the course of a hospital admission. This formulation is convenient for retrospective cohort analysis, but does not reflect clinical practice as it is incapable of dynamically modeling patient trajectories — a patient's physiological state and phenotypic membership may change substantially over the course of hours or days. From a dynamic point of view, existing methods are also unable to provide a mechanism to predict or estimate future physiological trajectories, a necessary precondition for prospective clinical monitoring that could be used for early intervention planning [13, 24, 29]. In clinical practice, observations are continuously accumulated throughout admission, and it is important for TPC models to dynamically update their representations to account for this updated information.

Overall, the identified limitations above are not independent. Collectively, they reflect a wide gap between the current state of TPC research and the requirements of clinical deployment to real-world settings. Addressing these limitations in a principled and unified manner constitutes the central aim of this thesis, as detailed in the following section.

1.4 Datasets and Research Scope

The empirical contributions of this thesis are evaluated on two real-world EHR datasets derived from distinct clinical environments. The first is a proprietary dataset that consists of retrospective hospital admissions collected across four hospitals within the Oxford University Hospitals NHS Foundation Trust in the UK [35, 36], denoted as HAVEN. HAVEN constitutes a rich data source for the TPC setting due to the extensive acquisition of vital-sign measurements and other

physiological variables recorded during each patient's hospital admission. In addition, HAVEN contains information related to the occurrence of adverse clinical events, such as unplanned escalation to intensive care or in-hospital mortality, which are of direct clinical interest and can naturally be encoded within the TPC framework. In addition, a subcohort of HAVEN, determined by those patients with or at risk of type II respiratory failure, is considered in subsequent chapters due to the existence of well-characterized respiratory phenotypes in this population. The second dataset is MIMIC-IV-ED, a publicly available dataset comprising data from emergency department admissions to the Beth Israel Deaconess Medical Center in Boston, Massachusetts, United States [37]. MIMIC-IV-ED captures a distinct patient population and clinical workflow relative to HAVEN, with variables primarily consisting of vital-sign measurements and demographic information from the patient. For MIMIC-IV-ED, the outcome component is defined based on the patient disposition following emergency department discharge, including hospital discharge and admission to a non-intensive care ward, resulting in a multiclass outcome structure in both empirical settings.

The two datasets were selected to provide evidence of the generalizability of the methodology developed in this thesis across heterogeneous clinical environments. HAVEN and MIMIC-IV-ED differ along multiple dimensions relevant to TPC: clinical setting (general ward *versus* emergency department), population cohorts (based in the UK *versus* in the US), data collection practices and definition and prevalence of outcome classes. HAVEN provides a richer set of physiological variables and a broader clinical cohort than MIMIC-IV-ED; accordingly, it serves as the primary dataset for developing the TPC framework and the analysis of the clinical relevance of its outputs. Nevertheless, the inclusion of MIMIC-IV-ED is important as evidence of generalizability to distinct clinical settings and, as a publicly available dataset, it additionally supports the reproducibility of the experiments conducted. Finally, it is important to note that all experiments are retrospective, providing proof-of-concept demonstrations of the feasibility and utility of the proposed TPC methods in real-world EHR data.

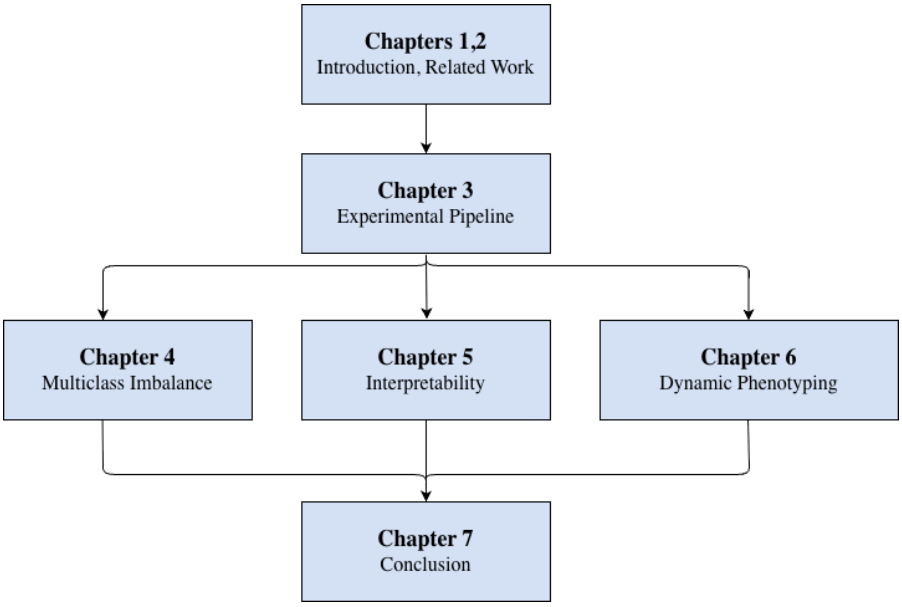


Figure 1.1: Schematic visualization of the thesis structure. Each chapter develops a different component of the TPC framework and its empirical evaluation.

1.5 Thesis Layout

A schematic visualization of the thesis structure is shown in Figure 1.1. Each of the following chapters develops a different component of the TPC framework and its empirical evaluation, as follows:

Chapter 2 reviews existing clustering methodologies relevant to TPC using a top-down approach. First, it introduces core concepts in clustering, including different clustering paradigms and common algorithmic families. Second, it discusses models designed for clustering univariate and multivariate temporal data, highlighting challenges such as irregular sampling, missingness, and high dimensionality. Finally, it surveys applications of clustering to healthcare data, with a particular focus on TPC, and synthesizes their limitations to motivate the contributions of the thesis.

Chapter 3 describes the experimental framework used throughout the thesis. First, the chapter introduces the two datasets considered in this work, HAVEN and MIMIC-IV-ED, and then presents a general data processing framework for

standardizing raw EHR data into model-ready inputs. Finally, it introduces the quantitative evaluation procedure used to assess TPC model-based development in both temporal and outcome data components.

Chapter 4 investigates the behavior of existing TPC models and related clustering benchmarks in settings characterized by severely imbalanced multiclass outcome data. Based on these findings, the chapter proposes an adapted model [38] that incorporates class-frequency weighting and a modified loss function to improve phenotypic diversity and stratification of minority classes. Empirical evaluation demonstrates improved performance on outcome-related metrics relative to established benchmarks, while maintaining competitive performance on temporal metrics.

Chapter 5 introduces a novel model for TPC, CAMELOT [39], which learns feature-time relevance maps as part of its internal representation, providing intrinsic interpretability with respect to phenotype assignments. Empirical results in both HAVEN and MIMIC-IV-ED demonstrate improved performance against established benchmarks, and ablation studies are used to isolate the contribution of each component. Finally, a qualitative analysis of the learned relevance maps and phenotypes is conducted.

Chapter 6 extends the TPC framework to a dynamic phenotypic setting and introduces DeepGC [40], a probabilistic generative model with phenotypic assignments that evolve over time. Empirical evaluations on HAVEN and MIMIC-IV-ED cohorts exhibit competitive performance in the temporal and outcome components. Furthermore, post hoc analyses demonstrate that the generative component is beneficial in settings with more limited input data and enables the synthesis of plausible future observations, which can provide meaningful insights to clinicians.

Chapter 7 concludes the thesis by summarizing the methodological and empirical findings of the research. The chapter also discusses the clinical implications and practical considerations of the applications of TPC methods in real-world EHR settings. Finally, it outlines the main limitations of the present work and proposes directions for future research in TPC and its clinical translation.

1.5.1 Publications

Portions of the work presented in this thesis have been published in the following peer-reviewed venues, matching, respectively, with Chapters 4 to 6:

- H. Aguiar et al. 'Phenotyping Clusters of Patient Trajectories Suffering from Chronic Complex Disease'. In: *Proceedings of the Machine Learning for Health (ML4H) Workshop at NeurIPS 2020*. 2020
- H. Aguiar et al. 'Learning of Cluster-based Feature Importance for Electronic Health Record Time-series'. In: *Proceedings of the 39th International Conference on Machine Learning*. Ed. by K. Chaudhuri et al. Vol. 162. Proceedings of Machine Learning Research. PMLR, July 2022, pp. 161–179
- H. Aguiar et al. 'Variational Modelling of Temporal EHRs for Phenotypic Clustering'. In: *ICLR 2023 Workshop on Time Series Representation Learning for Health*. 2023

The greatest part of a writer's time
is spent in reading, in order to write:
a man will turn over half a library
to make one book.

— Samuel Johnson,
as recorded by James Boswell (1791)

2 Related Work

2.1 Introduction

Healthcare systems around the world face the challenge of providing personalized care to heterogeneous patient populations, where treatment standardization often fails to account for individual patient variability. In particular, patients with identical clinical diagnoses and initial presentation may often exhibit vastly different clinical trajectories, treatment responses, and outcomes — a phenomenon seen in chronic conditions such as Chronic Obstructive Pulmonary Disease (COPD), in which patients exhibit distinct exacerbation patterns, response to bronchodilators, and varied progression rates [41–43]. The growing availability of Electronic Health Record (EHR) data [44, 45] presents opportunities to identify clinically meaningful patient subgroups through computational phenotyping[1] approaches that could inform personalized treatment protocols and reduce adverse patient outcomes [46–48].

Clustering techniques have emerged as a powerful initial approach to capture latent patient information and subgroup structures within datasets [49], or as a preliminary step towards downstream tasks, for instance K-$means$ and

[1] See Section 2.4.2 for a more detailed definition of phenotyping.

Spectral Clustering [50, 51]. However, the application of these traditional clustering methods to EHR data faces fundamental challenges such as irregular temporal sampling patterns, extreme class imbalances, missing data, and, critically, requirements for clinically interpretable results [52–54]. Current clustering approaches typically address some of these challenges in isolation, e.g., via methods that can handle complex data structures such as temporal irregularities and class imbalances [32, 55], or that have shown clinical relevance when applied to components of EHR data sources [56], such as temporal measurement observations [57]. However, significant gaps remain in providing an integrated solution necessary for real-world clinical implementation, particularly in the context of phenotype identification. Specifically, existing methods lack the ability to simultaneously handle temporal irregularity, leverage outcome information for phenotype discovery, and provide clinically interpretable cluster characterizations [24, 30, 31], an issue that this thesis aims to address.

This chapter will provide a review of existing clustering methodologies relevant to phenotypic clustering in healthcare, applying a top-down approach (see also the diagram in Figure 2.1). Section 2.2 introduces the clustering problem, including definitions of soft and hard clustering (Section 2.2.1), clustering objectives (Section 2.2.2), and general algorithmic families (Section 2.2.3). Section 2.3 explores methods for univariate (Section 2.3.1) and multivariate (Section 2.3.2) temporal data, discussing the particular challenges of each. Section 2.4 examines healthcare applications, from general clustering approaches to specific (Section 2.4.1), concluding with existing research in temporal phenotypic clustering (Section 2.4.2). Finally, Section 2.5 synthesizes limitations of the current literature and positions this thesis' contributions.

2.2 The Clustering Problem

Cluster Analysis is 'the formal study of methods and algorithms for grouping, or clustering, objects according to measured or perceived intrinsic characteristics or similarity' [22]. Clustering algorithms have a long history, dating as far back as

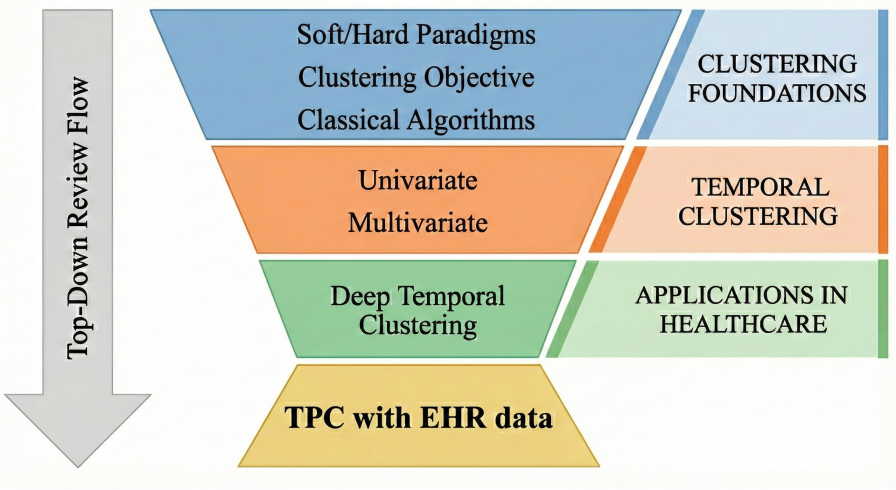


Figure 2.1: Top-down structure of Chapter 2. Clustering foundations and classical algorithms are first introduced (Section 2.2). The particularities of clustering temporal data are then discussed in Section 2.3. Finally, deep learning methods for temporal clustering in healthcare are presented in Section 2.4, and the chapter culminates with the formulation of the Temporal Phenotypic Clustering problem in Section 2.4.2.

the 1940s with the famous work of R. B. Cattell in the field of trait theory classification [58]. Despite extensive research in the field, clustering unfortunately has no universally agreed definition; indeed, there is not even consensus on the notion of *cluster* (the objects identified by clustering) [23]. However, most proposals exhibit a common approach to defining clusters [59], leading to a general-purpose definition that is followed in this thesis.

Definition 1 *A cluster is defined as a subset of data objects that share high mutual similarity according to a specific similarity measure. Clustering refers to the task of discovering a set of clusters in the data according to the target properties of interest.*

Individual clusters can be characterized by the data objects associated with them; alternatively, clustering algorithms may identify representative cluster objects during the learning process. Throughout this thesis, *clustering* may refer to either the process of identifying clusters or to the resulting set of learned clusters; the intended meaning will be clear from context.

A critical challenge in clustering is that of evaluation [60, 61]: determining the quality of learned clusters requires assessing intra-cluster cohesion, inter-cluster separation, and often no domain-specific criteria or external information

is present in the input data. In general, evaluation approaches can be broadly categorized into four groups of techniques [62]: (a) Internal Evaluation, which evaluates the quality of the cluster intrinsically from the data structure; (b) External Evaluation, which compares clusters against ground-truth labels; (c) Manual Evaluation, where domain experts validate the meaningfulness of the cluster; and (d) Indirect Evaluation, which measures cluster utility through downstream task performance.

Each approach has distinct advantages and limitations. External and indirect evaluation are particularly relevant when outcome information is available, although care must be taken to avoid conflating clustering objectives with downstream task metrics. In practice, a combination of these approaches is typically used to evaluate clustering performance. Due to its convoluted nature, Section 3.4 presents the evaluation framework used in this thesis in more detail.

2.2.1 Soft and Hard Clustering

There are many subfields within clustering that result from different interpretations of the definition introduced; however, not all are naturally suitable for the research question under discussion [59, 63, 64]. In this thesis, both *hard* [65] and *soft clustering* [66] tasks are considered (see below for formal definitions). Both paradigms offer distinct advantages within the context of phenotypic clustering for healthcare applications, specifically the ease of interpretation and the modeling of phenotypes as discrete or continuous subsets of a wider population [3, 67].

Consider a dataset of the form $\mathcal{X} = \{x_i\}_{i=1}^{N}$, where i is a generic index, N is the total number of samples. Then, hard clustering is concerned with identifying a partition of $\mathcal{X}$ into K disjoint subsets, also called clusters. Mathematically, clusters are written as $C_1, \ldots, C_K \subset \mathcal{X}$ and must satisfy:

(i) $\bigcup_{i=1}^{K} C_i = \mathcal{X}$ (coverage), and

(ii) $C_i \cap C_j = \emptyset$ for all $i \neq j$ (disjointness).

The assignment of samples to clusters is represented via the function $\phi : \mathcal{X} \to [K]$, where $[K] := \{1, 2, \ldots, K\}$, and $\phi(x_i) = k$ indicates that the sample x_i belongs to cluster C_k.

In general, the definition specifies that the clusters may not be empty ($C_i \neq \emptyset,\ \forall i \in [K]$). In practice, empty clusters often arise during iterative optimization when no samples are assigned to a particular cluster representation. This thesis adopts the convention that the number of learned clusters equals the number of non-empty subsets, while K denotes the target number of clusters specified as a hyperparameter.

The number of target clusters is one of the most important components of clustering and remains an active research challenge [68, 69]. Lower values of K can be associated with broad generalizations, while higher values of K can represent fine-grained oversegmentation. At an extreme, both the whole-set clustering ($K = 1, C_1 = \mathcal{X}$) and the single-sample clustering ($K = N, C_k = x_k$ for all $k \in [K]$) are optimal, if not perfect, in capturing trends within the learned clusters. However, this disregards the practical application of clustering for discovery, as clusterings are useless in practice.

For most of the proposed methods, K is a hyperparameter tuned according to the particular application and selected based on some criteria, such as the Gap Statistic [70] or the Elbow Method [71]. Researchers have proposed techniques capable of automatic learning of K [72], however, they introduce alternative hyperparameters to determine whether a higher or lower number of clusters is better. In general, the choice of K is nuanced and requires domain expertise for interpretation. Section 3.4 provides more clarity with respect to K selection.

Another paradigm, soft clustering [73, 74], introduces additional ambiguity. The task here is to identify the set of clusters $\{C_i\}_{i=1}^{K}$, as well as the membership function $P_{\text{soft}} : \mathcal{X} \to \mathbb{R}_{\geq 0}^{K}$, which indicates the degree to which each data point belongs to each cluster. Here, $\mathbb{R}_{\geq 0}^{K}$ represents the set of K-dimensional vectors with nonnegative real entries. If P satisfies the normalization constraint

of $\sum_{j=1}^{K} P_{\text{soft}}(x_i)_j = 1$ for all samples i, then P_{soft} can be interpreted as a probability function over the set of clusters, known as *probabilistic clustering* [75]. Alternatively, P_{soft} can be interpreted as a membership degree function, which indicates the degree to which each data point belongs to each cluster, known as *fuzzy clustering*.

Although there are subtle differences between the two paradigms, note that soft cluster assignment can be 'hardened' by assigning each data sample to the most probable cluster according to $\phi_{\text{soft}}(x_i) = \arg\max_{k \in [K]} P_{\text{soft}}(x_i)_k$. With this viewpoint, it is possible to define the set of cluster samples as the set of data points that belong to the cluster (after hardening in a soft setting). This is useful for defining cluster representation objects via summarization functions, for example, the cluster centroid. Alternatively, soft clustering researchers have proposed an alternative and more detailed characterization by weighting according to P [76]. For each cluster C_k, its representative object, $c_{\text{repr}}(k)$, can be defined as:

$$c_{\text{repr}}(k) = \begin{cases} \dfrac{1}{\#C_k} \displaystyle\sum_{x_i \in C_k} x_i, & \text{Hard clustering,} \\ \dfrac{\sum_{i=1}^{N} P_{\text{soft}}(x_i)_k \, x_i}{\sum_{i=1}^{N} P_{\text{soft}}(x_i)_k}, & \text{Soft clustering.} \end{cases} \tag{2.1}$$

In the context of a healthcare application, both hard and soft paradigms have shown promise in extracting clinically meaningful information [24, 63]. Soft clustering is particularly valuable in healthcare settings where patients may exhibit characteristics of multiple phenotypes simultaneously (e.g., overlapping comorbidities or transitional disease states). However, hard clustering provides simpler interpretation and is often preferred when clinicians require discrete assignment of phenotypes for treatment planning.

Other clustering paradigms exist, such as hierarchical clustering [77] and subspace clustering [78]. However, there are inherent limitations to the application of these methods in clinical settings. For example, hierarchical methods impose a nested cluster structure that does not align with the clinical utility of phenotypes as distinct, non-hierarchical patient subgroups. Similarly, subspace clustering's

focus on feature-specific groupings is less suitable when seeking holistic patient phenotypes that span multiple clinical domains.

2.2.2 The Clustering Objective

Clustering algorithms learn by optimizing an objective function that encodes the desired cluster properties for a given task, domain, and dataset [79]. In both soft and hard clustering, a common formulation seeks to maximize clustering quality, typically defined through two competing goals: (i) the maximization of intra-cluster cohesion (samples within a cluster should be similar), and (ii) the maximization of inter-cluster separation (samples in different clusters should be dissimilar) [80]. Most clustering objectives decompose this intuition into two components: a distance function (or similarity measure) that quantifies the relationships between individual data points, and an aggregation mechanism that combines pairwise distances into a global objective. For example, the $K\text{-}means$ objective minimizes the sum of squared Euclidean distances between points and their assigned cluster centroids [81]:

$$\mathcal{L}_{\text{k-means}} = \sum_{k=1}^{K} \sum_{x_i \in C_k} \left\| x_i - \mu_k^{\text{centroid}} \right\|_2^2 \tag{2.2}$$

where μ_k^{centroid} is the centroid of the cluster C_k, a learnable parameter included in the optimization process. Alternative objectives use different distance functions and aggregation strategies suited to specific data characteristics.

Distance Functions and Similarity Scores Distance functions and similarity scores are used to quantify the similarity between individual data points and clusters. Intrinsically, they exhibit an inverse relationship — smaller distances indicate higher similarity, and vice versa — and the transformation between distances and similarities can typically be performed via a suitable mapping function. Formally, distance functions must satisfy specific mathematical axioms (e.g., the triangle inequality), while similarity scores may not. For practical purposes, they are treated as inversely related: transformations such as $s(\cdot, \cdot) =$

$1/(1 + d(\cdot, \cdot))$ can be used to convert between them. Throughout this thesis, the terms distance function and similarity score are used interchangeably and the choice of the term is based on context and common usage.

The choice of distance function profoundly affects the clustering results [82] and represents a critical design decision. More generally, if $\mathbf{v}^1, \mathbf{v}^2$ are generic, real-valued vectors with the same dimension, the family of L_p (Minkowski) distances is defined as:

$$L_p(\mathbf{v}^1, \mathbf{v}^2) := \left\|\mathbf{v}^1 - \mathbf{v}^2\right\|_p = \left(\sum_i |v_i^1 - v_i^2|^p\right)^{1/p} \tag{2.3}$$

where $p = 1$ gives the Manhattan distance, $p = 2$ the Euclidean distance (most commonly used), and $p = \infty$ (defined as $\lim_{p\to\infty} L_p$) gives the Chebyshev distance. p controls sensitivity to large deviations: smaller p values emphasize differences between many dimensions, while larger p is more sensitive to the maximum absolute deviation between dimensions. Minkowski distances have been applied successfully in various clustering scenarios [83, 84]. Another commonly used measure is Cosine Similarity [85], defined as $s_{\cos}(\mathbf{v}^1, \mathbf{v}^2) := \frac{\mathbf{v}^1 \cdot \mathbf{v}^2}{\|\mathbf{v}^1\|_2 \|\mathbf{v}^2\|_2} = \frac{\sum_i v_i^1 v_i^2}{\|\mathbf{v}^1\|_2 \|\mathbf{v}^2\|_2}$ — where $\mathbf{v}^1 \cdot \mathbf{v}^2$ is the vector dot product between $\mathbf{v}^1$ and $\mathbf{v}^2$ — that measures the angle between vectors and is particularly effective for high-dimensional sparse data [86]. The corresponding Cosine Distance is defined as $d_{\cos}(\mathbf{v}^1, \mathbf{v}^2) = 1 - s_{\cos}(\mathbf{v}^1, \mathbf{v}^2)$.

The type of data lends itself to designing different classes of distance functions. An example is the Jaccard Similarity Index, introduced in Jaccard [87], which is commonly used with finite sample sets, particularly in binarized datasets [88]. Another family of similarity scores is derived from the Mahalanobis Distance [89], which can take into account unequal variances and correlations between features, as discussed in [90]. Similarly, Dynamic Time Warping [91], or DTW, for short, has shown promise in clustering by successfully modeling the relationship between temporal data objects by taking into account different temporal speeds [92]. It is discussed in more detail in Section 2.3.1.

Aggregation Functions Once pairwise distances are defined, clustering objectives aggregate these scores into a single optimization target. The choice of aggregation function determines which clustering properties are emphasized. Sum-based aggregation, used in K-$means$ and hierarchical clustering, minimizes the total within-cluster distance [77, 81]. Ratio-based aggregation balances intra- and inter-cluster distances, for example, the Calinski-Harabasz Index [93]. Extremal aggregation [94] considers worst-case scenarios (e.g., linkage-based hierarchical clustering), while *probabilistic aggregation* maximizes likelihood in mixture models [75]. Different aggregation strategies discover different cluster geometries [95].

In general, choosing the right distance and aggregation functions is challenging and requires careful consideration, particularly the distance definition. This is further compounded by data structures that are non-Euclidean. To tackle this, some researchers have proposed solutions that learn a distance function as part of the training process [96–98], using kernel methods, and usually under the assumption that data lie on a manifold. Alternatively, other researchers use deep learning models that learn to represent data in a latent space where clustering is easier [99–101]. This is largely the approach followed by the methods developed in this thesis due to its flexibility, although it requires a more adequate approach to interpret performance results as discussed in Ros, Riad and Guillaume [102].

2.2.3 General Families of Clustering Algorithms

Arguably, the most popular and one of the simplest hard clustering algorithms is the K-$means$ model [81]. As part of the algorithm, it learns cluster centroids — representative points that characterize each cluster, but do not need to correspond to actual data samples. The objective function is the Within-Cluster Sum of Squares (WCSS), which measures the total squared distance between the points and their assigned centroids. Global optimization of WCSS is intractable (it is an NP-hard problem [103]), so a practical iterative implementation, called Lloyd's algorithm [104], is commonly used. The method proceeds by initializing a set of

cluster centroids and iteratively alternating between two steps: assigning observations to the cluster with the closest centroid (measured by Euclidean distance) and updating centroids as an average of the data points assigned to the cluster. The algorithm stops once the WCSS has converged. Since WCSS decreases after each iteration, convergence is guaranteed; however, the convergence point might be a local optimum, rather than a global optimum [105]. The algorithmic complexity is linear as a function of the number of data points, clusters, and iterations, making it efficient for large datasets.

This method has been shown to be efficient [105] and requires only a few iterations to succeed when applied to data with a cluster structure [106]. It has been successfully deployed in a variety of areas [107–109]. However, it is known to have limitations related to distance function selection, initialization, sensitivity to outliers, and convergence to counter-intuitive results, particularly since it converges to local, rather than global, optima [110]. To address these issues, several variants of $K\text{-}means$ have been proposed, which include methods such as $K\text{-}medians$ [111], $K\text{-}medoids$ [112], $K\text{-}means{+}{+}$ [113], and Fuzzy $C\text{-}means$ [76], adapting initialization strategies, centroid assignment, centroid update, and aggregation functions.

Another family of common clustering methods are Mixture Models [75]. Instead of solving a discrete optimization problem, mixtures use probabilistic structures that model subpopulations, as well as membership to each subpopulation. Mixture model approaches represent data distributed according to a mixture distribution of K subgroups, each following the same distribution family (e.g., Gaussian) with different parameters. Furthermore, the models leverage a set of unseen, or latent, variables that estimate the membership to each component through soft assignment [75].

Maximizing the log-likelihood function is generally intractable, so optimization is typically done via an Expectation-Maximization algorithm (EM [114]). EM alternates between computing posterior membership probabilities (E-step) and updating distribution parameters (M-step), mirroring the iterative structure

of assignment update in K-$means$, but in a probabilistic framework. The most common application of mixture approaches to clustering is the Gaussian Mixture Model (GMM [115]), which uses the (multivariate) Gaussian distribution family to model subgroup data [116–118]. Although GMMs provide flexibility in cluster shape through the covariance matrix, they introduce increased optimization challenges compared to the spherical assumptions of K-$means$. Other alternatives have been suggested to improve model optimization [119, 120] and representation capability [121, 122], among others. In general, mixtures are more flexible than K-$means$, however, they are associated with more challenging optimization processes [123–125].

The choice between clustering families involves trade-offs between computational efficiency, interpretability, and modeling assumptions. K-$means$ excels in speed and simplicity but assumes well-separated spherical clusters. Mixture models provide probabilistic frameworks suitable for overlapping clusters, but require careful initialization and more computation. In healthcare applications, both have shown promise for the clustering of medical data, as can be seen in [16, 126–128]. However, both paradigms assume static, tabular data. Their adaptation to temporal and multivariate time-series settings requires specialized modifications, as discussed in Section 2.3.

2.3 Handling Temporal Data

The clustering methods introduced in Section 2.2 assume fixed-dimensional feature vectors where all measurements are observed simultaneously. Temporal data, sequences of observations indexed by time, violate these assumptions as they exhibit variable sequence lengths, irregular sampling, temporal dependencies, and information encoded in both shape and magnitude [129, 130]. In healthcare, temporal datasets occur ubiquitously in clinical settings [131], from continuous monitoring of vital signs to longitudinal tracking of disease progression, and

can provide unique clinical insights [132–134]. However, their distinctive properties — high dimensionality of multiple concurrent measurements, internal temporal dependencies (autocorrelation), irregular sampling patterns, and complex intermittent patterns — require specialized clustering methodologies [25].

A time series is any sequence of data points that are indexed and ordered by time indices, whereas a temporal dataset is any collection of data samples where each sample is represented by one or more time series. This thesis assumes that all samples share the same set of time-series features, although individual measurements may be missing. Time indices may differ for any pair of time series; in fact, even the length might be different. For a particular sample, *dimensionality* refers to the number of different time series associated with it. Each time series is also called a feature. If the total number of features is one, the data is said to be univariate; all others are denoted as multivariate.

A fundamental categorization of clustering methods for time-series data is related to the structure of output clusters. *Whole time-series clustering* targets the complete set of time-series information, so that each data sample must be assigned, via hard or soft assignment, to a single cluster. On the other hand, an algorithm can also cluster based on subsegments of time series (*subsequence clustering*), or individual points (*time point clustering* [135]).

This thesis focuses on whole time-series clustering, as the discovery of phenotypes requires the holistic characterization of patient trajectories rather than the identification of isolated temporal patterns or anomalies of individual measurements. As for subsequence and time point clustering, although they are valuable for motif discovery or anomaly detection [135], they do not naturally align with the clinical goal of stratifying patients into distinct phenotypic subgroups based on their complete temporal profiles. Sections 2.3.1 and 2.3.2 address univariate and multivariate temporal clustering, respectively, introducing similarity-based and representation-based techniques.

2.3.1 Univariate Data

Applying the clustering concepts of Section 2.2 to temporal data introduces challenges in extracting meaningful information from raw sequences [136]. This issue is particularly acute in healthcare settings [137], where irregular sampling, observation sparsity, and differences in scaling complicate the assessment of similarity. Research suggests three different types of approaches [25] — similarity-based approaches, representation-based methods, and model-based algorithms.

2.3.1.1 Similarity-Based Techniques

Similarity-based techniques directly model the raw temporal data (its 'shape') with suitable data preprocessing and definition of similarity function [25]. Time series exhibit unique properties that add to the difficulty in computing the similarity between two objects. For example, sensitivity to temporal shifts, variable lengths, and awareness of scaling and translation on the temporal and observation axes, as discussed in Kotsakos et al. [138]. The complexity cost of similarity computation is likewise more problematic than in the case of non-temporal data, particularly for longer time series.

Under the assumption of equal length, the L_p norms (particularly Euclidean and Manhattan distances) can be naturally extended to time series by summing over the temporal dimension. As one of the most commonly used similarity metrics, they have shown relevance in capturing similarity based on the proximity of two objects in time [139–141]. Although computationally efficient, these similarity measures show a lack of robustness to temporal misalignment, reducing their suitability in healthcare settings where patients progress at different rates or exhibit temporal changes in their trajectories [129].

Sakoe and Chiba [91] introduced DTW as a shape-based alternative with success in a variety of applications [142–144]. DTW calculates an optimal match between two time series by considering realignments of the time indexes of both time series (under certain restrictions), with optimality determined by the L_p norm of the realignment. In healthcare settings, this can be useful, as different

patients can (and do) progress at different speeds. Recent work has shown that DTW is sensitive to the choice of p, and suggested tuning of this parameter [145]. Unlike L_p measures, DTW does not require time series to be equal in length. DTW is more computationally expensive; however, it exhibits a near quadratic bound for sequences of length n [146, 147]. Faster techniques for approximating DTW have been proposed, such as PrunedDTW [148], which incorporates a pruning strategy in the realignment search space, and FastDTW and SoftDTW [149, 150], which efficiently approximate the objective of DTW. The latter is particularly useful due to its differentiability, which allows its deployment as part of models that learn via gradient updates. Methods like $K\text{-}means$ can naturally be extended to time-series data by leveraging the above metrics. TSKM [151] is a temporal extension of the $K\text{-}means$ algorithm that implements distances of L_p, DTW or SoftDTW.

Another commonly used metric is the Longest Common SubSequence distance (LCSS [152]). LCSS implements a realignment strategy similar to DTW; however, unlike DTW, LCSS does not focus on matching all individual observations and instead uses parameters ϵ_{LCSS} and δ_{LCSS}, which impose restrictions on the temporal and observation dimensions for allowable realignments, respectively. Vlachos, Kollios and Gunopulos [152] argue that the flexibility of LCSS makes it more amenable to outliers compared to DTW at the expense of sensitivity to threshold values, and LCSS has been applied to clustering of time series with some success [153–155].

Many other variants have been proposed for LCSS and DTW using a similar realignment approach, such as Edit Distances [130, 156]. Other researchers have proposed solutions based on different properties of time series, such as correlation [141, 157]. In general, extensive research has not shown conclusively that any metric performs optimally across different objectives and goals; however, DTW is commonly regarded a strong choice in most use cases [158–162].

2.3.1.2 Representation-Based Techniques

An alternative paradigm proposes mapping the input time-series data to a simpler space, also known as time-series representation, instead of directly computing similarity measures. After transformation, the represented data points can be analyzed more directly; in particular, for clustering purposes, these representations can be fed into more traditional clustering algorithms. This is also interpreted as feature extraction, where the complex information of time-series data is 'distilled' to a feature vector that captures the most relevant information associated with each object. Feature extraction can be performed manually [163, 164], or as part of the algorithmic learning process [165, 166]. The selection of a representation is challenging, as the overall performance of the clustering model is sensitive to the choice of the representation function, as argued in [167]. However, representation is also essential to reduce the size and dimensionality of the data and to remove noise.

One of the most popular families of representation techniques consists of Data-Adaptive Methods [130], where a given representation map is applied to all temporal objects in the data with the goal of minimizing a global reconstruction error. Examples include local methods such as Piecewise Polynomial Interpolation [168] and Adaptive Piecewise Constant Approximation [169], which construct piecewise approximations to temporal objects via local polynomial functions. Other methods employ global dimensionality reduction techniques. For example, Korn, Jagadish and Faloutsos [170] propose to use singular value decomposition to represent time-series data, while Lesch, Caille and Lowe [171] uses principal component analysis instead to conduct linear dimensionality reduction. Symbolic methods offer another approach, transforming time series into compact sequences of discrete symbols [172, 173]. Overall, the researchers in Aghabozorgi, Shirkhorshidi and Wah [25] argue that data-adaptive representations can provide good approximations; however, they lead to challenges in comparing different time-series objects due to the data-adapted transformation.

To address this limitation, non-data-adaptive approaches that use representation functions suitable for time series of equal length have been proposed [25]. Some of the more successful techniques leverage a discrete Fourier transform to transform a temporal object into a set of time-frequency characteristics [174, 175]. An alternative representation, based on wavelets and wavelet transforms, has seen particular success in representation, such as [176–178]. Unlike the Fourier transform, wavelets capture more localized information, rather than only global patterns. The main limitation of non-data-adaptive techniques is their dependence on prior specification of transformation functions, which may require extensive experimentation and domain knowledge. Non-data-adaptive methods offer consistency across datasets but require domain expertise to select appropriate transformations, while data-adaptive methods optimize for specific datasets at the cost of cross-dataset comparability.

Many representation approaches have been proposed, although it is important to mention that existing research suggests that no method outperforms all other representation techniques across most settings [179, 180]. With the advent of deep learning, researchers have begun exploring models based on neural networks to represent time-series data. Researchers argue that manual selection of features may be suboptimal and that time-series representations should be learned based on the particular data and application [181, 182]. DL methods allow researchers to extend the above paradigms to optimize representations based on properties other than a simple reconstruction error. DL models have shown particular success in clustering time series according to this approach, and multiple types of network have been developed to handle temporal data. A further description of models in this paradigm is provided in Section 2.3.2, due to the extensive number of proposed methods that have shown success in both univariate and multivariate data.

Finally, model-based algorithms employ a family of parametric methods to represent the input time-series data. The input data are transformed into distribution parameters that are the targets of the clustering algorithms. Traditional para-

metric time-series models (such as ARIMA [183] or standard Hidden Markov Models [184, 185]) are generally not well suited for phenotypic clustering due to scalability (with respect to the number of time series and the number of data samples) and sensitivity to noise [186, 187], so they are not considered in more detail. Modern probabilistic approaches using Deep Learning (DL) attempt to address these limitations and are discussed more extensively in Section 2.3.2 and Section 6.2.

Note that the representation and similarity techniques are components of a global clustering model. Thus, a number of the works below focus on the task of extracting information from the input time-series data, and, wherever possible, clustering applications of such techniques are mentioned. Finally, the usage of representation or similarity approaches need not be exclusive, and many researchers have proposed mixed algorithmic methods, combining a similarity definition with a type of representation, followed by a simpler clustering model. Examples include work by Abonyi et al. [188] and Ratanamahatana et al. [189], and Liu and Shao [190].

2.3.2 Multivariate Data

The challenges mentioned above that relate to time-series clustering are further compounded when considering multivariate time-series data. The increase in data size and dimensionality is linear in the number of time-series trajectories, which exacerbates computational difficulties. Moreover, each trajectory exhibits not only an internal correlation, but also a correlation with other trajectories for the same sample. Other challenges include potential misalignment of time indices, different data types, and different signal frequencies [25]. This heterogeneity has led researchers to mainly adopt deep learning methods as they possess the ability to learn the highly complex information extraction functions that are required [182]. Similarly to Section 2.3.1, part of the discussion below focuses

on the representation learning capabilities of DL models, as they form a critical component of clustering algorithms, with clustering use-cases mentioned wherever possible.

One of the most successful families of DL architectures for temporal data is Recurrent Neural Networks (RNNs Rumelhart, Hinton and Williams [191] and Sherstinsky [192]), models that use recurrent layers. At its core, an RNN layer maintains an internal state (represented as a vector), which intuitively summarizes information from the previous time steps. Time-series observations are processed sequentially, and the internal state is updated at each time step based on the current observation input and the previous state. This structure allows RNNs to capture multivariate temporal dependencies effectively, provided that the data have been preprocessed appropriately. Multiple researchers have shown success using RNNs for clustering tasks; for example, Trosten et al. [193] apply an RNN separately for each input time series and then add a divergence-based cluster loss function to identify clusters within the representation space. Alternatively, researchers in Tavakoli et al. [194] instead generate latent representations using an RNN model for multivariate financial data built with domain knowledge and consequently train a $K\text{-}means$ algorithm for cluster identification.

However, several limitations of standard RNNs have been identified, including difficulties in learning long-term dependencies due to vanishing or exploding gradients during training [195]. To address these issues, more specialized RNN architectures have been developed. Gated-Recurrent Units (GRUs [196, 197]) and Long Short-Term Memory networks (LSTMs [198]) introduce gating mechanisms that regulate information flow within the network (namely input, internal cell state, and output interactions), allowing better retention of long-term dependencies [192], as well as interpretability [199]. These architectures have shown success in various temporal tasks, including clustering, either by adding clustering loss functions, or as part of a downstream clustering objective [200–203].

Another popular family of DL architectures for temporal data are Convolutional Neural Networks (CNNs LeCun et al. [204]). Originally designed for imaging data, each CNN layer applies filters (also called kernels) over local temporal or spatial neighborhoods, followed by nonlinear activation functions. This carries two major advantages: CNNs naturally support hierarchically extracted local features and, compared with vanilla RNNs, are less prone to exploding or vanishing gradients during optimization. At the end of the CNN layers, the outputs are traditionally flattened to a wide vector, and a set of dense layers is used for the target task. Different approaches have been introduced for the representation of multivariate time-series inputs. One can leverage 1D-CNN layers along the temporal axis to learn feature representations that are subsequently clustered [181, 205], or use 2D convolutional filters that jointly operate across time and feature dimensions [182, 206]. Some researchers propose another fascinating approach, first transforming time-series data into 2D image representations, and then applying image-based networks for clustering [207]. Similarly to RNNs in their applications to clustering, convolutional architectures have shown some success, typically by learning latent representations with convolutional backbones and then applying clustering losses or downstream clustering algorithms on the learned embeddings [208].

Other DL methods have been proposed to represent time-series data, with the goal of learning representations suitable for clustering. Some authors propose probabilistic DL models, such as Variational AutoEncoders (VAEs [101]) for better embedding generation [209]. Other researchers combine VAEs with Self-Organizing Maps (SOMs [210]) to better handle temporal trajectories [211, 212], with the limitation that representation learning does not consider the entire temporal sequence, rather learning embeddings that characterize individual observation time points. More recently, Transformer architectures [213] have gained traction in time-series representation due to their proven success in natural language processing via the usage of attention blocks. Although most research focuses on

the learning of representations, some authors have proposed methods specifically designed to tackle multivariate temporal clustering. For example, Zerveas et al. [214] proposes a Transformer-based architecture for general unsupervised representation learning of time series that is suitable for a variety of downstream tasks. Alternatively, Liu et al. [215] introduce a graph clustering transformer with the target of forecasting. Although useful for unsupervised representation, transformer models are typically data-hungry, computationally expensive, and struggle with irregular sampling, which limits their applicability in healthcare settings where such patterns are common [216, 217].

Addressing the irregular properties of temporal data has also motivated specialized architectures. The simplest approaches are interpolation methods that model the relationship between time points for different variables [218, 219]. The authors in Che et al. [220] introduce GRU-D, a GRU network that incorporates trainable decay terms that model the time since the last observation. This enables the model to account for measurement sparsity and use that information to reduce or increase the contribution of the previous cell state to the current time index. Both approaches require parameterization of the relationship between time index observations. Neural Ordinary Differential Equations [221] provide continuous-time representations, naturally handling arbitrary sampling patterns. More recently, set-based approaches treat irregularly-sampled observations as sets rather than sequences [222], applying permutation-invariant architectures. These methods demonstrate varying trade-offs between modeling flexibility and computational efficiency, with RNN-based approaches generally preferred in resource-constrained clinical settings.

There is substantial evidence suggesting that the most suitable deep learning architecture is highly dependent on the specific task, the characteristics of the input data, and other practical factors such as the interpretability of the model output and the robustness to missingness [223, 224]. However, other researchers have shown, based on extensive empirical evaluation in their respective studies, that LSTMs perform well when applied to healthcare settings with temporal data

compared to alternative deep learning architectures [225–227], provided that appropriate design alterations are introduced. This suitability arises from an ability to handle irregular sampling, variable sequence lengths, and heterogeneous feature spaces — that are common in EHR data [225, 227].

2.4 Applications in the Healthcare Domain

Unlike previous sections, which focus on the methodological component of clustering methods together with challenges associated with particular data properties, this section provides an overview of how DL clustering methods are applied in the healthcare domain. Successful methodological design requires an understanding of the clinical context, practice, and its requirements, which, in turn, justifies modifications and design choices relative to the baseline or 'vanilla' methods [26, 228]. A particular focus is given to DL methodologies, due to their ability to learn representations from high-dimensional multivariate temporal data (see also Section 2.3.2) and their growing use in temporal phenotypic clustering, the central theme of this thesis.

In healthcare, clustering has been widely used for a range of patient- and disease-level tasks in various settings. For example, risk and outcome stratification methods aim to divide populations into subgroups with different prognoses or risks of complications, which is useful to support the monitoring and improvement of patient outcomes [229, 230]. For modeling of treatment response, clustering has been used to identify groups with similar trajectories under specific therapies or interventions, which provides information to inform more personalized treatment strategies [231, 232]. Phenotyping[2] and disease subtyping are naturally associated with clustering, where the goal is to discover patient subgroups that differ in underlying physiological characteristics, progression patterns, and long-term outcomes [24, 233, 234].

At the same time, applying clustering methods to healthcare raises specific challenges and requirements that differ from many generic time-series or tabular

[2] See Section 2.4.2 for a more precise definition.

domains. At the data level, healthcare data are typically high-dimensional, multimodal, and noisy. When considering time series, issues of sparsity, irregular sampling, and varying observation windows are exacerbated [216, 235]. Integrating static variables, temporal data, genomic data, imaging information, and unstructured text is another common challenge [26, 228]. EHR data, in particular, do not typically measure latent disease state or patient health status, rather capturing presentation and care processes [236]. Outside of data particularities, healthcare applications often require model properties such as interpretability, reproducibility in different cohorts or settings, and clinical validity [237]. Unfortunately, standard DL architectures, in their vanilla form, often struggle to provide these properties without additional constraints or post hoc explanation mechanisms [238]. These considerations motivate specialized architectural design choices and modifications to baseline clustering algorithms and are discussed in the following Section 2.4.1.

2.4.1 Deep Learning for Clustering Temporal Data

Section 2.4 provided a general overview of the challenges associated with the clustering of temporal healthcare data. This section examines the adaptations and design choices that DL-based clustering methods employ to address these challenges. Broadly, these adaptations can be categorized into two groups: (a) data-driven, and (b) clinically-driven. Note that the categories are not mutually exclusive, and many methods introduce components that fall into both groups.

2.4.1.1 Data-Driven Adaptations

Data-driven adaptations include research that attempts to model data properties directly. Ren et al. [239] propose a DL network with additional layers capable of temporal interpolation to handle irregular sampling of vital signs in the context of phenotype identification for acute diseases. The time-series data are first interpolated according to the interpolation approach proposed in Shukla and Marlin [240], and an AE architecture is used to generate embeddings for clustering in

a subsequent step [241]. On the other hand, Ruan et al. [242] directly modify a GRU-based architecture to take into account missing values and time decay to the last observation as part of an AE architecture, while Merkelbach et al. [243] directly encodes the time index of each data point as a positional encoding, so that GRU-AE can distinguish not only value patterns, but also their timing during stay in Intensive Care Units (ICUs). Latent representations are then clustered via further dimensionality reduction and a classical clustering algorithm in a subsequent step. In a different biomedical domain, a separate approach introduced a DL framework capable of modeling temporal data using ordinary differential equations in the context of longitudinal microbiome studies [244].

Other approaches have instead focused on learning 'cluster-friendly' latent representations, where the embedding space is explicitly shaped to facilitate clustering. The authors in Qiu et al. [55] introduce a transformer-based variational AE architecture that receives diagnosis sequences as input and estimates longitudinal survival data. The encoded embeddings are consequently clustered with a Gaussian mixture prior, with performance shown on synthetic data, as well as a large-scale longitudinal study dataset. Alternatively, Kok et al. [245] adapt deep embedded clustering to an AE architecture that handles mixed data types (continuous and categorical) with application in intensive care settings.

2.4.1.2 Clinically-Driven Adaptations

In contrast, clinically-driven approaches denote modifications that were introduced to DL models to help address specific healthcare challenges. Tsai et al. [246] propose NaviDCN, a clustering framework with an additional loss function ('navigator') that encourages groups to align with clinically significant sepsis phenotypes; the overall model is trained jointly with reconstruction, clustering and navigator losses. The authors validate this approach in the early intensive care cohort in the context of sepsis phenotyping during the first hours of ICU admission. Similarly, the temporal SOM and Hidden Markov structure in Manduchi et al. [212] was designed to support clinical interpretation: it provides a

2D map of cluster prototypes and an explicit state-transition structure, enabling clinicians to visualize typical patient trajectories and their transitions between phenotypic states.

Rather than focusing on intrinsic model design, other researchers have instead focused on other components of the experimental pipeline to help understand the outputs of DL models. One such example are post hoc interpretability techniques which can be applied after model training (or inference) to provide insights into learned clusters, representations, and assignments. Castela Forte et al. [247] use a DL clustering model to discover high-risk phenotypes in intensive care cohorts; consequently, they apply a SHAPley additive explanation (SHAP [34]) technique to identify contributions of different variables to cluster assignments. Zheng et al. [248] attempt to extend SHAP usage by incorporating it as part of the DL model training process, using SHAP outputs as labels passed to a prediction DL network. The authors in [239] instead describe the clusters clinically by averaging the clinical properties of interest over the set of patients associated with each cluster. This approach is simple; however, it is widely used and accepted.

Alternatively, given limitations in generalizability and practical deployment, there has been an increased emphasis on demonstrating that the results of the DL model are not artifacts of a single dataset or data cohort. Separate cohort validation, in which an independent separate test set is used for testing and evaluation, has been adopted in a variety of experimental settings. Different separation strategies have been proposed, depending on the availability of data and task; for example, temporal separation, where training and testing cohorts are derived from the same hospital at distinct time periods [239], or geographic separation, in which models are trained and tuned in different hospital data altogether [245]. Finally, complementary work has targeted changes to the evaluation process itself, proposing longitudinally aware metrics to better assess and quantify cluster quality [249].

2.4.2 Temporal Phenotypic Clustering with EHR data

Building on this, the present subsection focuses on the central problem discussed in this thesis: *Temporal Phenotypic Clustering* (*TPC*) approaches in EHRs. To facilitate discussion, the following terms are defined according to their usage in the thesis.

Definition 2 *An aspect denotes a fundamental, semantically coherent view or dimension of clinical data that is modeled separately to enable interpretable phenotyping. Aspects are classified as observable if available at inference time or as unobservable otherwise.*

The aspects decompose complex EHR data into manageable components for clustering and analysis. Example aspects include temporal clinical presentations, the longitudinal profile of a patient's physiological data, disease progression patterns, such as deterioration trajectories, patient response to different treatments or interventions, and outcome occurrences like death events, all of which are available within EHRs [24, 228, 250]. Furthermore, with respect to observability, the temporal profile of a patient's vital signs can be seen as an observable aspect, while the occurrence of future adverse events is an unobservable aspect. At inference time, the former is available, while the latter has not yet occurred. The distinction between observable and unobservable aspects is critical for model design as the different aspects are not symmetrically available for clustering, in the sense that, mathematically, symmetrically switching aspect dimensions would not be applicable to the same clinical problem [13].

Definition 3 (*Clinical*) *Phenotyping is the process of identifying and characterizing meaningful subgroups of patients from* (*clinical*) *data, such as EHRs, based on patterns/trends in their* (*clinical*) *characteristics. Identified subgroups are also denoted as phenotypes. When applied to a setting with at least one observable aspect of temporal or longitudinal data and a separate unobservable aspect, the problem is referred to as Temporal Phenotypic Clustering, or TPC, for short.*

In this thesis, clusters will also be used to denote phenotypes whenever the context is suitable, as the computational algorithms to learn phenotypes require clustering. In contrast to phenotyping in a genomic context, phenotyping focuses on clinical manifestations, as well as the interpretability of phenotypes. A descriptive diagram of a TPC model is provided in Figure 2.2.

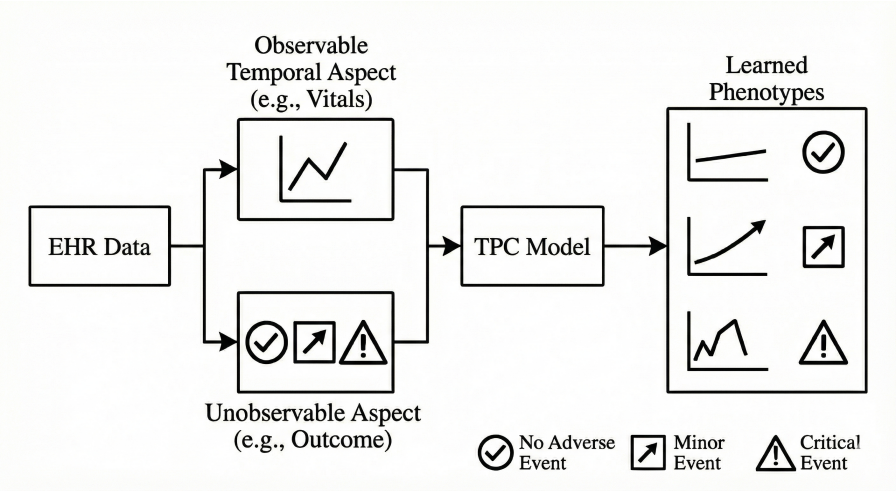


Figure 2.2: Schematic representation of the TPC framework for EHR data. EHR data are decomposed into an observable temporal aspect (longitudinal measurements available at inference time) and an unobservable outcome aspect (future adverse events). A TPC model jointly leverages both aspects to learn phenotypes characterized by temporal and outcome representations. The figure also highlights the asymmetry between the aspects: only observable temporal data are available for phenotype assignment at inference time. An example characterization of the outcome aspect is shown in the bottom-right corner.

Distinguishing the TPC paradigm from MultiView Clustering (MVC [27]) is important. MVC methods assume that each sample is described by multiple observed *views*, which are very similar to the aspect concept used in this thesis. The key difference is that they are all available for clustering at inference and are commonly treated symmetrically in the model objective. By contrast, in TPC settings, not all aspects are observed at inference or testing time, therefore, requiring more adequate handling of such aspects in model design. Applying MVC methods to TPC settings is possible by restricting the input only to observable aspects. However, this restricts the remaining aspects to be used primarily for post hoc evaluation [239] and not as part of the cluster identification. This distinction

highlights a potential advantage of TPC over MVC — TPC methods can leverage the full set of available data and use learned clusters to model the behavior of unobserved aspects, and that, potentially, can be translated into actionable information for clinical decision-making [13].

Most of the previously proposed methods for TPC in healthcare could not fully leverage unobserved aspects during cluster discovery, which were instead used for post hoc evaluation of cluster quality [24]. Wang, Sontag and Wang [251] develop a TPC method specifically for Parkinson's disease using a Markov Model of disease progression together with a component to handle an unobservable comorbidity aspect. Liu et al. [10] use a fundamentally similar approach through a temporal graph clustering model. Both are not DL models, however, are early examples of TPC problem modeling. More recently, Lee and Schaar [13] and Lee, Rashbass and Schaar [252] introduce a TPC method for EHR data with an event aspect based on the sequential occurrence of adverse events throughout patient admissions. The authors propose a DL architecture with two components: (a) an AE architecture to learn representations of the temporal aspect (in this case, the temporal evolution of vital signs), and (b) a prediction network to model the outcome aspect (longitudinal record of clinical events). Critically, both components are jointly trained with a clustering loss, which encourages the learned representations to be 'cluster-friendly' and also sensitive to both aspects. This was extended in Qin, Schaar and Lee [29] to improve the representation of the temporal observed aspect; the authors also propose a new evaluation metric (cluster consistency, or AUSIL) to facilitate comparison between the evaluations of the temporal and event aspects.

Existing TPC research in EHR settings still struggles to fully incorporate unobserved aspects during cluster discovery in a manner that fully unlocks its potential. Of particular note, the authors in Lee and Schaar [13], Qin, Schaar and Lee [29] and Lee, Rashbass and Schaar [252] introduce frameworks that address this issue; however, they mainly focus on technical feasibility and lack comprehensive practical validation in real-world cohorts. Specifically, they exhibit a lack

of (a) proven clinical utility and interpretability and (b) scalability to heterogeneous populations and data.

2.5 Conclusion

This chapter has provided a comprehensive introduction and review of clustering methodologies relevant to the problem of temporal phenotypic clustering using a top-down approach, particularly focused on deep learning clustering methods due to their ability to handle complex data structures and properties in EHR data. In particular, the latest research on TPC was discussed, with a particular focus on existing limitations of current methods. This thesis introduces research that expands on the aforementioned methods to add important components to healthcare applications, including a benchmark overview of TPC methods and their ability to handle severely imbalanced outcomes (Chapter 4), interpretability (Chapter 5), and the ability to forecast patient trajectories (Chapter 6).

Everything should be made as simple as possible,
but not simpler.

— Albert Einstein (attr. 1950)

3 Experimental Pipeline

3.1 Introduction

Chapter 2 introduced the TPC task and its associated challenges, and reviewed the existing literature on TPC methods. Given the inherent complexity of TPC problems, the establishment of a reproducible and scalable experimental framework is essential for a rigorous evaluation across clinical settings. This chapter describes the experimental pipeline followed throughout this thesis. First, Section 3.2 presents the two datasets used in this work: (a) HAVEN (Section 3.2.1) and (b) MIMIC-IV-ED (Section 3.2.2), each representing a distinct clinical context. Section 3.3 then introduces a general data processing framework for translating raw EHR data into structured inputs suitable for model training, including a set of transferable guidelines for handling complex EHR data. Finally, Section 3.4 describes the quantitative evaluation procedure used to assess the performance of the models in all data aspects.

3.2 Data

This thesis considers two datasets associated with distinct clinical settings. The primary dataset used for the empirical evaluation and analysis of all TPC models is HAVEN (Section 3.2.1), comprising hospital admissions collected from

multiple hospitals in the United Kingdom. Another dataset is also considered: MIMIC-IV-ED (Section 3.2.2), which covers admissions to Emergency Departments (EDs). A comparative summary of the two datasets is shown at the end of the section in Table 3.2. The inclusion of MIMIC-IV-ED provides evidence of generalizability, in the sense of assessing whether models suitable in a hospital population and a healthcare setting can also retain their performance in an alternative population or setting.

Before describing each dataset, key clinical terminology used throughout this chapter is defined as follows. *Patients* represent the individuals for whom data related to hospital visits were recorded. A hospital visit, defined by the contiguous period of time between hospital admission and discharge, is denoted as an *admission*. Thus, a single patient may have multiple admissions, but each admission belongs to exactly one patient. Each measurement recorded during admission is called an *observation*, e.g., a single recording of SpO_2 or heart rate. Each observation corresponds to a particular *variable*, or *feature*, which denotes the clinical property of interest that is being measured.

3.2.1 HAVEN

The Hospital Alerting via Electronic Noticeboard (HAVEN [35, 36]) is a multicenter collaborative project designed to develop clinical surveillance and Early Warning Score (EWS) systems for patients admitted to hospital care who may require treatment in an ICU. A retrospective cohort study of routinely collected EHR was conducted, including more than 200,000 adult (at least 16 years of age) hospital admissions to Oxford University Hospitals NHS Foundation Trust (OUH), a collection of four different hospitals within the Oxford region (UK), from 21 March 2014 to March 2018, denoted as HAVEN. The full scope of the HAVEN data is extremely large; therefore, this thesis considers a well-defined and clinically coherent subset, following the protocol from Pimentel et al. [19], with minor modifications. Firstly, the analysis was performed on the most recent extract of the HAVEN database, covering admissions up to March 2018. Only

data from non-ICU wards were collected; in addition, only admissions with a recorded discharge status were considered. Furthermore, admissions that satisfied (a) discharge from the hospital before midnight on the day of admission and (b) no vital-sign observations recorded 24 hours before the outcome time (defined below) were removed.

Chapter 4 uses the entire HAVEN cohort for TPC benchmarking purposes; however, subsequent analyses in Chapters 5 and 6 restrict to a clinically coherent subcohort of patients with (i) COPD-related Type II Respiratory Failure (T2RF) or (ii) at risk of T2RF. The complete clinical definition and criteria for this subcohort are presented in Pimentel et al. [19] and the corresponding data will be referred as HAVEN-T2RF for simplicity.

This restriction was motivated by clinical and methodological considerations. First, T2RF represents a well-defined high-risk inpatient population characterized by elevated rates of adverse outcomes, e.g., in-hospital mortality [19, 253]. The management of these patients poses challenges for standard EWS systems due to their atypical physiological baselines: they present chronically lower SpO_2 values, which often lead to false-positive alerts, with the negative effect of contributing to alarm fatigue in clinical practice. Worse, inappropriate oxygen therapy can instead induce hyperoxia and worsen hypercapnic failure [254]. Modified EWS systems have been proposed to ensure safer oxygen titration; however, evidence suggests that it does not lead to a quantitative improvement in the prediction of key outcomes [19]. This highlights the need for more expressive modeling approaches capable of capturing subtle patterns of deterioration in this population.

Second, from a methodological perspective, this restriction helps to mitigate cross-disease heterogeneity and potential confounding effects arising from multimorbidity [255], which can improve the identification of clinically meaningful phenotypes. Importantly, T2RF is characterized by substantial heterogeneity, with distinct exacerbation patterns and disease subtypes, although not yet fully recognized, influencing the prognosis and treatment response [255, 256].

This setting is therefore well-suited for TPC methods, and this thesis hypothesizes that such methods can identify previously unidentified subgroups of the T2RF population, with potential implications for the early detection of deterioration and improved clinical decision-making. *Vital signs* were recorded by clinical

Serum			
Variable	**Shorthand**	**Units**	**Indicative Range**
UREa	URE	mmol/L	3.7 – 23.6
ALBumin	ALB	g/L	18.0 – 32.0
CREatinine	CRE	μmol/L	48.0 – 289.0
SODium	SOD	mmol/L	133.0 – 142.0
POTassium	POT	mmol/L	3.5 – 5.0
C-Reactive Protein	CRP	mg/L	12.8 – 213.7
Hematological			
Variable	**Shorthand**	**Units**	**Indicative Range**
White Cell Count	WCC	10^9/L	7.1 – 17.3
HemoGloBin	HGB	g/L	8.8 – 13.2
EOSinophils	EOS	10^9/L	0.0 – 0.4
BASophils	BAS	10^9/L	0.0 – 0.1
NEUtrophils	NEU	10^9/L	4.8 – 14.6
LYMphocytes	LYM	10^9/L	0.5 – 2.0
Eosinophil-to-Basophil Ratio	EBR		0.0 – 7.6
Neutrophil-to-Lymphocyte Ratio	NLR		3.1 – 22.3

Table 3.1: Laboratory test variables included in the temporal aspect of HAVEN. Measurements are typically recorded once or twice per day. Variables are grouped into serum (top) or hematological (bottom) categories, with the corresponding shorthand, units, and indicative range. The final two hematological variables are derived and unitless.

staff every $2-6$ hours via the System for Electronic Notification Documentation (SEND [257]), including heart rate (HR), respiratory rate (RR), systolic blood pressure (SBP), diastolic blood pressure (DBP), temperature (TEMP), blood oxygen saturation (SpO_2), neurological status (via the derived 'Alert, Verbal, Pain, Unresponsive', or AVPU, score [258]), and the estimated fraction of inspired oxygen (FiO_2) for patients receiving supplemental oxygen. Vital signs constituted the main temporal features due to their higher frequency of recording and their well-established utility to capture the physiological status of patients and predict the occurrence of adverse events [259, 260].

Laboratory test results were also included and recorded once or twice per day for each patient. Two laboratory test measures were derived from existing meas-

urements rather than directly recorded: the eosinophil-to-basophil ratio and the neutrophil-to-lymphocyte ratio, where, if the denominator is zero, undetectable, or missing, the ratio is set to missing. A complete list of laboratory variables is shown in Table 3.1 and was selected in consultation with the collaborating clinical team, reflecting routinely available tests that are informative for understanding general ward deterioration and acute physiological status. For brevity, *vitals* will be used interchangeably with vital signs and *labs* with laboratory test measurements. The time of each observation was aggregated with an hourly granularity to allow for discrepancies and delays in measurement. Static variables of age and sex were also included, along with binary variables that indicated whether the type of admission was elective and/or surgical.

The temporal aspect comprised vital-sign and laboratory test trajectories, together with static variables concatenated at each time point and represented as constant time series to facilitate modeling. The outcome aspect was defined following an extension of the approach in Pimentel et al. [19]. *Adverse events* were defined as the occurrence of death (*Death*), cardiac arrest (*Cardiac*), or unanticipated admission to the ICU (*ICU*). The *outcome label* was defined as the first adverse event that occurred; for admissions without an adverse event, the outcome label was set as *NoEvent*. The outcome time of the first adverse event, in the former case, or the time of hospital discharge, in the latter, was used as *outcome time*. HAVEN thus constitutes a MultiClass Setting (MCS), in which each admission is assigned exactly one outcome label from a finite set of *classes* (four, in this case). Any observations recorded within the 24-hour window before the outcome time were excluded from the model inputs to prevent bias toward the future and to produce clusters that are predictive of the outcome, rather than concurrent with the outcome. This is consistent with existing research and clinical guidelines [19, 261], although it introduces a limitation that clinical staff, at any time point, can only interpret the output of the model with respect to events that could occur more than 24 hours in the future and not earlier. A summary description of the initial handling of the HAVEN data is shown in Figure 3.1.

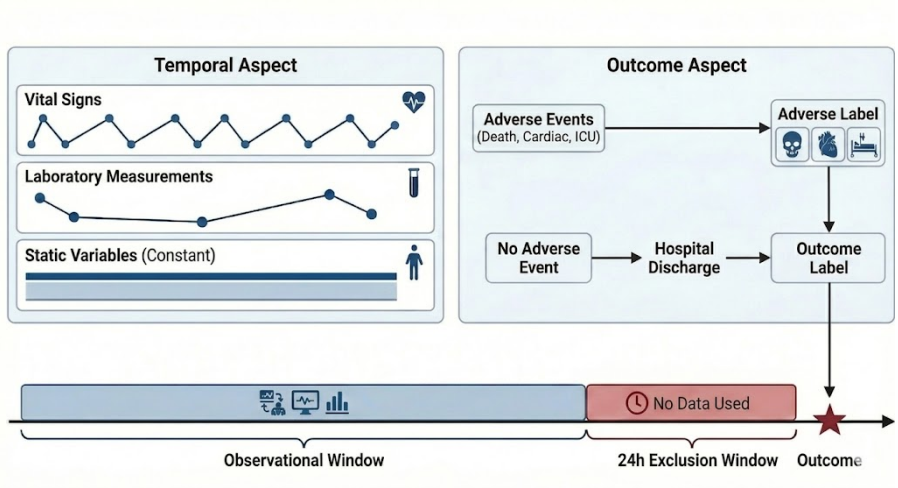


Figure 3.1: Overview of the initial processing steps for HAVEN, including the definition of the temporal and outcome aspects and the data sources used for each. The temporal aspect comprises vital signs, laboratory tests, and static variables, while the multiclass outcome labels are derived from the occurrence of adverse events including death, cardiac arrest, and unanticipated ICU admission, or their absence. The exclusion window applied prior to the outcome time is shown in red.

3.2.2 MIMIC-IV-ED

The Medical Information Mart for Intensive Care (MIMIC [262]) is a collection of databases of de-identified health-related data associated with patients admitted to a single center, the Beth Israel Deaconess Medical Center, in Boston, MA, USA. The MIMIC-IV database links patient data across a variety of different modalities and hospital modules, including the core hospital, ED, ICU, and imaging modalities, and each department maintains its own data collection and recording approach. The MIMIC-IV ED module, MIMIC-IV-ED [37], includes ED admissions between 2011 and 2019, totaling 448,972 ED admissions (version 1.0). MIMIC-IV-ED was considered as a secondary dataset to validate the generalizability of the models developed in this thesis. It constitutes an external setting distinct from HAVEN as it represents tertiary care emergency department admissions in the USA. This allows the evaluation of the models in a separate clinical environment, and thus provides supporting evidence to the generalizability of the model to other clinical applications. Furthermore, MIMIC-IV-ED is publicly accessible after approval from PhysioNet [263] and completion of the relevant

data use agreement and training, which supports reproducibility and independent validation.

The ward-level vitals in MIMIC-IV do not have the temporal resolution of HAVEN, which leverages the SEND system to record vital signs at semi-periodic intervals; consequently, a direct mapping of the HAVEN methodology to MIMIC-IV is not possible. However, the ED module manually records vital-sign observations at aperiodic intervals during the ED admission of a patient, consistent with standard ED practice [264, 265]. In the case of MIMIC-IV-ED, this corresponded to observations every $60 - 120$ minutes, depending on the patient's acuity. The recorded vital signs include heart rate, respiratory rate, SpO_2, systolic blood pressure, diastolic blood pressure, and temperature. Additionally, MIMIC-IV-ED records the Emergency Severity Index (ESI [266]) — a nurse-assessed triage score that indicates the acuity of a patient on a scale from 1 to 5, where 1 denotes the highest level of concern. ESI is recorded once at the beginning of the ED admission during triage. FiO_2 is not available in this module. The static variables included in the analysis were age and gender. The linkage of the ED module to MIMIC-IV further enables the inclusion of medication information and, critically, discharge diagnoses and follow-up information for each patient.

The analysis cohort was defined by the following inclusion and exclusion criteria. Admissions were required to have valid admission and discharge times (i.e., non-missing values and with the admission time preceding the discharge time) and at least one recorded observation of vital signs; moreover, the minimum allowable age at the time of admission to the ED was set at 18 years. When the reason for admission or discharge was related to psychiatry or childbirth, such as labor, the corresponding admissions were excluded, as these represent distinct clinical scenarios that differ substantially from the general ED population; empirically, such admissions exhibited a limited number of observations and had a relatively short admission time within the ED, so their inclusion was not suitable for the TPC research. Admissions with an ESI score of 1 or 5 were also excluded: an ESI score of 1 encodes patients of extreme acuity who rapidly

escalate from the ED, while scores of 5 encode patients with extremely low acuity, almost all of whom were immediately discharged without proceeding to the main hospital [266]. Finally, patients whose admissions comprised consecutive ED admissions followed by hospital discharge were also excluded. These occurrences were rare, with approximately 30 admissions in total in the final stage of postprocessing satisfying this condition, and the occurrences often reflected data artifacts or short revisits.

For each admission, the outcome label was defined based on the patient status after a horizon window of 12 hours following the discharge time from the ED. The horizon was empirically selected: a range of values (4, 12, 24, and 48 hours) was evaluated, and 12 hours was the shortest horizon for which the minimum proportion of results on all outcome labels exceeded 0.5% of the total cohort. The lower bound of 0.5% was chosen as approximately half the prevalence of the rarest outcome class in the postprocessed HAVEN data. Adverse events were defined as the occurrence of death (*Death*) or admission to an ICU (*ICU*, determined by admission of the patient to an intensive care unit of any type in the hospital). *NoEvent* was defined as hospital discharge (different from ED discharge) within the horizon window, which is interpreted as a proxy of stable patient status. Lastly, *Ward* denoted admission to a non-ICU hospital ward within the target window.

The temporal aspect comprised the trajectories of the vital-sign variables described above, together with static variables, again modeled as static time series. Laboratory test results, although available in MIMIC-IV, were not included in the temporal aspect — in clinical ED practice, laboratory measurements were recorded with a frequency too low relative to the duration of the ED admission, and many observations had a recorded time many hours after discharge from the ED. To define the outcome aspect, the four outcome classes of Death, ICU, NoEvent, and Ward were used; similar to HAVEN, MIMIC-IV-ED constitutes an example of an MCS. In cases where multiple outcome labels were applicable within the target window, the most severe label was assigned according to the

priority hierarchy: Death > ICU > Ward > NoEvent, consistent with the standard severity assessment during triage, and consistent with previous analyses [267–269]. Consequently, the outcome time was defined as: (a) the time of the first adverse event that occurred that matched the outcome label, in the case of Death or ICU; (b) the time of discharge from the hospital, in the case of NoEvent; (c) the time of admission to a non-ICU ward, in the case of Ward.

	HAVEN	**HAVEN-T2RF**	**MIMIC-IV-ED**
Clinical setting	Hospital Ward	Hospital Ward	ED
Location	UK (Oxford)	UK (Oxford)	USA (Boston)
Period	2014–2018	2014–2018	2011–2019
Number of Admissions	251,266	49,173	448,972
Number of Patients	168,794	31,495	216,877
Observations	6,229,740	1,231,378	8,890,459
Temporal variables	22*	22*	6
Static variables	3	3	3
Outcome classes	4	4	4
Outcome setting	MCS	MCS	MCS
Public availability	No	No	Yes

* Includes 8 vital-sign variables and 14 laboratory test variables; see Table 3.1.

Table 3.2: Comparative summary of the two datasets used in this thesis. Temporal variable counts exclude static variables. Outcome classes refer to the distinct labels defined for each dataset. The setting denotes the outcome formulation used in each dataset.

3.3 A Unified Approach for Preprocessing

TPC problems can be used in a wide range of applications with data from highly heterogeneous sources [270–272]. Although dataset-specific criteria exist, this section describes a set of general guidelines for handling TPC data. A diagram of the proposed framework is shown in Figure 3.2, organized as a sequence of steps. This framework assumes that the base cohort is already defined, as well as the temporal and outcome aspects.

3.3.1 Outlier Removal

The removal of outliers is applied using two criteria. First, observations with values that fall outside the clinically admissible ranges are removed. For example,

a SpO_2 value of 110% is physically impossible and is therefore excluded. Some range endpoints are universal; others depend on the population under study. Second, the top and bottom 2.5% quantiles of observed values are set as missing, effectively removing them, following established practice for quantile-based outlier removal [273, 274]. In TPC settings, quantile thresholds are computed per feature according to the assumption that each recorded value is an independent sample of that variable. This is appropriate for the relatively short horizons used in this thesis; for longer trajectories, it may be preferable to compute thresholds using a time-varying approach [275].

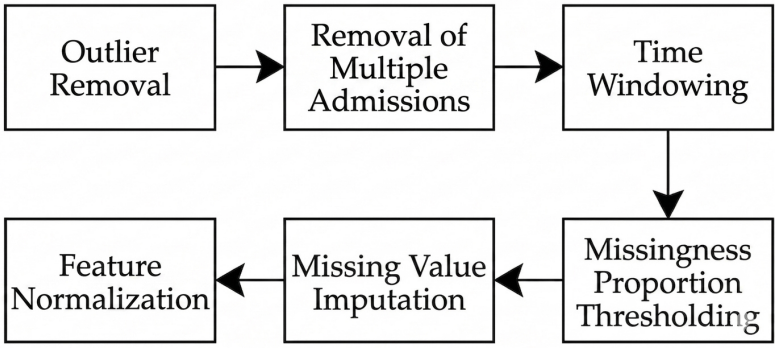


Figure 3.2: Overview of the processing steps used to transform TPC EHR data into a format suitable for modeling. Arrows indicate the flow of data through the pipeline.

3.3.2 Removal of Multiple Admissions

ML models typically assume data points are Independent and Identically Distributed (IID). However, in EHR data, the same patient often has multiple admissions, which violates the IID assumption. A common mitigation is to retain only one admission per patient. The literature varies on the selection criteria (e.g., first, last, or random admission). This thesis selects the *most recent* (last) admission per patient, determined by the timestamp corresponding to the admission entry. This choice is clinically motivated: in outcome-imbalanced cohorts such as HAVEN, the last admission is most likely to reflect peak acuity and thus the

rarest (associated with the worst outcome) classes, thus partially mitigating the class imbalance.

3.3.3 Time Windowing

In clinical practice, observations of temporal variables can be recorded at irregular intervals and variables are collected at different time points. The number of observations can therefore differ between (a) variables and (b) admissions. Some models can handle this irregularity [220], but most models require a fixed number of time points for each variable and patient. Time windowing refers to the process of converting irregularly sampled data into a regular format. First, for each admission, the time axis is partitioned into consecutive and disjoint window blocks of fixed length, and observations of the same variable are aggregated within each block. The aggregation function followed in this thesis is the median; however, the mean or any other suitable function may also be used. For each admission, the window blocks are shared between different variables and provide a common time index for the admission. A subsequent step fixes the number of time points by determining the maximum number of allowable window blocks, and truncating or padding the time series data accordingly, resulting in data of (potentially-missing) observations that can be represented as a matrix of dimensions $N \times T \times D_f$, where N is the number of samples (i.e., the number of patients or number of admissions), T is the total number of time points, and D_f is the total number of time series.

In HAVEN, a window length of 4 hours was introduced to mimic the expected average of the in-hospital vital-sign recording frequency [19]. Window blocks were defined backward from the outcome time and only observations between 72 and 24 hours before the outcome time were included during training, although longer time periods are used in visualization. Thus, HAVEN exhibits a 48-hour observation length for models. These decisions were informed by an empirical analysis of the average length of admissions and confirmed by clinical experts. In MIMIC-IV-ED, 1-hour window blocks were used, starting backward from the

discharge time of the ED. Due to the shorter duration of ED admissions, only observations within 10 hours prior to ED discharge were included in the training.

3.3.4 Missingness Proportion Thresholding

Even after the above steps, missing values may remain. Removing admissions that contain missing values for any feature often results in a very substantial reduction in the size of the dataset, particularly in TPC, where the number of features and time points is large [276, 277]. Imputation is therefore preferred; however, imputing variables with a very high proportion of missing values can introduce bias and produce unreliable estimates, e.g., as in Grigoroff et al. [278]. Therefore, thresholding rules are applied to determine when features are suitable for imputation and when admissions should be excluded due to the high number of missing values. Of all steps described here, this step is the most dataset-specific, as the amount and structure of missingness depend strongly on the clinical environment and the definition of the window blocks.

HAVEN processing adapted the protocol in Pimentel et al. [19], and only admissions that satisfy (a) at least 3 observations per day, (b) at least 24 hours between the first and last available observations, and (c) an average rate of 2 observations for each set of 3 window blocks were included for analysis. These criteria were applied across all vital signs only as they are recorded more frequently and are more informative, as discussed in Section 3.2.1. In MIMIC-IV-ED, a more lenient threshold was used and instead the inclusion criteria consisted of at least 4 total observations for each of the six vital-sign variables.

3.3.5 Missing Value Imputation

This thesis imputes the time series data by applying the following three steps in sequence: (a) forward-filling, where missing values are filled with the last observed value for the same variable and admission; (b) backward-filling, where missing values are filled with the next observed value for the same variable and admission; and (c) median imputation, where any remaining missing values are

filled with the median value of the corresponding variable across the training data to avoid information sharing[1]. The choice of the median was informed by the non-normality of the distribution of some variables; moreover, the computation of the median is done prior to steps (a) and (b) to avoid data leakage. Note that step (b) only applies to missing values at the beginning of the time series, since all the others would have been filled in step (a). This was determined to be suitable because the focus was on clustering the full length of the time series, rather than time series forecasting or prediction. This three-step approach has a limitation in that it assumes state persistence; nevertheless, this is reasonable for the short time horizons considered.

3.3.6 Feature Normalization

Normalization is performed to ensure that all features are on the same scale and dynamic range, which can improve model learning. Min-max normalization is used *feature-wise* according to the following equation:

$$z_{\text{norm}} = \frac{x_{\text{obvs}} - \min_f}{\max_f - \min_f} \tag{3.1}$$

where x_{obvs}, z_{norm} denote an observed value pre- and post-normalization, respectively, and $\min_f$ and $\max_f$ denote the minimum and maximum values of the feature f over all admissions *and* at all time points, respectively. Min-max normalization was preferred to standardization for the same reason as in Section 3.3.5: some features did not follow a normal distribution.

An illustrative example of fully processed data is shown in Figure 3.3, which displays the mean vital-sign trajectories per outcome class in HAVEN after all processing steps have been run. The trajectories across outcome classes/labels exhibit substantial overlap, with no clear separation between groups. This describes a core challenge in TPC problems and motivates the development of models that take advantage of both the temporal and the outcome aspects.

[1] See Section 4.4 for clarification on training data.

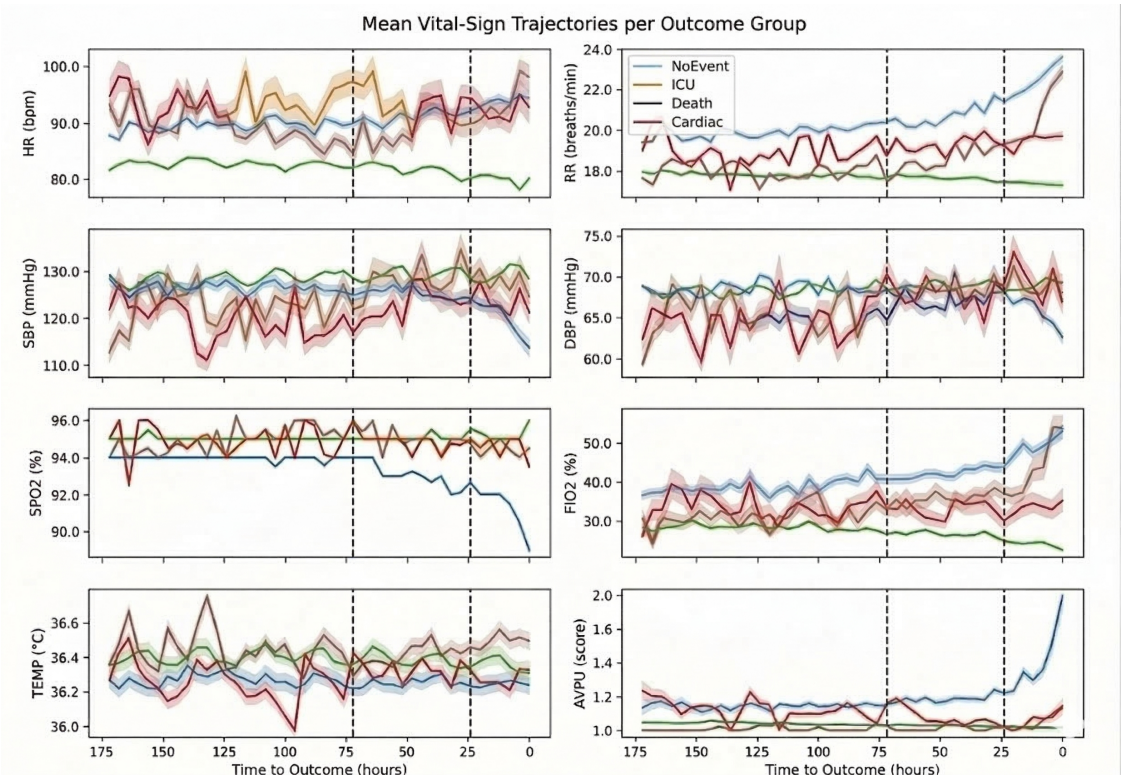


Figure 3.3: Mean vital-sign trajectories for each outcome class in HAVEN after the application of the processing framework described in Section 3.3. Eight panels are shown, one per vital sign. Colors represent outcome labels: blue, *Death* ($N = 788$); red, *Cardiac* ($N = 58$); brown, *ICU* ($N = 159$); and green, *NoEvent* ($N = 15464$). Vertical dashed lines indicate the input cutoff points for model training. Shaded regions denote the standard error of the mean.

HAVEN and MIMIC-IV-ED exhibit a severe outcome class imbalance, a characteristic common in real-world clinical data (see also Chapter 4 for a more in-depth discussion). In HAVEN, the postprocessed cohort includes $16,469$ admission ($4,266$ for HAVEN-T2RF), of which 86.8% are assigned NoEvent, while the remaining classes account for 10.3% (Death), 1.8% (ICU), and 1.1% (Cardiac) proportion of admissions in the data, respectively. In MIMIC-IV-ED, the distribution of the outcome classes is 72.96% (Ward), 24.72% (ICU), 1.54% (NoEvent), and 0.78% (Death) over 3831 admissions.

3.4 Evaluation Procedure

The TPC models introduced in this thesis were evaluated following a standardized approach to facilitate the comparison of results and to avoid the need for dataset-specific evaluation approaches. As discussed in Sections 2.4.2 and 3.2, this thesis considers TPC settings with two aspects: a temporal aspect that is observed and an outcome aspect that represents information about the future status of a patient and unobserved at inference time. The discussion below addresses the evaluation of these aspects and is naturally extended to each particular setting, with any setting-specific particularities indicated accordingly. With that in mind, Section 3.4.1 focuses on evaluating the temporal aspect and Section 3.4.2 addresses the evaluation of the outcome aspect. Section 3.4.3 introduces two metrics that propose an approach to combine both aspects. Finally, the end of this section considers the combination of the two sets of results with respect to the overall performance and clinical utility of the TPC models.

3.4.1 Temporal Aspect

In general, the evaluation of the temporal aspect focuses on quantifying the quality of the derived clusters, i.e., whether the clusters are *consistent* (data points in each cluster are 'close'), and *distinct* (data points in different clusters are 'far'). This evaluation is performed without access to any 'true' cluster labels. The approach here leverages clustering metrics commonly used in the literature with adaptations introduced to handle the temporality of the data. The distinction is important to leverage each metric with different methods to handle temporal data. The notation from Chapter 2 is extended. The set of all clusters, $\mathcal{C} = \{C_k\}_{k=1}^{K}$, is called the clustering output. In the discussion that follows, a *sample* relates to a data point that is arbitrarily selected. Finally, assume that there exists a distance function $d : X \times X \to \mathbb{R}_{\geq 0}$ that quantifies the proximity between any two samples in the input space.

Definition 4 *For each sample x_i, define $a_{SIL}(x_i)$ as the average distance between x_i and all other samples in $C_{\phi(x_i)}$, where $C_{\phi(x_i)}$ is the cluster to which x_i is assigned. Then, for each sample x_i and cluster $k \neq \phi(x_i)$, define $b_{SIL}(x_i, k)$ as the average distance between x_i and all samples in cluster k. Finally, define $b_{SIL}(x_i) = \min_{k \neq \phi(x_i)} b_{SIL}(x_i, k)$. The silhouette score for each sample is then defined as:*

$$S_{SIL}(x_i) = \frac{b_{SIL}(x_i) - a_{SIL}(x_i)}{\max(a_{SIL}(x_i), b_{SIL}(x_i))} \tag{3.2}$$

and the SILhouette Coefficient Score (SIL [279]) for clustering $\mathcal{C}$ is defined as the average silhouette score across all samples:

$$S_{SIL}(\mathcal{C}) = \frac{1}{N} \sum_{i=1}^{N} S_{SIL}(x_i) \tag{3.3}$$

The functions a_{SIL} and b_{SIL} are a proposed representation of the average *intra-cluster* and *inter-cluster* distances, respectively. For each sample x_i, $S_{\text{SIL}}(x_i)$ ranges between -1 and 1 —, this can be derived by noting that $a_{\text{SIL}}, b_{\text{SIL}}$ are always nonnegative and therefore the absolute value of the numerator is smaller than that of the denominator. The score is positive if $b_{\text{SIL}} > a_{\text{SIL}}$, i.e., the sample is closer to other samples in its own cluster, $C_{\phi(x_i)}$, than other clusters, which describes a good clustering result. In particular, $S_{\text{SIL}}(x_i)$ close to 1 if $a_{\text{SIL}}(x_i) \ll b_{\text{SIL}}(x_i)$ represents a nearly ideal clustering. The opposite is true in the case of negative scores (and when $S_{\text{SIL}}(x_i) \approx -1$). Since the SIL of a given clustering is obtained as the average silhouette score, all the above properties are naturally extended from a single sample to the entire clustering, so higher SIL values (up to a maximum of 1) indicate better cluster outputs.

Definition 5 *Suppose $d_{intra} : \mathcal{C} \to \mathbb{R}_{\geq 0}$ is a function that measures the intra-cluster proximity[2] for each cluster, while $d_{inter} : \mathcal{C} \times \mathcal{C} \to \mathbb{R}_{\geq 0}$ is a function that measures the inter-cluster distance between any two clusters. The Dunn-Index family of metrics (DI [76]) is defined as:*

$$S_{DI}(\mathcal{C}) = \frac{\min\limits_{k_1 \neq k_2} d_{inter}(C_{k_1}, C_{k_2})}{\max\limits_{k} d_{intra}(C_k)} \tag{3.4}$$

[2]Denoted as *diameter* in other sources.

with k_1, k_2 ranging over $[K]$ such that $k_1 \neq k_2$.

The original formulation considered particular implementations of $d_{\text{intra}}, d_{\text{inter}}$, however, this was then generalized to other formulations [280]. For the purposes of this thesis, the average formulations are assumed, defined as follows:

$$d_{\text{intra}}(C_k) = \frac{1}{\#C_k\big(\#C_k - 1\big)} \sum_{\substack{x_{i_1}, x_{i_2} \in C_k \\ i_1 \neq i_2}} d(x_{i_1}, x_{i_2}) \tag{3.5}$$

$$d_{\text{inter}}(C_{k_1}, C_{k_2}) = \frac{1}{\#C_{k_1}\,\#C_{k_2}} \sum_{x_{i_1} \in C_{k_1}} \sum_{x_{i_2} \in C_{k_2}} d(x_{i_1}, x_{i_2}) \tag{3.6}$$

The DI score is nonnegative, since both the numerator and the denominator are nonnegative. Higher DI values indicate higher values of inter-cluster distance relative to intra-cluster proximity, therefore indicating better clustering outputs. Unlike the SIL score, the DI score is unbounded from above, and the absolute values of the score depend on the choices of d_{intra} and d_{inter}, and are not directly comparable between settings.

For the following metrics, suppose that each cluster C_k has an associated cluster representation in input space, $z_k^{\text{repr}} \in X$. Examples include the cluster centroid, defined as the average of all samples in the cluster, the medoid, defined as the sample in the cluster with the smallest average distance to all other samples in the cluster, cluster representations learned in parallel, etc.

Definition 6 *For each cluster, C_k, define $\sigma_k^{DBI} := \frac{1}{\#C_k} \sum_{x_i \in C_k} d(x_i, z_k^{repr})$ as the average distance between cluster samples and their representation. The Davies-Bouldin Index (DBI [281]) is then defined as:*

$$S_{DBI}(\mathcal{C}) = \frac{1}{K} \sum_{k_1=1}^{K} \max_{k_2 \neq k_1} \left(\frac{\sigma_{k_1}^{DBI} + \sigma_{k_2}^{DBI}}{d(z_{k_1}^{repr}, z_{k_2}^{repr})} \right) \tag{3.7}$$

The DBI score is nonnegative, as all terms in the formula are nonnegative. Intuitively, $\sigma_{k_1}^{\text{DBI}} + \sigma_{k_2}^{\text{DBI}}$ corresponds to the sum of 'spread' of the clusters k_1 and k_2, while $d(z_{k_1}^{\text{repr}}, z_{k_2}^{\text{repr}})$ corresponds to the distance between each pair of cluster representations. The ratio $(\sigma_{k_1}^{\text{DBI}} + \sigma_{k_2}^{\text{DBI}})/d(z_{k_1}^{\text{repr}}, z_{k_2}^{\text{repr}})$ is smaller when clusters are

more compact and/or farther apart. Therefore, smaller values of DBI indicate better clusterings (and cluster representations), while higher values indicate worse clusterings. Similarly to the DI, the DBI is unbounded and is not directly comparable across experimental settings.

Definition 7 *Given cluster representations, z_k^{repr}, as defined above, define the centroid of centroids as the average of all cluster representations, $z^{VRC} := \frac{1}{K}\sum_{k=1}^{K} z_k^{repr}$. The between-cluster distance is defined as $\Delta_{VRC} := \sum_{k=1}^{K} \#C_k d(z_k^{repr}, z^{VRC})$, while the within-cluster distance is defined as $\delta_{VRC} := \sum_{k=1}^{K}\sum_{x_i \in C_k} d(x_i, z_k^{repr})$. The Calinski-Harabasz Index or Variance Ratio Criterion (VRC [93]) can be computed according to:*

$$S_{VRC}(\mathcal{C}) = \frac{\Delta_{VRC}}{\delta_{VRC}} \frac{N-K}{K-1} \tag{3.8}$$

In the definition of the VRC, Δ_{VRC} is a weighted average of the distance from each cluster representation to the overall centroid, while δ_{VRC} is a measure of the overall distance between samples and their cluster representations, summed over all clusters. The ratio $\frac{\Delta_{\text{VRC}}}{\delta_{\text{VRC}}}$ is higher when clusters are more compact and/or farther apart; therefore, higher VRC values indicate better clusterings, while lower values indicate worse clusterings. Similarly to DI and DBI, VRC scores are nonnegative and unbounded above, and absolute values are not directly comparable between settings. The normalization term $\frac{N-K}{K-1}$ is a correction factor that accounts for the number of clusters and samples, and is included to allow comparison of the VRC scores between different numbers of clusters.

Applications to TPC In the special case of TPC settings, the input space X is the space of multidimensional trajectories, and patient n's data can be represented as $\mathbf{X}_n \in \mathbb{R}^{T \times D_f}$, where T is the maximum number of time steps and D_f is the number of features. This formulation is possible due to the sequence of steps described in Section 3.3, specifically, time windowing and missing value imputation, which allow the representation of trajectories as fixed-length vectors. For a consistent evaluation across different models, the distance function considered is the Euclidean distance or L_2 (see also Equation (2.3)), defined for real-valued

vectors. In order to adapt the distance function to the temporal nature of the data, the trajectories are 'flattened' into vectors of dimension $T \cdot D_f$, with each dimension of the flattened vector denoting a particular combination (feature, time step). Unfortunately, this approach is naturally biased in favor of models that apply clustering operations directly in the input space due to the definition of the distance function. Models such as TSKM, as described in Section 4.2.1.1, thus benefit from this bias, while models that implement clustering in a latent space are associated with worse-performing results. However, the alternative approach of computing metrics on the latent state representations would also not be suitable due to the mismatch in the number of dimensions ($T \times D_f$ against the dimension of the latent space) and due to the fact that representation in the latent space distorts the original data. Finally, the cluster representations (particularly relevant for VRC and DBI) were defined as the cluster centroid, unless cluster representation vectors were learned as part of the model.

3.4.2 Outcome Aspect

This work proposes leveraging commonly-used evaluation metrics for MCS classification [282] defined below. Consequently, mimicking Section 3.4.1, the subsection concludes with an explanation of how these can be applied in TPC settings. To discuss MCS classification metrics, first, the Binary Classification Setting (BCS) is introduced. Assume that each sample x_i is associated with a binary target $y_i^{\mathrm{b}} \in \{0, 1\}$. Assume a classification model is trained, and let $\tilde{y}_i^{\mathrm{b}} \in [0, 1]$ be the predicted value for the sample i. For a given threshold $\epsilon \in [0, 1]$, the predicted label for the sample i is defined as $\hat{y}_i^{\mathrm{b}}(\epsilon) = 1$ if $\tilde{y}_i^{\mathrm{b}} \geq \epsilon$, and $\hat{y}_i^{\mathrm{b}}(\epsilon) = 0$ otherwise, commonly denoted as $\hat{y}_i^{\mathrm{b}}$ when ϵ is clear from context. According to $\left\{y_i^{\mathrm{b}}, \hat{y}_i^{\mathrm{b}}(\epsilon)\right\}_{i=1}^{N}$, the set of *true positives* (TP), *false positives* (FP), *true negatives* (TN) and *false negatives* (FN) refer to the set of samples that are correctly labeled, respectively, as positive, incorrectly labeled as positive, correctly labeled as negative, and incorrectly labeled as negative. When written in a 2×2 matrix, these counts are also called the *confusion matrix*.

Definition 8 *The following metrics are commonly used in binary classification:*

- $ACC = \frac{TP+TN}{TP+FP+TN+FN}$ *is the ACCuracy score, which measures the overall proportion of correctly classified samples.*
- $PRE = \frac{TP}{TP+FP}$ *denotes the PREcision score, which measures the proportion of true positives among all predicted positives.*
- $REC = \frac{TP}{TP+FN}$ *is used to compute the RECall score, which measures the proportion of true positives among all actual positives.*
- $F1 = 2 \times \frac{PRE \times REC}{PRE+REC} = \frac{2 \times TP}{2 \times TP+FP+FN}$ *is the F1 score, the harmonic mean between precision and recall.*

All metrics are bounded between 0 and 1, with higher values indicating better performance. They reflect different viewpoints on classifier performance, e.g. accuracy is a measure of overall performance, while precision and recall penalize different types of errors. Furthermore, they require specification of ϵ, which is not ideal — intuitively, a predicted value of 0.8 for a single sample is more indicative of the occurrence of a true label, as opposed to a value of 0.1, so these should be penalized differently. The following metric addresses this issue:

Definition 9 *The Receiver-Operating Characteristic Curve* (ROC) *is a graphical plot of REC, or true positive rate* (TPR)*, on the* $y-$*axis against the false positive rate* $(\frac{FP}{TN+FP})$*, or FPR, on the* $x-$*axis that shows the performance of a binary classifier as one varies* ϵ*. The Area-Under-the-Receiver-Operating Characteristic Curve* $(AUROC)$ *then denotes the area of this curve over the interval* $[0, 1]$*.*

To see why AUROC is useful, note first that, as functions of ϵ, both the FPR and the REC are monotonically non-increasing. Furthermore, edge cases, $\epsilon = 0$ and $\epsilon = 1$, correspond to $(0, 0)$ and $(1, 1)$, respectively, in the ROC curve, so that AUROC $\in [0, 1]$. Moreover, it can be shown that AUROC is equal to the probability that a randomly drawn positive sample will have a higher predicted value than a randomly drawn negative sample [283], which is a desirable property.

Threshold Selection The AUROC is a metric capable of capturing more nuance than other metrics; however, it has the disadvantage that it does not actually convert predicted values into predicted labels, i.e., it cannot be used to infer predicted labels from a model. Instead, a target performance metric is first chosen, e.g. F1, and then optimal ϵ^* is selected as the value that maximizes the target. Before optimization of ϵ, it is typically assumed that $\epsilon^* = 0.5$, however, this is not recommended practice for the analysis of predicted labels [284]. In binary classification, under the standard assumptions that predictions can be inverted, it can be shown that, for a given model that does not produce a constant prediction, AUROC ≥ 0.5 — this can be derived via 'flipping', where a symmetric classification model with values $\tilde{y}^{\text{b}}_{i,2} = 1 - \tilde{y}^{\text{b}}_i$, with the effect that positive and negative predictions are switched, and the ROC curve is reflected across the diagonal $y = x$. Therefore, one of the two classifiers has an area under the curve bigger than that of $y = x$, 0.5, and the best classifier can be picked based on whichever is the largest. Critically, this property does not scale to the MCS setting, as discussed below.

3.4.2.1 MCS Classification

Unlike BCS, MCS (relevant to HAVEN and MIMIC-IV-ED data) cannot treat label predictions independently, as each data point is associated with a single class. In MCS, data samples are associated with a single class, $o_i \in [O]$, commonly represented as a one-hot vector, $\mathbf{y}^{\text{mc}}_i \in \Delta^{O-1}$, where O is the number of classes. An MCS model produces a row-normalized vector of predicted values, $\tilde{\mathbf{y}}^{\text{mc}}_i \in \Delta^{O-1}$, where each entry corresponds to the predicted value for the corresponding label. In MCS, threshold selection is irrelevant, as the predicted label is defined as the label with the highest predicted value, i.e., $\hat{y}^{\text{mc}}_i = \underset{o=1,\ldots,O}{\arg\max}(\tilde{\mathbf{y}}^{\text{mc}}_i)_o \in [O]$. While helpful for metrics such as accuracy, F1, recall, ..., note that a higher score on the correct label ideally corresponds to a better prediction than otherwise. Consider the following illustrative example. Suppose two models with predicted scores $\tilde{\mathbf{y}}^{\text{mc}}_{i,1}$ and $\tilde{\mathbf{y}}^{\text{mc}}_{i,2}$ for a data point i with a true label $\mathbf{y}^{\text{mc}}_i$. If $(\tilde{\mathbf{y}}^{\text{mc}}_{i,1})_{\mathbf{y}^{\text{mc}}_i} > (\tilde{\mathbf{y}}^{\text{mc}}_{i,2})_{\mathbf{y}^{\text{mc}}_i}$, then

the first model is more confident in its prediction than the second and should be rewarded accordingly. However, standard metrics such as accuracy or F1 would not capture this difference, as they only consider whether the predicted label matches the true label, not the confidence of the prediction. This is not a fully solvable issue [285, 286], however, it is important to critically analyze results. Due to the inability to treat labels independently, MCS evaluation is more complex than BCS. Two strategies are used:

Definition 10 *Given an MCS problem, decomposition into multiple binary classification problems can be done via:*

- *One vs Rest (OvR) approach — for each class, o, a BCS problem is defined by considering o as a positive label, and all others as negative.*
- *One-vs-One (OvO) approach — for every pair of classes, o_1 and o_2, a BCS problem is defined by considering o_1 as a positive label, and o_2 as a negative label. Furthermore, only samples belonging to the classes o_1 or o_2 are included for each BCS. This results in $\binom{O}{2}$ separate BCS problems.*

To obtain a single score, two different methods are followed. The first, called *macro-average*, computes metrics such as PRE, REC, and F1 for each label, and then returns the average score as the overall metric score. However, metrics that are macro-averaged are not equivalent to the average performance of fully independent binary classifiers. This discrepancy arises because, in practice, the underlying model often produces a shared scoring function across classes, leading to correlated predictions and errors. Furthermore, class imbalance and the choice of decision thresholds can significantly affect per-class performance, and these effects are amplified under macro-averaging, where each class is weighted equally regardless of its prevalence. This effect is particularly pronounced for models that produce a single scalar risk score (e.g. Early Warning Scores), where the same ranking is reused across all classes. Alternatively, one can use *micro-average*, which computes the total TP, FP, TN, and FN on all labels and then

computes the target metric using the totals. In general, micro-average is more suitable for imbalanced datasets, as it gives more weight to the performance on the most common labels, while macro-average treats all labels equally [282]. Independently of the strategy used, metrics such as accuracy, precision, recall, F1, and AUROC can be calculated for each binary classification problem, and then averaged across classes using micro- or macro-average. The experiments in this thesis considered the OvR strategy for metric computation. The MCS accuracy metric is an exception to the above extension and is calculated as the proportion of samples for which the predicted label matches the true label.

Applications to TPC Since outcomes are forward-facing and unobserved, it is important that learned clusters can be used in clinical practice to identify clear outcome profiles, rather than simply being associated with different outcome distributions. In HAVEN, clusters associated with high likelihoods of Death or ICU outcome labels are of greater clinical utility than clusters that do not discriminate between outcome classes. The evaluation should therefore be adapted to identify this property. To do this, the evaluation of the outcome aspect is performed via a *proxy task*, that is, outcome prediction via cluster assignment, i.e., the prediction of patient outcomes through cluster assignments. The rationale is that if clusters are associated with relevant outcome profiles, then it should be possible to take advantage of the cluster assignments to predict particular outcome labels.

Proxy task construction is done in two steps. First, the prediction vector for each cluster is derived: for models that implicitly model this information during training, then the prediction vector is taken directly from the model; alternatively, for models such as TSKM, the prediction vector is computed from the empirical distribution of outcomes for the patients assigned to each cluster:

$$\mathbf{y}_k^{\text{proxy}} = \frac{1}{\#C_k} \sum_{\mathbf{x}_n \in C_k} \mathbf{y}_n \tag{3.9}$$

where $\mathbf{y}_n$ is the outcome label vector for patient n (e.g., $\mathbf{y}_n^{\text{mc}}$). Second, given a clustering assignment, ϕ, and the prediction vector for each cluster, a prediction pipeline is built by the assignment $x_n \rightarrow \phi(\mathbf{X}_n) \rightarrow \mathbf{y}_{\phi(\mathbf{X}_n)}^{\text{proxy}}$, where $\phi(\mathbf{X}_n)$

denotes the cluster assignment for a given input trajectory $\mathbf{x}$. In other words, any given patient trajectory, $\mathbf{X}_n$, is first assigned to a cluster, $\phi(\mathbf{X}_n)$, and then the corresponding outcome for that patient is obtained directly from the cluster prediction vector. Thus, the performance of the outcome prediction pipeline is directly related to the quality of the cluster assignments and the relevance of the cluster outcome profiles. The performance of this pipeline is then evaluated using the metrics described above for the MCS setting, depending on the nature of the outcome labels.

3.4.3 Combined Evaluation

Combined evaluation metrics are unfortunately not standardized in the TPC literature, and there is no single proposed unified approach. Consider the following naive approach. Given a cluster assignment, ϕ, for each patient, concatenate the patient's outcome labels with the input trajectory variables, and then evaluate the cluster performance using the metrics described in Section 3.4.1 on the expanded vectors. This naive approach is unsuitable as it conflates variables of fundamentally different types and dimensionality, suppressing the contribution of each aspect to the overall distance.

Instead of the naive solution, a different approach is followed. First, consider the following set of clustering metrics, commonly used when true cluster labels are available. For the following definitions, suppose that true cluster labels, $\{c_i^{\text{true}}\}_{i=1}^N$, are available for each sample, where $c_i^{\text{true}} \in [K]$.

Definition 11 *Define a_{RI} as the number of pairs of samples that belong to the same (predicted) cluster and have the same true cluster. Similarly, define b_{RI} as the number of pairs of samples that lie in different (predicted) clusters and have different true clusters. Then, the Rand Index (RI [287]) can be computed as:*

$$S_{RI}(\mathcal{C}) := \frac{a_{RI} + b_{RI}}{N(N-1)/2} \tag{3.10}$$

The adjusted Rand Index (ARI [288]) is then defined as:

$$S_{ARI}(\mathcal{C}) := \frac{S_{RI}(\mathcal{C}) - \mathbb{E}(S_{RI}(\mathcal{C}))}{\max(S_{RI}(\mathcal{C})) - \mathbb{E}(S_{RI}(\mathcal{C}))} \tag{3.11}$$

where the expectation is taken over random cluster assignments.

The RI score is bounded between 0 and 1, as $a_{\text{RI}} + b_{\text{RI}}$ is at most the total number of pairs of samples, $N(N-1)/2$. Higher RI values indicate better clustering, with a maximum of 1 corresponding to a perfect clustering, since that would mean that all pairs of samples are correctly assigned to the same cluster. However, a known limitation of RI is that it does not account for chance, as even random cluster assignments can yield high RI scores [289]. The ARI addresses this issue by adjusting the RI score based on the expected RI score under random cluster assignments. In practice, the ARI computation can be done by combinatorial methods and does not require sampling of random cluster assignments [290]. ARI can take values between -1 and 1, with higher values indicating better clusterings, and a value of 0 corresponding to random cluster assignments.

Definition 12 *Let N_{k_1,k_2} be the number of samples that are assigned to the cluster k_1 and have a true cluster k_2, AC_{k_1} be the number of samples assigned to the cluster k_1, and TC_{k_2} be the number of samples with a true cluster k_2. Mutual information (MI [291]) is defined as:*

$$S_{MI}(\mathcal{C}) := \sum_{k_1=1}^{K} \sum_{k_2=1}^{K} \frac{N_{k_1,k_2}}{N} \log\left(\frac{N_{k_1,k_2}/N}{(AC_{k_1}/N)(TC_{k_2}/N)}\right) \tag{3.12}$$

The Normalized Mutual Information (NMI) is then defined as:

$$S_{NMI}(\mathcal{C}) := \frac{S_{MI}(\mathcal{C})}{\sqrt{H_{pred} H_{true}}} \tag{3.13}$$

where H_{pred} and H_{true} are the entropies of the predicted and true cluster distributions, respectively, and $\sqrt{H_{pred} H_{true}}$ their geometric mean.

The MI score is non-negative as it is defined as a Kullback-Leibler divergence, which is always non-negative. Higher MI values indicate better clusterings, with a maximum of $\log(K)$ corresponding to a perfect clustering, since that would mean that the predicted cluster distribution perfectly matches the true cluster distribution. However, similar to RI, MI does not account for chance, as even

random cluster assignments can produce high MI scores, and, furthermore, exhibit bias for larger numbers of clusters [292]. The NMI addresses this issue by normalizing the MI score based on the entropies of the predicted and true cluster distributions. The NMI can take values between 0 and 1, with higher values indicating better clusterings, and a value of 0 indicating statistical independence between predicted and true cluster assignments.

Applications to TPC In the settings considered in this thesis, the true cluster labels are not available. However, the logic used in the introduction of the proxy task can be applied, and outcome labels are used in the above metrics instead of true cluster labels. The reasoning is similar: Samples in the same (predicted) cluster should ideally have the same outcome labels, while samples in different (predicted) clusters should ideally have different outcome labels. Therefore, higher RI/ARI and MI/NMI scores indicate better clusterings. The evaluation approach of the temporal and outcome aspects, whether separate (Sections 3.4.1 and 3.4.2), or combined (Section 3.4.3), represents a proposal that can be generalized to other TPC models and is a contribution of this thesis to the scientific literature. However, it is important to mention that it is not a holistic solution and that the results of any experiment should also involve a more detailed qualitative analysis, particularly in clinical settings. For example, clinical interpretability in clustering outputs is not necessarily associated with better metric performance [293, 294].

3.5 Conclusion

This chapter provided three key contributions for the remainder of the thesis. The two different datasets considered were introduced in Section 3.2, while general guidelines for data preprocessing for TPC settings were presented in Section 3.3. Finally, a standardized evaluation framework for TPC models was proposed in Section 3.4. This provides a meaningful segue to the following chapter,

which uses these tools to illustrate practical challenges of TPC in imbalanced settings, and attempts to address some of the existing limitations.

Your assumptions are your windows
on the world. Scrub them off
every once in a while, or
the light won't come in.

— Alan Alda

4

Temporal Phenotypic Clustering with Imbalanced Data

4.1 Introduction

This chapter presents the first empirical evaluation, within this thesis, of State-Of-The-Art (SOTA) TPC benchmarks on a real-world clinical dataset, HAVEN (see also Section 3.2.1), focusing on the challenges associated with severely imbalanced multiclass settings, a regime in which existing methods were not originally designed to operate. To partially alleviate these challenges, this chapter introduces two modifications to the training objective: a class-frequency-weighted loss function and a reformulated loss for cluster separation. The goals of the chapter, therefore, are two-fold: (a) characterizing benchmark behavior and limitations within the context of HAVEN, and (b) proposing a derived method (Section 4.3) which is designed to handle multiclass imbalance and improve sensitivity to under-represented outcome classes. This chapter serves as a building block to address the limitations of existing methods and as a natural transition to the other methods proposed in Chapters 5 and 6.

This chapter is organized as follows. Firstly, current TPC SOTA methods are identified and described in Section 4.2, which is divided into two sections using the evaluation strategy proposed in Section 3.4. Regarding the benchmarking

of the TPC approaches in Section 4.2.1, three SOTA methods are discussed for TPC. Separately, benchmarks for predictive performance in the proxy task used for evaluation of the outcome aspect (see Section 3.4.2), are described in Section 4.2.2, including classical classification approaches, such as Support Vector Machines (SVM) and eXtreme Gradient Boosting (XGBoost), as well as a baseline clinical rule-based system, the National Early Warning System (NEWS2 [295]), a purely clinical benchmark.

4.2 Benchmark Methods

Benchmark methods are derived from distinct methodological approaches, with TSKM representing classical clustering methods for time-series data, SOM-VAE a DL approach for learning interpretable representations of EHR temporal data, and AC-TPC, which is capable of handling both the temporal and outcome aspects and was designed specifically for a particular TPC task[1]. Second, due to evaluation of the proxy task introduced in Section 3.4.2, two classical classification methods (SVM and XGBoost) and a clinically-driven method (NEWS2) are used as benchmark comparisons for this particular component.

The temporal aspect is represented as a set $\mathcal{X} \coloneqq \{\mathbf{X}_n\}_{n=1}^N$, where $\mathbf{X}_n \in \mathbb{R}^{T \times D_f}$ denotes the trajectory matrix for patient n, with (t, f)-entry $(X_n)_{t,f}$. Here, N is the total number of patients, and it is assumed that $\mathbf{X}_n$ does not contain missing values, which follows from the preprocessing guidelines established in Section 3.3. T is the total number of time steps, while D_f is the number of variables, also denoted as *features*, and the entry (t, f) of $\mathbf{X}_n$ represents the observation of feature f at time step t for patient n. Furthermore, $\mathbf{X}_n$ can be decomposed into T *observation vectors*, each of which corresponds to a temporal snippet of the data, i.e., data of the form $\mathbf{x}_{n,t} \in \mathbb{R}^{D_f}$. The data corresponding to the outcome aspect is written as $\mathcal{Y} = \{\mathbf{y}_n^{\text{mc}}\}_{n=1}^N$, where $\mathbf{y}_n^{\text{mc}} \in \Delta^{D_o - 1}$ is the outcome label for the patient n and D_o

[1]Note that AC-TPC uses a different modeling approach for the outcome aspect from that followed in this thesis; see Section 4.2.1.3 for further details.

is the number of possible outcome classes ($D_o = 4$ in the HAVEN dataset, with the outcome classes assigned to integers $1, \ldots, D_o$, and then one-hot encoded).

The objective of TPC is to identify clinically meaningful clusters with respect to the temporal and outcome aspects; mathematically, the goal is to learn a map that assigns data points to clusters, $\phi\colon \mathbb{R}^{T\times D_f} \to [K]$, where K is the number of clusters. The selection and learning of the parameter K is nontrivial and model-dependent, so it is deliberated in the corresponding model descriptions. A cluster, e.g., the cluster k, is the set of samples assigned to it, i.e., $C_k \coloneqq \{\mathbf{X}_n \mid \phi(\mathbf{X}_n) = k\}$, and the set of clusters is the output of the clustering, $\mathcal{C} \coloneqq \{C_k\}_{k=1}^{K}$. Clusters are expected to have an associated cluster representation vector, a mathematical summary of the cluster available in the input space, or in a lower-dimensional representation (also called *latent*) space. For each cluster k, its representation vector is denoted as $\mathbf{c}_k$, and the set of cluster representation vectors is written as $\mathcal{E} = \{\mathbf{c}_k\}_{k=1}^{K}$.

4.2.1 Clustering and TPC SOTA Models

The following models instantiate three different methodological families: classical clustering methods (TSKM), DL-based approaches for EHR phenotyping (SOM-VAE), and TPC SOTA (AC-TPC). Note that mathematical descriptions are preserved; however, notation has been adapted to fit the structure of this thesis.

4.2.1.1 TSKM

The model TSKM [151] adapts the classical K-*means* algorithm to multivariate time-series data by replacing the Euclidean distance with distance functions specific to time-series data. The algorithm learns cluster centroids $\mathbf{c}_k \in \mathbb{R}^{T\times D_f}$ that lie in the input space and represent multivariate temporal trajectories, making TSKM the only benchmark considered here that operates entirely without a learned latent representation. The target number of clusters K must be pre-specified, and TSKM is guaranteed to return exactly K non-empty clusters. A sketch of the algorithm output is shown in Figure 4.1.

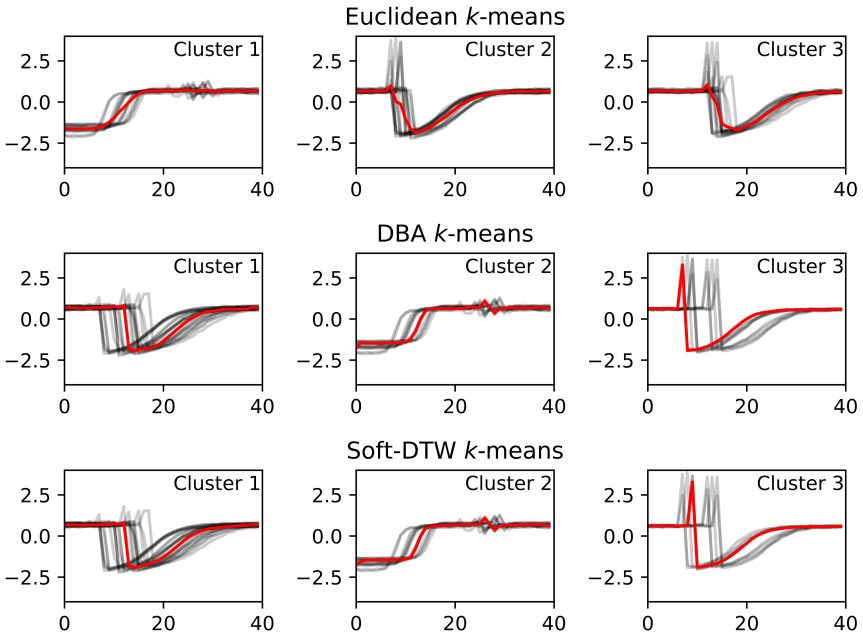


Figure 4.1: Schematic visualization of TSKM outputs and centroids on a toy example consisting of a univariate time-series dataset Tavenard et al. [151]. The TSKM algorithm can be instantiated with Euclidean, DTW Barycenter Averaging (DBA) [92], or Soft-DTW distance measures, which differ in how cluster centroids are computed; the outputs of each variant are shown in separate rows. Within each row, the algorithm identifies 3 clusters, which are visually depicted in separate plots — each plot shows the centroid (red) learned by TSKM and some of the samples assigned to that cluster (grey). Because each data sample is a univariate time series, centroids and assigned samples are visualized as time series. Axis ranges are derived from the toy dataset and kept consistent across distance measures and output clusters, with the x-axis representing time and the y-axis the value of the univariate time series.

Model Objective TSKM minimizes an objective equivalent to the sum of intra-cluster pairwise distances:

$$\mathcal{L}_{\text{TSKM}} = \sum_{k=1}^{K} \sum_{\mathbf{X}_{n_1}, \mathbf{X}_{n_2} \in C_k} d(\mathbf{X}_{n_1}, \mathbf{X}_{n_2}) \tag{4.1}$$

which reduces to standard WCSS when d is the temporal L_2 distance [151]. The choice of d is discussed in the 'Distance Function' paragraph below. The objective is minimized iteratively, as exact optimization is intractable.

Model Training The model initializes the cluster centroids according to one of two strategies. The first uses *random initialization*, where each cluster centroid $\mathbf{c}_k$

is sampled randomly, without replacement, from the data, $\mathcal{X}$. An alternative approach, $K\text{-}means++$ [113], has been shown to improve the quality of initialized centroids. $\mathbf{c}_1$ is initialized by random sampling from $\mathcal{X}$, and the other centroids are set by an iterative process: given $\mathbf{c}_1, \ldots, \mathbf{c}_k$, the distance to the nearest already-selected centroid is computed for each data sample that has not yet been selected as a centroid. Then, $\mathbf{c}_{k+1}$ is sampled from a categorical distribution over non-assigned data points with probability proportional to their squared distance which gives higher sampling preference to data samples that are farther from known cluster centroids, intending to initialize more 'spaced out' centroids.

Following the initialization of $\mathcal{E}$, the training alternates between two steps. The *assignment step* maps each trajectory to its nearest centroid:

$$\phi_{\text{TSKM}}(\mathbf{X}_n) \coloneqq \underset{k=1,\ldots,K}{\arg\min}\ d(\mathbf{X}_n, \mathbf{c}_k) \tag{4.2}$$

The *centroid update step* recomputes each centroid as the mean of its assigned trajectories under the Euclidean distance, or by DTW barycenter averaging [92] when the distance function is DTW. Iteration continues until the objective improvement falls below a threshold or a maximum number of steps is reached, both of which are treated as hyperparameters.

Distance Function Tavenard et al. [151] propose three distance functions, all of which are considered in this thesis. The first extends L_2, i.e., Euclidean, to the multivariate time-series by summing across time steps:

$$d_{L_2}(\mathbf{X}_i, \mathbf{X}_j) \coloneqq \sum_{t=1}^{T} L_2(\mathbf{x}_{i,t}, \mathbf{x}_{j,t}) \tag{4.3}$$

This is computationally efficient at $O(T \cdot D_f)$, and is equivalent to treating all feature-time observations as independent. It assumes that time series are temporally aligned, which holds under the preprocessing framework of Section 3.3, but it may be unsuitable in clinical practice when patients deteriorate at different rates.

Dynamic Time Warping (DTW [91]) addresses misalignment by optimizing over index realignment paths. For univariate trajectories $\mathbf{m}, \mathbf{o} \in \mathbb{R}^T$ corresponding to feature f of two patients in TPC settings, it simplifies to:

$$\text{DTW}(\mathbf{m}, \mathbf{o}) = \min_{\pi} \; L_2(\mathbf{m}, \mathbf{o}_\pi) \tag{4.4}$$

where $\pi : [T] \to [T]$ is a monotonically non-decreasing path satisfying $\pi(1) = 1$ and $\pi(T) = T$, which can be interpreted as trajectory stretching or compression. For multivariate input, the DTW is computed per feature and summed across features. The computational complexity of the algorithm has a lower bound of $O(T^2/\log\log T)$ [146], which can be prohibitive for large values of T. In addition, the min operator is non-differentiable, precluding direct gradient-based optimization.

Soft-DTW [150] addresses both drawbacks via a differentiable approximation that replaces min with a soft-minimum controlled by the smoothing hyperparameter $\gamma > 0$. Smaller values of γ yield closer approximations to DTW but increase the risk of numerical overflow due to the underlying exponential operator. Soft-DTW has similar complexity to DTW whilst yielding smoother loss landscapes, and is therefore more suitable for gradient-based optimization.

Outcome Aspect TSKM does not incorporate the outcome aspect during cluster formation; assignments are determined solely by the distance to centroids in the temporal input space. As a consequence, the model does not produce a direct estimate of outcome probabilities per cluster, and a direct evaluation of the outcome aspect is not possible without additional steps. Therefore, the proxy outcome prediction pipeline of Section 3.4.2 is applied: cluster assignments are used as discrete features from which a downstream classifier predicts outcome labels, allowing the evaluation of the outcome aspect using the same framework as outcome-aware methods.

Advantages The most distinctive advantage of TSKM for TPC is that centroids $\mathbf{c}_k \in \mathbb{R}^{T \times D_f}$ lie in the input space, making them directly interpretable as representative temporal trajectories. Similarly to K-$means$, centroids partition the input space into Voronoi regions [110], providing an intuitive geometric characterization of the cluster boundaries that does not require post hoc analysis of a latent space. Under the L_2 distance, TSKM is computationally efficient and does not incur additional memory overhead from learned network parameters, a significant practical advantage over SOM-VAE and AC-TPC. The inclusion of DTW and Soft-DTW as alternative distance functions further provides flexibility to handle time series that evolve at different speeds, which is common in healthcare settings where patient deterioration rates vary [91]. Finally, TSKM is guaranteed to converge to exactly K non-empty clusters and is fully reproducible given a fixed seed, which simplifies experimental analysis.

Limitations TSKM exhibits several limitations that constrain its applicability to TPC settings [296]. As the number of features increases, the pairwise distances become less discriminative (a manifestation of the curse of dimensionality) making it harder to identify meaningful cluster boundaries in high-dimensional temporal data. Under the L_2 distance, cluster boundaries correspond to linear Voronoi hyperplanes, which are poorly suited to the non-linear structure common in clinical EHR data; whilst DTW and Soft-DTW relax this constraint, they do so at increased computational cost. The K-$means$-style objective also encourages clusters of approximately equal size, which is ill-suited to the severely imbalanced class distributions in HAVEN (see Section 3.2.1). Furthermore, the sensitivity of centroid initialization requires multiple restarts, which adds computational overhead. Most critically for TPC, TSKM does not incorporate the outcome aspect during cluster formation, partitioning the input space solely based on temporal similarity. Clusters that are coherent with respect to trajectory patterns need

not correspond to clinically significant phenotypes, a limitation empirically confirmed in Section 4.5. Addressing this limitation motivates the outcome-aware methods introduced from Chapter 5 onward.

4.2.1.2 SOM-VAE

The Self-Organizing Map Variational Auto-Encoder (SOM-VAE [211]) is an unsupervised DL model that combines a VAE, a SOM, and a Markovian temporal model to learn discrete, topologically structured latent representations of time-series data. In contrast to TSKM, which operates directly in the input space, SOM-VAE performs clustering in a latent space and, unlike AC-TPC (Section 4.2.1.3), it does not incorporate the outcome aspect during training. Its inclusion of a two-dimensional SOM grid over cluster representations (or prototypes, using the authors' notation) provides an interpretable structure that is relevant to the TPC paradigm and, therefore, serves as the SOTA unsupervised DL baseline in this thesis. A schematic diagram of the model is provided in Figure 4.2.

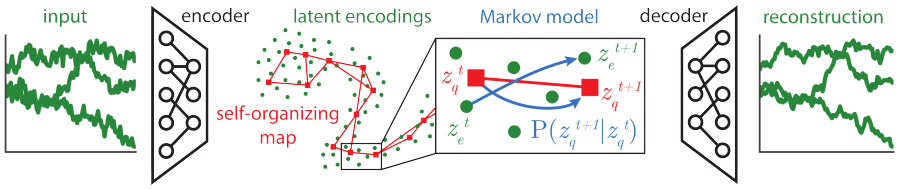


Figure 4.2: Schematic visualization of the SOM-VAE architecture Fortuin et al. [211].

Model Architecture The neural architecture of SOM-VAE includes an encoder, a decoder, and a SOM; separately, a Markov component is added to the assignments of the SOM nodes. The encoder and decoder are part of a VAE [101], in which the encoder outputs parameters of an approximate posterior, based on a Gaussian prior $\mathbf{z}_{n,t}^{\text{SV}} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, with $\boldsymbol{\mu} \in \mathbb{R}^d$, diagonal $\boldsymbol{\Sigma} \in \mathbb{R}^{d \times d}$, and d the dimensionality of the latent space — the original implementation set $\boldsymbol{\mu} = \mathbf{0}_d, \boldsymbol{\Sigma} = \boldsymbol{I}_d$. Crucially, the VAE operates on individual observation vectors $\mathbf{x}_{n,t}$, rather than full trajectories, so each time step is encoded and decoded independently. The SOM is a graph (V, G) with K nodes $V = \{v_k\}_{k=1}^K$ and edges G between adjacent

nodes. In this formulation, nodes are similar to clusters or phenotypes. Each node v_k is associated with two learned representations: latent representation $\mathbf{e}_k^{\text{SV}} \in \mathbb{R}^d$ and $\mathbf{m}_k^{\text{SV}} \in \mathbb{R}^2$, located in a two-dimensional grid that facilitates visualization. Edges connect adjacent grid points, enforcing a topological neighborhood structure on the cluster prototypes.

The forward pass proceeds as follows. Each observation vector $\mathbf{x}_{n,t}$ is passed to the encoder, producing representations $\mathbf{z}_{n,t}^{\text{SV}} \in \mathbb{R}^d$. The nearest node is then:

$$k_{n,t}^* \coloneqq \arg\min_{k=1,\dots,K} L_2(\mathbf{e}_k^{\text{SV}}, \mathbf{z}_{n,t}^{\text{SV}}) \tag{4.5}$$

with associated embedding $\mathbf{e}_{n,t}^{\text{SV}} \coloneqq \mathbf{e}_{k_{n,t}^*}^{\text{SV}}$. The decoder then reconstructs the original input observation vector, $\mathbf{x}_{n,t}$, from both $\mathbf{z}_{n,t}^{\text{SV}}$ and $\mathbf{e}_{n,t}^{\text{SV}}$, which produces, respectively, reconstructions $\hat{\mathbf{x}}_{n,t,z}^{\text{SV}}$, $\hat{\mathbf{x}}_{n,t,e}^{\text{SV}} \in \mathbb{R}^{D_f}$.

Model Objective The SOM-VAE objective decomposes into a loss per-time step and a loss at the trajectory-level that incorporates the Markov component. The former further comprises three terms. The reconstruction loss, $\mathcal{L}_{\text{SV, rec}}$, penalizes discrepancies between the input $\mathbf{x}_{n,t}$ and both reconstructions $\hat{\mathbf{x}}_{n,t,z}^{\text{SV}}$ and $\hat{\mathbf{x}}_{n,t,e}^{\text{SV}}$, simultaneously encouraging faithful encoder and node representations. The commitment loss, $\mathcal{L}_{\text{SV, commit}}$, penalizes the distance between the encoder output $\mathbf{z}_{n,t}^{\text{SV}}$ and its assigned node embedding $\mathbf{e}_{n,t}^{\text{SV}}$. The SOM loss, $\mathcal{L}_{\text{SV, SOM}}$, pushes the embeddings of neighboring nodes towards the encoder output, $\mathbf{z}_{n,t}^{\text{SV}}$, thereby reinforcing the local topological structure of the grid[2]. Together, the per-time step loss is:

$$\mathcal{L}_{\text{SV, ts}} \coloneqq \mathcal{L}_{\text{SV, rec}} + \alpha_{\text{SV}}\, \mathcal{L}_{\text{SV, commit}} + \beta_{\text{SV}}\, \mathcal{L}_{\text{SV, SOM}} \tag{4.6}$$

Temporal evolution is modeled via a Markov model over node assignments, with a transition matrix P_M. Two additional losses are introduced: a *transition loss* $\mathcal{L}_{\text{SV, tran}} \coloneqq -\log P_M(\mathbf{e}_{n,t-1}^{\text{SV}}, \mathbf{e}_{n,t}^{\text{SV}})$, which encourages node assignments to follow Markovian dynamics, and a *smoothness loss*:

$$\mathcal{L}_{\text{SV, smooth}} \coloneqq \mathbb{E}_{\mathbf{e}^{\text{SV}} \sim P_M(\mathbf{e}^{\text{SV}} | \mathbf{e}_{n,t}^{\text{SV}})}[L_2(\mathbf{e}^{\text{SV}}, \mathbf{e}_{n,t}^{\text{SV}})] \tag{4.7}$$

[2]This differs from the original SOM formulation [210]; Fortuin et al. [211] provide evidence that this variant is better suited to the clustering task.

which minimizes the expected distance between the current node embedding and likely successor nodes, encouraging smooth temporal transitions in latent space. The full trajectory objective is as follows:

$$\mathcal{L}_{\text{SV}}(\mathbf{X}_n) = \sum_{t=1}^{T} \left[\mathcal{L}_{\text{SV, ts}} + \gamma_{\text{SV}} \, \mathcal{L}_{\text{SV, tran}} + \tau_{\text{SV}} \, \mathcal{L}_{\text{SV, smooth}} \right] \tag{4.8}$$

where $\alpha_{\text{SV}}, \beta_{\text{SV}}, \gamma_{\text{SV}}, \tau_{\text{SV}} \geq 0$ are hyperparameters controlling the relative contribution of each term, and the Markov terms are set to zero at $t = 1$.

Model Training SOM-VAE is trained via Stochastic Gradient Descent (SGD), with all network parameters treated as trainable. Additionally, node embeddings $\mathbf{e}_k^{\text{SV}}$ are updated per observation via a rule specific to the SOM component:

$$\mathbf{e}_k^{\text{SV}} \leftarrow \mathbf{e}_k^{\text{SV}} + h(v_k, v_{n,t}) \, \eta_{\text{SV}} \left(\mathbf{z}_{n,t}^{\text{SV}} - \mathbf{e}_k^{\text{SV}} \right) \tag{4.9}$$

where η_{SV} is a specific learning rate for the SOM component and $h(\cdot, \cdot)$ is the neighborhood function. The neighborhood function is determined by the specification of the SOM grid, for instance, the square grid the authors use. Because the model does not explicitly capture temporal dynamics, in inference, the cluster assigned to the patient n is the most frequent assignment of nodes throughout the trajectory:

$$\phi(\mathbf{X}_n) = \underset{k=1,\ldots,K}{\arg\max} \sum_{t=1}^{T} \mathbb{1} \left[k_{n,t}^* = k \right] \tag{4.10}$$

Note also that not all nodes in the SOM are guaranteed to have samples assigned to them, so the number of learned clusters can be lower than K.

Outcome Aspect SOM-VAE does not incorporate the outcome aspect during training; cluster assignments are determined solely by proximity to node embeddings in the temporal latent space. Consequently, the proxy outcome prediction pipeline described in Section 3.4.2 is applied to generate the proxy task used to evaluate performance on the outcome aspect.

Advantages The primary advantage of SOM-VAE for TPC is its topologically structured latent space: the two-dimensional SOM grid preserves neighborhood relationships between cluster prototypes, allowing intuitive visualization of patient similarity and trajectory evolution, e.g., transition from normal to abnormal physiology patterns. In addition, the probabilistic VAE latent space offers greater representational flexibility than the deterministic input-space centroids of TSKM, which may facilitate discovery of a more complex phenotypic structure. The model does not require pre-specification of the target number of clusters; in fact, only the SOM grid dimensions need specification, which provides the ability to discover natural groupings within the data.

Limitations Nevertheless, SOM-VAE exhibits important limitations for TPC. As an unsupervised method, it does not incorporate the outcome aspect during training, so learned phenotypes may not correspond to clinically relevant outcome groups. It balances five hyperparameters $(\alpha_{\text{SV}}, \beta_{\text{SV}}, \gamma_{\text{SV}}, \tau_{\text{SV}}, \eta_{\text{SV}})$, requiring careful tuning and contributing to training instability, particularly in sparse or imbalanced data settings common in healthcare. This instability also reduces the reproducibility between runs. The ability to naturally select a potential subset of the available clusters can also be disadvantageous: it is not directly controllable, as the node population depends on initialization and training dynamics. Finally, while the SOM grid provides structural interpretability, the relationship between node embeddings and the outcome aspect remains opaque without post hoc analysis.

4.2.1.3 AC-TPC

The Actor-Critic for Temporal Phenotypic Clustering (AC-TPC [13]) is a supervised DL method for TPC that leverages an Actor-Critic (AC) training framework [297] to jointly learn cluster representations and a cluster assignment policy, with the outcome aspect explicitly incorporated into the clustering process. Unlike TSKM and SOM-VAE, which operate solely on the temporal aspect, AC-TPC

directly optimizes for outcome-informed cluster formation, it is therefore treated as the primary TPC benchmark in this thesis[3]. A sketch of the model architecture is provided in Figure 4.3.

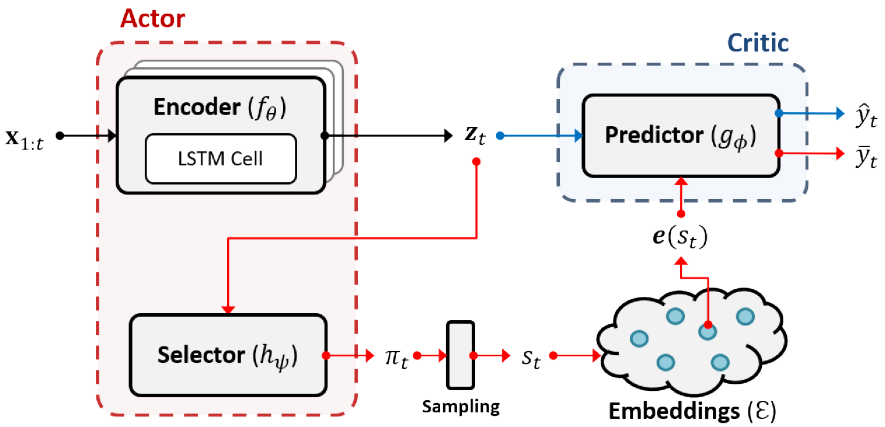


Figure 4.3: Schematic visualization of the AC-TPC architecture [13].

Model Architecture AC-TPC consists of three neural network components: an *encoder*, a *selector*, and a *critic*, together with a set of trainable cluster representation vectors $\mathcal{E}_{\text{ACTPC}} = \{\mathbf{c}_k^{\text{AC}}\}_{k=1}^{K}$, situated in latent space rather than the input space of temporal trajectories. The encoder is an LSTM network capable of producing sequential latent representations of multivariate temporal data, while the selector and critic are both MLPs that model the assignment policy and outcome prediction, respectively. To ensure an output suitable for MCS, the critic's output layer uses a softmax activation, yielding predicted probabilities over outcome classes. The selector's output layer also uses a softmax activation, producing a probability distribution over cluster assignments for each time step.

The forward pass proceeds as follows. A trajectory $\mathbf{X}_n$ is passed to the encoder, which outputs a sequence of latent representations $\mathbf{z}_{n,t}^{\text{AC}}$, for $t = 1, \ldots, T$. Each $\mathbf{z}_{n,t}^{\text{AC}}$ is passed to the selector, which outputs a probability vector $\boldsymbol{\pi}_{n,t}^{\text{AC}} \in \Delta^{K-1}$. A cluster index is sampled from this distribution, $\mathbf{s}_{n,t}^{\text{AC}} \sim \text{Cat}(\boldsymbol{\pi}_{n,t}^{\text{AC}})$, and

[3]The original AC-TPC formulation models the outcome aspect as events that occur *concurrently* with the temporal aspect data. Mathematically, adaptation to the TPC setting in this thesis is straightforward and is discussed below.

the corresponding cluster representation vector $\mathbf{c}_{n,t}^{\text{AC}} \coloneqq \mathbf{c}_{\mathbf{s}_{n,t}^{\text{AC}}}^{\text{AC}}$ is retrieved from $\mathcal{E}$. The critic network then maps $\mathbf{c}_{n,t}^{\text{AC}}$ to a vector of predicted outcome probabilities $\hat{\mathbf{y}}_{n,t}^{\text{AC}} \in \Delta^{D_o-1}$, under a BCS or MCS prediction with D_o classes.

Model Training Training proceeds in three stages: two initialization phases followed by the core AC training procedure. In *first initialization*, the encoder and critic networks are jointly trained in an outcome prediction task, $\mathbf{X}_n \rightarrow \mathbf{z}_{n,t}^{\text{AC}} \rightarrow \hat{\mathbf{y}}_{n,t}^{\text{AC}}$, using categorical cross-entropy, $\mathcal{L}_{\text{CatCE}}$, as a loss function (MCS). This encourages the encoder to produce latent representations that are informative of the outcome aspect prior to cluster formation. The *second initialization* runs $K\text{-}means$ on the full set of training-data[4] latent representations $\left\{\mathbf{z}_{n,t}^{\text{AC}}\right\}_{n=1,t=1}^{N,T}$. The resulting $K\text{-}means$ centroids are used to initialize $\mathcal{E}$, and the selector network is subsequently trained to replicate the $K\text{-}means$ cluster assignments via a cluster prediction task $\mathbf{X}_n \rightarrow \mathbf{z}_{n,t}^{\text{AC}} \rightarrow \boldsymbol{\pi}_{n,t}^{\text{AC}}$. Training continues until convergence or a maximum number of epochs is reached [13], and the resulting model is used as a starting point for the core AC training procedure in the next paragraph.

Model Objective Following initialization, the model is trained end-to-end using a composite loss function:

$$\mathcal{L}_{\text{AC}} = \mathcal{L}_{\text{AC},1} + \alpha_{\text{AC}}\,\mathcal{L}_{\text{AC},2} + \beta_{\text{AC}}\,\mathcal{L}_{\text{AC},3} \tag{4.11}$$

where $\alpha_{\text{AC}}, \beta_{\text{AC}} \geq 0$ are hyperparameters controlling the relative contribution of auxiliary losses, $\mathcal{L}_{\text{AC},2}$, $\mathcal{L}_{\text{AC},3}$. The *primary loss*, $\mathcal{L}_{\text{AC},1}$, is defined as $\mathcal{L}_{\text{CatCE}}$ (in the case of MCS) between the true outcome label of patient n, $\mathbf{y}_n^{\text{mc}}$, and the predicted probabilities $\hat{\mathbf{y}}_{n,t}^{\text{AC}}$, $\mathcal{L}_{\text{CatCE}}(\mathbf{y}_n^{\text{mc}}, \hat{\mathbf{y}}_{n,t}^{\text{AC}})$. Auxiliary terms contribute towards stabilization of training and discovery of more clinically-meaningful phenotypes. $\mathcal{L}_{\text{AC},2}$ is the mean sample-wise entropy of $\boldsymbol{\pi}_{n,t}^{\text{AC}}$:

$$\mathcal{L}_{\text{AC},2} = \frac{1}{NT}\sum_{n=1}^{N}\sum_{t=1}^{T}\left(-\sum_{k=1}^{K}\left(\boldsymbol{\pi}_{n,t}^{\text{AC}}\right)_k \log\left(\left(\boldsymbol{\pi}_{n,t}^{\text{AC}}\right)_k\right)\right) \tag{4.12}$$

[4] see also Section 4.4 on training and testing data splits.

which is minimized when $\boldsymbol{\pi}_{n,t}^{\text{AC}}$ is a one-hot vector, thus encouraging confidence in the cluster assignments. Furthermore, $\mathcal{L}_{\text{AC},3}$ is the sum of pairwise *negative* $\mathcal{L}_{\text{CatCE}}$ losses between the cluster label distributions, defined as the critic output at each $\mathbf{c}_k$:

$$\mathcal{L}_{\text{AC},3} = \sum_{k_1 \neq k_2} \left[-\mathcal{L}_{\text{CatCE}} \left(\hat{\mathbf{y}}_{k_1}^{\text{AC}}, \hat{\mathbf{y}}_{k_2}^{\text{AC}} \right) \right] \tag{4.13}$$

where, slightly abusing notation, $\hat{\mathbf{y}}_{k_i}^{\text{AC}} \in \Delta^{D_o - 1}$ denotes the critic's predicted outcome probability vector when applied to cluster representation vector $\mathbf{c}_{k_i}$, and k_1, k_2 range over all ordered pairs of distinct cluster indices. It thereby encourages diversity in the cluster labels and therefore pushes $\mathcal{E}$ to span diverse regions of the outcome space.

Direct backpropagation through the loss is not applicable due to the non-differentiable categorical sampling of $\mathbf{s}_{n,t}^{\text{AC}}$. The model is therefore trained using a policy-gradient strategy [297], viewing the cluster assignment (encoder and selector) as the actor and the critic (network) as the critic, so that the selector gradients are computed as:

$$\nabla_{\theta_{\text{sel}}} \mathcal{L}_{\text{AC, actor}} = -\hat{V}(\mathbf{c}_{n,t}^{\text{AC}}) \nabla_{\theta_{\text{sel}}} \log \boldsymbol{\pi}_{n,t}^{\text{AC}}(\mathbf{s}_{n,t}^{\text{AC}}) \tag{4.14}$$

where $\hat{V}(\mathbf{c}_{n,t}^{\text{AC}})$ is the predicted outcome value of the critic for the selected cluster, which serves as the advantage signal, and θ_{sel} are the parameters of the selector network. Critic gradients are computed via standard backpropagation on $\mathcal{L}_{\text{AC},1}$. Cluster representation vectors $\mathcal{E}$ are treated as trainable parameters and updated jointly throughout main training.

Outcome Aspect The TPC setting of this thesis requires adaptation of AC-TPC to a different outcome aspect. AC-TPC models events throughout the patient's stay, while TPC considers a prediction of future outcomes in the last time step, $t = T$. Adaptation is achieved by restricting the computation of losses and the

prediction of the outcome to the final index, so that $\hat{\mathbf{y}}^{\text{AC}}_{n,T}$ is the only output used in $\mathcal{L}_{\text{AC},1}$ during training. At inference time, the cluster assigned to the patient n is:

$$\phi(\mathbf{X}_n) = \underset{k=1,\ldots,K}{\arg\max} \, (\boldsymbol{\pi}^{\text{AC}}_{n,T})_k \tag{4.15}$$

so the number of learned (non-empty) clusters is upper bounded by K, but may be lower, as with SOM-VAE. Critically, unlike TSKM and SOM-VAE, AC-TPC directly produces outcome predictions from the cluster representations in $\mathcal{E}$ via the critic network. Therefore, the proxy prediction pipeline described in Section 3.4.2 is not required.

Advantages The primary advantage of AC-TPC for TPC is its explicit incorporation of the outcome aspect during cluster formation: by jointly optimizing cluster assignment and outcome prediction, learned clusters are directly associated with clinical outcomes rather than temporal patterns alone. Additionally, the probabilistic formulation of the selector network provides soft cluster assignments $\boldsymbol{\pi}^{\text{AC}}_{n,t}$, which offer greater modeling flexibility than the hard assignments of TSKM, and allow for principled handling of patients with ambiguous phenotype membership. It also provides a natural framework for incorporating clinically desirable properties directly into the model objective. Losses such as $\mathcal{L}_{\text{AC},1}$ and $\mathcal{L}_{\text{AC},2}$ penalize unwanted properties such as the misalignment between clusters and their corresponding characterizations in the outcome aspect. Finally, the LSTM encoder is well-suited to capture sequential dependencies at the trajectory level, unlike SOM-VAE, which processes individual observation vectors in isolation.

Limitations Nevertheless, AC-TPC has important limitations. The AC optimization process is inherently unstable, due to competing actor and critic objectives, and the method is sensitive to both initialization and hyperparameter selection instabilities that are shared with SOM-VAE, but further compounded by the policy-gradient training scheme. Moreover, the original model was developed and validated primarily in binary outcome settings; its extension to MCS imbalanced

data is not trivial, as the primary loss based on $\mathcal{L}_{\text{CatCE}}$ does not account for class prevalence. This limitation is empirically confirmed in Section 4.5 and directly motivates the weighted extension proposed in Section 4.3. Additionally, cluster representations are not interpretable in input space, particularly since representations are in latent space and have no direct match to input features, an issue addressed in Chapter 5. Finally, the cluster representation vectors $\mathcal{E}$ are static and do not exhibit temporal structure, which precludes direct modeling of the evolution of patient trajectories, as discussed further in Chapter 6.

4.2.2 Proxy Task Benchmarks

The clustering methods described in Section 4.2.1 are not designed for the outcome prediction task, and the performance on the outcome aspect is evaluated through the proxy pipeline of Section 3.4.2. To provide a meaningful performance reference for this proxy task, three additional benchmarks are included, which are *specifically designed* to predict the outcome: SVM, XGBoost, and NEWS2. The first two are well-established classical machine learning classifiers readily adapted to MCS tasks that have shown success in a wide range of clinical prediction tasks [298–301], while NEWS2 is a clinically-derived early warning score widely used in UK hospital practice [19] that serves as a benchmark method grounded in clinical decision-making. It should be noted that more powerful supervised models could in principle achieve higher predictive performance; however, the purpose of this comparison is not to identify the best possible outcome predictor conditional on $\mathbf{X}_n$, but rather to establish whether cluster-based phenotypes can simultaneously support outcome information alongside intrinsic phenotypic properties. Since all three methods are natively defined for tabular input, a temporal adaptation is required to apply them to the trajectory data $\mathbf{X}_n \in \mathbb{R}^{T \times D_f}$; the adaptation strategy for each method is described in the corresponding paragraph below.

4.2.2.1 SVM

Support Vector Machines [302] are discriminative classifiers that identify a decision boundary of maximum margin between classes, regularized by a hyperparameter C_{SVM}, which controls the penalty for margin-violation. To handle non-linear decision boundaries, SVMs can operate in a higher-dimensional, kernel-transformed feature space, where the kernel imposes prior beliefs on data relationships. Common kernels include linear (equivalent to not using any kernel), polynomial, and radial-basis function (RBF), which uses kernel $\kappa(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\gamma L_2(\mathbf{x}_i, \mathbf{x}_j)^2\right)$ with hyperparameter $\gamma > 0$. They are inherently binary classifiers; in MCS, both OvO and OvR strategies can be used and at inference time, OvR chooses the class with the largest decision function value, while OvO uses a majority vote for all pairwise classifiers.

Since SVM operates on tabular input data, two temporal adaptations are considered. The first uses a *flattening* strategy, which maps the input $\mathbf{X}_n \in \mathbb{R}^{T \times D_f}$ to a one-dimensional vector of size $T \cdot D_f$, which is then fed to the SVM model. Although straightforward, this treats each feature-time observation as an independent feature, discarding temporal ordering and scale differences across time steps. In particular, the challenges associated with high-dimensionality become more apparent, potentially degrading model performance. The second approach trains an ensemble of independent SVM classifiers, one per feature or per time step, and aggregates their predicted probability scores to obtain a final prediction. The best-performing adaptation is selected during hyperparameter tuning (see Section 4.4).

4.2.2.2 XGBoost

eXtreme Gradient Boosting [303] is an ensemble method that constructs a predictive model as a weighted additive sequence of shallow decision trees. Each tree is trained to approximate the negative gradient of a differentiable loss function with respect to the current ensemble prediction, so that each successive tree corrects the residual errors of its predecessors through a procedure known as

gradient boosting [304]. At each node, trees are grown by selecting the feature split that maximizes a regularized gain criterion, which penalizes tree complexity to reduce overfitting. The final prediction is obtained by summing the outputs of all trees, scaled by a learning rate hyperparameter. The key hyperparameters include the number of estimators, the maximum tree depth, and the minimum child weight. In MCS, $\mathcal{L}_{\text{CatCE}}$ is used as the training objective, with the predicted class determined by arg max over the per-class scores. XGBoost is also known to be robust to class imbalance via sample reweighting, which is applied here given the severe class imbalance observed in HAVEN (see Section 3.2.1).

The same two temporal adaptations described for SVM are considered: trajectory flattening to a vector of size $T \cdot D_f$, and aggregation of the ensemble per-feature / time step. The best-performing adaptation is selected during hyperparameter tuning (see Section 4.4).

4.2.2.3 NEWS2

The National Early Warning Score 2 [295] is a rule-based clinical deterioration score widely used in acute care settings of the NHS in the UK. Integer scores from 1 to 3 are assigned to deviations from clinically defined normal ranges in each of six physiological parameters (SBP, RR, HR, level of consciousness — through AVPU —, TEMP, and SpO_2), together with an optional component for supplemental oxygen use (FiO_2), which adds a score of 2. SpO_2 has an alternative scale that is applied to patients with hypercapnic respiratory failure (usually due to COPD) who have a clinically recommended SpO_2 value between $88 - 92\%$ and for whom higher SpO_2 values may put at risk. The final score is obtained by aggregating all scores and applying a clinical threshold (score at least 5 or any parameter score value of 3), although, when deriving the performance of NEWS2, the aggregated score without threshold was also evaluated. The aggregate score stratifies patients into low, medium, and high risk of clinical deterioration, and has been validated as a predictor of in-hospital adverse events, e.g. Pimentel et al. [19].

Unlike SVM and XGBoost, NEWS2 involves no learning and has no free parameters; it is a fixed clinical protocol. In the context of this thesis, NEWS2 is applied to the final observation vector $\mathbf{x}_{n,T}$ only, consistent with the intended use as an assessment snapshot, rather than a trajectory-level modeling tool. For HAVEN, the final observation vector corresponds to the time point 24 hours prior to the outcome time, which matched the horizon time for early warning on which NEWS2 has been validated. Furthermore, NEWS2 is evaluated exclusively in HAVEN, as its component variables are not fully available in other datasets, particularly FiO_2 and AVPU, although other clinical alternatives, such as ESI, are introduced accordingly.

4.3 Improving Multiclass Outcome Modeling

This section introduces AC-TPC$_w$, a modification of AC-TPC designed to address the challenges of severely imbalanced multiclass outcome distributions observed in HAVEN. The model retains the core architecture of AC-TPC but proposes changes to the loss function and the training procedure to better capture underrepresented outcome classes, with full details in the following paragraphs.

Model Architecture AC-TPC$_w$ consists of the same three neural network components as AC-TPC: an *encoder*, a *selector*, and a *critic*, together with a set of trainable cluster representation vectors $\mathcal{E}_{\text{ACw}} = \{\mathbf{c}_k^{\text{ACw}}\}_{k=1}^{K}$ in latent space. This returns a single trajectory-level representation, $\mathbf{z}_n^{\text{ACw}} \in \mathbb{R}^d$, similar to selecting the last representation vector in the AC-TPC adaptation mentioned in Section 4.2.1.3. The forward pass otherwise follows AC-TPC directly. The representation $\mathbf{z}_n^{\text{ACw}}$ is passed to the selector, which outputs a cluster assignment probability vector $\boldsymbol{\pi}_n^{\text{ACw}} \in \Delta^{K-1}$. A cluster index is sampled as $\mathbf{s}_n^{\text{ACw}} \sim \text{Cat}(\boldsymbol{\pi}_n^{\text{ACw}})$, and the corresponding cluster representation $\mathbf{c}_n^{\text{ACw}} := \mathbf{c}_{\mathbf{s}_n^{\text{ACw}}}^{\text{ACw}}$ is retrieved from $\mathcal{E}$. The critic network maps $\mathbf{c}_n^{\text{ACw}}$ to a vector of predicted outcome probabilities $\hat{\mathbf{y}}_n^{\text{ACw}} \in \Delta^{D_o-1}$.

Model Objective The AC-TPC$_w$ training objective introduces two modifications to AC-TPC: a weighted primary loss and a reformulated separation loss. Each is described in turn.

Recall that $\mathcal{L}_{\text{CatCE}}$ treats all outcome classes symmetrically: the loss incurred by a sample that is assigned a wrong outcome label is independent of which outcome class it belongs to. Thus, mini-batch gradients are dominated by the majority class, since the expected outcome distribution per batch mirrors that of the training dataset. The consequence is that gradient updates largely ignore minority classes, and the model collapses towards predicting the dominant outcome. Metrics such as F1, which are sensitive to performance on minority classes, are used to investigate model performance; however, they are not directly part of gradient updates.

To address this, AC-TPC$_w$ introduces class-specific weights inversely proportional to class prevalence [305], defining the *weighted categorical cross-entropy* loss:

$$\begin{aligned}\mathcal{L}^{w}_{\text{CatCE}}(\mathbf{y}^{\text{mc}}_n, \hat{\mathbf{y}}^{\text{ACw}}_n \mid \boldsymbol{\alpha}) &\coloneqq -\sum_{o=1}^{D_o} \alpha_o (\mathbf{y}^{\text{mc}}_n)_o \log(\hat{\mathbf{y}}^{\text{ACw}}_n)_o \\ &= -\alpha_{o(\mathbf{y}^{\text{mc}}_n)} \log(\hat{\mathbf{y}}^{\text{ACw}}_n)_{o(\mathbf{y}^{\text{mc}}_n)}\end{aligned} \tag{4.16}$$

where $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_{D_o})$ with $\alpha_o > 0$ is a weight vector for each outcome class, and $o(\mathbf{y}^{\text{mc}}_n)$ is the index of the true outcome class for the patient n. In general, $\sum_{o=1}^{D_o} \alpha_o = 1$ is a common normalization, but any fixed scale is sufficient since the loss is optimized by gradient descent, and the remaining loss terms are scaled by additional hyperparameters. $\mathcal{L}^{w}_{\text{CatCE}}$ sets $\alpha \propto N_o^{-1} = \frac{A}{N_o}$, where N_o is the number of patients with outcome o and is estimated from the training dataset to prevent information leakage. A is a normalization factor such that the weights sum to 1.

$\mathcal{L}^{w}_{\text{CatCE}}$ shares the same optimum as $\mathcal{L}_{\text{CatCE}}$: as functions of $\hat{\mathbf{y}}^{\text{ACw}}_n$, it is minimized if and only if $\hat{\mathbf{y}}^{\text{ACw}}_n = \mathbf{y}^{\text{mc}}_n$, with an attainable minimum 0. However, minority classes now incur proportionally higher penalties for misclassification. Concretely, for two samples n_1, n_2 with equal predicted confidence but belonging to different outcome classes, the ratio of their $\mathcal{L}^{w}_{\text{CatCE}}$ losses equals $\alpha_{o(\mathbf{y}^{\text{mc}}_{n_2})}/\alpha_{o(\mathbf{y}^{\text{mc}}_{n_1})}$ —

exactly the inverse ratio of their class frequencies. This directly counteracts the gradient imbalance described above.

Separately, AC-TPC separation loss $\mathcal{L}_{\text{AC},3}$ encourages diversity by penalizing similarity between cluster label distributions. However, under severe class imbalance, cluster label distributions naturally collapse towards the dominant outcome class, rendering this loss ineffective. AC-TPC$_\text{w}$ instead enforces separation directly in latent embedding space, defining:

$$\mathcal{L}_{\text{ACw},3}(\mathcal{E}_{\text{ACw}}) \coloneqq -\frac{1}{\binom{K}{2}} \sum_{k<k'} L_2(\mathbf{c}_k, \mathbf{c}_{k'}) \tag{4.17}$$

Minimizing $\mathcal{L}_{\text{ACw},3}$ directly encourages $\mathcal{E}_{\text{ACw}}$ to span a large volume of latent space, promoting cluster diversity independently of outcome predictions and without relying on label distributions that collapse under imbalance.

Combining both modifications, the AC-TPC$_\text{w}$ training objective is the same as AC-TPC, with $\mathcal{L}_{\text{CatCE}}$ replaced by $\mathcal{L}^w_{\text{CatCE}}$ in the primary loss, and the separation loss replaced by $\mathcal{L}_{\text{ACw},3}$.

$$\mathcal{L}_{\text{ACw}}(\mathbf{y}^{\text{mc}}_n, \hat{\mathbf{y}}^{\text{ACw}}_n) \coloneqq \mathcal{L}^w_{\text{CatCE}}(\mathbf{y}^{\text{mc}}_n, \hat{\mathbf{y}}^{\text{ACw}}_n \mid \boldsymbol{\alpha}) + \alpha_{\text{ACw}}\, \mathcal{L}_{\text{AC},2} + \beta_{\text{ACw}}\, \mathcal{L}_{\text{ACw},3} \tag{4.18}$$

where $\mathcal{L}_{\text{AC},2}$ is the confidence loss from Equation (4.11), retained unchanged, and $\alpha_{\text{ACw}}, \beta_{\text{ACw}} \geq 0$ are hyperparameters.

Model Training Training of AC-TPC$_\text{w}$ follows the three-stage pipeline of AC-TPC (see Section 4.2.1.3), with $\mathcal{L}^w_{\text{CatCE}}$ substituted for $\mathcal{L}_{\text{CatCE}}$ wherever outcome prediction losses appear. In *first initialization*, the encoder and the critic are jointly trained in the outcome prediction task $\mathbf{X}_n \to \mathbf{z}^{\text{ACw}}_n \to \hat{\mathbf{y}}^{\text{ACw}}_n$ using $\mathcal{L}^w_{\text{CatCE}}$ as the loss function. In *second initialization*, K-*means* is applied to the set of training-data representations $\{\mathbf{z}^{\text{ACw}}_n \mid n = 1, \ldots, N\}$ to initialize $\mathcal{E}_{\text{ACw}}$, and the selector is trained to replicate the K-*means* assignments using $\mathcal{L}_{\text{CatCE}}$[5]. Following initialization, the model is trained end-to-end using $\mathcal{L}_{\text{ACw}}$ and a policy-gradient strategy,

[5] $\mathcal{L}^w_{\text{CatCE}}$ is not required at this stage since the K-*means* assignments are approximately balanced between clusters by construction.

identically to AC-TPC. Training continues until convergence or a maximum number of epochs is reached and $\mathcal{E}_{\text{ACw}}$ is updated jointly throughout. As in AC-TPC, the number of learned clusters is upper bounded by K and may be strictly lower.

4.4 Implementation

This section describes the implementation details specific to the experiments carried out in this chapter, which constitute the first instantiation of the general evaluation framework of Section 3.4 for the HAVEN setting. The full HAVEN cohort is used in all experiments for the purpose of benchmarking.

4.4.1 Model Training

For the purposes of a controlled benchmark analysis, only vital signs were considered as input features for the models, as the critical role of vital signs in clinical deterioration is well-established. For each experiment, data were split patient-wise into optimization data (60%), and hold-out test data (40%). Optimization data was further divided patient-wise into a training set (80%) and a validation set (20%) to allow for the fair comparison of different models and hyperparameter configurations, as shown in the subsequent subsection Section 4.4.2. The distributional parameters used in the data processing steps, such as normalization or imputation, were estimated from training data only and applied to all other sets to prevent information leakage, where information from the 'unseen' data is used to inform the training process. Furthermore, all neural network components were trained using mini-batch SGD with an Adam optimizer [306], with all model parameters treated as trainable. Training was carried out until convergence or until a maximum number of epochs was reached, where convergence was determined as the lack of significant improvements in the target metric (unweighted average of AUROC) for more than 10 epochs. Early stopping was used, based on a minimum loss improvement of 0.1 and patience of 30 epochs, with model checkpoints saved at the epoch with the best performance. The implementations of SOM-VAE and AC-TPC (jointly with AC-TPC$_{\text{w}}$) were adapted

from the original code repositories provided by the authors, with modifications to accommodate the TPC setting and MCS. All other models were implemented using standard libraries, such as `scikit-learn` [307], `XGBoost` [303], and `TSLearn` [151]. Finally, experiments were run with 1 Intel Core $i7 - 8700K$ Central Processing Unit (CPU), and a single NVIDIA GeForce RTX 2080 Graphics Processing Unit (GPU), and GPU training was added wherever applicable.

4.4.2 Hyperparameter Selection

Hyperparameters were explored by using a grid search strategy over the ranges reported in Table A.1, in the Appendix, and selection of the best-performing hyperparameter configuration proceeded as follows. For each model and hyperparameter choice, 5 runs were conducted by varying the random seed over a fixed set of 5 seeds, i.e., shared across all models and hyperparameters, where each seed corresponds to a different random split of the data into training, validation and test sets, as well as any random initialization of model parameters. For each seed, the model was fitted to the training data and evaluated in the validation set. The set of hyperparameters that performed the best was defined as the one with the highest AUROC, averaged between the different seeded experiments. Subsequently, models were re-trained on the combined training and validation datasets, and the performance was evaluated on the test set. The average and standard deviation scores on the test set are reported in Table A.1.

Hyperparameter optimization followed an additional constraint: any configuration that resulted in a fitted model that produced no predictions for one or more classes in the proxy task was disqualified, regardless of its AUROC score. This was deemed necessary because, in severely imbalanced MCS, a model that collapses to predicting only a subset of the classes, particularly the most dominant ones, can achieve a spuriously high average AUROC score. Such a model is clinically incoherent as it fails to provide actionable outcome stratification across all classes, the primary purpose of the proxy task.

4.4.3 Selection of K

The optimal value of K was defined, following an approach adapted from the Elbow Method [71], as the smallest candidate value of K whose unweighted mean macro-averaged AUROC was not statistically significantly lower than that of the best-performing candidate value identified across the full range considered; all candidate values of K were evaluated over 5 seeded experiments each before this comparison was made. Statistical significance of improvements was assessed using a Friedman test with Bonferroni correction [308]: Friedman's hypothesis test framework is a non-parametric approach that tests whether the distributions of ≥ 2 related groups (in this case, experiment scores) are derived from the same distribution, while a Bonferroni correction is needed to control the family-wise error rate under multiple comparisons at once [309]. This outcome-based criterion was chosen for the selection of K because, in the TPC setting considered, clusters are ultimately used as phenotypes to understand and potentially stratify the clinical risk of the patient, so that marginal gains in internal clustering inertia do not necessarily translate into more clinically meaningful clusters. For simplicity of discussion, this approach to selecting the optimal K will henceforth be referred to as the *adapted Elbow method*.

The above procedure determines the optimal value for the hyperparameter K; separate consideration is needed for visualization and qualitative interpretation. TSKM is the only method that returns a fixed number of clusters, so the number of learned clusters is the same in all experiments with varying seeds. This is not the case for other clustering benchmarks, where K serves as an upper bound on the number of non-empty clusters. For visualization and qualitative interpretation of a given model and hyperparameter set, the set of 5 experiments was analyzed, and the most common number of learned clusters across the 5 seeds was identified. For example, suppose that 5 different-seeded runs resulted in $4, 4, 3, 4, 3$ learned clusters, respectively, across experiments with $K = 4$. Then, the most common number of learned clusters is 4, and, for visualization and qualitative interpretation, a representative experiment with 4 learned clusters

was selected, i.e. from seeds $\{1, 2, 4\}$. This ensures that the visualized results reflect a representative and reproducible model behavior.

4.5 Results

As part of this controlled benchmark evaluation, the evaluation framework of Section 3.4 is applied to the HAVEN dataset. The results are organized by aspect: the temporal aspect is discussed in Section 4.5.1, while Section 4.5.2 covers the outcome aspect, with qualitative analysis presented in each subsection.

SOM-VAE, AC-TPC, and AC-TPC$_w$ all returned $K = 4$ as the optimal number of clusters by the adapted Elbow method described in Section 4.4.3. TSKM required separate treatment, as described in Section 4.4.3, and is fixed at $K = 4$ for ease of comparison. Despite this, SOM-VAE frequently exhibited experimental runs without predictions for one or more minority classes, a challenge largely absent in AC-TPC and AC-TPC$_w$ except in configurations disqualified by the constraint described in Section 4.4.2.

4.5.1 Temporal Aspect

The performance results of all benchmark models and AC-TPC$_w$ in the temporal aspect are reported in Table 4.1, according to the VRC, SIL and DBI metrics. The best-performing model for each metric is highlighted in bold. Taking into account

Model	SIL ↑	VRC ↑	DBI ↓
TSKM	**0.35 (.01)**	**554.6 (2.50)**	**1.19 (.08)**
SOM-VAE	0.02 (.08)	12.8 (9.32)	3.89 (.63)
AC-TPC	0.04 (.01)	66.5 (18.7)	2.34 (.80)
AC-TPC$_w$	0.03 (.02)	62.8 (15.2)	2.23 (.53)

Table 4.1: Temporal aspect performance of AC-TPC$_w$ and benchmarks in HAVEN, evaluated using SIL, VRC, and DBI. For each model, results correspond to the best-performing hyperparameter configuration, and are reported as the mean across 5 random seeds (standard deviations in parentheses). Bold values indicate statistically significant best-performing model for each metric ($p < 0.05$, Friedman test with Bonferroni correction). Arrows next to metric names indicate whether higher (↑) or lower (↓) values are preferred.

only the DL models, AC-TPC$_w$ is competitive with AC-TPC in all three metrics, with no statistically significant differences observed using a Friedman test with Bonferroni correction. SOM-VAE exhibits lower performance on VRC and DBI; SIL scores are comparable across DL models from around 0.02 to 0.04, although SOM-VAE exhibits substantially higher variance (± 0.08) compared to AC-TPC (± 0.01) and AC-TPC$_w$ (± 0.02), reflecting the instability of SOM-VAE in learning consistent temporal representations.

TSKM outperforms all DL models in all three metrics, with a particularly large margin in the VRC (554.6 *versus* 12.8, 66.5, and 62.8 for SOM-VAE, AC-TPC, and AC-TPC$_w$, respectively). This result is consistent and not unexpected as DL methods perform clustering in a learned latent space, unlike TSKM, which clusters directly in the input space. Since VRC, DBI, and SIL are computed in the input space and carry an inherent bias towards convex cluster geometry [279, 310], this directly favors the convex input space structure of the clusters learned by TSKM and limits the interpretability of these metrics as a comparative measure across methods.

A qualitative illustration is provided in Figure 4.4, which plots the mean RR trajectory for patients assigned to each cluster. AC-TPC$_w$ identifies four qualitatively distinct temporal trajectories, reflecting meaningful separation in respiratory behavior between clusters. TSKM, by contrast, produces one cluster with a clearly distinct trajectory (cluster 4), while clusters 1–3 exhibit a nearly identical evolution, which is consistent with their correspondence with the dominant outcome class. In contrast, the SOM-VAE cluster outputs form pairs of similar trajectories (clusters 1 and 2 and clusters 3 and 4) and the overall variation of the trajectory over time is markedly lower than in AC-TPC$_w$, consistent with the limited capacity of SOM-VAE to capture temporal dynamics.

4.5.2 Outcome Aspect

Performance in the outcome aspect is shown in Table 4.2. This is a particularly challenging setting for all models, likely due to the severe outcome imbalance

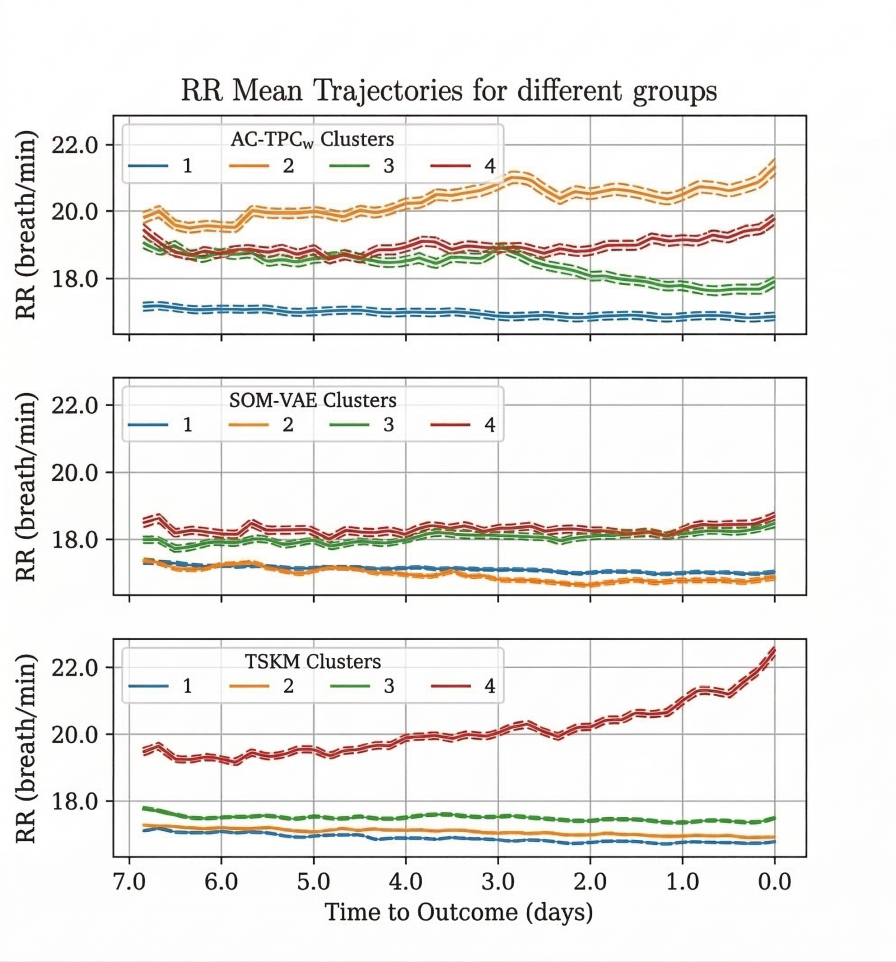


Figure 4.4: Mean respiratory rate (RR, breaths per minute) trajectories for patients assigned to each cluster over time for AC-TPC$_w$ (top), SOM-VAE (third row), and TSKM (bottom). AC-TPC produces similar trajectories to AC-TPC$_w$ and is omitted for clarity. Colors denote cluster membership and are not comparable across models. Means are computed at each time step and shown over the 7 days prior to outcome time, although models were trained using only the 24–72 hour window.

in HAVEN: even well-established supervised classifiers exhibit limitations here, with SVM and XGBoost achieving average AUROC scores of 0.59 and 0.65, respectively. Among the clustering methods, TSKM (0.55) performs below both supervised baselines, while SOM-VAE (0.61) is broadly comparable to SVM and NEWS2. The advantages of outcome-informed TPC are evident in AC-TPC and AC-TPC$_w$, both of which surpass all other benchmarks. AC-TPC$_w$ achieves the highest AUROC (0.70), REC (0.37), and NMI (0.18), and matches AC-TPC in F1 (0.38). All improvements of AC-TPC$_w$ over AC-TPC were statistically significant under a Friedman test with Bonferroni correction. The improved capture of out-

comes relative to TSKM and SOM-VAE is attributable to the end-to-end outcome-informed training shared by both AC-TPC and AC-TPC$_w$, with the weighted loss in AC-TPC$_w$ further addressing class imbalance directly, improving the stratification of minority outcome classes.

Model	AUROC ↑	F1 ↑	REC ↑	NMI ↑
TSKM	0.55 (.01)	0.24 (.03)	0.26 (.02)	0.01 (.03)
SOM-VAE	0.61 (.09)	0.27 (.05)	0.27 (.03)	0.05 (.03)
AC-TPC	0.68 (.01)	**0.38 (.01)**	0.36 (.01)	0.17 (.02)
AC-TPC$_w$	**0.70 (.01)**	**0.38 (.00)***	**0.37 (.01)**	**0.18 (.02)**
SVM	0.59 (.02)	0.23 (.00)*	0.25 (.00)*	–
XGBoost	0.65 (.01)	0.23 (.00)*	0.22 (.00)*	–
NEWS2	0.61	0.29	0.34	–

* Standard deviation rounded to two decimal places.

Table 4.2: Outcome aspect performance of AC-TPC$_w$ and benchmarks in HAVEN, evaluated using AUROC, F1, REC, and NMI. For each model, results correspond to the best-performing hyperparameter configuration and are reported as the mean across 5 random seeds (standard deviations in parentheses). NEWS2 is deterministic and therefore does not report variability. Clustering-models (top) are separated from predictive baselines (bottom). NMI is not applicable to predictive models and is denoted by –. Bold values indicate statistically-significant best-performing model for each metric ($p < 0.05$, Friedman test with Bonferroni correction). Arrows next to metric names indicate whether higher (↑) or lower (↓) values are preferred.

A phenotypic characterization of the learned clusters further supports the quantitative findings. Figure 4.5 illustrates the outcome distribution within each cluster for AC-TPC$_w$, SOM-VAE, and TSKM. TSKM identifies three quasihomogeneous healthy clusters (clusters 1–3) and a single cluster concentrating all adverse events (cluster 4), which does not differentiate between the types of adverse outcomes. SOM-VAE performs worse with respect to outcome separability and all four of its clusters contain a mixture of outcomes without meaningful differences in composition. By contrast, AC-TPC$_w$ identifies two large clusters predominantly composed of patients without events (clusters 1 and 3) and two smaller clusters with distinct adverse event profiles (clusters 2 and 4). In fact, in combination with Figure 4.4, it could be interpreted that cluster 3 contains patients who were poor and then recovered. Moreover, approximately 40% of cardiac arrest patients

are concentrated in cluster 4, while the benchmark models distribute cardiac arrest patients approximately uniformly between clusters, reflecting their inability to capture the structure of the minority class. This concentration is also reflected in the cluster size distribution: the largest AC-TPC$_w$ cluster comprises $N = 14{,}438$ patients, while the second largest contains only $N = 723$, demonstrating the ability of the model to identify clinically meaningful minority phenotypes rather than producing equally sized partitions.

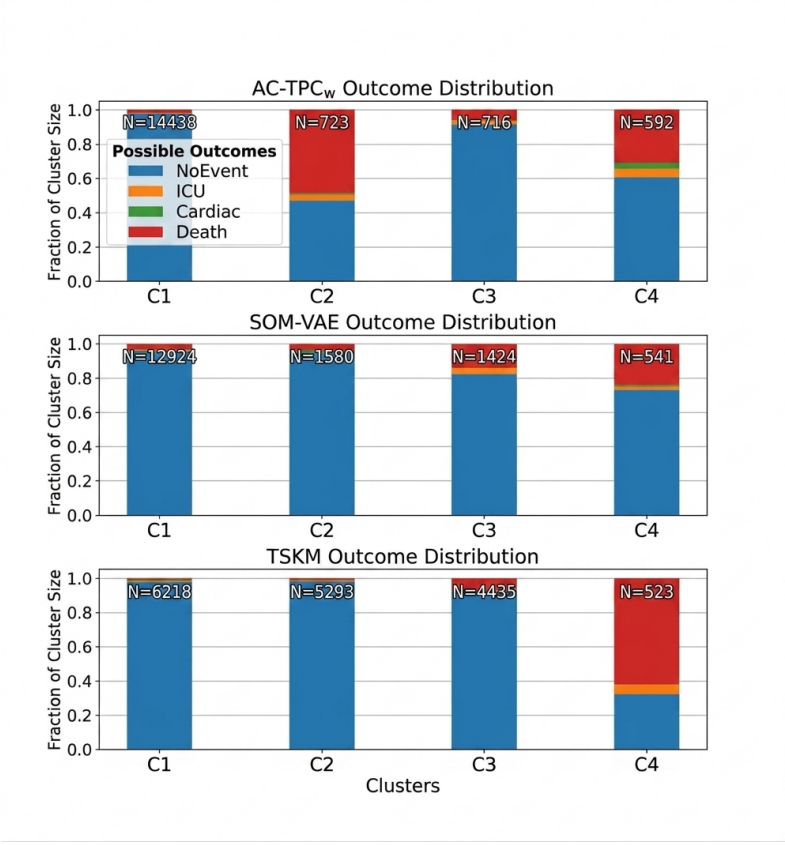


Figure 4.5: Outcome distribution across clusters learned by each model in HAVEN (AC-TPC$_w$, top row; SOM-VAE , middle row; TSKM, bottom row). Clusters are ordered from largest to smallest, with the number of patients indicated above each bar. Colors denote outcomes: NoEvent (blue), ICU (orange), Cardiac (green), and Death (red).

4.6 Conclusion

This chapter carried out a controlled benchmark evaluation of TPC methods in the HAVEN dataset and proposed AC-TPC$_w$ as a principled first extension of TPC

to the multiclass imbalanced setting. The results demonstrate that while classical clustering methods (TSKM) can compete in the temporal aspect, they struggle to identify clinically meaningful phenotypes with respect to the outcome aspect. On the other hand, the outcome-informed training of AC-TPC and AC-TPC$_w$ addresses this gap, with the weighted loss modification in AC-TPC$_w$ further improving the stratification of the minority class, leading to more separable and clinically useful cluster phenotypes. The performance of the temporal aspect of AC-TPC$_w$ remains competitive with AC-TPC, suggesting that loss modification does not compromise cluster quality. Subsequent chapters explore further modifications to address other clinical properties of the TPC problem, including the interpretability of the assignment of patients to clusters, as introduced in Chapter 5.

We are drowning in information
but starved for knowledge.

— John Naisbitt

5 Interpretable Deep Clustering

5.1 Introduction

In Chapter 4, limitations of existing SOTA methods for the TPC task were discussed; moreover, AC-TPC$_w$ was introduced to partially alleviate the challenges associated with multiclass imbalance in TPC settings, through two modifications to the AC-TPC training regimen and objective. However, other limitations are still present across all models, and this chapter considers two particular issues. Firstly, the empirical results of Section 4.5 suggest that meaningful performance gains can be achieved through architectural improvements, specifically through mechanisms that better navigate the heterogeneous nature of the EHR data. Secondly, the interpretability of the model remains unaddressed: no understanding of the reasoning underlying patient assignment to a given cluster is provided, which constitutes an important consideration for clinical deployment.

Both of the aforementioned challenges are addressed in this chapter. A novel end-to-end DL model, CAMELOT, is introduced, which includes a novel encoder architecture that jointly learns weights over the temporal and feature dimensions of the input, and is trained jointly with the remainder of the model. The performance of CAMELOT in the HAVEN-T2RF and MIMIC-IV-ED datasets demonstrates an improved empirical performance in identifying meaningful pheno-

types (relative to the benchmarks), and a novel framework is proposed to visualize and understand the relevance of each feature and time step of the input data to cluster assignment.

5.2 Background

During the past decade, largely driven by the complexity of the EHR data, an ever-increasing number of models have been introduced in the literature specifically designed to improve modeling performance when dealing with EHR patient trajectory data [311]. Such approaches have shown improved success in a variety of clinical tasks compared to benchmarks [312]. Most solutions leverage DL models adapted from standard NN frameworks such as RNNs and CNNs ([313–316]; see also Section 2.3.2). Adaptations to standard architectures are used for two reasons — on the one hand, maintaining the high modeling capacity of DL models, which is needed to handle heterogeneous EHR data, and, on the other hand, employing architectural solutions to address inherent limitations of DL in the context of modeling particular types of EHR data and phenotyping [317, 318].

Despite improvements in the learning of meaningful representations, most solutions fail to address one of the most pervasive criticisms pertaining to DL models and their usefulness in deployment to clinical settings: interpretability. Despite (or perhaps due to) their high capacity, DL models are inherently not interpretable, in the sense that it is not straightforward to understand how a certain model output was derived from the input data [319]. Interpretability is of particular importance in the healthcare domain, where clinical staff must have a good understanding of all information related to the patient to decide on a course of action.

Techniques to address the limitation of interpretability without leading to a significant reduction in modeling capacity have generally followed two separate approaches. The first approach maintains the original, complex, and non-linear model as is and, instead, introduces mechanisms that attempt to extract meaningful information from the model outputs that can be interpreted. Examples

include techniques such as SHAPley additive explanations (SHAP [34, 320]), saliency maps [321], among others [322]. Methods within this avenue of investigation have been shown to lead to erroneous conclusions due to their post hoc approach to interpretability, while also not providing any benefits related to improving the capacity to model EHR data [33].

A separate strategy has focused on architectural changes that internally incorporate subcomponents that can be leveraged to derive some degree of interpretability. One of the most prominent ideas includes the attention mechanism [213]. Given data for patient n, $\mathbf{X}_n$, three matrices of learned linear projections, queries ($\mathbf{Q}$), keys ($\mathbf{K}$), and values ($\mathbf{V}$), are computed as part of the training process. The *attention weights*, and the output are respectively derived according to the equation:

$$\mathbf{W}_{\text{att}} = \text{Softmax}(\frac{1}{\sqrt{d_k}}\mathbf{Q}\mathbf{K}^\top), \qquad \mathbf{O} = \mathbf{W}_{\text{att}}\mathbf{V} \tag{5.1}$$

where d_k is the dimension of the key vectors. The weights are sample-specific and dynamic: in temporal settings, the model can focus selectively on the most critical time steps. As the computation in Equation 5.1 relies almost entirely on linear operations, with the sole non-linearity being the Softmax() function, attention weights admit a natural interpretation as a relevance map, indicating the relative importance of each component of the input to the model output. For simplicity, attention will be used throughout to interchangeably refer to the attention mechanism and the attention weights, with the intended meaning derived from context.

Attention mechanisms and variants have shown success in many areas [323–325], including temporal EHR modeling [133, 313] and [326], demonstrating better quality in the learned representations and generating meaningful clinical insights. Nevertheless, they remain limited in describing the relevance of all components of the input data. More specifically, attention was originally designed for tasks related to language processing, where a sequence of words is preprocessed into a sequence of high-dimensional token vectors; consequently, attention identifies the most relevant tokens to the task at hand. Standard attention

mechanisms, therefore, provide interpretability along a single dimension only — in TPC, this translates to temporal attention weights.

In clinical settings, it is very useful to understand the relevance maps across both feature and temporal dimensions, i.e. to estimate the relevance of a particular feature and time step pair to the output of the model. This gives rise to a dimensionality limitation in the attention mechanism, as this requires providing an interpretable explanation along two dimensions. To address this, the authors in Shamout et al. [327] proposed an attention-variant by considering a set of trajectory-specific RNNs that are followed by an attention layer, with the layer outputs concatenated as the final output vector. A two-dimensional attention map was derived by aggregating the maps of each layer. Other works [328, 329] instead proposed learning attention weights across the feature dimension, prior to being relayed to a temporal RNN/attention layer. Existing methodologies fail to simultaneously consider both feature and time dimensions when estimating representations of patient data. Furthermore, to the best of our knowledge, no methods have been proposed that are capable of determining specific two-dimensional relevance maps within the context of TPC.

5.3 CAMELOT

This chapter introduces a novel model termed Cluster-bAsed iMportancE Learning fOr Time series, henceforth denoted as CAMELOT. CAMELOT proposes several new contributions over previous work. First, a novel interpretable model is introduced, an Attention RNN Encoder, denoted AttentionRNNEncoder, designed to characterize the relevance of each feature-time pair in the assignment to a cluster. This is done using an attention-based mechanism capable of joint modeling of both the feature and temporal dimensions (see also Section 5.3.2). Modified loss functions are also incorporated to effectively manage multiclass imbalance and to encourage cluster exploration and assignment, based on the strategy used in Chapter 4. Lastly, an improved training framework is introduced to allow joint learning of model weights, attention weights, and cluster representation

vectors, as part of a fully end-to-end pipeline. A high-level diagram of the model architecture is shown in Figure 5.1.

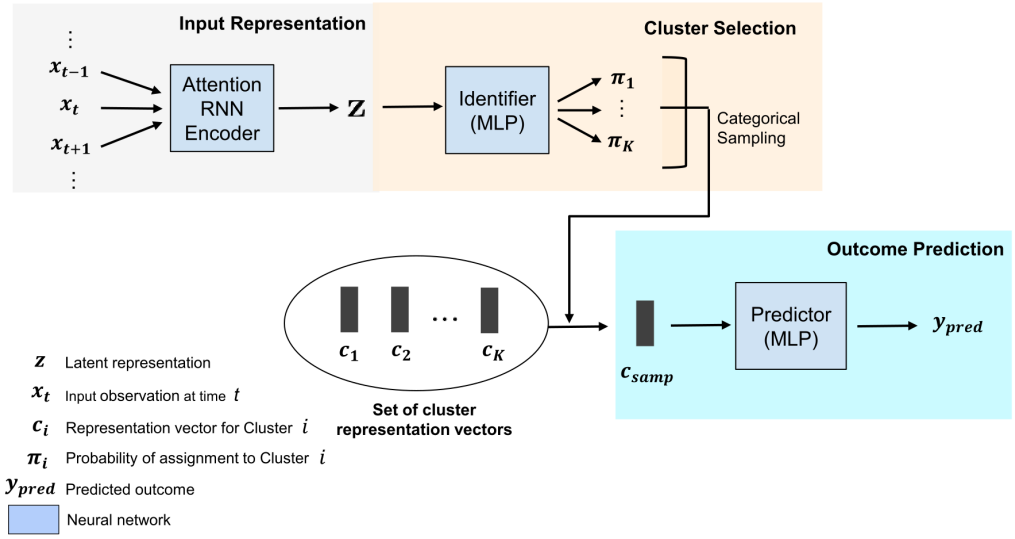


Figure 5.1: Schematic visualization of the CAMELOT model architecture. Input patient trajectories are first encoded by the proposed AttentionRNNEncoder module to produce a latent representation. This representation is passed to the Identifier network, which outputs a probability distribution over clusters. A cluster representation vector is sampled according to this distribution and provided as input to the Predictor, which produces the final outcome prediction.

Data notation from Chapter 4 is adopted — data for the temporal aspect of the patient n are written as $\mathbf{X}_n$, and data for the outcome aspect are written as $\mathbf{y}_n^{\text{mc}} \in \Delta^{D_o-1}$. Finally, for any given model or network block, M, act_M denotes the function associated with a pass through the model, i.e., $\text{act}_M(\cdot)$ is the output of the model M for a given input.

5.3.1 Model architecture

CAMELOT is made up of three subcomponents, an Encoder (Enc), an Identifier (Iden), and a Predictor (Pred), together with a set of learnable cluster representation vectors, $\mathcal{E}^{\text{CAM}} := \{\mathbf{c}_k^{\text{CAM}}\}_{k=1}^{K}$. The Identifier and Predictor are implemented as MLPs, each comprising a stack of feed-forward dense layers with independently set hyperparameters. The Encoder is implemented as an AttentionRNNEncoder

model (see Section 5.3.2), which is a novel framework based on an RNN architecture, designed to improve the quality of representation of input data and also to introduce a mechanism for feature-time interpretability.

A model pass, or call, for CAMELOT proceeds as follows. Given the input trajectory for patient n, $\mathbf{X}_n$, a latent representation is first computed as $\mathbf{z}_n^{\text{CAM}} := \text{act}_{Enc}(\mathbf{X}_n) \in \mathbb{R}^l$, where l is a latent dimension hyperparameter set as described in Section 5.3.3. The latent representation is then mapped by the Identifier, which computes the assignment probability to clusters, $\boldsymbol{\pi}_n^{\text{CAM}} := \text{act}_{Iden}(\mathbf{z}_n^{\text{CAM}}) \in \Delta^{K-1}$, where $\pi_{n,k}^{\text{CAM}}$ models the assignment probability of patient n to cluster k.

During the training phase, a cluster index is sampled from $\boldsymbol{\pi}_n^{\text{CAM}}$ according to a categorical distribution, i.e., $k_n^{\text{CAM,samp}} \sim \text{Cat}(\boldsymbol{\pi}_n^{\text{CAM}})$. Then, the corresponding cluster representation vector $\mathbf{c}_{k_n^{\text{CAM,samp}}}^{\text{CAM}}$ is retrieved from $\mathcal{E}^{\text{CAM}}$. The outcome of CAMELOT is then defined as $\hat{\mathbf{y}}_n^{\text{CAM}} := \text{act}_{Pred}(\mathbf{c}_{k_n^{\text{CAM,samp}}}^{\text{CAM}}) \in \Delta^{D_o-1}$, where $\hat{y}_{n,o}^{\text{CAM}} \in \mathbb{U}$ models the estimated probability of the outcome o for the patient n. On the other hand, at inference time (post-training), the stochastic sampling of the cluster index is replaced in order to obtain a deterministic output. Instead, CAMELOT assigns patient n to the cluster to which it is most likely to belong, according to the rule:

$$\phi(\mathbf{X}_n) = k_n^{\text{CAM,pred}} := \underset{k \in [K]}{\arg\max}(\pi_{n,k}^{\text{CAM}}) \tag{5.2}$$

The remainder of the pipeline then proceeds as described above.

After training, each cluster k is directly associated with an outcome prediction vector, $\mathbf{o}_k^{\text{CAM,clus}} := \text{act}_{Pred}(\mathbf{c}_k^{\text{CAM}}) \in \Delta^{D_o-1}$, referred to henceforth as the outcome characterization of cluster k. Furthermore, the temporal characterization of the cluster k is defined by the mean trajectory of the patients assigned to cluster k according to the inference rule for cluster assignment applied to the training data. Lastly, a feature-time relevance map is assigned to each cluster. This consists of a two-dimensional heatmap indicating the relevance of each feature-time pair of values for the assignment of each patient to the cluster. This is derived from the Encoder subcomponent according to Section 5.3.4.

A key architectural property of CAMELOT is that outcome predictions are mediated entirely through cluster representation vectors, restricting the Predictor input to one of K discrete possibilities. This clustering bottleneck, analogous to that discussed in Section 4.3, implies that strong performance is contingent on two conditions: confident assignment of patients to clusters must be achievable, and the combination of temporal and outcome characteristics should be sufficiently distinct between clusters.

5.3.2 Encoder Architecture

More detail on the AttentionRNNEncoder architecture underlying the encoder subcomponent and its custom attention mechanism is provided in this section. The Encoder is designed to simultaneously achieve two goals: deriving a latent patient representation $\mathbf{z}_n^{\text{CAM}}$ and producing relevance maps, also called attention maps. These are weight matrices that characterize the relative contribution of each feature-time pair to the assignment of patient n to the cluster k. At a high level, the proposed attention mechanism differs from standard attention-based mechanisms as follows. Unlike generating near-linear representations of the input data via the key, query, and value matrices (Equation (5.1)), the approach considered leverages the full modeling complexity of an LSTM model and, instead, approximates the resulting RNN outputs. Approximation is done via a set of single feedforward layers; while each layer does not use keys, values, or queries, they also represent near-linear maps, in the sense that they apply an activation function to an affine transformation. A schematic diagram of the AttentionRNNEncoder layer is provided in Figure 5.2.

5.3.2.1 Custom Attention Mechanism

The AttentionRNNEncoder layer comprises two substructures: an RNN block of stacked LSTM layers, and D_f independent feedforward layers without hidden components, $\mathbf{U}_1, \ldots, \mathbf{U}_{D_f}$. Each network $\mathbf{U}_f$ is *feature-specific* and maps a scalar

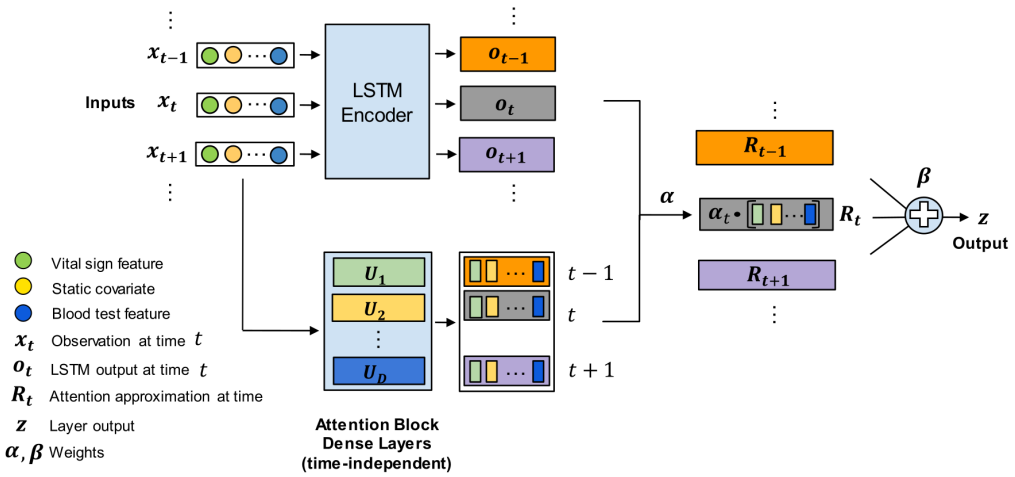


Figure 5.2: Overview of the AttentionRNNEncoder layer. Input $\mathbf{X}_n$ is processed by stacked LSTM layers to produce temporal outputs $\mathbf{o}_1, \ldots, \mathbf{o}_T$. At each time step t, the observation $\mathbf{x}_{n,t}$ is mapped to a feature matrix $\mathbf{R}_{n,t}$ via a feedforward network. The LSTM outputs are expressed as linear combinations of the columns of $\mathbf{R}_{n,t}$, and the resulting aggregation forms the Encoder output $\mathbf{z}_n^{\text{CAM}}$.

$x \in \mathbb{R}$ to a vector in $\mathbb{R}^l$ via:

$$\text{act}_{\mathbf{U}_f}(x) \coloneqq \boldsymbol{\sigma}(\mathbf{D}_f\, x + \mathbf{b}_f) \tag{5.3}$$

where $\mathbf{D}_f \in \mathbb{R}^l$ and $\mathbf{b}_f \in \mathbb{R}^l$ are the learnable weight and bias vectors of $\mathbf{U}_f$, respectively. The function $\boldsymbol{\sigma}(\cdot)$ applies an elemental scalar activation σ to its argument. Two conditions are imposed on σ: it must match the output activation of the final LSTM layer, ensuring that outputs of both substructures lie within a commensurate range; and it is encouraged to be monotonic (e.g., $\sigma(\cdot)$), which supports the near-linear relevance interpretation, in analogy with the standard attention mechanism of Section 5.2.

Individual weight and bias vectors are aggregated column-wise into matrix objects $\mathbf{D} \coloneqq \left[\mathbf{D}_1, \ldots, \mathbf{D}_{D_f}\right] \in \mathbb{R}^{l \times D_f}$ and $\mathbf{B} \coloneqq \left[\mathbf{b}_1, \ldots, \mathbf{b}_{D_f}\right] \in \mathbb{R}^{l \times D_f}$, respectively. For a single time slice, $\mathbf{x}_{n,t} \in \mathbb{R}^{D_f}$, a pass through the set of feedforward layers is as follows:

$$\mathbf{R} \coloneqq \text{act}_{\text{AttL}}(\mathbf{x}_{n,t}) = \boldsymbol{\sigma}(\mathbf{D} \odot \mathbf{x}_{n,t} + \mathbf{B}) \in \mathbb{R}^{l \times D_f} \tag{5.4}$$

where $\odot$ denotes element-wise column scaling, so $(\mathbf{D} \odot \mathbf{x}_{n,t}) \in \mathbb{R}^{l \times D_f}$ has (i, f)-entry $D_{i,f} \cdot x_{n,t,f}$. Then, $\mathbf{r}_f = \boldsymbol{\sigma}(\mathbf{D}_f\, x_{n,t,f} + \mathbf{b}_f)$, where the f-th column of $\mathbf{R}$ is written as $\mathbf{r}_f \in \mathbb{R}^l$, confirming that each column is derived solely from $x_{n,t,f}$; this independence between features is a deliberate design choice, as discussed below in Section 5.3.2.2.

5.3.2.2 Encoder Pass

A pass through the Encoder proceeds as follows. The input $\mathbf{X}_n$ is fed into the RNN block, producing output vectors $\mathbf{o}_{n,1}^{\text{CAM}}, \ldots, \mathbf{o}_{n,T}^{\text{CAM}}$, where $\mathbf{o}_{n,t}^{\text{CAM}} \in \mathbb{R}^l$ encodes a high-complexity representation of patient information up to time t. Separately, at each time step t, $\mathbf{x}_{n,t}$ is given as input to the feedforward layers $\{\mathbf{U}_i\}$, which produces the matrix $\mathbf{R}_{n,t} \in \mathbb{R}^{l \times D_f}$ with columns $\mathbf{r}_{n,t,1}, \ldots, \mathbf{r}_{n,t,D_f}$. The LSTM output at time t, $\mathbf{o}_{n,t}^{\text{CAM}}$, is approximated as a linear combination of these columns, with weight vector $\boldsymbol{\alpha}_{n,t}^{\text{CAM}} \in \mathbb{R}^{D_f}$:

$$\mathbf{o}_{n,t}^{\text{CAM}} \approx \sum_{f=1}^{D_f} \mathbf{R}_{n,t,f} \alpha_{n,t,f}^{\text{CAM}} = \mathbf{R}_{n,t} \boldsymbol{\alpha}_{n,t}^{\text{CAM}} \tag{5.5}$$

Equation (5.5) can be minimized via a well-known least-squares criterion, with closed-form solution:

$$\hat{\boldsymbol{\alpha}}_{n,t}^{\text{CAM}} = \left(\mathbf{R}_{n,t}^\top \mathbf{R}_{n,t}\right)^{-1} \mathbf{R}_{n,t}^\top \mathbf{o}_{n,t}^{\text{CAM}} \tag{5.6}$$

under the assumption that $\mathbf{R}_{n,t}^\top \mathbf{R}_{n,t} \in \mathbb{R}^{D_f \times D_f}$ is invertible. In practice, Equation (5.6) is evaluated using the `LinearRegression` class of Scikit-Learn [330], which handles the case where invertibility does not hold. The optimal approximation to $\mathbf{o}_{n,t}^{\text{CAM}}$ is then $\hat{\mathbf{o}}_{n,t}^{\text{CAM}} := \mathbf{R}_{n,t} \hat{\boldsymbol{\alpha}}_{n,t}^{\text{CAM}} \in \mathbb{R}^l$.

Finally, a weighted linear combination of $\hat{\mathbf{o}}_1^{\text{CAM}}, \ldots, \hat{\mathbf{o}}_T^{\text{CAM}}$ is calculated to obtain the representation vector of the patient n, as follows:

$$\mathbf{z}_n^{\text{CAM}} := \sum_{t=1}^{T} \beta_{n,t}\, \hat{\mathbf{o}}_{n,t}^{\text{CAM}} \tag{5.7}$$

where scalar weights $\beta_{n,t}$ are patient-specific, learnable parameters of AttentionRNNEncoder, optimized as part of the global training process. The resulting $\mathbf{z}_n^{\text{CAM}}$ constitutes the Encoder output and is subsequently passed to the Identifier.

5.3.2.3 Design Rationale

The approximation of $\hat{\mathbf{o}}_{n,t}^{\text{CAM}}$ as a linear combination of feature representations serves two purposes. Firstly, it acts as a regularization on the RNN block: the high-capacity LSTM is encouraged to produce outputs that are approximately expressible as near-linear combinations of input features, which is expected to improve optimization stability. Secondly, the near-linearity of Equation (5.3) makes the learned weights directly interpretable as feature relevance indicators and this is exploited in the derivation of cluster-level feature-time relevance maps, as described in Section 5.3.4. The time-independence of $\mathbf{U}_1, \ldots, \mathbf{U}_{D_f}$ is also a deliberate design choice. Were the feature transformations to vary across time steps, the relative feature-time relevance would be confounded with differences in the transformation itself, preventing clean disentanglement of the two effects.

5.3.3 Model Training

CAMELOT is an end-to-end model optimized by considering three distinct loss functions: a prediction loss, which acts as the primary loss, and two auxiliary losses. The first auxiliary loss encourages cluster exploration, while the second encourages separation of cluster representation vectors. Prior to the main training phase, two rounds of pre-training initialization are also performed. In general, the training process is similar to that of AC-TPC$_w$ (Chapter 4), although with some modifications to accommodate the architectural differences of CAMELOT and the empirical observations during training.

5.3.3.1 Main Training Phase

During training, the model can be trained by backpropagation with respect to a global loss function, which is a weighted sum of three loss functions. Each is described in turn below.

Prediction Loss $\mathcal{L}_{\text{pred}}$ is relabeled from the weighted cross-entropy formulation of Section 4.3 (Equation (4.16)). Define the weights ω_c as normalized and inversely proportional to the class distribution, then:

$$\mathcal{L}_{\text{pred}}(\mathbf{y}_n, \hat{\mathbf{y}}_n^{\text{CAM}}) = -\sum_{c=1}^{D_o} \omega_c \, y_{n,c} \log \hat{y}_{n,c}^{\text{CAM}} \tag{5.8}$$

As discussed in Section 4.3, the weighting scheme helps the model handle the high level of class imbalance in the datasets considered. Furthermore, it is minimized if and only if $\hat{\mathbf{y}}_n^{\text{CAM}} = \mathbf{y}_n$ (in which case $\mathcal{L}_{\text{pred}} = 0$), which is the desired outcome for the model to achieve.

Distribution and Cluster Separation Losses The distribution loss encourages cluster exploration, discouraging degenerate solutions in which all patients are assigned to a limited number of clusters. Given the probability vector representing the patient n's cluster assignment, $\boldsymbol{\pi}_n^{\text{CAM}}$, which is the output of the Identifier block, the mean probability of assignment at the population-level is defined as:

$$\hat{\boldsymbol{\pi}}^{\text{CAM}} := \frac{1}{N} \sum_{n=1}^{N} \boldsymbol{\pi}_n^{\text{CAM}} \tag{5.9}$$

This vector represents the average probability of assignment to each cluster k in the patient population. Note that it forms a well-defined distribution over the set of clusters ($[K]$). Then, the distribution loss, $\mathcal{L}_{\text{dist}}$, is defined as the negative entropy of $\hat{\boldsymbol{\pi}}^{\text{CAM}}$:

$$\mathcal{L}_{\text{dist}}(\hat{\boldsymbol{\pi}}^{\text{CAM}}) := \sum_{k=1}^{K} \hat{\pi}_k^{\text{CAM}} \log \hat{\pi}_k^{\text{CAM}} = -\mathcal{H}(\hat{\boldsymbol{\pi}}^{\text{CAM}}) \tag{5.10}$$

where $\mathcal{H}$ denotes Shannon entropy [291].

From Equation (5.10), the minimization of $\mathcal{L}_{\text{dist}}$ is equivalent to the maximization of the entropy of $\hat{\boldsymbol{\pi}}^{\text{CAM}}$. It is well known that the entropy of a discrete distribution over K categories is maximized by a uniform distribution, with a maximal value $\log K$ [331]. Consequently, $\mathcal{L}_{\text{dist}}$ attains its minimal value of $-\log K$ at $\hat{\boldsymbol{\pi}}^{\text{CAM}} = [1/K, \ldots, 1/K]$, i.e., when no cluster is preferentially assigned at the

population level. Thus, minimization of $\mathcal{L}_{\text{dist}}$ encourages the model to explore all available clusters, reducing the risk of the degenerate solutions mentioned above.

Finally, the cluster representation loss introduced in Section 4.3 is used to encourage the dispersion of cluster representation vectors across the latent space, rather than the convergence towards a narrow region. Here, it is relabeled as $\mathcal{L}_{\text{clus}}$, and denotes the average L_2 distance between the representation vectors of the clusters.

Global Loss and Optimization The global loss, $\mathcal{L}_{\text{CAMELOT}}$, is defined as the sum of the three losses:

$$\mathcal{L}_{\text{CAMELOT}} = \mathcal{L}_{\text{pred}} + \lambda_1 \mathcal{L}_{\text{dist}} + \lambda_2 \mathcal{L}_{\text{clus}} \tag{5.11}$$

where $\lambda_1, \lambda_2 \in \mathbb{R}_{\geq 0}$ are hyperparameters that control the relative importance of the auxiliary losses with respect to the primary prediction loss. The optimization of CAMELOT is performed with respect to all the learnable parameters, including those of the Encoder, Identifier, Predictor, and the cluster representation vectors $\mathcal{E}^{\text{CAM}}$.

Unfortunately, $\mathcal{L}_{\text{CAMELOT}}$ is not directly differentiable with respect to all parameters, due to the non-differentiable categorical sampling used to select a cluster representation vector during training. Specifically, the issue lies in the propagation of gradients through the Encoder and Identifier with respect to $\mathcal{L}_{\text{pred}}$, as the sampling step is non-differentiable. Unlike the AC gradient matching strategy introduced in Equation (4.14), CAMELOT uses a relaxation technique: the sampled cluster index $k_n^{\text{CAM,samp}}$ is represented by its one-hot encoding $\mathbf{e}_{k_n^{\text{CAM,samp}}}$, and the selected cluster vector is expressed as the weighted sum $\sum_{k=1}^{K} (\boldsymbol{\pi}_n^{\text{CAM}})_k \mathbf{c}_k^{\text{CAM}}$. It is noted that the gradient of $\mathcal{L}_{\text{dist}}$ with respect to the Predictor parameters is identically zero; other loss-parameter pairs that are decoupled in this way also exist; however, this does not affect the overall training process.

Training proceeds in mini-batches of patient data from the training data. At each iteration, a batch is passed through CAMELOT, gradients are computed

with respect to $\mathcal{L}_{\text{CAMELOT}}$, and parameters are updated via Adam optimizer [306] with a learning rate hyperparameter. After all batches have been processed, the model is evaluated on the validation data, the batches are reshuffled, and training continues for a new epoch. The number of epochs constitutes a further hyperparameter.

5.3.3.2 Initialization

A pre-training routine is implemented prior to main training, consisting of two consecutive subprocedures, adapted from that of Section 4.3. In the first stage, the Encoder and Predictor networks are trained on a pure classification task, with the clustering component disregarded. This is achieved by passing the Encoder output directly to the Predictor, $\mathbf{X}_n \rightarrow \text{act}_{Enc}(\mathbf{X}_n) \rightarrow \text{act}_{Pred}(\text{act}_{Enc}(\mathbf{X}_n))$, and optimization is performed only with respect to the $\mathcal{L}_{\text{pred}}$ loss. Subsequently, latent representations are computed for all patients in the training data, $\mathbf{z}_n^{\text{CAM}} = \text{act}_{Enc}(\mathbf{X}_n)$, and clustered by an $K\text{-}means$ algorithm with K clusters. The hyperparameter K is shared between the initialization and the main training phases, and it is set at the beginning of the experiment. Cluster representation vectors $\mathbf{c}_k^{\text{CAM}}$ are initialized as the resulting $K\text{-}means$ centroids. The second initialization phase involves training the Identifier block to replicate the $K\text{-}means$ cluster assignments using $\mathcal{L}_{\text{CatCE}}$ (introduced in Section 4.2.1.3) as the loss function; importantly, the Encoder weights are frozen during this phase, ensuring that the latent representations $\mathbf{z}_n^{\text{CAM}}$ remain constant.

5.3.3.3 Cluster Collapse

The TPC objective encompasses two aspects, the temporal and the outcome aspects, whose learning signals typically conflict during their joint optimization. During the development of CAMELOT, a recurring phenomenon was regularly observed, termed *cluster collapse*, which refers to the premature convergence of the model into one of two types of degenerate configurations:

1. Convergence of cluster representation vectors towards a narrow region of the latent space.

2. Assignment of all patients to a strict subset of available clusters.

Note that the former is defined in qualitative terms, although it would not be difficult to formalize the definition by setting thresholds on the distance between cluster representation vectors. However, in practice, the phenomenon was easily identifiable and the discussion that follows is not affected by the lack of a mathematical definition.

Both configurations are undesirable. In the first, the cluster representation vectors coalesce, which causes modeling issues with the Predictor and Identifier. More specifically, it must be the case that the outcome prediction vectors are similar, or, as an alternative, the Predictor output must be highly variable for close inputs, which suggests it denotes a model with high variance or with low confidence in outcome prediction. The Identifier is also affected, as the proximity of cluster representation vectors leads to higher uncertainty in cluster assignment, with small changes in the latent representation of patients leading to different cluster assignments.

The second configuration (Item 2) is problematic for a different set of reasons. A small number of distinct clusters are explored, reducing phenotypic separability. In particular, this affects the outcome aspect strongly, as it is of a much lower dimensionality than the temporal aspect. An additional optimization pathology arises in this configuration: for those clusters without assigned patients, in other words $\{\mathbf{c}_k^{\text{CAM}} \mid \nexists n\colon\ \phi(\mathbf{X}_n) = k\}$, the loss $\mathcal{L}_{\text{CAMELOT}}$ can be decreased without any lower bound by moving its representation vectors further apart, since $\mathcal{L}_{\text{clus}}$ has no finite minimizer. This is mathematically possible *if* $l > K$, a condition satisfied in most practical deployments of CAMELOT due to the expressivity requirements of the latent space, although it is not necessarily the case that it will occur. It was also empirically confirmed during model development. This constitutes a

degenerate training path in which $\mathcal{L}_{\text{CAMELOT}}$ decreases without any update to the clinically relevant model parameters.

The empirical occurrences of the above degenerate cases are believed to be caused by the severe class imbalance in the data, analogous to the failure mode of TSKM to identify only the most prevalent outcome class (Section 4.5). The auxiliary losses address each collapse type directly: $\mathcal{L}_{\text{clus}}$ promotes the separation of cluster representation vectors in each training iteration, targeting the first type; and the gradients of $\mathcal{L}_{\text{dist}}$ encourage the Identifier to assign patients uniformly across all clusters, targeting the second.

An alternative strategy to mitigate cluster collapse would be to modify the training procedure so that clusters with no assigned patients are excluded from the clustering loss $\mathcal{L}_{\text{clus}}$. However, this approach would make the objective function discontinuous across iterations (batches or epochs): As the clusters switch between populated and unpopulated, their contribution to $\mathcal{L}_{\text{clus}}$ would appear or disappear similarly. In practice, this risks leaving some clusters permanently unexplored once they become vacant. On the other hand, the proposed formulation maintains a globally differentiable loss that penalizes cluster collapse without the introduction of such discontinuities.

5.3.4 Attention Map Visualization

Per Section 5.3.2, the Encoder output $\mathbf{z}_n^{\text{CAM}}$ is expressed as a weighted linear combination, with weights $[\hat{\boldsymbol{\alpha}}_{n,t}, \beta_t]_{t=1}^{T}$, of approximated observation representations $\hat{\mathbf{o}}_{n,t}^{\text{CAM}}$. The latter are near-linear transforms of the input data, obtained by composing an element-wise activation with an affine transformation (layers $[\mathbf{U}_f]_{f=1}^{D_f}$). In this section, the procedure for amalgamating and normalizing the learned Encoder weights into a two-dimensional feature-time relevance map for each cluster is described. This procedure does not affect the training or inference pipelines of CAMELOT; it operates solely on the learned parameter weights after training.

Step 1: Feature-Weight Normalization For each time step t, the feature weights $\hat{\boldsymbol{\alpha}}_{n,t}^{\text{CAM}}$ are normalized across the feature dimension by applying a softmax applied to their absolute values:

$$\mathbf{s}_{n,t}^{\text{CAM}} := \text{Softmax}(|\hat{\boldsymbol{\alpha}}_{n,t}^{\text{CAM}}|) = \frac{\exp|\hat{\alpha}_{n,t,f}^{\text{CAM}}|}{\sum_{f'=1}^{D_f} \exp|\hat{\alpha}_{n,t,f'}^{\text{CAM}}|} \tag{5.12}$$

so that $\sum_{f=1}^{D_f}(s_{n,t,f}^{\text{CAM}}) = 1$ for each n and t, by construction. Three properties motivate this normalization. First, softmax renders the weights $\mathbf{s}_{n,t}^{\text{CAM}}$ comparable between time steps, since all entries are standardized to sum to one. Second, exponential terms produce greater contrast between high- and low-relevance features relative to other normalization schemes, making the most relevant features more clearly distinguishable. Third, the absolute value is applied to account for negative weights: under the near-linear interpretation of Section 5.3.2, a large-magnitude weight in either direction indicates a strong relevance to cluster assignment.

Step 2: Cluster-Specific Temporal Weights For each cluster $k \in [K]$, the cluster representation vector $\mathbf{c}_k^{\text{CAM}}$ is approximated as a linear combination of the matrix $\hat{\mathbf{O}}_n \in \mathbb{R}^{l \times T}$, whose t-th column is $\hat{\mathbf{o}}_{n,t}^{\text{CAM}}$:

$$\mathbf{c}_k^{\text{CAM}} \approx \sum_{t=1}^{T} \hat{\mathbf{o}}_{n,t}^{\text{CAM}} \gamma_{k,t}^{\text{CAM}} = \hat{\mathbf{O}}_n \boldsymbol{\gamma}_{n,k}^{\text{CAM}} \tag{5.13}$$

where $\boldsymbol{\gamma}_{n,k}^{\text{CAM}} \in \mathbb{R}^T$ collects the scalar weights $\gamma_{k,t}$ for $t = 1, \ldots, T$. Analogously to Equation (5.6), the optimal weights are obtained using a least-squares criterion:

$$\boldsymbol{\gamma}_{n,k}^{\text{CAM}} = \left(\hat{\mathbf{O}}_n^\top \hat{\mathbf{O}}_n\right)^{-1} \hat{\mathbf{O}}_n^\top \mathbf{c}_k^{\text{CAM}} \tag{5.14}$$

This is a *patient-specific* approximation. Subsequently, the estimates for optimal weights are normalized across the temporal dimension via a softmax on absolute values, yielding cluster-specific temporal weights:

$$e_{n,t}^k := \text{Softmax}(|\boldsymbol{\gamma}_{n,k}^{\text{CAM}}|) = \frac{\exp|\hat{\gamma}_{n,t,k}|}{\sum_{t'=1}^{T} \exp|\hat{\gamma}_{n,t',k}|} \tag{5.15}$$

so that $\sum_{t=1}^{T} e_{n,t}^k = 1$ for each n and k.

Step 3: Feature-Time Relevance Map The patient feature-time relevance map for the cluster k is encoded as a matrix $\mathbf{M}_n^k \in \mathbb{R}^{T \times D_f}$, whose (t, f)-entry is defined as the product of the cluster-specific temporal weight and the normalized feature weight:

$$(M_k)_{t,f} \coloneqq e_{n,t}^k \cdot s_{n,t,f}^{\text{CAM}} \qquad \forall\, t \in [T],\ f \in [D_f] \tag{5.16}$$

Each matrix $\mathbf{M}_n^k$ is L_1-normalized and can therefore be directly interpreted as a probability distribution over feature-time pairs, without further transformation. To verify this, note that both $e_{n,t}^k \geq 0$ and $(s_{n,t,f}^{\text{CAM}}) \geq 0$ are softmax outputs, so absolute values can be dropped:

$$\left\|\mathbf{M}_n^k\right\|_1 = \sum_{t=1}^{T} \sum_{f=1}^{D_f} e_{n,t}^k \cdot s_{n,t,f}^{\text{CAM}} = \sum_{t=1}^{T} e_{n,t}^k \sum_{f=1}^{D_f} s_{n,t,f}^{\text{CAM}} = \sum_{t=1}^{T} e_{n,t}^k = 1 \tag{5.17}$$

where $\sum_{f=1}^{D_f} s_{n,t,f}^{\text{CAM}} = 1$ follows from the softmax normalization in Equation (5.12), and $\sum_{t=1}^{T} e_{n,t}^k = 1$ from that in Equation (5.15).

Cluster-Level Averaging The construction above yields a patient-specific relevance map $\mathbf{M}_n^k$ for each patient n and cluster k. Since patient-level maps may be noisy and sensitive to individual outliers, the population-level effect is of greater relevance to characterize a cluster phenotype. Therefore, a cluster-level relevance map is obtained by averaging all patients assigned to cluster k:

$$\mathbf{M}^k \coloneqq \frac{1}{|C_k|} \sum_{\phi(\mathbf{X}_n)=k} \mathbf{M}_n^k \tag{5.18}$$

where $C_k \coloneqq \{n \mid \phi(\mathbf{X}_n) = k\}$ denotes the set of patients assigned to the cluster k under the inference rule of Equation (5.2).

The procedure described in this section represents one of several strategies for combining the learned weights $\hat{\boldsymbol{\alpha}}_{n,t}^{\text{CAM}}$, β_t, and $\mathbf{c}_k^{\text{CAM}}$ into an importance map. The specific choices made — about the application of softmax to absolute values, the definition of $\hat{\gamma}_{n,t,k}$ via Equation (5.14), and the average of only those patients assigned to the cluster — were made based on qualitative analysis of empirical results and clinical expert validation. These choices can be modified without affecting the quantitative evaluation of CAMELOT.

5.4 Implementation

This section describes the details of implementation specific to the experiments completed in the development of CAMELOT. The general evaluation framework is similar to that of Section 4.4; only the differences relevant to this chapter are described in full. Section 5.4.1 covers the experimental setup, including dataset scope, data split, initialization, and selection of hyperparameters. To validate the design choices in the model, Section 5.4.2 describes the design of the ablation studies conducted to assess the individual contribution of each proposed component.

5.4.1 Experimental Setup

Dataset and Input Features In contrast to Chapter 4, which employed the full HAVEN cohort for controlled benchmarking, experiments in this chapter are conducted on HAVEN-T2RF (see Section 3.2.1). This subcohort was selected due to its known clinical properties; furthermore, the application of CAMELOT to this subpopulation may allow the modeling of patient subgroups that differ significantly in their baseline SpO_2 values (Section 3.2.1) — these can then be analyzed via feature-time relevance maps. In order to evaluate the generalizability of CAMELOT, experiments are also conducted on the MIMIC-IV-ED dataset (see Section 3.2.2). In both settings, $D_o = 4$ outcome classes are considered. All available temporal features are used as input, in contrast to Chapter 4, which restricted input to vital signs only for controlled comparison purposes.

Data Splits and Initialization Data splitting and initialization strategies followed a similar approach to the one in the previous chapter Section 4.4.1. For each experiment, data were split into optimization (60%), and hold-out test (40%) sets by random selection of patients. Consequently, optimization data was split patient-wise into a training set $(80\%$ of the optimization data$)$ and a validation set (the remaining 20% of the optimization data). The distributional parameters

for the preprocessing steps, such as normalization and imputation, were estimated from the training data only and applied to all other splits, according to the guidelines for preventing data leakage discussed in Section 3.3. All neural network weights were initialized using Glorot initialization [332], except bias weights, which were initialized from a standard normal distribution. In practice, the same hyperparameters were used for the Predictor and Encoder subcomponents of CAMELOT, as a simplifying design choice. The Adam optimizer [306] was used for all DL model training, as described in Section 5.3.3.

Hyperparameter Selection For each model and hyperparameter configuration, 10 experiments were conducted using a fixed set of 10 distinct seeds, each seed determining a different random data split and parameter initialization. Performance metrics were averaged across seeds, and the standard deviation was reported. This extends the 5-seed protocol of Chapter 4 to reduce the variance in evaluation, given the larger hyperparameter search space. The selection of hyperparameters largely followed the same procedure described in Section 4.4. Hyperparameters were explored by grid search in the ranges reported in Table A.2 — the best-performing configuration was defined as the one with the highest mean validation set AUROC over the 10 seeds. For the number of clusters K, the adapted Elbow method of Section 4.4.3 was followed instead, using a Friedman test with Bonferroni correction with a 0.05 level to assess statistical significance. Lastly, as previously mentioned, any configuration that did not produce predictions for one or more outcome classes was disqualified. After hyperparameter selection, all models were retrained on the combined training and validation data, and the final performance was evaluated on the hold-out test set.

The same clustering benchmarks as Chapter 4 were considered: TSKM, SOM-VAE, and AC-TPC. For the outcome prediction proxy task, SVM and XGBoost are retrained; NEWS2 and ESI are also evaluated in HAVEN-T2RF and MIMIC-IV-ED training and held-out sets, respectively (see also Section 3.2.2). The hyperparameters for the DL benchmarks were kept consistent with those of CAMELOT

wherever applicable. All experiments and statistical tests were implemented and analyzed in Python using Tensorflow2, Scikit-Learn, Matplotlib, and Numpy. A single Tesla V100 GPU and eight Intel Xeon Gold 6246 CPUs at 3.30 GHz were used.

5.4.2 Ablation Studies

Ablation studies were conducted to assess the individual contribution of each novel component of CAMELOT to overall performance. The following variants were considered:

- CAMELOT: the full proposed model.
- ABL1: CAMELOT with $\mathcal{L}_{\text{dist}}$ replaced by the sample-wise entropy loss of AC-TPC and AC-TPC$_{\text{w}}$ [13], to evaluate the effect of the proposed formulation for the distribution loss.
- ABL2: the clustering component is removed, retaining only the Encoder and Predictor in a direct classification pipeline to evaluate the contribution of the clustering bottleneck. This mimics the architecture of the first stage of pre-training initialization, except that it is evaluated in the context of TPC.
- ABL3: CAMELOT with $\lambda_1 = 0$ (no $\mathcal{L}_{\text{dist}}$).
- ABL4: CAMELOT with $\lambda_2 = 0$ (no $\mathcal{L}_{\text{clus}}$).
- ABL5: CAMELOT with $\lambda_1 = \lambda_2 = 0$ (neither auxiliary loss).
- ABL6: CAMELOT without the AttentionRNNEncoder Encoder, replaced by a standard LSTM encoder, to evaluate the contribution of the proposed attention mechanism.

ABL1 and ABL2 represent structural alternatives to CAMELOT and ABL3–ABL5 isolate the effect of each auxiliary loss and their combination. ABL6 isolates the contribution of the AttentionRNNEncoder architecture. Ablation results are presented alongside the main results in Section 5.5.4.

5.5 Results

The same structure for reporting results as in Chapter 4 is followed, with separate sections for the temporal aspect (Section 5.5.1), the outcome aspect (Section 5.5.2). The final two sections investigate the learned attention maps (Section 5.5.3) and the ablation studies (Section 5.5.4).

5.5.1 Temporal Aspect

Cluster performance with respect to the temporal aspect is evaluated using SIL, DBI, and VRC metrics, following the framework of Section 3.4. The results for HAVEN-T2RF and MIMIC-IV-ED are jointly reported in Table 5.1. Temporal aspect metrics are not applicable to purely predictive benchmarks (SVM, XGBoost, NEWS2, ESI) and are therefore not included in the table.

Dataset	Model	SIL ↑	VRC ↑	DBI ↓
HAVEN-T2RF	TSKM	**0.27 (.03)**	**372.4 (12.32)**	**1.91 (.04)**
	SOM-VAE	0.17 (.02)	52.8 (9.32)	2.89 (.23)
	AC-TPC	0.06 (.02)	66.5 (18.7)	2.75 (.14)
	CAMELOT	0.11 (.04)	116.7 (6.2)	2.68 (.38)
MIMIC-IV-ED	TSKM	**0.16 (.03)**	**471.9 (80.7)**	**1.59 (.10)**
	SOM-VAE	0.07 (.02)	342.1 (14.5)	2.58 (.12)
	AC-TPC	0.08 (.03)	245 (15.2)	3.36 (.17)
	CAMELOT	0.11 (.05)	302.5 (25.1)	2.77 (.20)

Table 5.1: Temporal aspect performance of clustering models in the HAVEN-T2RF and MIMIC-IV-ED datasets, evaluated using SIL, VRC, and DBI. Results are reported as the mean across 10 random seeds (standard deviations in parentheses). Predictive baselines (SVM, XGBoost, NEWS2, ESI) are excluded as temporal aspect metrics are not applicable. Bold values indicate statistically significant best-performing models for each metric ($p < 0.05$, Friedman test with Bonferroni correction). Arrows next to metric names indicate whether higher (↑) or lower (↓) values are preferred.

Similarly to the temporal results reported in Section 4.5.1, these findings are consistent with the convexity of the clusters learned by TSKM and the bias of the clustering metrics — in particular, the empirical visualization of the average trajectories for the learned clusters of TSKM shows that the mean trajectories are not constant over time and do not lead to properly separated clusters with respect to the outcome aspect. More notably, CAMELOT achieves competitive

performance in all metrics compared to deep learning benchmark models. In HAVEN-T2RF data, CAMELOT is the best DL approach according to VRC (116.7 vs 66.5 for AC-TPC and 52.8 for SOM-VAE), while in MIMIC-IV-ED it is the best DL approach according to SIL (0.11 vs 0.08 for AC-TPC and 0.07 for SOM-VAE). In other metrics, SOM-VAE exhibits slightly better performance (although not always statistically significant, as is the case of DBI in MIMIC-IV-ED), however, CAMELOT consistently outperforms AC-TPC.

Interestingly, comparing with the results in the entire HAVEN cohort in Section 4.5.1, it can be seen that (a) TSKM exhibits a worse performance in all metrics and (b) the performance of SOM-VAE and AC-TPC is improved, except for DBI for AC-TPC. Although a direct quantitative comparison is not possible due to the different data sizes and its impact on the corresponding metrics, it is qualitatively surprising given the expectation that the reduced cohort would be more suitable for the discovery of relevant clinical phenotypes. A possible explanation relies on the fact that the HAVEN-T2RF subcohort is more homogeneous, which may lead to higher levels of difficulty in identifying clusters with respect to the temporal aspect, while the larger cohort may facilitate the discovery of average trends which, while not necessarily representing clinically significant phenotypes, are more easily captured by convex clustering methods such as TSKM.

5.5.2 Outcome Aspect

Performance in the outcome aspect is reported in Table 5.2, according to AUROC, F1, REC, and NMI. As in Chapter 4, any experimental run in which the model failed to produce predictions for one or more outcome classes was excluded from the reported results, except for TSKM and ESI, which were incapable to produce predictions for all outcome classes. For purely unsupervised models (TSKM, SOM-VAE, NEWS2, and ESI), outcome predictions were obtained via the proxy pipeline of Section 3.4.2.

In the HAVEN-T2RF dataset, CAMELOT achieves the highest AUROC (0.73), REC (0.38), and NMI (0.20) among all models. CAMELOT outperforms AC-TPC

Dataset	Model	AUROC ↑	F1 ↑	REC ↑	NMI ↑
HAVEN-T2RF	TSKM	0.55 (.01)	0.24 (.03)	0.26 (.02)	0.01 (.03)
	SOM-VAE	0.61 (.09)	0.27 (.05)	0.27 (.03)	0.05 (.03)
	AC-TPC	0.68 (.01)	**0.38 (.01)**	0.36 (.01)	0.17 (.02)
	CAMELOT	**0.73 (.02)**	0.36 (.01)	**0.38 (.02)**	**0.20 (.03)**
	SVM	0.59 (.02)	0.30 (.00)	0.32 (.00)	–
	XGBoost	0.65 (.01)	0.34 (.00)	0.35 (.00)	–
	NEWS2	0.61	0.29	0.34	–
MIMIC-IV-ED	TSKM	0.64 (.03)	0.22 (.00)	0.20 (.00)	0.00 (.01)
	SOM-VAE	0.62 (.03)	0.21 (.01)	0.20 (.00)	0.02 (.00)
	AC-TPC	0.67 (.02)	0.31 (.01)	0.35 (.01)	0.09 (.00)
	CAMELOT	**0.72 (.02)**	**0.34 (.02)**	**0.36 (.02)**	**0.11 (.03)**
	SVM	0.68 (.00)	0.25 (.00)	0.27 (.00)	–
	XGBoost	**0.73 (.01)**	0.29 (.00)	0.28 (.00)	–
	ESI	0.53	0.20	0.20	–

Table 5.2: Outcome aspect performance of CAMELOT and benchmark models in the HAVEN-T2RF and MIMIC-IV-ED datasets, evaluated using AUROC, F1, REC, and NMI. Results are reported as the mean across 10 random seeds (standard deviations in parentheses). NMI is not applicable to predictive baselines (SVM, XGBoost, NEWS2, ESI) and is denoted by –. ESI and NEWS2 are clinical baselines for MIMIC-IV-ED and HAVEN-T2RF, respectively, and are deterministic. Bold values indicate statistically significant best-performing models for each metric ($p < 0.05$, Friedman test with Bonferroni correction). Arrows next to metric names indicate whether higher (↑) or lower (↓) values are preferred.

by at least 5% in AUROC and matches it in F1, and outperforms the strongest supervised classifiers (XGBoost, SVM) by at least 8% in AUROC. In MIMIC-IV-ED, CAMELOT achieves the highest F1 (0.34), REC (0.36), and NMI (0.11). In AUROC, CAMELOT (0.72) is generally comparable to XGBoost (0.73), with no statistical differences detected, although CAMELOT outperforms XGBoost by 5% in F1 and 8% in REC, indicating a superior ability to identify minority outcome classes — empirically confirmed by the observation that XGBoost and SVM failed to assign patients to the two smallest outcome classes in this dataset.

The improved performance on the outcome aspect obtained by CAMELOT in relation to the benchmarks is attributed to its improved capacity to model the complex and heterogeneous structure of temporal EHR data. It is particularly notable that strong predictive performance is achieved despite the clustering bottleneck: all outcome predictions for a given patient are mediated through a single cluster representation vector rather than tailored to the precise input

trajectory. This contrasts to purely predictive models, which may achieve higher performance on the direct prediction task but lack the ability to combine information between clinically similar patients, identify relevant temporal trends, or provide an interpretable characterization of how and why a particular outcome is expected to occur.

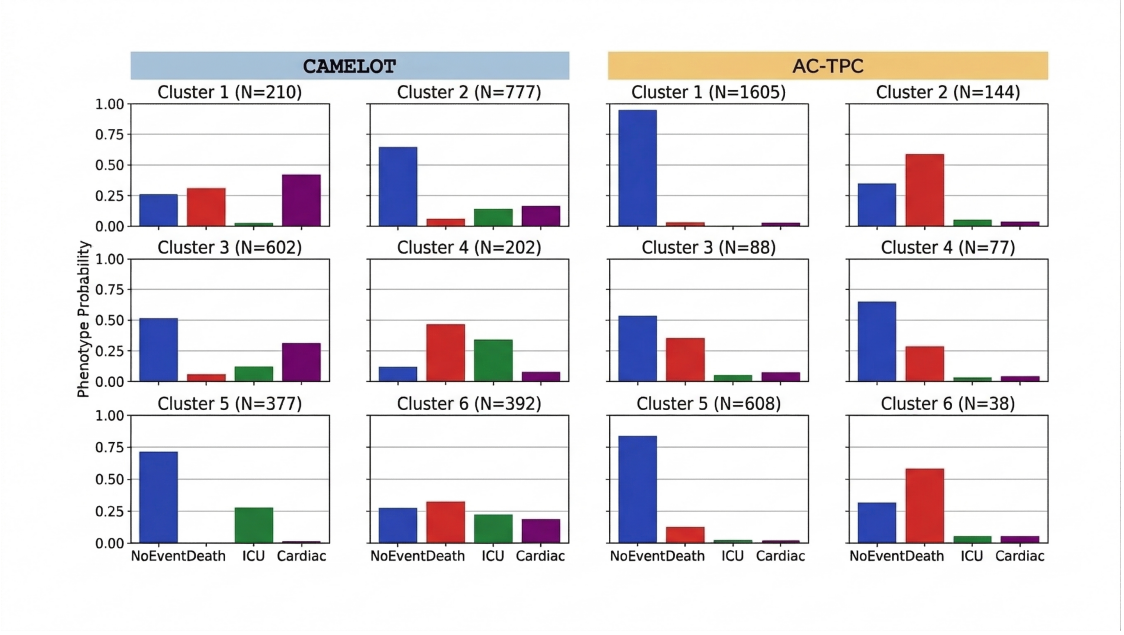


Figure 5.3: Cluster outcome distributions $\mathbf{o}_k^{\text{CAM,clus}}$ learned by CAMELOT (top) and AC-TPC (bottom) in HAVEN-T2RF. For each cluster k, bars show predicted class probabilities over $D_o = 4$ outcomes: No Event (blue), Death (red), ICU (green), and Cardiac (purple). Subplots indicate cluster index and size (in parentheses).

Figure 5.3 illustrates the learned cluster outcome phenotypes for CAMELOT and AC-TPC in the HAVEN-T2RF dataset. For each cluster k, the outcome characterization $\mathbf{o}_k^{\text{CAM,clus}}$ is shown as a bar plot on the $D_o = 4$ outcome classes. CAMELOT identifies six clusters with clearly separable outcome distributions: clusters 1 and 4 concentrate patients at high risk of death and cardiac arrest in the subsequent 24 hours; clusters 3 and 6 are associated with a lower probability of adverse events; and clusters 2 and 5 are predominantly healthy, with a markedly reduced risk of the most severe outcomes. By contrast, AC-TPC only identifies 3 distinct outcome distributions and fails to recover the ICU and Cardiac outcome

classes, reflecting its limitations under severe class imbalance. Furthermore, AC-TPC cluster sizes are much more extreme than those of CAMELOT, suggesting a more pronounced difficulty in handling the minority classes. The learned outcome distributions of CAMELOT also align closely with the empirical outcome frequencies within each cluster (see Table B.1 in the Appendix), although they slightly exaggerate the true frequencies of the smaller classes, which provides additional validation of the model's cluster assignments.

5.5.3 Attention Maps

The cluster-level feature-time relevance maps $\mathbf{M}^k$, as in Equation (5.18) and displayed in Figure 5.4, are examined in this section. As in Section 5.3.4, each map $\mathbf{M}^k \in \mathbb{R}^{T \times D_f}$ is obtained by averaging patient-specific relevance matrices over all patients assigned to cluster k, and is L_1-normalized so that its entries can be interpreted directly as a probability distribution over feature-time pairs. The analysis presented here is restricted to vital-sign features, as these constitute the primary temporal input modality in HAVEN-T2RF and have established clinical relevance for patient deterioration [259, 260]. Laboratory test features, while included in model training, are not displayed; their lower recording frequency and sparser temporal coverage relative to vital signs reduce the interpretive utility of their corresponding attention weights. The results are shown in HAVEN-T2RF, as the primary dataset of clinical motivation.

Clusters 1, 4, and 6, identified in Section 5.5.2 as those with the highest chance of death and Cardiac Arrest, exhibit the most concentrated and clinically informative relevance patterns. The respiratory rate (RR), FiO_2 and SpO_2 consistently receive the highest attention weights in multiple time steps in these clusters. This is consistent with the established clinical literature, in which respiratory function deterioration is a well-recognized early warning signal for adverse outcomes in patients with T2RF [19]. The temporal structure of the maps further reveals that attention weights for these variables are not uniformly distributed across time: elevated weights are concentrated in the 24–72 hour window prior to the outcome

time, coinciding with the input window defined in Section 3.2.1, suggesting that the model has learned to attend to the period with the greatest physiological instability, although this is not a formal proof. This observation is corroborated by the mean trajectory visualizations in Figure B.1, which show an increase in RR and a decrease in SpO_2 in the hours preceding the outcome event for these clusters, and by the descriptive statistics in Table B.2, in which RR, FiO_2, and SpO_2 exhibit the most pronounced inter-cluster separation.

The relevance maps for clusters 2 and 5, which are predominantly composed of patients without events (see Table B.1), show a qualitatively distinct pattern. SpO_2 receives consistently high attention throughout the admission window, and RR and FiO_2 receive lower weights relative to the adverse outcome clusters. This is clinically interpretable: in patients who do not deteriorate, stable blood oxygenation is a primary indicator of physiological health, and its sustained relevance throughout the observation window reflects the model's capacity to identify the absence of deterioration as a phenotypically meaningful signal. The pattern for the clusters 3 and 5 (Figure B.1) further suggests that the model assigns clinical significance to the maintenance of normal SpO_2 values, rather than exclusively to deviations from normality.

The feature-time relevance maps can capture clinically meaningful patterns across the full range of phenotypic profiles, extending the model interpretability beyond the identification of high-risk patients to include the characterization of stable phenotypes. However, it should be noted that the relevance maps are derived post hoc from the learned encoder weights and represent a near-linear approximation to the full model computation. As such, they should be interpreted as indicative relevance scores rather than formal measures of feature importance with associated uncertainty quantification. The clinical validity of the identified patterns was assessed through a qualitative comparison with the knowledge of experts in the domain, as described in Section 5.3.4; formal clinical

validation in a prospective setting remains a direction for future work. For completion, cluster characterizations in the MIMIC-IV-ED dataset (empirical counts, attention maps) are shown in the Appendix in Section B.2.

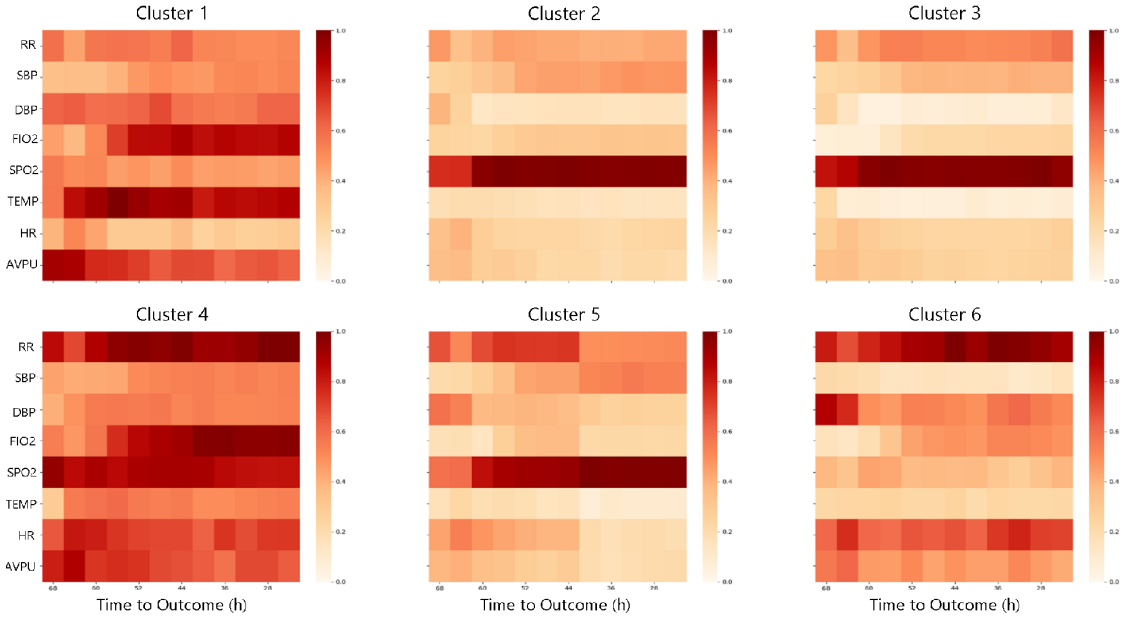


Figure 5.4: Cluster-level feature–time relevance maps $\mathbf{M}^k$ learned by CAMELOT in HAVEN-T2RF, shown as normalized heatmaps. Columns correspond to time steps and rows to vital-sign features. Warmer colors indicate higher relevance for cluster assignment. Laboratory features are omitted (see text). Maps are L_1-normalized to sum to one across all feature–time entries.

5.5.4 Ablation Studies

The ablation results reported in Table 5.3 are discussed in this section. Each ablation variant isolates the contribution of a distinct component of CAMELOT, and the results are interpreted jointly between both datasets to distinguish robust effects from dataset-specific ones.

The removal of the entire clustering component (ABL2), retaining only the Encoder and Predictor in a direct classification pipeline, produces the largest single performance drop across all metrics and both datasets. In HAVEN-T2RF, the AUROC decreases by 0.16 relative to CAMELOT (0.57 vs 0.73), and NMI by 0.14 (0.06 vs 0.20). In MIMIC-IV-ED, the AUROC decreases by 0.07 (0.65 vs

Dataset	Model	AUROC ↑	F1 ↑	REC ↑	NMI ↑
HAVEN-T2RF	ABL1 ($\mathcal{L}_{\text{dist}}$ replaced)	0.67 (.02)	0.36 (.02)	0.36 (.02)	0.16 (.03)
	ABL2 (clustering removed)	0.57 (.02)	0.25 (.02)	0.26 (.02)	0.06 (.03)
	ABL3 ($\lambda_1 = 0$, no $\mathcal{L}_{\text{dist}}$)	0.65 (.01)	0.30 (.02)	0.32 (.02)	0.15 (.01)
	ABL4 ($\lambda_2 = 0$, no $\mathcal{L}_{\text{clus}}$)	0.65 (.02)	0.28 (.01)	0.30 (.00)	0.15 (.01)
	ABL5 ($\lambda_1 = \lambda_2 = 0$)	0.61 (.04)	0.25 (.02)	0.27 (.01)	0.10 (.03)
	ABL6 (Encoder replaced)	0.69 (.01)	0.36 (.01)	0.36 (.01)	0.18 (.01)
	CAMELOT	**0.73 (.02)**	**0.36 (.01)**	**0.38 (.02)**	**0.20 (.03)**
MIMIC-IV-ED	ABL1 ($\mathcal{L}_{\text{dist}}$ replaced)	0.69 (.01)	0.25 (.01)	0.33 (.02)	0.07 (.01)
	ABL2 (clustering removed)	0.65 (.03)	0.23 (.01)	0.26 (.01)	0.04 (.00)
	ABL3 ($\lambda_1 = 0$, no $\mathcal{L}_{\text{dist}}$)	0.70 (.01)	0.30 (.01)	0.34 (.00)	0.07 (.00)
	ABL4 ($\lambda_2 = 0$, no $\mathcal{L}_{\text{clus}}$)	0.70 (.02)	0.24 (.03)	0.31 (.04)	0.05 (.02)
	ABL5 ($\lambda_1 = \lambda_2 = 0$)	0.67 (.03)	0.30 (.01)	0.33 (.01)	0.06 (.00)
	ABL6 (Encoder replaced)	0.65 (.05)	0.24 (.01)	0.30 (.02)	0.04 (.01)
	CAMELOT	**0.72 (.02)**	**0.34 (.02)**	**0.36 (.02)**	**0.11 (.03)**

Table 5.3: Outcome aspect performance of CAMELOT ablation variants (ABL1–ABL6) and the full model in the HAVEN-T2RF and MIMIC-IV-ED datasets, evaluated using AUROC, F1, REC, and NMI. Results are reported as the mean across 10 random seeds (standard deviations in parentheses). Definitions are provided in Section 5.4.2. Bold values indicate the best-performing variant per metric within each dataset block. Arrows next to metric names indicate whether higher (↑) or lower (↓) values are preferred.

0.72). These results confirm that the clustering bottleneck, far from being a constraint on predictive performance, provides a meaningful inductive bias: by restricting outcome predictions to be mediated through a finite set of cluster representation vectors, the model is encouraged to learn phenotypically separable latent representations, which in turn improves both clustering quality and outcome prediction.

The individual and combined effects of the two auxiliary losses are assessed by comparing ABL3 ($\lambda_1 = 0$, no $\mathcal{L}_{\text{dist}}$), ABL4 ($\lambda_2 = 0$, no $\mathcal{L}_{\text{clus}}$), and ABL5 ($\lambda_1 = \lambda_2 = 0$, neither loss) against the full CAMELOT model. On HAVEN-T2RF, removal of $\mathcal{L}_{\text{dist}}$ alone (ABL3) reduces AUROC by 0.08 and NMI by 0.05 relative to CAMELOT, while removal of $\mathcal{L}_{\text{clus}}$ alone (ABL4) produces a comparable reduction in AUROC (0.08) but a smaller reduction in NMI (0.05). The removal of both losses (ABL5) yields a further decrease in all metrics, with AUROC at 0.61 and NMI at 0.10 in HAVEN-T2RF, consistent with the two losses addressing complementary failure modes of cluster collapse described in Section 5.3.3.3. In MIMIC-IV-ED, a similar but less pronounced pattern is observed, with ABL5 pro-

ducing an AUROC of 0.67 and an NMI of 0.06. The consistent ordering ABL5 $<$ ABL3 $\approx$ ABL4 $<$ CAMELOT between both datasets suggests that both losses contribute independently and approximately equally to overall performance.

Replacing $\mathcal{L}_{\text{dist}}$ with the sample-wise entropy loss of AC-TPC [13] (ABL1) produces a performance below CAMELOT in both datasets: AUROC of 0.67 vs 0.73 in HAVEN-T2RF and 0.69 vs 0.72 in MIMIC-IV-ED. In particular, ABL1 consistently outperforms ABL3 (no $\mathcal{L}_{\text{dist}}$ at all) on HAVEN-T2RF (0.67 vs 0.65), confirming that some form of entropy-based cluster exploration loss is beneficial. The margin between ABL1 and CAMELOT validates the proposed population-level formulation of $\mathcal{L}_{\text{dist}}$ (Equation (5.10)) as superior to the sample-wise alternative: by encouraging a uniform population-level cluster assignment distribution rather than confident per-sample assignments, $\mathcal{L}_{\text{dist}}$ more directly addresses the collapse failure mode described in Section 5.3.3.3.

The removal of the AttentionRNNEncoder Encoder in favor of a standard LSTM encoder (ABL6) has a dataset-dependent effect. In MIMIC-IV-ED, the performance drop is substantial: AUROC decreases by 0.07 (0.65 vs 0.72), F1 by 0.10 (0.24 vs 0.34), and NMI by 0.07 (0.04 vs 0.11). In HAVEN-T2RF, the impact is more modest: AUROC decreases by 0.04 (0.69 vs 0.73), with F1 and REC unchanged and NMI decreasing by 0.02. The stronger effect on MIMIC-IV-ED suggests that the regularization properties of the near-linear approximation in AttentionRNNEncoder (Section 5.3.2.3) are particularly beneficial in the more heterogeneous ED setting, where the LSTM may otherwise overfit to spurious temporal patterns. The smaller effect on HAVEN-T2RF may reflect the greater homogeneity of the T2RF subcohort, which reduces the benefit of additional regularization. Together, these results confirm the contribution of the AttentionRNNEncoder architecture, with the caveat that its benefit is most pronounced in settings with greater temporal heterogeneity.

In general, the ablation results confirm that all four proposed components of CAMELOT — the clustering bottleneck, the distribution loss, the cluster separation loss, and the AttentionRNNEncoder architecture — contribute positively to

overall performance. The clustering bottleneck provides the largest individual contribution, followed by the combined effect of the auxiliary losses. The AttentionRNNEncoder architecture provides a consistent but dataset-dependent benefit, most pronounced in settings with high temporal heterogeneity. These findings validate the design choices of CAMELOT and motivate the retention of all components in the final model.

5.6 Conclusion

This chapter introduced CAMELOT, a novel end-to-end DL model for TPC of EHR data, proposing three contributions over prior SOTA: a novel AttentionRNNEncoder encoder architecture providing joint feature-time interpretability, two auxiliary loss functions to address cluster collapse under severe class imbalance, and a procedure for deriving cluster-level feature-time relevance maps to characterize learned phenotypes. Experiments on HAVEN-T2RF and MIMIC-IV-ED confirmed improved performance with respect to the outcome aspect and relative to benchmarks. In particular, the model showed improved AUROC performance over clinical approaches such as NEWS2 and ESI. Ablation studies validated the contribution of each component. Qualitative analysis suggested translating the learned attention maps and phenotype characterizations into clinically meaningful insights.

Two limitations of the current framework motivate the subsequent chapter. First, the cluster representation vectors in CAMELOT, $\mathbf{c}^{\text{CAM}}$, are static and do not encode an explicit temporal structure; as a result, the model characterizes the phenotypes by their average trajectory rather than by the dynamics of how patients change within a cluster over time. Secondly, CAMELOT cannot inform how patients with adverse outcomes deteriorate over time, which can be crucial to understand how and when to intervene. The next chapter addresses these limitations by proposing a novel model architecture that learns dynamic cluster representations and provides patient-level trajectory forecasting.

There are two kinds of forecasters:
those who don't know and
those who don't know they don't know.
— John Kenneth Galbraith

6 Dynamic Phenotyping

6.1 Introduction

The models discussed in Chapters 4 and 5 share a common structural assumption: cluster assignments and outcome predictions are defined for a single time point, resulting in *static* phenotypes. This formulation is practically convenient, as it is well-suited to retrospective cohort analysis, as well as to the clinical characterization of learned phenotypes. However, it does not reflect the fundamentally dynamic nature of patient trajectories. In particular, clinical decision-making requires continuous reasoning about the patient's state throughout their hospital admission; furthermore, it is important to anticipate how a patient's physiological condition is likely to change over time, e.g., whether they are at risk of deterioration in key physiological variables. Intervention planning, which additionally requires reasoning about causal effects and counterfactual scenarios, is beyond the scope of this thesis; nevertheless, a necessary precondition is the ability to model the temporal evolution of patient trajectories.

This chapter addresses the above limitation by extending the TPC framework in the context of *dynamic phenotyping*. Here, cluster assignment is defined at each time step of admission rather than at a single point; furthermore, learned phenotypes and per-step assignments, through the methodology introduced, can be

used to generate future patient trajectories that are plausible. To this end, the DeepGC model is introduced: a deep generative model that is capable of producing time-varying cluster assignments within the TPC framework described in Section 2.4.2, and also incorporates a generative component capable of synthesizing sequences of patient observations for future time steps. By generating observations beyond the last recorded time step, DeepGC enables reasoning about the future evolution of a patient's phenotypic state without requiring additional clinical data, directly supporting the relevance of the dynamic phenotyping objective. The model is evaluated in the HAVEN-T2RF and MIMIC-IV-ED datasets, consistent with the experimental setting of Chapter 5, and the notation of the preceding chapters is carried forward throughout.

6.2 Background

Identifying dynamic patient phenotypes from temporal EHR data requires models capable of simultaneously representing the sequential structure of physiological trajectories and recovering time-varying cluster assignments. A central difficulty is that cluster membership is inherently discrete: a patient belongs to one phenotype or another at any given time, yet this discreteness obstructs gradient-based optimization and complicates the modeling of smooth transitions between phenotypic states. A further challenge is the disentanglement of whether an observed change in patient data reflects a genuine shift in the underlying phenotype, the natural temporal evolution within the phenotype, or measurement noise. These considerations collectively motivate the probabilistic approaches reviewed in the following.

6.2.1 Dynamic Clustering

Dynamic Phenotyping is naturally associated with temporally-aware clustering methods designed for sequential data. Classical models such as K-$means$ and GMM operate on fixed-dimensional feature vectors and are not suitable for sequential data, neither capturing the temporal dependencies in the input data

nor accounting for the varying lengths of patient admissions. Extensions to time-series data, such as TSKM (Section 4.2.1.1), typically address temporality via a different distance metric, such as DTW, or via the imposition of a regular temporal grid on the input observations via the standardization of time steps and missing value imputation. It has been argued in Chapter 4 that these methods scale poorly to high-dimensional, heterogeneous EHR data. More importantly, these solutions do not easily allow for dynamic updates to cluster assignment as new data are observed, therefore making them unsuitable for dynamic phenotyping. However, DL methods and LSTM-based architectures, in particular, are more naturally suited to address these challenges due to the incorporation of gated mechanisms that selectively retain relevant historical information and update the patient state representation at each time step. By maintaining a hidden state that evolves over time, long-term dependencies in the input data can, in principle, be leveraged to dynamically assign patients to clusters via the hidden state representation.

Unfortunately, the non-continuous nature of cluster assignment poses difficulties for gradient-based model optimization, as gradients cannot be propagated through discrete assignment variables directly [333, 334], so that the literature approaches this problem via the use of probabilistic models [335]. In the context of DL methods, VAEs are one of the most common successful DL architectures that handle probabilistic modeling [101, 336]. The VAE framework uses a standard encoder-decoder architecture, where the encoder maps the input data to a latent space and the decoder converts the latent representations back to the input space. Critically, VAEs model the latent space as a probability distribution (a Gaussian, $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, in the original formulation), with parameters derived from the encoder output. The latent representations are then sampled from this distribution and passed to the decoder. Training of a VAE is done via variational inference [101, 337], which parameterizes the intractable posterior distribution of the latent variables with a tractable distribution derived from the encoder network. Further details of variational inference in the context of

DeepGC are provided in Section 6.3.3. Importantly, the probabilistic treatment of the latent space allows uncertainty in the representation to be explicitly modeled, which is a desirable property in the context of soft cluster assignment, avoiding the discrete requirements of a hard assignment to a cluster.

As VAEs are not designed to handle sequential data, several extensions have been proposed to incorporate temporal modeling capabilities. The Variational Recurrent Neural Network (VRNN [338]) uses a recurrent architecture combined with a VAE framework at each time step, modeling the temporal evolution of the latent distribution by conditioning both the prior and posterior parameters on the hidden state of the RNN. Other variants include the Stochastic Recurrent Neural Network [339] and the Recurrent Latent Variable model, which differ in the conditional equations that model dependencies between the posterior and prior distributions. The above models have been successfully applied to various tasks, including speech and handwritten recognition, however, their application to clustering and phenotyping remains limited. Of particular relevance is the Gaussian Mixture Model-Variational Recurrent Neural Network (GMM-VRNN [338]), which extends the VRNN model by placing a Gaussian mixture prior over the latent space at each time step, allowing time-varying modeling of mixture probabilities. Nevertheless, GMM-VRNN does not incorporate supervised outcome information into the clustering objective; furthermore, it does not model the temporal evolution of mixture coefficients through a learned prior, so that the connection between consecutive mixture coefficients is minimal. These limitations motivate the design of DeepGC, which addresses the above limitations via a Dirichlet prior and a joint training objective over the observation sets, probability vectors of cluster assignments and outcomes.

Existing deep clustering methods such as SOM-VAE and AC-TPC incorporate soft cluster assignment mechanisms and serve as comparative benchmarks in this chapter, consistent with the approach of Chapters 4 and 5. Probabilistic state-space models [340, 341] offer a counterpoint to the above DL methods, by modeling the state of the patient as a latent variable that evolves over time according to

a Markov process. Approaches include the Kalman filter [342] and HMMs [184]. Deep variants of these models have also been proposed, such as the Deep Kalman filter [343] and deep latent factor models [344, 345], which replace the parametric assumptions of the original models with neural networks. However, despite the demonstrated empirical utility in certain settings, these approaches suffer from restrictive distributional assumptions, scalability and inference complexity, and lack of explicit phenotypic semantics in the latent space [343, 344], so they have shown limited relevance in temporal healthcare settings and are not considered further as benchmarks.

Other solutions use Bayesian NonParametric (BNP [346, 347]) methods. BNP allows for a flexible number of clusters, allowing the number of mixtures (i.e., clusters) to grow or reduce as the data require. An example includes the Dirichlet Process Mixture Model (DPMM [348–350]), which places a Dirichlet process prior over the mixture components. The Hierarchical Dirichlet Process Hidden Markov Model (HDP-HMM [351, 352]) extends the DPMM approach to sequential latent states. Although theoretically appealing, BNP approaches are computationally demanding, require a great number of simulations to approximate the model objective, and pose significant optimization challenges on the scale of EHR datasets considered in this thesis [349, 353]; furthermore, they do not easily allow the incorporation of other aspects of the TPC task, such as the outcome aspect, and therefore are not suitable for the task at hand. The work in this chapter builds on the above literature, particularly that of the VRNN architecture, by introducing the DeepGC model. The method builds a probabilistic encoder-decoder model that directly incorporates outcomes in its internal architecture. See Section 6.3 for further details on the model architecture and the training procedure.

6.2.2 Dirichlet Distribution

The Dirichlet distribution is the canonical probability distribution defined on the set of probability vectors, Δ^{K-1}, and plays a central role in the probabilistic

formulation of DeepGC as it is the core component that allows the temporal modeling of cluster assignment vectors. Its key properties are reviewed here, and the results that are necessary to derive the training objective of DeepGC are also shown. To facilitate discussion and analysis, the gamma function and distributions are defined as follows.

Definition 13 (Definition of the Gamma Function and Distribution) *For a positive real number x, the gamma function is defined as per the integral:*

$$\Gamma(x) = \int_0^\infty t^{x-1} e^{-t} dt \tag{6.1}$$

Consequently, the gamma distribution with shape parameter $a > 0$ and rate parameter $b > 0$ is defined as the probability distribution over $\mathbb{R}_{>0}$ with Probability Density Function (PDF) given by:

$$p(x \mid a, b) = \frac{b^a}{\Gamma(a)} x^{a-1} e^{-bx} \tag{6.2}$$

and one writes $X \sim \Gamma_d(a, b)$.

The key properties of the distribution include its expectation, $\mathbb{E}[X] = \frac{a}{b}$, and its variance, $\mathbb{V}\text{ar}[X] = \frac{a}{b^2}$.

Definition 14 (Definition of the Dirichlet Distribution) *Let $\boldsymbol{\alpha}^{DIR} \in \mathbb{R}^K_{>0}$. A random variable is Dirichlet distributed with concentration parameter $\boldsymbol{\alpha}^{DIR}$ if it is a probability distribution over the set of probability vectors of dimension K, Δ^{K-1} and its PDF satisfies:*

$$p(\mathbf{x}^{DIR} \mid \boldsymbol{\alpha}^{DIR}) = \frac{1}{B(\boldsymbol{\alpha}^{DIR})} \prod_{k=1}^{K} (x_k^{DIR})^{\alpha_k^{DIR}-1} = \frac{\Gamma(\alpha_s^{DIR}) \prod_{k=1}^{K} (x_k^{DIR})^{\alpha_k^{DIR}-1}}{\prod_{k=1}^{K} \Gamma(\alpha_k^{DIR})} \tag{6.3}$$

where $B(\boldsymbol{\alpha}^{DIR})$ is the normalization constant and $\alpha_s^{DIR} := \sum_{k=1}^{K} \alpha_k^{DIR}$ is the sum of elements of the concentration parameter.

The Dirichlet distribution is a multivariate generalization of the beta distribution and is commonly used as a prior distribution in Bayesian statistics, particularly in the context of mixture models whose support is the simplex set. The distribution is parameterized by a concentration parameter, $\boldsymbol{\alpha}^{\text{DIR}}$, which controls the shape

of the distribution: the relative weight of each dimension in $\boldsymbol{\alpha}^{\text{DIR}}$, $\alpha_k^{\text{DIR}}/\alpha_s^{\text{DIR}}$, determines the expected value of the corresponding dimension in $\mathbf{x}^{\text{DIR}}$, while the sum of the concentration controls the variance of the distribution. In particular, when $\alpha_k^{\text{DIR}} < 1$ for all $k \in [K]$, the distribution is sparse, with most of the probability mass concentrated on vectors with few nonzero dimensions; alternatively, when $\alpha_k^{\text{DIR}} \gg 1$ for all k, the distribution is dense. The expected value of an $\boldsymbol{\alpha}^{\text{DIR}}$-parameterized Dirichlet distribution can be readily computed so that its moments are given by:

$$\mathbb{E}[x_k^{\text{DIR}}] = \frac{\alpha_k^{\text{DIR}}}{\alpha_s^{\text{DIR}}}, \quad \mathbb{V}\text{ar}[x_k^{\text{DIR}}] = \frac{\alpha_k^{\text{DIR}}(\alpha_s^{\text{DIR}} - \alpha_k^{\text{DIR}})}{(\alpha_s^{\text{DIR}})^2(\alpha_s^{\text{DIR}} + 1)}, \quad k \in [K] \tag{6.4}$$

Logarithms and Marginal Distributions The Dirichlet distribution is defined over the simplex set; thus, it is important to understand how the distributions over each dimensional element, i.e., the marginal distributions, behave. As each individual dimension has support over the unit interval, it is generally useful to apply a logarithmic transformation, which maps the support to the negative or positive (if a negative sign is applied) real numbers. The following results are useful for understanding DeepGC:

Lemma 1 (Marginal Dirichlet Distributions) *Given a Dirichlet distribution with parameter $\boldsymbol{\alpha}^{DIR}$, the marginal distribution of each component of the random vector $\mathbf{x}^{DIR}$ is a beta distribution such that:*

$$p(x_k^{DIR} \mid \boldsymbol{\alpha}^{DIR}) = Beta(\alpha_k^{DIR}, \alpha_s^{DIR} - \alpha_k^{DIR}) \tag{6.5}$$

where $Beta$ represents a Beta distribution.

Proof 1 *The proof can be found in Gelman et al.* [354].

The previous result quantifies the above mentioned relevance of $\boldsymbol{\alpha}^{\text{DIR}}$ in the distribution of each dimension in the output vector $\mathbf{x}^{\text{DIR}}$. In particular, instead of using manual integral computation through the formal definition of expectation and variance properties, the values can instead be derived from analogous formulae

for the beta distribution, matching the values shown in Equation (6.4). On the other hand, the Dirichlet distribution moments also exhibit closed-form formulae when mapped via a logarithmic transformation:

Lemma 2 (Expectation of Dirichlet Under Logarithm) *Let* $\log \mathbf{x}^{DIR}$ *be the natural logarithm of* $\mathbf{x}^{DIR} \sim \text{Dir}(\boldsymbol{\alpha}^{DIR})$. *Then:*

$$\mathbb{E}[\log x_k^{DIR}] = \psi(\alpha_k^{DIR}) - \psi(\alpha_s^{DIR}), \quad k \in [K] \tag{6.6}$$

where ψ *denotes the digamma function, the logarithmic derivative of the gamma function.*

Proof 2 *The proof is straightforward from applications of the Dirichlet PDF and the definition of expectation. Full details can be found in Gelman et al.* [354].

KL Divergence between Dirichlet Distributions The usefulness of the above results is that they allow computation of the divergence between two Dirichlet distributions with different concentration parameters, a key component of the training objective of DeepGC, as it uses variational inference as part of its training approach. Let $D_{\text{KL}}(\cdot\|\cdot)$ denote the Kullback-Leibler (KL) divergence between two distributions. Suppose $\boldsymbol{\alpha}_1^{\text{DIR}}, \boldsymbol{\alpha}_2^{\text{DIR}} \in \mathbb{R}_{>0}^K$ are two concentration parameters such that $\mathbf{X}_1^{\text{DIR}} \sim \text{Dir}(\boldsymbol{\alpha}_1^{\text{DIR}})$ and $\mathbf{X}_2^{\text{DIR}} \sim \text{Dir}(\boldsymbol{\alpha}_2^{\text{DIR}})$.

Lemma 3 (KL Divergence between Dirichlet Random Variables) *The KL divergence between* $\mathbf{X}_1^{DIR}$ *and* $\mathbf{X}_2^{DIR}$ *can be computed as:*

$$D_{KL}(\mathbf{X}_1^{DIR} \| \mathbf{X}_2^{DIR}) = \log \frac{\Gamma(\alpha_{1,s}^{DIR})}{\Gamma(\alpha_{2,s}^{DIR})} + \sum_{k=1}^{K} \Bigg[- \log \frac{\Gamma(\alpha_{1,k}^{DIR})}{\Gamma(\alpha_{2,k}^{DIR})} + (\alpha_{1,k}^{DIR} - \alpha_{2,k}^{DIR}) \big[\psi(\alpha_{1,k}^{DIR}) - \psi(\alpha_{1,s}^{DIR}) \big] \Bigg] \tag{6.7}$$

where ψ *is the Digamma function and* $\alpha_{1,k}^{DIR}, \alpha_{2,k}^{DIR}$ *are, respectively, the* $k-$*entry of the vectors* $\boldsymbol{\alpha}_1^{DIR}, \boldsymbol{\alpha}_2^{DIR}$, *while* $\alpha_{1,s}^{DIR}, \alpha_{2,s}^{DIR}$ *denote the sums of the corresponding parameters.*

Proof 3 *The proof is straightforward, if laborious, from the definition of KL divergence and substituting for the P.D.F. of the Dirichlet distribution. See Gelman et al.* [354] *for complete details.*

Dirichlet Reparameterization via Gamma Distributions The final property of the Dirichlet distribution that is leveraged in DeepGC is related to reparameterization. The following result is noted [354]:

Lemma 4 *Suppose* $X_1^{GAM} \sim \Gamma_d(\alpha_1^{DIR}, 1), \cdots, X_K^{GAM} \sim \Gamma_d(\alpha_K^{DIR}, 1)$ *are gamma distributed random variables. Let* $X_s^{GAM} := \sum_{k=1}^{K} X_k^{GAM}$. *Then:*

$$\left(\frac{X_1^{GAM}}{X_s^{GAM}}, \cdots, \frac{X_K^{GAM}}{X_s^{GAM}}\right) \sim \text{Dir}(\boldsymbol{\alpha}^{DIR}) \tag{6.8}$$

6.3 DeepGC

This section introduces a novel model termed the *DEEP Generative outcome-based Clustering* model, henceforth denoted as DeepGC. DeepGC extends the TPC framework to the dynamic phenotyping setting discussed in the introduction in Section 6.1, adapting and building directly on the VRNN architecture. Three core contributions are made over prior work. Firstly, cluster assignment is probabilistically modeled as a time-varying Dirichlet random variable at each time step, unlike previous TPC models, which model it as a single static assignment per admission. Secondly, a generative component is incorporated that reconstructs observed patient trajectories during training and, critically, synthesizes future observations beyond the last recorded time step — this is done via a forward autoregressive procedure, with outcome prediction derived from the latent representation in the final generated step. Finally, supervised outcome information is incorporated into the training objective, ensuring that the data corresponding to the outcome aspect are relevant for modeling.

6.3.1 Model Architecture

A high-level sketch of the model architecture is shown in Figure 6.1. Globally, DeepGC is broadly similar to CAMELOT in that both models derive latent representation of patient trajectories and use these to predict the outcome aspect. However, DeepGC differs fundamentally in the way the representation vectors are computed. Rather than a single admission-level encoding, a time-varying

cluster assignment vector is estimated at each time step; consequently, the latent representation is derived as a weighted combination of the learnable cluster means. Another critical difference is that DeepGC extends the modeling capabilities beyond the input data time horizon, using these data to generate future observations, from which the outcome is predicted.

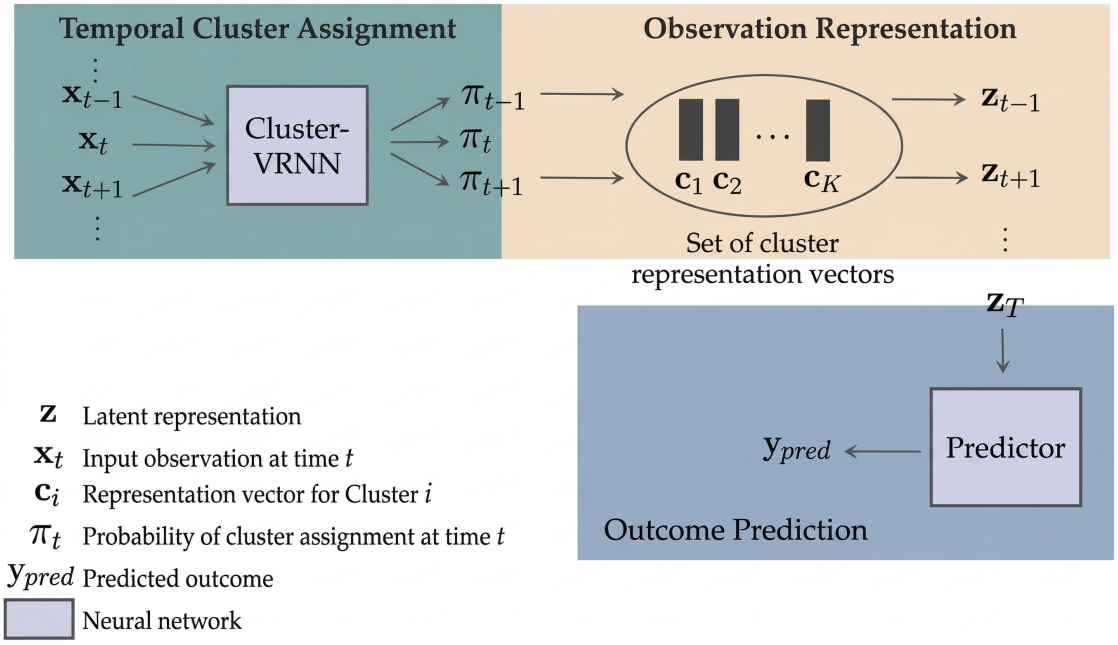


Figure 6.1: Schematic visualization of the DeepGC model architecture. Input trajectories are processed sequentially by the Cluster-VRNN network which, at each time step, estimates cluster assignment probabilities through the prior and inference components and uses them to compute latent patient representations. Beyond the last observed time step T, the model iteratively generates unobserved observations and latent representations. The final representation is passed to the Predictor to produce the outcome estimate for each patient.

The model is composed of three functional components: (a) a Cluster-VRNN network (described in further detail in Section 6.3.2), a Predictor network, and a set of learnable cluster mean and variance vectors, also called cluster representation vectors. Cluster-VRNN can be further decomposed into separate networks, the Prior, Inference, and Generation networks. Each is implemented as a feed-forward NN with independent parameters, the *gate functions*, (g_i^{DGC}), and interact jointly as part of a shared RNN backbone structure. The Predictor is also implemented as a feed-forward neural network, mapping latent representations

$\mathbf{z}_{n,t}^{\text{DGC}} \in \mathbb{R}^l$ to an outcome prediction of dimension D_o. Similarly to Chapter 5, l is used to represent the dimensionality of the latent space and is a hyper-parameter to the model. Finally, the cluster representations are modeled according to a set of Gaussian distributions:

$$\mathcal{E}^{\text{DGC}} := \left\{ (\boldsymbol{\mu}_k^{\text{DGC}}, (\sigma_k^{\text{DGC}})^2 \mathbf{I}_l) \right\}_{k=1}^{K} \tag{6.9}$$

where $\boldsymbol{\mu}_k^{\text{DGC}} \in \mathbb{R}^l$ and $(\sigma_k^{\text{DGC}})^2 > 0$ are, respectively, the mean and variance of the cluster k and a diagonal covariance structure is imposed on the cluster means, so that each dimension is modeled independently.

A forward pass of DeepGC proceeds in two sequential stages. Firstly, in *observed stage*, the observation sets, $\mathbf{x}_{n,t}$, are modeled within an RNN-like structure. For patient n, the RNN backbone maintains a hidden state, $\mathbf{h}_{n,t}^{\text{DGC}} \in \mathbb{R}^s$, initialized as $\mathbf{0}$ at time $t = 0$, which continuously encodes all prior observation and assignment data for each time step prior to time step t. For each t, the Inference network models the conditional probability of the cluster assignment vectors given the latest observation, $\mathbf{x}_{n,t}$, and the previous cell state, $\mathbf{h}_{n,t-1}^{\text{DGC}}$, resulting in a posterior distribution on $\boldsymbol{\pi}_{n,t}^{\text{DGC}} \in \Delta^{K-1}$. The posterior distribution is then sampled and the patient latent representation, $\mathbf{z}_{n,t}^{\text{DGC}}$ is computed as a weighted combination of the cluster means in $\mathcal{E}^{\text{DGC}}$ and the sampled probability of assignment. The Generative network models the reverse process: latent representations $\mathbf{z}_{n,t}^{\text{DGC}}$ and $\mathbf{h}_{n,t-1}^{\text{DGC}}$ are fed to a generative gate which returns a reconstructed observation, $\hat{\mathbf{x}}_{n,t}^{\text{DGC}}$. Finally, the hidden state of the RNN, $\mathbf{h}_{n,t}^{\text{DGC}}$, is updated as a function of the observed data, the latent representation, and the previous cell state.

As no observation is present in the input for $t > T$ and the outcome has a lag with the observation temporal indices, DeepGC includes a second phase, denoted as *generative phase*. Here, the model proceeds similarly as above with one caveat: the generated observations from the Generation network are synthesized (reconstructed), and these replace all dependencies in the observations $\mathbf{x}_{n,t}$ within the first stage (specifically, the Inference network and the hidden state update). Intuitively, the model iteratively generates a reconstructed observation set, then

determines the new latent representation, followed by an updated cell state. The hyperparameter T' governs the extent of forward generation and is discussed as part of the post hoc investigation studies of DeepGC. Once the generative step is completed, the Predictor receives the latent representation in the final generated step, $\mathbf{z}_{n,T'}^{\text{DGC}}$, and generates an outcome prediction, $\hat{\mathbf{y}}_n^{\text{DGC}}$.

DeepGC shares with CAMELOT the property that outcome predictions are modeled through a phenotypic bottleneck: the Predictor input is restricted to the convex hull of $\mathcal{E}^{\text{DGC}}$ (in DeepGC) or the set $\mathcal{E}^{\text{CAM}}$ (in CAMELOT). However, the bottleneck in CAMELOT is much more severe as the Predictor receives only one of K cluster vectors. On the other hand, the DeepGC bottleneck is soft — $\mathbf{z}_{n,T'}^{\text{DGC}}$ is a convex combination of all the cluster means, as weighted by the assignment probabilities in the final generated step. This facilitates the continuous flow of gradient information through the network, which is likely to lead to easier optimization and can help mitigate the cluster collapse issue discussed in Section 5.3.3.3.

6.3.2 Probabilistic Modeling

DeepGC defines a fully generative probabilistic model in the joint space consisting of (a) patient observation vectors and (b) outcomes, $p(\mathbf{X}_n, \mathbf{y}_n^{\text{mc}})$ through the introduction of a sequence of cluster assignment vectors, $\boldsymbol{\pi}_{n,\leq T'}^{\text{DGC}} := (\boldsymbol{\pi}_{n,1}^{\text{DGC}}, \ldots, \boldsymbol{\pi}_{n,T'}^{\text{DGC}})$, which are latent, i.e. "hidden" and unobservable. The indexing "$<, >, \leq, \geq$" will henceforth be used to describe the temporal set of objects that occur at time steps earlier, later, earlier or equal, or later or equal, respectively. The joint likelihood factorizes over time steps, and each factor is conditional on all preceding information (see also Equation (6.17) for an analytical description). A key modeling assumption in DeepGC is that the RNN hidden state $\mathbf{h}_{n,t-1}^{\text{DGC}}$ is a sufficiently relevant summary of all information prior to t, namely the observed trajectories $\mathbf{x}_{n,<t}$ and the latent assignments $\boldsymbol{\pi}_{n,<t}^{\text{DGC}}$. Under this Markovian assumption, each factor reduces to a conditional probability on $\mathbf{h}_{n,t-1}^{\text{DGC}}$ alone, allowing long-range

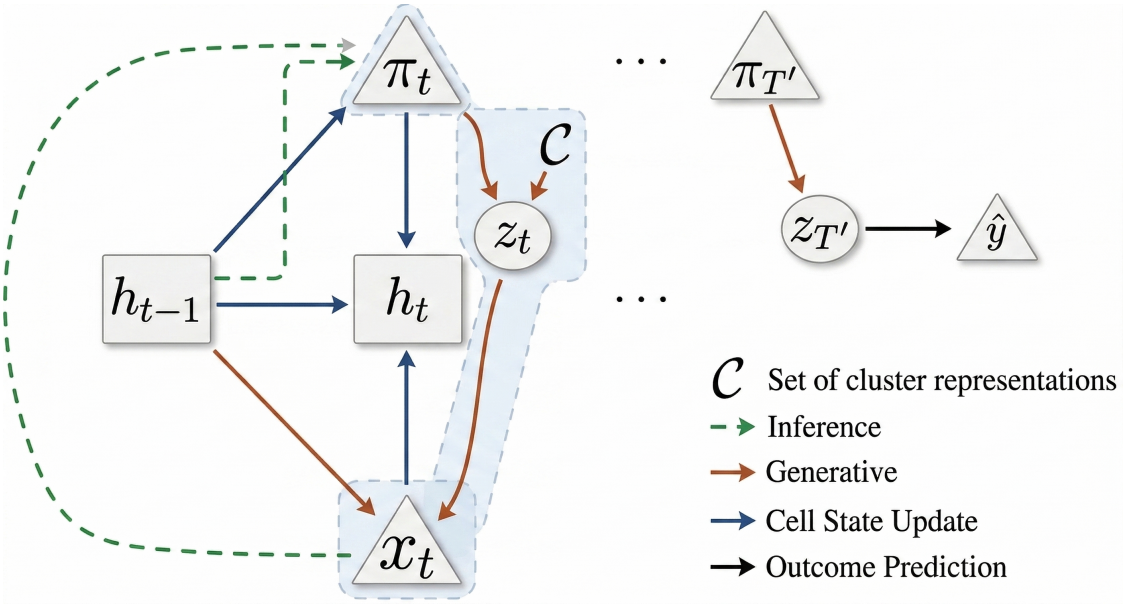


Figure 6.2: Overview of the probabilistic model in Cluster-VRNN. Square nodes denote internal cell states, triangular nodes denote model outputs (observation, cluster assignment, and outcome prediction), and circular nodes denote latent variables requiring sampling. The generative, inference, state update, and prediction steps are indicated by colored arrows, together with their inputs and outputs.

temporal dependencies to be more easily captured without requiring explicit conditioning of the data at time t on the entire previous history. The distributional components of each time step are formalized below using the gate functions, $g^{\text{DGC}}_{(\cdot)}$. Furthermore, a sketch diagram of the action of these components is shown in Figure 6.2.

Prior The model imposes a prior distribution on the set of cluster assignments at time t. This distribution varies over time and depends on the cell state $\mathbf{h}^{\text{DGC}}_{n,t-1}$, according to:

$$
\begin{aligned}
f_p(t) &:= p\big(\boldsymbol{\pi}^{\text{DGC}}_{n,t} \mid \mathbf{x}_{n,<t},\, \boldsymbol{\pi}^{\text{DGC}}_{n,<t}\big) = \text{Dir}(\boldsymbol{\alpha}^{\text{DGC,p}}_{n,t}), \\
\boldsymbol{\alpha}^{\text{DGC,p}}_{n,t} &:= \text{ReLU}(g^{\text{DGC}}_1(\mathbf{h}^{\text{DGC}}_{n,t-1})) + \epsilon^{\text{DGC}}.
\end{aligned} \tag{6.10}
$$

where $\boldsymbol{\alpha}^{\text{DGC,p}}_{n,t} \in \mathbb{R}^K_{>0}$ is the prior concentration parameter vector for patient n at time t, obtained from the prior network component. The activation of $\text{ReLU}(\cdot)$ enforces nonnegativity, while ϵ^{DGC} is a small threshold value added to ensure

that any value is not exactly 0, as required by Definition 14.. In practical experimentation, ϵ^{DGC} was set to 10^{-6}.

Inference The posterior distribution over the cluster assignments, given observation data $\mathbf{x}_{n,t}$, denoted as $q(\cdot)$ due to its role in the variational approximation in Section 6.3.3, has a structure similar to the prior; however, includes a component derived from the value of the current observation set:

$$\begin{aligned} q(t) &\coloneqq q\big(\boldsymbol{\pi}_{n,t}^{\text{DGC}} \mid \mathbf{x}_{n,\leq t},\ \boldsymbol{\pi}_{n,<t}^{\text{DGC}}\big) = \text{Dir}(\boldsymbol{\alpha}_{n,t}^{\text{DGC,i}}), \\ \boldsymbol{\alpha}_{n,t}^{\text{DGC,i}} &\coloneqq \text{ReLU}(g_3^{\text{DGC}}\big(\mathbf{h}_{n,t-1}^{\text{DGC}},\ g_2^{\text{DGC}}(\mathbf{x}_{n,t})\big)) + \epsilon^{\text{DGC}}. \end{aligned} \tag{6.11}$$

where $\boldsymbol{\alpha}_{n,t}^{\text{DGC,i}} \in \mathbb{R}_{\geq 0}^{K}$ is the inference concentration parameter vector derived from the Inference network, and the "feature extraction" network transforms the input $\mathbf{x}_{n,t}$ according to g_2^{DGC}. For future steps $t > T_n$, $\mathbf{x}_{n,t}$ is replaced by $\hat{\mathbf{x}}_{n,t}^{\text{DGC}}$ in the argument of g_2^{DGC}. Once the posterior has been estimated, a cluster assignment vector can be sampled, and during training, this is the cluster assignment at time t. However, at inference time, the mean is selected instead as the deterministic output is preferred.

Latent Representation Given a cluster assignment for the current time step t, the latent representation of the patient n is computed as the expectation of a mixture random variable with weights given by the current cluster assignment, and the mixtures corresponding to the cluster representations. This can be written as:

$$\mathbf{z}_{n,t}^{\text{DGC}} \coloneqq \sum_{k=1}^{K} \pi_{n,t,k}^{\text{DGC}}\, \boldsymbol{\mu}_k^{\text{DGC}} \ \in\ \mathbb{R}^l, \tag{6.12}$$

where $\pi_{n,t,k}^{\text{DGC}}$ denotes the k-th component of $\boldsymbol{\pi}_{n,t}^{\text{DGC}}$. Since $\boldsymbol{\pi}_{n,t}^{\text{DGC}} \in \Delta^{K-1}$, the representation lies in the convex hull of $\mathcal{E}^{\text{DGC}}$ and interpolates smoothly between cluster means according to the probabilities of assignment.

Generation DeepGC also includes a generative model over observations at time t conditioned on $\mathbf{h}_{n,t-1}^{\text{DGC}}$ and $\mathbf{z}_{n,t}^{\text{DGC}}$. More specifically, the model makes the following Gaussian assumption:

$$f_g(t) \coloneqq p\big(\mathbf{x}_{n,t} \mid \mathbf{x}_{n,<t},\, \boldsymbol{\pi}_{n,\leq t}^{\text{DGC}},\big) = \mathcal{N}(\boldsymbol{\mu}_{g,n,t}, \boldsymbol{\sigma}_{g,n,t}^2\, \mathbf{I}_{D_f}), \tag{6.13}$$

with parameters estimated from the gate functions:

$$\boldsymbol{\mu}_{g,n,t},\ \boldsymbol{\sigma}_{g,n,t}^2 \coloneqq g_5^{\text{DGC}}\big(\mathbf{h}_{n,t-1}^{\text{DGC}},\ g_4^{\text{DGC}}\big(\mathbf{z}_{n,t}^{\text{DGC}}\big)\big)\,, \tag{6.14}$$

where g_4^{DGC} extracts a feature embedding from $\mathbf{z}_{n,t}^{\text{DGC}}$, and g_5^{DGC} maps the concatenation of this embedding and $\mathbf{h}_{n,t-1}^{\text{DGC}}$ to the Gaussian parameters. The output of g_5^{DGC} implements an activation function ReLU() on the output dimensions that represent the variance, as it is not allowed to be negative; however, the dimensions that represent the mean are unrestricted. Similarly to the inference step, for future steps $t > T$, the generated observations $\hat{\mathbf{x}}_{n,t}^{\text{DGC}}$ replace $\mathbf{x}_{n,t}$ in all subsequent computations.

Hidden State Update. The hidden state is updated at each time step after the latent representation has been computed:

$$\mathbf{h}_{n,t}^{\text{DGC}} \coloneqq g_6^{\text{DGC}}\big(\mathbf{h}_{n,t-1}^{\text{DGC}},\ g_2^{\text{DGC}}(\mathbf{x}_{n,t})\,,\ g_4^{\text{DGC}}\big(\mathbf{z}_{n,t}^{\text{DGC}}\big)\big)\,, \tag{6.15}$$

where g_6^{DGC} corresponds to the recurrent update function of the RNN backbone. The feature extractors g_2^{DGC} and g_4^{DGC} are shared with the Inference and Generation components, respectively, and they are intended to be "feature extractors" relevant for the different modeling steps.

Outcome Prediction. Following the generative stage, the outcome is predicted from the latent representation in the final generated step:

$$\hat{\mathbf{y}}_n^{\text{DGC}} \coloneqq \text{Softmax}(g_7^{\text{DGC}}\big(\mathbf{z}_{n,T'}^{\text{DGC}}\big)) \ \in\ \Delta^{D_o-1}, \tag{6.16}$$

where $\hat{y}_{n,o}^{\text{DGC}} \in \mathbb{U}$ models the estimated probability of outcome o for patient n, and $\text{Softmax}(g_7^{\text{DGC}}(\cdot))$ maps the latent space to the outcome simplex. The use of $\mathbf{z}_{n,T'}^{\text{DGC}}$ ensures that the outcome prediction incorporates information from the generative stage.

Joint Likelihood. DeepGC's joint likelihood model over observations and outcomes can be factorized as:

$$p(\mathbf{X}_n,\, \mathbf{y}_n^{\text{mc}}) = \int p\big(\mathbf{y}_n^{\text{mc}} \mid \mathbf{x}_{n,\leq T'},\, \boldsymbol{\pi}_{n,\leq T'}^{\text{DGC}}\big) \times \prod_{t=1} p\big(\mathbf{x}_{n,t} \mid \mathbf{x}_{n,<t},\, \boldsymbol{\pi}_{n,\leq t}^{\text{DGC}}\big) \cdot p\big(\boldsymbol{\pi}_{n,t}^{\text{DGC}} \mid \mathbf{x}_{n,<t},\, \boldsymbol{\pi}_{n,<t}^{\text{DGC}}\big)\; \mathrm{d}\boldsymbol{\pi}_{n,\leq T'}^{\text{DGC}} \tag{6.17}$$

Applying the hidden state assumption, under which $\mathbf{h}_{n,t-1}^{\text{DGC}}$ summarizes information pertaining to the set $(\mathbf{x}_{n,<t}, \boldsymbol{\pi}_{n,<t}^{\text{DGC}})$, and the assumption that the outcome depends solely on $\mathbf{z}_{n,T'}^{\text{DGC}}$, this simplifies to:

$$p(\mathbf{X}_n,\, \mathbf{y}_n^{\text{mc}}) = \int p\big(\mathbf{y}_n^{\text{mc}} \mid \mathbf{z}_{n,T'}^{\text{DGC}}\big) \prod_{t=1} f_g(t)\, f_p(t)\; \mathrm{d}\boldsymbol{\pi}_{n,\leq T'}^{\text{DGC}}, \tag{6.18}$$

where the product runs over all time steps, including those in the generative stage. Note that a slight abuse of notation is used: $\mathbf{x}_{n,t}$ is set to $\hat{\mathbf{x}}_{n,t}^{\text{DGC}}$ for $t > T$. Unfortunately, this integral is generally intractable, motivating the variational training objective described in Section 6.3.3.

6.3.3 Model Training

To train the model, instead of maximizing the likelihood, DeepGC leverages the typical solution of maximization of an Evidence Lower Bound (ELBO) of the log-likelihood function, adapted to include the generative component of DeepGC. To do so, generated trajectories are included in the modeling objective to ensure plausible generation of future observations jointly with the patient's outcome.

The inference equation in Equation (6.11) is taken as the variational approximation model, which approximates the true posterior distribution. In the following result, Q will be used to refer to the variational distribution over all time steps, $Q \sim q(\boldsymbol{\pi}_{n,\leq T'}^{\text{DGC}} \mid \mathbf{x}_{n,\leq T'})$, which can be factorized using the Markovian assumption:

$$Q = \prod_{t=1}^{T'} q(\boldsymbol{\pi}_{n,t}^{\text{DGC}} \mid \mathbf{x}_{n,\leq t}, \mathbf{h}_{n,t}^{\text{DGC}}) \tag{6.19}$$

Consequently, the ELBO for the adapted (log-)likelihood can be derived as follows:

Lemma 5 (ELBO for DeepGC) *The log-likelihood* $\log p(\mathbf{X}_n, \mathbf{y}_n^{mc})$ *is lower-bounded by:*

$$\mathcal{L}_{DGC} \coloneqq \mathbb{E}_{\boldsymbol{\pi}^{DGC}_{n,\leq T'} \sim Q} \left[\log p\big(\mathbf{y}_n^{mc} \mid \mathbf{z}^{DGC}_{n,T'}\big) + \sum_{t=1}^{T'} \Big[\log f_g(t) - D_{KL}(q(t) \| f_p(t)) \Big] \right] \quad (6.20)$$

Proof 4 *The proof is nontrivial, but the proof for the ELBO of VRNN-GMM can be extended; a full proof is provided in the Appendix in Section C.1.*

ELBO $\mathcal{L}_{\text{DGC}}$ decomposes into three interpretable terms. Firstly, $\log p(\mathbf{y}_n^{\text{mc}} \mid \mathbf{z}^{\text{DGC}}_{n,T'})$, represents the log-likelihood for the outcome prediction, which encourages the final latent representation to be predictive of the observed outcome. Secondly, $\sum_t \log f_g(t)$, is the reconstruction log-likelihood, accumulated over all time steps, which encourages the generation network to be accurate in reproducing observed trajectories. Finally, the third term, $-\sum_t D_{\text{KL}}(q(t) \| f_p(t))$, quantifies the difference between the variational posterior and the prior at each step, so that a larger difference (in the sense of the definition $D_{\text{KL}}(\|)$) is penalized. In general, the terms balance predictive accuracy, trajectory reconstruction, and regularization of phenotype assignments.

KL Divergence Computation. At each time step t, both $q(t) = \text{Dir}(\boldsymbol{\alpha}^{\text{DGC,i}}_{n,t})$ and $f_p(t) = \text{Dir}(\boldsymbol{\alpha}^{\text{DGC,p}}_{n,t})$ are Dirichlet distributions, so that the KL divergence admits the closed-form expression derived in Equation (6.7):

$$\begin{aligned} D_{\text{KL}}(q(t) \| f_p(t)) = &\log \frac{\Gamma\left(\alpha^{\text{DGC},i}_{n,t,s}\right)}{\Gamma\left(\alpha^{\text{DGC},p}_{n,t,s}\right)} - \sum_{k=1}^{K} \log \frac{\Gamma\left(\alpha^{\text{DGC},i}_{n,t,k}\right)}{\Gamma\left(\alpha^{\text{DGC},p}_{n,t,k}\right)} \\ &+ \sum_{k=1}^{K} \left(\alpha^{\text{DGC},i}_{n,t,k} - \alpha^{\text{DGC},p}_{n,t,k}\right) \left(\psi\left(\alpha^{\text{DGC},i}_{n,t,k}\right) - \psi\left(\alpha^{\text{DGC},i}_{n,t,s}\right)\right), \end{aligned} \quad (6.21)$$

where $\alpha^{\text{DGC},i}_{n,t,s} \coloneqq \sum_{k=1}^{K} \alpha^{\text{DGC},i}_{n,t,k}$ and $\alpha^{\text{DGC},p}_{n,t,s} \coloneqq \sum_{k=1}^{K} \alpha^{\text{DGC},p}_{n,t,k}$ denote the respective concentration sums and $\psi(\cdot)$ is the digamma function. This closed form avoids Monte Carlo estimation of the KL term, reducing variance in the training signal, as well as errors associated with alternative optimization techniques.

Dirichlet Reparameterization In order to train DeepGC, one more important component remains. The expectation can be approximated by sampling, as is the process of selecting a cluster assignment during training, a well-known technique in probabilistic DL literature that shows it is a sufficiently good approximation to the expectation as it is averaged over a mini-batch. However, maximization of the inner term of $\mathcal{L}_{\text{DGC}}$ requires differentiation with respect to the inference network parameters — because a sampling operation was used to sample a cluster assignment, no gradients can be computed directly. All other terms are clearly differentiable, so no optimization concerns are raised.

In order to allow gradient backpropagation, Lemma 4 can be used to reparameterize samples from $\text{Dir}(\boldsymbol{\alpha}_{n,t}^{\text{DGC,i}})$ according to a normalization of gamma variables. Define a sequence of random variables $G_{n,t,k} \sim \Gamma_{\text{d}}(\alpha_{n,t,k}^{\text{DGC},i}, 1)$, and assume independence for each $k = 1, \ldots, K$. Then, $\pi_{n,t,k}^{\text{DGC}} = G_{n,t,k} / \sum_{k'} G_{n,t,k'}$ is a Dirichlet distribution with the target concentration parameter, $\boldsymbol{\alpha}_{n,t}^{\text{DGC,i}}$, due to Lemma 4. This disentangles the gradients through $\boldsymbol{\alpha}_{n,t}^{\text{DGC,i}}$ into components of each dimension of the concentration parameter. Unfortunately, the same solution does not apply to each Gamma distribution, and no known reparameterization is known that allows for gradient computation of the shape parameter. Instead, a common solution is to use an approximate inverse CDF approximation to the Gamma distribution:

Lemma 6 (Approximate Gamma Reparameterization) *Let $u \sim \mathcal{U}[0,1]$ and let $F_{\alpha}^{approx}(u) \coloneqq (u\alpha\Gamma(\alpha))^{1/\alpha}$ be a function of the unit interval. Then:*

$$G_{n,t,k} \approx F_{\alpha_{n,t,k}^{DGC,i}}^{approx}(u) \coloneqq \left(u\,\alpha_{n,t,k}^{DGC,i}\,\Gamma\left(\alpha_{n,t,k}^{DGC,i}\right)\right)^{1/\alpha_{n,t,k}^{DGC,i}} \tag{6.22}$$

is an approximate sample of $\Gamma_d(\alpha_{n,t,k}^{DGC,i}, 1)$ that is differentiable with respect to $\alpha_{n,t,k}^{DGC,i}$.

Proof 5 *By the inverse transform sampling theorem [355], if F_{α}^{-1} denotes the true inverse CDF of $\Gamma_d(\alpha, 1)$, then $F_{\alpha}^{-1}(u)$ is an exact sample from $\Gamma_d(\alpha, 1)$ for $u \sim \mathcal{U}[0,1]$. The true inverse CDF does not admit a closed form; however, the expression $(u\,\alpha\,\Gamma(\alpha))^{1/\alpha}$ has been shown to constitute a reasonable approximation for small values of α [355], so that $F_{\alpha_{n,t,k}^{DGC,i}}^{approx}(u) \approx F_{\alpha_{n,t,k}^{DGC,i}}^{-1}(u)$, and the result follows.*

Since Equation (6.22) is differentiable with respect to $\alpha_{n,t,k}^{\text{DGC},i}$, gradients propagate through the Dirichlet sampling step to the inference network parameters via Lemma 4.

Optimization The training objective of DeepGC is to maximize $\mathcal{L}_{\text{DGC}}$ over all learnable parameters: the gate functions $g_1^{\text{DGC}}, \ldots, g_7^{\text{DGC}}$ and the set of cluster representations, $\mathcal{E}^{\text{DGC}}$. Training is done in mini-batches via the Adam optimizer using a stochastic mini-batch gradient approach. Gradients are computed through time, from the last to the first time steps, according to a backpropagation over time technique used to train RNNs. All gate functions are implemented as shallow feed-forward networks with two hidden layers and a hyperbolic tangent activation, as well as a weight initialization given by the Glorot distribution, with the exception of the bias weights, which are initialized from a standard normal distribution. The remaining optimization details are detailed in Section 6.4.

6.3.4 Cluster Assignment and Trajectory Generation

Once training has been completed, DeepGC supports two relevant procedures: (a) dynamic cluster assignment and (b) generation of patient trajectories.

Dynamic Cluster Assignment For a new patient n who has not been seen and is available in the testing data, the input observation data are fed to DeepGC. The model executes the observed and generation stages, producing a sequence of posterior cluster assignment vectors $\boldsymbol{\pi}_{n,1}^{\text{DGC}}, \ldots, \boldsymbol{\pi}_{n,T'}^{\text{DGC}}$. As mentioned in Section 6.3.2, during inference, each cluster assignment vector is derived as the mean of the posterior distribution, rather than sampled:

$$\boldsymbol{\pi}_{n,t}^{\text{DGC}} := \mathbb{E}_{q(t)}\left[\text{Dir}(\boldsymbol{\alpha}_{n,t}^{\text{DGC,i}})\right] = \frac{\boldsymbol{\alpha}_{n,t}^{\text{DGC,i}}}{\alpha_{n,t,s}^{\text{DGC},i}} \in \Delta^{K-1}, \tag{6.23}$$

A hard assignment at each time step is then obtained by taking the most likely cluster:

$$\phi_t(\mathbf{X}_n) := \underset{k \in [K]}{\arg\max} \left(\pi_{n,t,k}^{\text{DGC}}\right). \tag{6.24}$$

so that the resulting sequence $\{\phi_t(\mathbf{X}_n\}_{t=1}^{T'}$ constitutes the dynamic trajectory of the patient's phenotype. The dynamic phenotype trajectory is used to characterize the clusters in both temporal and outcome aspects, extending the procedure in Section 5.3.1. Specifically, the cluster mean $\boldsymbol{\mu}_k^{\text{DGC}}$ encodes the latent centroid of the phenotype k; additionally, the mean observed trajectory in all patients and the time steps assigned to the cluster k is computed from the training data to derive a temporal representation of the cluster phenotype. On the other hand, the outcome aspect is derived from the empirical distribution of the outcomes within the cluster k.

Trajectory Generation In order to leverage its generative capabilities, the generative stage of DeepGC is also applied to generate time steps that occur after the last observed time step, T. The autoregressive procedure allows for sequential generation of future observations and estimated phenotypes, although generally predicting further into the future is increasingly more noisy and prone to erroneous behavior. In the testing set, observation values for $t > T$ are available (simply not fed to the model) in the HAVEN-T2RF data, so the quality of generated trajectories can be visually analyzed and investigated. This is further explored in the next section.

6.4 Implementation

This section describes the details of implementation specific to the experiments conducted for DeepGC, including anything related to benchmarking. The general evaluation structure follows that of Section 5.4; only the differences relevant to this chapter are described in full. Section 6.4.1 covers the experimental setup, including data processing and hyperparameter selection. Section 6.4.2 describes two post hoc analyses conducted to investigate the generative capabilities of DeepGC: an evaluation of the forward generation hyperparameter T', and a restricted input window experiment to assess the utility of forward generation of observations.

6.4.1 Experimental Setup

Dataset and Input Features Experiments were conducted in the HAVEN-T2RF and MIMIC-IV-ED datasets. In both settings, $D_o = 4$ outcome classes are considered; furthermore, all available temporal features are used as part of the model input. The clustering benchmarks (TSKM, SOM-VAE, and AC-TPC) of Chapter 5 are retained with the addition of VRNN-GMM as the most closely related clustering generative benchmark, and GMM, as a classical alternative for probabilistic clustering. Furthermore, CAMELOT is included in the list of models to compare as it shows improved performance in the results of Section 5.5. For the outcome prediction proxy task, SVM, XGBoost, NEWS2 (HAVEN-T2RF) and ESI (MIMIC-IV-ED) are computed similarly for comparison.

Data Splits and Initialization The data splitting protocol, similar to that of Chapter 5 Section 5.4.1, first split the data into an optimization set (60%) and a hold-out test set (40%) by random patient selection, with the optimization data further divided into training (80%) and validation (20%) sets. The distributional parameters for preprocessing, such as feature normalization, were estimated from training data only, according to Section 3.3. All gate functions in DeepGC were initialized with Glorot initialization, and bias weights were initialized according to a standard normal distribution. Cluster representation vectors were randomly initialized according to multivariate gaussians and optimized jointly with gate parameters during training, in contrast to the initialization based on $K\text{-}means$ used in CAMELOT(Section 5.3.3.2).

Temporal Aspect Metric Computation The framework of Section 3.4 defines the temporal aspect metrics (SIL, DBI, and VRC) with respect to a single cluster assignment per patient. However, for DeepGC, the assignment to a cluster is dynamic and each patient n is associated with assignments at each time step, represented as $\phi_1(\mathbf{X}_n), \ldots, \phi_{T'}(\mathbf{X}_n)$. To allow direct comparison with reference models within the framework of Section 3.4, the temporal aspect metrics for DeepGC were

computed using the assignment in the final observed time step, $\phi_T(\mathbf{X}_n)$, rather than the final generated step $\phi_{T'}(\mathbf{X}_n)$. This choice was preferred because it avoids conflating the metric performance with the quality of the generative component; furthermore, it provides a more sensible comparison with other benchmarks that follow this approach.

Hyperparameter Selection For each model and hyperparameter configuration, 10 experiments were conducted using a fixed set of 10 distinct seeds, each seed determining a different random number seed that was used in multiple operations such as initialization, sampling, and data splitting. Performance metrics were averaged across each seed, with the standard deviation reported in parentheses. Hyperparameters were explored by grid search in the ranges reported in Table A.3; the best-performing configuration is defined as that with the highest mean validation set AUROC over the 10 seeds. For the K hyperparameter the adapted Elbow method of Section 4.4.3 was applied, using a Friedman test with Bonferroni correction. T' was set at $T + 24$ hours for the HAVEN-T2RF data and $T + 4$ hours for MIMIC-IV-ED. This selection was empirically derived — in the HAVEN-T2RF context, representing observations occurring approximately concurrently with the potential adverse event.

A post hoc analysis related to T' is discussed in Section 6.5.3. Any configuration that did not produce predictions across one of the seeds for one or more outcome classes is disqualified, although exceptions were allowed for benchmark methods such as TSKM due to a complete inability to detect the lower-frequency outcome classes. After selecting the best hyperparameter configuration, all models were retrained on combined training and validation data, and the final performance was evaluated on the hold-out test set across the same 10 seeds. The reference hyperparameter grids were retained from Chapter 5, shown in Table A.2. All experiments were implemented in Python using Pytorch, Scikit-Learn, Matplotlib, and Numpy, with a single Tesla V100 GPU and eight Intel Xeon Gold 6246 CPUs at 3.30 GHz.

6.4.2 Post Hoc Analysis

Two post hoc analyses were conducted to investigate the properties of DeepGC that are not captured by the main evaluation metrics and that analyze the usefulness of the model with respect to clinical application. Firstly, an analysis of the choice of the forward generation horizon, T', was conducted considering different options for this hyperparameter. Secondly, Section 6.4.2.2 assessed the practical utility of the trajectory generation component by evaluating all models under a more restricted observation window, showcasing the ability of DeepGC to better leverage shorter time windows.

6.4.2.1 Predictive Horizon Analysis

As described in Section 6.3.1, T' governs the number of synthetic time steps produced beyond the last observed step in the input data; these steps include the estimation of observations sets and, in particular, latent patient representations, the last of which, $\mathbf{z}_{n,T'}^{\text{DGC}}$, is fed to the Predictor to compute an observation prediction. If $T' \approx T$ then the future generation capabilities are minimal and not relevant to improving the prediction of the outcome aspect, while if $T' \gg T$ the consecutive accumulation of generation error can lead to a noisy or even erroneous prediction. To investigate this, DeepGC was evaluated in HAVEN-T2RF across a range of T', including the degenerate case $T' = T$, where no forward generation is used. Writing $T' = T + \delta$ where $\delta \geq 0$ is *lag*, lags of $24, 20, 16, 12, 8, 4, 0$ were explored (note that the lag 0 corresponds to $T' = T$) and performance metrics are reported for the various lag factors. This analysis is shown only for HAVEN-T2RF, as it is harder to conduct such an ablation study in MIMIC-IV-ED. HAVEN-T2RF manually restricts the input window based on the occurrence of adverse events; therefore, while the input data are temporally restricted, there is known information that can be used to compare the future generation of trajectories. Unfortunately, this is not the case in MIMIC-IV-ED, as no data are available post-ED admission, and some patients are immediately transferred to other hospital wards at the last temporal step of admission. This also introduces challenges in qualitatively

visualizing the quality of the generated trajectories in MIMIC-IV-ED. Therefore, mainly HAVEN-T2RF trajectories are explored, although quantitative performance and benchmarking support the usefulness of DeepGC in different clinical settings. The results are shown in Section 6.5.3.1.

6.4.2.2 Utility of the Generative Component

A separate study was conducted to assess the practical utility of DeepGC with respect to generation under input restrictions motivated by clinical insight. Observations recorded closer to the outcome time are more likely to exhibit out-of-sample behavior, i.e., physiological variables are less similar to previously seen observations. This occurs because variables tend to worsen prior to an adverse event or, alternatively, improve prior to discharge from an admission. Therefore, it was hypothesized that restricting the available input window illustrates the usefulness of DeepGC over the benchmarks, since the generative component can synthesize the observations withheld better and therefore perform better with smaller amounts of required data.

To test this approach, all models were evaluated and retrained in HAVEN-T2RF using only the first 24 hours of the input window, corresponding to the period of $72-48$ hours that occurred before the outcome time. Under this restriction, the final observation available as part of the input data occurs approximately 48 hours prior to the outcome time. DeepGC received the same input, however, additionally synthesized observations over subsequent 24 hours through its generative framework and derived the outcome prediction with T' occurring 24 hours prior to the outcome time. The performance gap between all models is computed and discussed in Section 6.5.3.2.

6.5 Results

The results are organized according to the following structure. First, Section 6.5.1 covers the evaluation of the temporal aspect (Section 6.5.1), while the evaluation of the outcome aspect is discussed subsequently (Section 6.5.2), concluding with

a description of the learned clusters of DeepGC. The results of the post hoc studies are shown in Section 6.5.3. Note that the benchmark methods also considered in Chapter 5 have results reproduced from Section 5.5 as the experimental protocol is identical (not applicable to post hoc analyses).

6.5.1 Temporal Aspect

Clustering performance with respect to the temporal aspect is evaluated using SIL, DBI, and VRC metrics (Section 3.4.1). The results in both HAVEN-T2RF and MIMIC-IV-ED datasets are jointly reported in Table 6.1. Pure predictive benchmarks (SVM, XGBoost, NEWS2, and ESI) are excluded because the metrics suitable for the temporal aspect are not applicable to these benchmarks.

Dataset	Model	SIL ↑	VRC ↑	DBI ↓
HAVEN-T2RF	TSKM	**0.27 (.03)**	**372.4 (12.32)**	**1.91 (.04)**
	GMM	0.01 (.00)	98.7 (11.32)	3.56 (.28)
	SOM-VAE	0.17 (.02)	52.8 (9.32)	2.89 (.23)
	AC-TPC	0.06 (.02)	66.5 (18.7)	2.75 (.14)
	VRNN-GMM	0.11 (.03)	103.2 (4.56)	3.13 (.87)
	CAMELOT	0.11 (.04)	116.7 (6.2)	2.68 (.38)
	DeepGC	0.22 (.04)	189.9 (29.2)	1.97 (.15)
MIMIC-IV-ED	TSKM	**0.16 (.03)**	**471.9 (80.7)**	**1.59 (.10)**
	GMM	0.03 (.06)	143.4 (138.2)	4.41 (1.26)
	SOM-VAE	0.07 (.02)	342.1 (14.5)	2.58 (.12)
	AC-TPC	0.08 (.03)	245.0 (15.2)	3.36 (.17)
	VRNN-GMM	0.02 (.03)	41.6 (6.5)	3.80 (.62)
	CAMELOT	0.11 (.05)	302.5 (25.1)	2.77 (.20)
	DeepGC	0.10 (.01)	316.7 (15.1)	2.65 (.24)

Table 6.1: Temporal aspect performance of DeepGC and benchmarks in the HAVEN-T2RF and MIMIC-IV-ED datasets, evaluated using SIL, VRC, and DBI. Results are reported as the mean across 10 random seeds (standard deviations in parentheses). Predictive baselines (SVM, XGBoost, NEWS2, ESI) are excluded. For DeepGC, cluster assignments are taken at the final observed time step T. SOM-VAE, AC-TPC, and CAMELOT results are reproduced from Table 5.1. Bold values indicate statistically-significant best-performing model per metric within each dataset ($p < 0.05$, Friedman test with Bonferroni correction). Arrows next to metric names indicate whether higher (↑) or lower (↓) values are preferred.

TSKM achieves the highest scores on all temporal metrics in both datasets. The same two justifications as in Section 5.5.1 likely apply in this case as well: (a) TSKM directly optimizes the cluster boundaries in the input space and (b)

the evaluation metrics exhibit a bias towards convex cluster structures (in the input space). The newly introduced GMM benchmark performs remarkably poorly (0.00 mean SIL score on HAVEN-T2RF data, for example), which illustrates the complexity associated with the handling of temporal EHR data. Among DL models, DeepGC exhibits the best performance according to all metrics in the HAVEN-T2RF data — it has an average SIL score of 0.22, compared to 0.17 for SOM-VAE, an average of 189.9 VRC compared to 116.7 for CAMELOT, and with respect to DBI, 1.97 compared to 2.68 (CAMELOT). Furthermore, DeepGC and CAMELOT achieve very similar performance in MIMIC-IV-ED (SIL 0.10 and 0.11; VRC, 316.7 and 302.5; and DBI 2.65 and 2.77). It is hypothesized that the performance improvement over CAMELOT (in the HAVEN-T2RF data) lies in the holistic clustering modeling used in DeepGC: CAMELOT uses a training process that includes separate loss functions that target cluster formation and outcome prediction, while the objective in DeepGC leverages all such components naturally. The improved performance of DeepGC over VRNN-GMM is notable (for example, 0.22 *versus* 0.11 SIL score in HAVEN-T2RF or 2.65 *versus* 3.80 DBI in MIMIC-IV-ED) as both share the same foundational VRNN structure with a clustering component. This suggests that the additional structure in the probabilistic formulation of DeepGC may help to identify meaningful cluster phenotypes.

6.5.2 Outcome Aspect

Performance in the outcome aspect is reported below in Table 6.2, according to macro-average AUROC, F1, REC, and NMI metrics (Sections 3.4.2 and 3.4.3). Models that were purely unsupervised (TSKM, GMM, SOM-VAE, VRNN-GMM) generated outcome predictions via the proxy pipeline described in Section 3.4.2. For AC-TPC, CAMELOT, and DeepGC, the corresponding outcome prediction was used directly, derived from the associated internal model architecture. Furthermore, as mentioned above, the benchmarks present in Chapter 5 carried the results to this section as the experimental protocol was identical.

Dataset	Model	AUROC ↑	F1 ↑	REC ↑	NMI ↑
HAVEN-T2RF	SVM	0.59 (.02)	0.30 (.00)	0.32 (.01)	–
	XGBoost	0.65 (.01)	0.34 (.00)	0.35 (.00)	–
	NEWS2	0.61	0.29	0.34	–
	TSKM	0.55 (.01)	0.24 (.03)	0.26 (.02)	0.01 (.03)
	GMM	0.62 (.01)	0.30 (.06)	0.26 (.02)	0.12 (.04)
	SOM-VAE	0.61 (.09)	0.27 (.05)	0.27 (.03)	0.05 (.03)
	AC-TPC	0.68 (.01)	**0.38 (.01)**	0.36 (.01)	0.17 (.02)
	VRNN-GMM	0.66 (.01)	0.32 (.02)	0.29 (.01)	0.18 (.01)
	CAMELOT	**0.73 (.02)**	0.36 (.01)	**0.38 (.02)**	**0.20 (.03)**
	DeepGC	**0.72 (.02)**	0.36 (.02)	0.35 (.02)	**0.19 (.01)**
MIMIC-IV-ED	SVM	0.68 (.00)	0.25 (.00)	0.27 (.00)	–
	XGBoost	**0.73 (.01)**	0.29 (.00)	0.28 (.00)	–
	ESI	0.53	0.20	0.20	–
	TSKM	0.64 (.03)	0.22 (.00)	0.20 (.00)	0.00 (.01)
	GMM	0.58 (.02)	0.25 (.01)	0.28 (.03)	0.08 (.01)
	SOM-VAE	0.62 (.03)	0.21 (.01)	0.20 (.00)	0.02 (.00)
	AC-TPC	0.67 (.02)	0.31 (.01)	**0.35 (.01)**	0.09 (.00)
	VRNN-GMM	0.63 (.01)	0.18 (.01)	0.27 (.00)	0.06 (.01)
	CAMELOT	**0.72 (.02)**	**0.34 (.02)**	**0.36 (.02)**	**0.11 (.03)**
	DeepGC	0.70 (.01)	**0.32 (.03)**	**0.35 (.01)**	**0.11 (.00)**

Table 6.2: Outcome aspect performance of DeepGC and benchmarks in the HAVEN-T2RF and MIMIC-IV-ED datasets, evaluated using macro-average AUROC, F1, REC, and NMI. The HAVEN-T2RF input window length is 48 hours. Results are reported as the mean across 10 random seeds (standard deviations in parentheses). Dashes indicate metrics not applicable to predictive models. Bold values indicate statistically-significant best-performing model per metric within each dataset ($p < 0.05$, Friedman test with Bonferroni correction). Arrows next to metric names indicate whether higher (↑) or lower (↓) values are preferred.

Regarding the outcome aspect, DeepGC achieves a performance comparable to CAMELOT in AUROC in both datasets (0.72 and 0.73 in HAVEN-T2RF and 0.70 and 0.72 in MIMIC-IV-ED, respectively). Note, in particular, that there was a significant difference between DeepGC and XGBoost in the MIMIC-IV-ED setting; no such difference was found between DeepGC and CAMELOT. Furthermore, the model consistently outperforms other methods, with the exception of XGBoost (AUROC, HAVEN-T2RF) and AC-TPC (F1 and REC, MIMIC-IV-ED). VRNN-GMM and GMM exhibit poor performance, despite the probabilistic frameworks, as is expected due to their inability to incorporate information from the outcome aspect in the model. In particular, the performance improvement of DeepGC against VRNN-GMM is consistent across both datasets (0.73 *versus* 0.66 on

HAVEN-T2RF and 0.70 *versus* 0.63 on MIMIC-IV-ED). The similarity in performance with CAMELOT is interesting as it suggests that (a) the addition of a generative component and the switch to a probabilistic framework do not significantly hamper performance; however, (b) the performance gains are not identifiable according to a significant hypothesis test. This is likely due to the particular input time window that was used for experimentation, as was mentioned in Section 6.4.2.2, due to the characteristics of the observations closer to the outcome time.

Finally, similar to CAMELOT, DeepGC performs better than almost all other purely predictive benchmarks in identifying relevant predictions, which supports (but not *proves*) the underlying TPC assumption of the existence of phenotypes in the data cohorts considered. Experimentally, a large part of this is driven by the ability to identify smaller sampled outcomes, which classification methods naturally struggle with, even methods such as XGBoost which incorporate natural modifications to penalize lower-sampled classes more heavily.

Learned Phenotype Characterization For a representative experimental seed, the phenotypes learned by DeepGC in the HAVEN-T2RF data were further characterized. The model identified $K = 6$ phenotypes, consistent with the cluster count selected for CAMELOT in Chapter 5. There is a particular nuance associated with DeepGC — the model identified 6 clusters as determined by the assignment at the final observed time step; however, there were clusters for whom patients were assigned to at earlier time steps, yet, had no patients at $t = T$. For the purposes of characterization, the experimental seed was not allowed to have this property; however, it is important to observe the empirical occurrences of such events, which can be difficult to analyze and complicate the characterization of the phenotypes.

To understand the learned clusters and the dynamics learned by DeepGC, the *mean cluster assignment trajectory* and the corresponding standard error are computed for each cluster as follows. Note that $\boldsymbol{\pi}_{n,t}^{\text{DGC}}$ is a vector shaped like $T \times K$,

where the entries of the last dimension sum to 1 as they represent the probabilities of assignment to clusters. Then, for each cluster k, compute the average of $\boldsymbol{\pi}_{n,t}^{\text{DGC}}$ over the set of patients assigned to k at time $t = T$, and this is denoted as the mean cluster trajectory. The mean cluster trajectory is of shape $TtimesK$ and they are displayed in Figure 6.3 — subplot k corresponds to the mean cluster trajectory of cluster k, where each column vector k' is shown as a univariate trajectory of average assignment to cluster k'. A standard error interval around the mean cluster trajectory is also shown for each trajectory. Several clinical observations

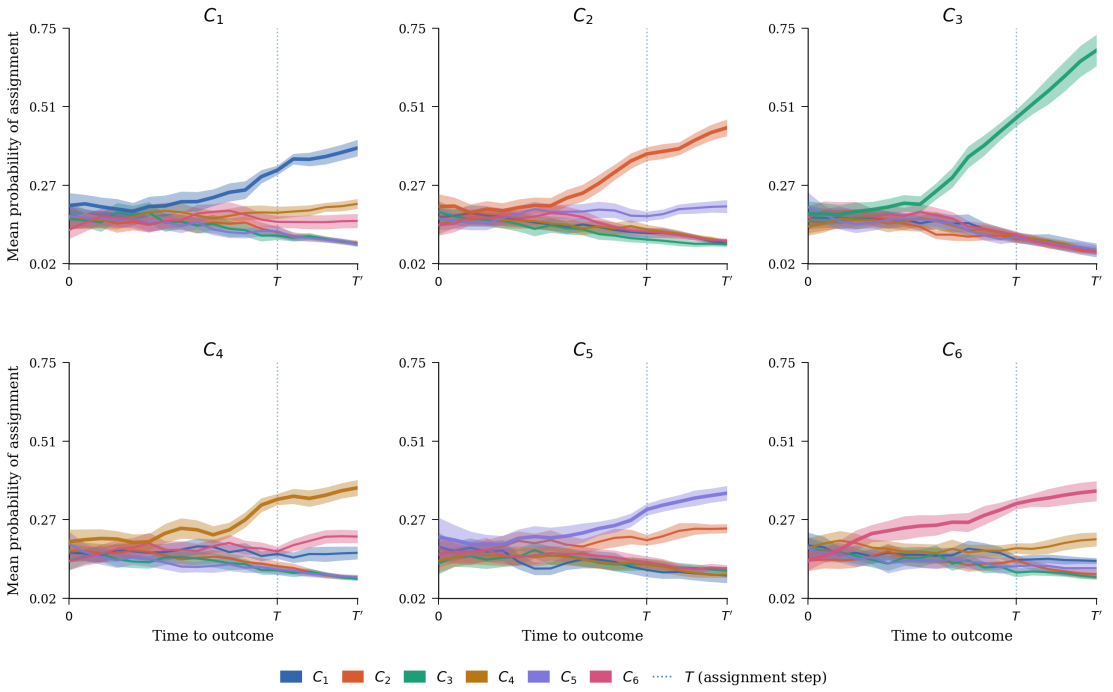


Figure 6.3: The mean cluster assignment trajectories learned by DeepGC in the HAVEN-T2RF data. $K = 6$ subplots are shown, corresponding to the mean cluster assignment trajectory for the corresponding cluster, which is determined based on assignment probabilities at $t = T$. In each subplot, column vectors are shown as univariate trajectories and labeled with the according target cluster. The mean trajectories are jointly plotted with a standard error deviation (shaded).

of interest can be derived from Figure 6.3. Across all phenotypes, the confidence in assignment appears low at admission onset, appearing as a roughly uniform assignment policy, and increases progressively over time. This is consistent with the expectation that physiological trajectories become more discriminative as

more data are observed. Interestingly, the level of confidence convergence differs across clusters, with cluster 3 exhibiting the clearest separation, in the sense that the average probability of assignment of patients in Cluster 3 to itself is high compared to other clusters. Clusters 4 and 6, on the contrary, retain non-negligible probability mass with respect to assignment to each other, which suggests representations are closer in the latent space. Cluster 1 also exhibits some interactions with this pair of clusters, however, it appears more limited. Similar behavior is observed between clusters 2 and 5, with an interesting observation that the confidence in assignment to the cluster 2 for patients in the cluster 2 is substantially higher than that of the cluster 5 (around 0.5 and 0.35 in $t = T'$, respectively), suggesting a nonsymmetric interaction between the clusters. A potential limitation of this analysis is that the size of the concentration parameter can heavily impact the variance, which can require strong parameterization and regularization of the gated functions to control. Complementary characterization of the dynamic nature of cluster assignment is shown in Figure C.1, which showcases, at different stages of admission, the probability of transition from the current assigned cluster to the assigned cluster at the next step. Finally, supplementary results for the clusters learned by the model in the MIMIC-IV-ED data are shown in Section C.3.

6.5.3 Post Hoc Analysis

6.5.3.1 Predictive Horizon Analysis

The sensitivity of DeepGC to the forward generation horizon T' is evaluated by varying *lag*, denoted as δ, which is the difference between T' and T. For each lag, performance metrics for the outcome aspect were calculated; the temporal performance metrics are identical due to the clustering evaluation at the last observed time step T. The results of the analysis on the HAVEN-T2RF data, with $\delta \in \{0, 4, 8, 12, 16, 20, 24\}$, are reported in Table 6.3.

The results in Table 6.3 indicate that DeepGC is relatively insensitive to the choice of δ in the current experimental setting. Within the range of lag values

δ (hours)	**AUROC** ↑	**F1** ↑	**REC** ↑	**NMI** ↑
0	0.70 (.01)	0.39 (.04)	0.37 (.01)	0.20 (.00)
4	0.69 (.03)	0.37 (.03)	0.37 (.01)	0.19 (.00)
8	0.70 (.02)	0.37 (.01)	0.37 (.02)	0.20 (.01)
12	0.72 (.01)	0.37 (.02)	0.36 (.01)	0.20 (.00)
16	0.72 (.02)	0.36 (.01)	0.35 (.02)	0.20 (.01)
20	0.73 (.03)	0.36 (.01)	0.34 (.02)	0.19 (.01)
24	0.72 (.02)	0.36 (.02)	0.35 (.02)	0.19 (.01)

Table 6.3: Outcome aspect performance of DeepGC in HAVEN-T2RF across forward generation horizons $\delta = T' - T \in \{0, 4, 8, 12, 16, 20, 24\}$ hours, evaluated using macro-average AUROC, F1, REC, and NMI. The case $\delta = 0$ corresponds to no forward generation. Results are reported as the mean across 10 random seeds (standard deviations in parentheses). Arrows next to metric names indicate whether higher (↑) or lower (↓) values are preferred.

considered, there are small positive improvements in AUROC as the lag increases (0.70 for $\delta = 0$ with 0.72 when $\delta = 24$), motivating the choice of $T' = T + 24$ hours in the main experiments. However, these gains in AUROC are associated with a small cost over F1 and REC (the highest F1 and REC scores obtained at $\delta = 0$), suggesting that extending the forward generation horizon introduces a slight degradation in the detection of minority outcome classes, but it is beneficial for majority classes. This is consistent with the accumulation of error at longer horizons in the autoregressive generation of DeepGC: as δ increases, the generated observations increasingly reflect the population-level prior rather than noisy, patient-specific trajectory dynamics.

When forward generation is disabled ($\delta = 0$), the performance of DeepGC on macro-average AUROC is slightly inferior to that of CAMELOTwhen both use 48 hours of input data (0.70 compared with 0.73); however, note that F1, REC and NMI are competitive, if not better (particularly REC). A suggested explanation for this effect is that the multi-term ELBO objective of DeepGC, which jointly balances reconstruction, KL regularization, and outcome prediction, is more challenging to optimize than the single-bottleneck objective of CAMELOT, and may produce, at time $t = T$, a more informative, but less discriminative representation of the state of the patient. As δ increases to moderate values, this gap narrows, with comparable performance at $\delta = 24$ as shown in Table 6.2,

which suggests that observations closer to the outcome time (likely to be more physiologically representative of the pre-outcome period) provide the Predictor with representations that are less noisy.

6.5.3.2 Generative Utility

The second post hoc analysis explored the performance of all models under a restricted input window and the results are shown in Table 6.4. All models were evaluated using only the first 24 hours of the HAVEN-T2RF input window, and DeepGC sets T' with a lag of 24 hours to the final observed time step.

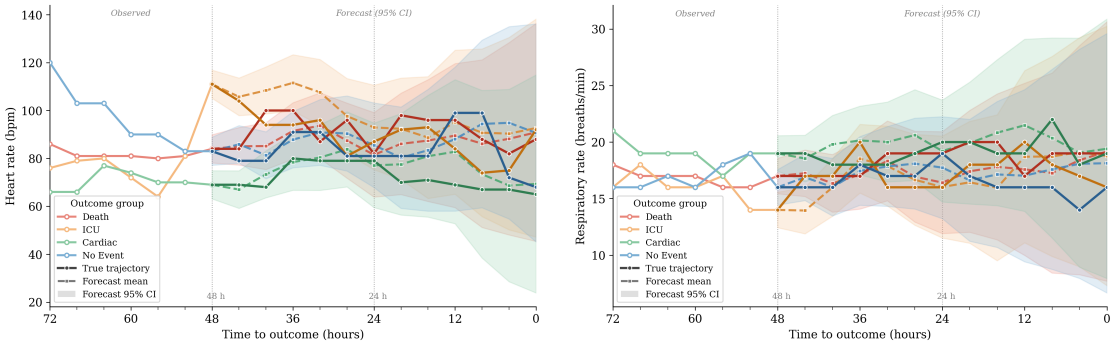
Model	AUROC ↑	F1 ↑	REC ↑	NMI ↑
SVM	0.56 (.01)	0.24 (.00)	0.26 (.01)	–
XGBoost	0.61 (.03)	**0.29 (.01)**	0.29 (.01)	–
NEWS2	0.53	0.20	0.20	–
TSKM	0.53 (.03)	0.20 (.02)	0.23 (.02)	0.03 (.00)
GMM	0.51 (.01)	0.13 (.05)	0.17 (.03)	0.14 (.01)
SOM-VAE	0.59 (.04)	0.22 (.03)	0.25 (.02)	0.07 (.04)
AC-TPC	0.58 (.03)	**0.30 (.01)**	0.30 (.00)	0.14 (.02)
VRNN-GMM	0.55 (.03)	0.26 (.01)	0.25 (.00)	0.12 (.01)
CAMELOT	0.63 (.02)	**0.29 (.01)**	**0.31 (.02)**	0.15 (.01)
DeepGC	**0.66 (.01)**	**0.30 (.03)**	**0.33 (.01)**	**0.17 (.02)**

Table 6.4: Outcome aspect performance of DeepGC and benchmarks in HAVEN-T2RF under a restricted 24-hour input window, evaluated using macro-average AUROC, F1, REC, and NMI. Results are reported as the mean across 10 random seeds (standard deviations in parentheses). For DeepGC, the following 24 hours are generated via the generative stage prior to prediction. Dashes indicate metrics not applicable to predictive models. Bold values indicate statistically-significant best-performing model per metric within each dataset ($p < 0.05$, Friedman test with Bonferroni correction). Arrows next to metric names indicate whether higher (↑) or lower (↓) values are preferred.

Restricting the input to 24 hours leads to a substantial degradation in performance in all models relative to the full 48-hour condition. This is naturally explained by the much smaller quantity of data, which heavily affects DL models, particularly, and the increase in noise in the data within this time period, as observations were missing more frequently than in the period of $48 - 24$ hours before the outcome time. Among all models considered, DeepGC achieves the highest AUROC (0.66) and the highest REC (0.33), while matching the best F1 score of AC-TPC (0.30) and slightly exceeding those of XGBoost and CAMELOT

(0.29 and 0.29, respectively). This is consistent with the hypothesis raised in Section 6.4.2.2: the generative component of DeepGC is capable of compensating for the time period that was removed, recovering some trajectory information through generation that is unavailable to all other models under the restriction.

In addition to the quantitative analysis, Figure 6.4 displays, in the restricted input settings, visualizations of the generated and observed trajectories of DeepGC over the subsequent period following the last observed time step. For each outcome class, a patient with the target outcome is sampled and plotted as representative of the class. For each patient, the 95% confidence interval was derived from the normal distribution modeling of the observation data. The figure illustrates the ability of DeepGC to produce physiologically plausible sequences of observations beyond the observed window. In particular, it can be seen that the generated observations are increasingly distinct from the true observed values as the auto-regression continues, particularly after T', which matches the expectations of the generative capabilities of the model. With further analysis and exploration, this could prove invaluable for clinicians in the future.



(**a**) Heart rate (bpm) (**b**) Respiratory rate (breaths min^{-1})

Figure 6.4: Observed and generated trajectories for representative patients from each outcome class (NoEvent, Death, ICU, Cardiac) of HAVEN-T2RF under a 24-hour input window. Solid lines denote observed data, while dashed lines denote the DeepGC forecast mean; shaded bands indicate 95% confidence intervals. Vertical markers at 48 h and 24 h indicate the end of the observed window and the start of the final pre-outcome period. Results are shown for heart rate (Figure 6.4a) and respiratory rate (Figure 6.4b).

6.6 Conclusion

This chapter introduced DeepGC, a novel DL model for dynamic phenotyping of EHR data, extending the static TPC framework considered in Chapters 4 and 5 to dynamic phenotyping, in which cluster assignment evolves over the course of a patient admission. Three main contributions were presented. Firstly, a fully generative probabilistic model was formulated in the joint space of patient observations, outcomes, and latent cluster assignment vectors. The latter were modeled as hidden variables following a Dirichlet distribution within a Cluster-VRNN architecture. Secondly, a closed-form ELBO objective was derived to train this approach, exploiting the analytic properties of the Dirichlet distribution to avoid costly and underperforming alternative methods such as Monte Carlo estimation. The objective of the model resulted in a decomposition into clearly interpretable components that correspond to the generation, prediction of the outcomes, and the quality of the variational approximation. Finally, a forward autoregressive procedure was introduced, capable of sequentially synthesizing observations for patients beyond the final observed time step. This is a capability absent from all other models considered in this thesis and denotes a step towards clinically actionable dynamic phenotyping.

Experiments on the HAVEN-T2RF and MIMIC-IV-ED datasets demonstrated that DeepGC exhibits competitive performance on the outcome aspect relative to CAMELOT and XGBoost, while outperforming all other benchmarks, with the exception of TSKM on the evaluation of the temporal aspect and AC-TPC in F1. This was consistent with the original expectation that dynamic generative cluster representations promote more coherent latent clusterings than a static bottleneck such as in CAMELOT. Post hoc analyses on HAVEN-T2RF confirmed that moderate forward generation horizons produce improvements in the context of the outcome aspect, consistent with the hypothesis that generated observations may help to encode early deterioration signals in the window preceding an outcome

time; furthermore, a subsequent analysis showed DeepGC is more robust and efficient than benchmarks when the available input window is restricted, providing evidence supporting the relevance of the generative capabilities of the model.

Scared to go onward,
but terrified to go back to what you were.

— Brandon Sanderson, The Way of Kings

7 Conclusion

This thesis explored the problem of Temporal Phenotypic Clustering (TPC), understood as the discovery of clinically meaningful subgroups of patient cohorts from EHR data that combine longitudinal physiological trajectories with information about future clinical outcomes. The motivation behind this research, established in Chapter 1, was that existing approaches to TPC suffered from four limitations that constrained their application to clinical practice. First, the absence of a formalized setting and a standardized evaluation procedure; second, insufficient validation in clinical settings characterized by severe imbalance in the outcome distributions; third, the limited interpretability in understanding the assignment of patients to clusters and the underlying phenotypes; and finally, the static modeling assumption that prevented updates to phenotype assignment as more patient data is observed. Four contributions were proposed to address each of the above limitations (Chapters 3 to 6). In particular, the latter three were evaluated on two real-world EHR datasets, HAVEN, a multihospital ward dataset, and its subcohort, HAVEN-T2RF, and MIMIC-IV-ED, publicly available data corresponding to admissions to the US emergency department, which provided supporting evidence of the methodological validity of the contributions, as well as initial indications of their generalizability.

7.1 Summary of Contributions

Contribution 1 — Chapter 3 — The first contribution of the thesis was the formalization of the experimental setting in which all subsequent work was conducted. The proposed framework was presented through three different components of an experimental pipeline. First, a formal definition of the TPC problem was introduced. The definition explicitly characterized the asymmetry between the different data aspects and provided terminology to distinguish TPC from related paradigms, such as multiview clustering and outcome prediction. The second proposed component consisted of a set of general-purpose preprocessing guidelines to process raw temporal EHR data in a format suitable for both model training and evaluation. The guidelines covered standard processing steps, such as outlier removal, admission selection, time windowing, thresholds to limit the proportion of missing values, imputation, and feature normalization, and examples were provided detailing the application of these guidelines in the context of HAVEN and MIMIC-IV-ED.

Chapter 3 proposed a standardized evaluation mechanism that could be applied to the data format specified by the preprocessing component. The evaluation of the performance of the clustering algorithm was separated into assessments of the temporal and outcome aspects, along with the corresponding suitable metrics. In particular, a proxy task was proposed for the evaluation of the model performance on the outcome aspect due to the future occurrence nature of this aspect. In general, the experimental setting framework proposed here provides a principled basis for the comparison of all models developed in later chapters; furthermore, it can support the development of novel TPC methods and their deployment to clinical settings.

Contribution 2 — Chapter 4 — The second contribution of the thesis was the development of a TPC model tailored to address the limitations associated with the presence of severe imbalances in multiclass outcome distributions. This is

common in real-life EHR cohorts, particularly with respect to minority classes being associated with worse clinical outcomes. The chapter first performed a benchmark comparison using the full HAVEN cohort, comparing three representative SOTA models from different algorithmic families: TSKM, a time-series aware version of the classical $K\text{-}means$ algorithm, SOM-VAE, a fully-unsupervised DL method for interpretable clustering of temporal data from EHR, and AC-TPC, the most prominent method in TPC that could take advantage of both data aspects. Using the proposed evaluation framework, the results demonstrated that all three methods consistently failed to stratify the minority outcome classes. To address this, a new method, AC-TPC$_w$, was introduced. AC-TPC$_w$ modified the training objective of AC-TPC by modifying two loss functions: a class-weighted categorical cross-entropy loss that downweights the majority class in the loss gradients and a latent-space cluster separation loss that promotes phenotypic diversity directly at the level of cluster representations. In the outcome aspect of the entire HAVEN cohort, AC-TPC$_w$ (0.70) achieved an improvement of 0.03 in the average AUROC on the best-performing benchmark, AC-TPC (0.67), and an improvement of 0.05 over the best classical benchmark (XGBoost, 0.65), with similar improvements observed in other metrics relevant to the outcome aspect. Moreover, the performance of the temporal aspect remained competitive with AC-TPC and other DL benchmarks. Qualitative analysis of identified phenotypes showed that AC-TPC$_w$ was more successful in the identification and phenotypic representation of minority outcome classes relative to benchmarks.

Contribution 3 — Chapter 5 — The third contribution introduced a novel deep learning model, CAMELOT, which used internal mechanisms to model the interpretability of clusters and cluster assignments, and that could be fully trained in an end-to-end fashion. The central architectural innovation was the AttentionRNNEncoder encoder, a neural network block that simultaneously produced latent representations for each patient through a high-capacity LSTM structure and then approximated its outputs by transformations applied per feature to each

observation of temporal aspect data. This approach yielded representations that exhibited the performance capabilities of an LSTM network and also allowed for the generation of a two-dimensional feature-time relevance map, providing interpretability without relying on post hoc explanation approaches. Two auxiliary loss functions were further introduced to address a failure mode that had been identified, cluster collapse, referring to convergence behavior to degenerate cluster configurations. The added loss functions encouraged full exploration of the set of available clusters.

Comparison with the benchmarks was conducted in a clinically defined subcohort of the HAVEN dataset (HAVEN-T2RF), and the results showed CAMELOT achieved better performance in the outcome aspect according to the macro-average AUROC (0.73, compared to 0.68 for AC-TPC), and matched the best performing benchmark in MIMIC-IV-ED data (0.72 versus 0.73 for XGBoost) while improving over other TPC benchmarks by at least 0.05 in HAVEN-T2RF (AC-TPC, 0.68). Similar behavior was observed in all other outcome metrics, with the exception of a 0.02 decrease in the average F1 score in the HAVEN-T2RF data compared to AC-TPC (0.36 and 0.38, respectively). Performance in the temporal aspect demonstrated a consistent improvement of CAMELOT over other DL approaches; however, performance remained below that of TSKM, which was attributed to the natural bias for convexity in temporal evaluation metrics and model design. Ablation studies confirmed the contribution of all proposed components by introducing new models extracted from CAMELOT by removing some of the architectural improvements and showing that ablated models exhibited weaker performance than CAMELOT in all metrics and datasets. Finally, qualitative analysis of the learned cluster and feature-time relevance maps on the HAVEN-T2RF data identified patterns related to FiO_2, SpO_2 and the respiratory rate, consistent with the established pathophysiology of T2RF and its underlying phenotypes. These results suggest that it is possible to design TPC methods that maintain strong quantitative performance while offering feature-time explanations of cluster assignment that can be used directly by clinical staff.

Contribution 4 — Chapter 6 — Finally, the fourth contribution proposed a framework that extended the TPC problem to a dynamic phenotypic setting, where phenotypes and phenotype assignments vary over time. The proposed method, DeepGC, was introduced as a fully generative probabilistic model in which cluster assignments are modeled according to Dirichlet distributions. Temporal modeling of the assignment of patients to clusters was performed using a recurrent variational architecture in which the cell state was used to derive the concentration parameters of the prior and posterior distributions at each time step. Furthermore, DeepGC proposed a component capable of synthesizing observations that have not yet been seen, allowing the plausible generation of future patient observation sequences beyond the last time step. In order to train the model, a closed-form ELBO was derived as the model objective, exploiting the analytic properties of the Dirichlet distribution and the architecture of DeepGC to avoid sources of instability associated with other variational optimization approaches such as Monte Carlo estimation. The objective could be decomposed into three natural components, specifically the error in modeling the outcome, the generation mismatch error, and the variational approximation error.

Empirically, DeepGC achieved the best performance in the temporal aspect metrics among all models and all data, with the sole exception of TSKM, with significant improvements relative to the benchmarks observed in the HAVEN-T2RF experiments, with increases of 0.05 in SIL (compared to SOM-VAE), 73.2 in VRC and a reduction of 0.71 in DBI (both compared to CAMELOT). Meanwhile, the performance in the outcome aspect was competitive and similar to that of CAMELOT (e.g. 0.72 *versus* 0.73 mean AUROC in the HAVEN-T2RF data). Post hoc analyses conducted with DeepGC investigated the effects of the forward generation horizon T', showing that increasing forward generation until the outcome time had a positive effect on AUROC while at the same time exhibiting a negative effect on the F1 and REC metrics, which is consistent with expectations related to accumulation of autoregressive generation error. Furthermore, when considering a restricted-input experiment in which the time window was

severely restricted (only the first 24 hours of the HAVEN-T2RF input window), DeepGC outperformed all benchmarks in AUROC (0.66 versus 0.63 for CAMELOT and 0.61 for XGBoost). Qualitative investigation of the learned clusters by DeepGC illustrated the behavior of dynamic cluster assignments as well as the quality of future generated observations, which deteriorates, as expected, as the lag with the last observed time step increases. In general, this provides evidence that the generative component and the dynamic phenotyping approach compensate for the limited presence of observational data, which can be directly relevant to early clinical monitoring and intervention decisions.

7.2 Limitations

The research conducted in this thesis should be interpreted in light of several limitations associated with different components of the work. Although these limitations do not invalidate the core findings, they deserve to be highlighted and considered carefully, particularly as they suggest avenues for future work.

7.2.1 Definition of the Computational Setting

The first set of limitations raised concerns the experimental and evaluation framework proposed in Chapter 3. The formulation of a proxy task for the evaluation of the outcome aspect provides a quantifiable avenue for investigating how a model performs with respect to the outcome aspect; however, it does not directly measure whether the discovered phenotypes are clinically meaningful or actionable to practitioners. The number of outcome classes is limited, constraining the space of possible outcome distributions per phenotype, and phenotypes naturally exhibit mixtures of outcome labels. Consequently, it is neither expected nor desirable for any model to achieve perfect performance in the proxy task. At the same time, because the outcome aspect is inherently future-facing and not available at inference time, designing an alternative evaluation procedure that reflects clinical utility more directly in a practical fashion is challenging.

A related issue concerns the absence of a joint scalar metric that aggregates performance on the temporal and outcome aspects simultaneously. As argued in Chapter 3, the metrics used for each aspect exhibit distinct dimensionalities and ranges and serve different purposes, so defining a single combination is not trivial. Nevertheless, this absence creates the possibility that different models optimize different subsets of metrics across aspects without a clear, scalable decision guideline to select the best model. In practice, model selection must therefore be conducted manually, a process that does not scale well and may introduce other forms of bias.

The foundational description of the TPC problem adopted in this thesis also does not resolve the challenge of choosing the number of clusters, K. In practice, an adapted Elbow method was used in experimentation; in turn, this required the use of a single metric on which to optimize this hyperparameter, rather than a holistic overview of performance. Moreover, even for a fixed model and hyperparameter configuration, the number of *learned* clusters (those with at least one patient assigned) can vary between runs, presenting core challenges to understanding the results. Finally, while the data processing guidelines were designed for general TPC settings, their concrete application to HAVEN and MIMIC-IV-ED revealed that key preprocessing criteria depended on dataset-specific thresholds, such as the minimum number of observations, the length of temporal windows, etc. The selection of these thresholds requires preliminary exploratory data analysis and domain knowledge, which limits the scalability of the guidelines.

7.2.2 Methodological Assumptions

The second group of limitations pertains to model design. All models introduced in this thesis, as well as the main TPC benchmark approaches, are deep learning models. Their high modeling capacity and compatibility with gradient-based optimization justify their use in this complex setting, but they are known to be prone to overfitting, sensitive to noise, and dependent on random initialization, so that results obtained under different seeds can be disparate (e.g. the number

of learned clusters). Furthermore, because training is formulated according to the optimization of a particular loss function, there is no guarantee that objective optimization leads to clinically meaningful phenotypes; in each chapter, a final qualitative analysis of the learned phenotypes was required to recognize the clinical relevance of the improvements in the individual components. Combined with the previously noted challenge of selecting K, this suggests that the reported phenotypes represent one of a set of potential admissible (i.e. plausible) phenotypic representations, rather than an underlying truth. This is further compounded by the fact that model objectives decompose into several competing components which are aggregated via scalar hyperparameters with no clinical significance. Optimization via a grid search approach with respect to a target metric only accentuates the aforementioned concerns about model alignment and the robustness of the identified patterns for clinical application.

Separately, a number of specific model assumptions or restrictions were imposed that may limit the suitability of these architectures to other settings. For example, CAMELOT does not allow for exact backpropagation of all loss terms with respect to all parameters due to its sampling component, and an approximate solution is needed to allow training. Similarly, DeepGC derives a closed-form ELBO objective as an approximation to the intractable marginal log-likelihood. Although this approach is supported by extensive theoretical work, the ELBO is a lower bound on the true objective. In addition, several distributional assumptions are imposed, such as Dirichlet-distributed cluster assignment parameters and Gaussian-distributed observation vectors, which may not occur in other datasets or clinical environments. Finally, it is important to note that these models retain many characteristics of black-box models that cannot be directly interpreted. Several architectural components were introduced to validate and understand how the models might behave in clinical practice, such as the AttentionRNNEncoder encoder in CAMELOT and the Cluster-VRNN block of DeepGC, but these mechanisms are fundamental approximations and cannot be guaranteed to fully represent the true underlying functions learned by the models.

7.2.3 Data Scope and Clinical Translation

The scope of the empirical evaluation conducted in this thesis is also associated with limitations that affect the translation of these models to clinical practice. All experiments were retrospective and observational. Although this is commonplace given the constraints of data availability and governance, prospective studies should be conducted to validate the performance of the models in live clinical environments. Furthermore, while the data used provided evidence to support the generalizability of the models, the empirical scope remains limited: HAVEN data were derived exclusively from a single NHS trust and MIMIC-IV-ED from a single hospital emergency department in the United States. The degree to which the learned phenotypic structures and associated methods generalize to other healthcare systems and disease areas remains to be established. This is further compounded by the preprocessing decisions required to transform raw EHR data into model-ready input formats, which included the exclusion or transformation of data that may not be handled similarly in real-time practice, and may reduce the direct applicability of the models without additional engineering.

Separately, it is possible that the identified phenotypes and relationships reflect primarily associations induced by the specific data used. No causal link was established between different temporal patterns and phenotype assignments, or between the phenotypes and the outcome distributions. This is important for clinical deployment, where understanding the underlying causal mechanisms is critical to determine which clinical actions are likely to positively alter patient trajectories. Integrating the TPC methods described in this thesis with causal inference studies and conducting a prospective evaluation would be important steps towards the deployment of these models.

7.3 Future Directions

The limitations identified in Section 7.2 motivate several concrete directions for future research. Five directions are discussed below, each representing a natural

extension of the work presented in this thesis, aligned with the three themes of the fundamental definition of the computational setting, methodology improvements, and the translation to clinical practice.

The first suggested direction directly addresses one of the previously identified limitations, namely the lack of a joint metric that simultaneously quantifies performance on the temporal and outcome aspects. A principled joint metric would enable the use of a single quantitative target against which different methods, models, and hyperparameter configurations could be compared, thus improving the scalability of model selection and reducing the dependence on manual interpretation of trade-offs across metrics. A well-designed joint metric would require the metrics for each aspect to be normalized or rendered comparable across dimensions and ranges. Furthermore, such metrics would be aggregated through a mathematical transformation whose parameters, if any, should admit a clear clinical interpretation. Existing work, such as the AUSIL metric [29], introduces a measure of temporal clustering performance intended to mimic the behavior of AUROC, standardizing the output range between 0 and 1; however, it does not address limitations associated with bias towards cluster convexity in the input space with respect to models that apply clustering in a latent space. When aggregating aspect-specific metrics, summary statistics derived from the Pareto front [356] provide a natural way to characterize trade-offs between aspects. Nevertheless, further validation is needed and the impacts of such approaches on the clinical relevance of the output are unknown.

A second avenue for future investigation refers to the K hyperparameter and the discrepancy between the target number of clusters and the learned number of clusters, which in turn can be separated into two distinct problems. First, K should reflect the underlying phenotypic structure of the data, rather than an analytical choice, estimated with high variance. Bayesian nonparametric approaches, such as Dirichlet process mixture models [348, 349], offer a suitable framework by allowing the number of mixture components to grow or shrink depending on the data requirements, thus removing the requirement to pre-specify K. The

probabilistic design of DeepGC is particularly amenable to this extension, but the optimization and scalability of such a model would be challenging in the high-dimensional context of EHR temporal data. To address the second problem, it would be desirable to modify training objectives or architectures to discourage the occurrence of empty clusters. Centroid-based approaches such as K-$means$ and TSKM are naturally relevant as they tend to produce non-empty clusters in practice, but their behavior does not transfer straightforwardly to deep clustering models. Penalties that depend on cluster assignment could be added to the training objective as a regularization factor, although such penalties would remain soft constraints rather than formal guarantees.

A third natural extension of this work is related to the expansion of the range of aspects and modalities considered within the TPC framework. In HAVEN and MIMIC-IV-ED, a natural direction to consider is the incorporation of a survival aspect that would model the time to the first occurrence of an adverse event using ideas from the field of Survival Analysis. This aspect would reflect the clinical reality that adverse events occur at patient-specific times rather than at a fixed horizon, and would introduce a wider variability range for each phenotype. Clustering methods that incorporate survival components have been explored in existing work [357, 358]. Adapting the proxy task and the training objectives of models such as CAMELOT and AC-TPC$_w$ to include a survival loss function such as the Cox partial likelihood is a natural consideration, although appropriately handling censoring is a nontrivial challenge. Separately, other aspects could be constructed by adding other EHR modalities, such as medication records and diagnostic or procedure codes, and this would precipitate architectural modifications to model them accordingly. As an example, diagnostic codes are typically recorded at the end of an admission, so, during the inference process, past admissions' codes would be available, however, not those corresponding to the current admission.

The fourth proposed direction concerns the robustness and uncertainty of qualitative outputs such as feature-time relevance maps (CAMELOT) and dy-

namic cluster assignments and generated trajectories (DeepGC). In the corresponding chapters, the outputs for particular seeds were primarily manually validated and, consequently, compared with domain knowledge. A formal framework for quantifying uncertainty and robustness across seeds, hyperparameter configurations, and models improves confidence in the derived results, thus increasing their clinical utility. Several key challenges would require addressing, such as how to compare the variable number of learned clusters, variable cluster membership, the introduction of internal components for uncertainty quantification, and the usage of external tools to analyze the stability of the results [69].

The direction most directly related to the long-term clinical impact of this work is the systematic validation of the proposed models. Future studies should include additional EHR datasets, disease contexts, and healthcare systems, to determine whether the phenotypic structures identified in HAVEN and MIMIC-IV-ED are robust to changes in the patient cohort [24]. Strategies for temporal validation, in which models are trained in different temporal periods or trained using variable window lengths or data availability, would be useful to determine the stability and relevance of learned phenotypes. An important next step is the incorporation of causal inference frameworks within TPC methods to model causal links in identified phenotypes and potentially inform clinical interventions, and parts of this idea are being explored [359, 360]. This work would move TPC from a purely descriptive paradigm to one that contributes directly to understanding causal pathways, potentially leading to impacts on clinical intervention decision-making and improving healthcare delivery.

7.4 Concluding Remarks

The research conducted in this thesis contributed to the growing literature on TPC through the formalization of a computational setting and the development of methods that address several key problems routinely seen in clinical practice. The retrospective experiments served as a proof-of-concept demonstration of the potential of TPC methods to inform clinical decision-making. This work is

intended to serve as a foundation for future research in TPC, with the long-term goal of improving patient outcomes in real-world settings.

References

[1] A. C. Gordon, N. Alipanah-Lechner, L. D. Bos, J. Dianti, J. V. Diaz, S. Finfer, T. Fujii, E. J. Giamarellos-Bourboulis, E. C. Goligher, M. N. Gong, E. Karakike, V. X. Liu, N. Lumlertgul, J. C. Marshall, D. K. Menon, N. J. Meyer, E. S. Munroe, S. N. Myatra, M. Ostermann, H. C. Prescott, A. G. Randolph, E. J. Schenck, C. W. Seymour, M. Shankar-Hari, M. Singer, M. R. Smit, A. Tanaka, F. S. Taccone, B. T. Thompson, L. K. Torres, T. van der Poll, J. L. Vincent and C. S. Calfee. 'From ICU Syndromes to ICU Subphenotypes: Consensus Report and Recommendations for Developing Precision Medicine in the ICU'. In: *American Journal of Respiratory and Critical Care Medicine* 210.2 (July 2024), pp. 155–166. DOI: 10.1164/rccm.202311-2086SO.

[2] J. Pathak, A. N. Kho and J. C. Denny. 'Electronic health records-driven phenotyping: challenges, recent advances, and perspectives'. In: *Journal of the American Medical Informatics Association* 20.e2 (Dec. 2013), e206–e211. DOI: 10.1136/amiajnl-2013-002428.

[3] A. Agusti, E. Bel, M. Thomas, C. Vogelmeier, G. Brusselle, S. Holgate, M. Humbert, P. Jones, P. G. Gibson, J. Vestbo, R. Beasley and I. D. Pavord. 'Treatable traits: toward precision medicine of chronic airway diseases'. In: *European Respiratory Journal* 47.2 (2016), pp. 410–419. DOI: 10.1183/13993003.01359-2015. eprint: http://erj.ersjournals.com/content/erj/47/2/410.full.pdf.

[4] J. Versnjak, T. Kuehne, P. Fahjen, N. Jovanovic, U. Löber, G. G. Schiattarella, N. Wilck, H. Gerhardt, D. N. Müller, F. Edelmann, P. Mertins, R. Eils, M. Gotthardt, S. K. Forslund, B. Wild and M. Kelm. 'Deep phenotyping of heart failure with preserved ejection fraction through multi-omics integration'. In: *European Journal of Heart Failure* 27.12 (Dec. 2025), pp. 3243–3259. DOI: 10.1002/ejhf.70041.

[5] E. H. Bel. 'Clinical phenotypes of asthma'. In: *Current Opinion in Pulmonary Medicine* 10.1 (Jan. 2004), pp. 44–50. DOI: 10.1097/00063198-200401000-00008.

[6] S. Mirza and R. Benzo. 'Chronic Obstructive Pulmonary Disease Phenotypes: Implications for Care'. In: *Mayo Clinic Proceedings* 92.7 (July 2017), pp. 1104–1112. DOI: 10.1016/j.mayocp.2017.03.020.

[7] J. R. Robinson, W.-Q. Wei, D. M. Roden and J. C. Denny. 'Defining Phenotypes from Clinical Data to Drive Genomic Research'. In: *Annual Review of Biomedical Data Science* 1 (July 2018), pp. 69–92. DOI: 10.1146/annurev-biodatasci-080917-013335.

[8] C. Åkerlund and A. Ercole. 'Data-driven approaches to reveal the pathobiological heterogeneity in patients with traumatic brain injury'. In: *Intensive Care Medicine* 49.9 (Sept. 2023), pp. 1107–1109. DOI: 10.1007/s00134-023-07156-y.

[9] D. W. Russell, J. M. Wells and J. E. Blalock. 'Disease phenotyping in chronic obstructive pulmonary disease: the neutrophilic endotype'. In: *Current Opinion in Pulmonary Medicine* 22.2 (Mar. 2016), pp. 91–99. DOI: 10.1097/MCP.0000000000000238.

[10] C. Liu, F. Wang, J. Hu and H. Xiong. 'Temporal Phenotyping from Longitudinal Electronic Health Records: A Graph Based Framework'. In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Sydney, NSW, Australia: Association for Computing Machinery, 2015, pp. 705–714. DOI: 10.1145/2783258.2783352.

[11] Y. Wang, J. N. Stroh, G. Hripcsak, C. C. Low Wang, T. D. Bennett, J. Wrobel, C. Der Nigoghossian, S. Mueller, J. Claassen and D. J. Albers. 'A methodology of phenotyping ICU patients from EHR data: high-fidelity, personalized, and interpretable phenotypes estimation'. In: *Journal of Biomedical Informatics* 148 (Dec. 2023), p. 104547. DOI: 10.1016/j.jbi.2023.104547.

[12] H. Estiri, T. H. McCoy, K. B. Wagholikar, A. P. Goodson, K. Murphy and S. N. Murphy. 'High-Throughput Phenotyping with Temporal Sequences'. In: *bioRxiv* (2019). DOI: 10.1101/590307. bioRxiv: 590307.

[13] C. Lee and M. V. D. Schaar. 'Temporal Phenotyping using Deep Predictive Clustering of Disease Progression'. In: *Proceedings of the 37th International Conference on Machine learning*. Journal of Machine Learning Research, June 2020, pp. 5767–5777.

[14] G. Cezard, C. T. McHale, F. Sullivan, J. K. F. Bowles and K. Keenan. 'Studying trajectories of multimorbidity: a systematic scoping review of longitudinal approaches and evidence'. In: *BMJ Open* 11.11 (2021). DOI: 10.1136/bmjopen-2020-048485. eprint: https://bmjopen.bmj.com/content/11/11/e048485.full.pdf.

[15] D. H. Kim, A. Jensen, K. Jones, S. Raghavan, L. S. Phillips, A. Hung, Y. V. Sun, G. Li, P. Reaven, H. Zhou and J. J. Zhou. 'A platform for phenotyping disease progression and associated longitudinal risk factors in large-scale EHRs, with application to incident diabetes complications in the UK Biobank'. In: *JAMIA Open* 6.1 (Apr. 2023), ooad006. DOI: 10.1093/jamiaopen/ooad006. eprint: https://academic.oup.com/jamiaopen/article-pdf/6/1/ooad006/49138927/ooad006.pdf.

[16] R. Pivovarov, A. J. Perotte, E. Grave, J. Angiolillo, C. H. Wiggins and N. Elhadad. 'Learning probabilistic phenotypes from heterogeneous EHR data'. In: *Journal of Biomedical Informatics* 58 (2015), pp. 156–165. DOI: 10.1016/j.jbi.2015.10.001.

[17] K. G. Aragam, A. Dobbyn, R. Judy, M. Chaffin, K. Chaudhary, G. Hindy, A. Cagan, P. Finneran, L.-C. Weng, R. J. F. Loos, G. Nadkarni, J. H. Cho, R. L. Kember, A. Baras, J. Reid, J. Overton, A. Philippakis, P. T. Ellinor, S. T. Weiss, D. J. Rader, S. A. Lubitz, J. W. Smoller, E. W. Karlson, A. V. Khera, S. Kathiresan, R. Do, S. M. Damrauer and P. Natarajan. 'Limitations of

Contemporary Guidelines for Managing Patients at High Genetic Risk of Coronary Artery Disease'. In: *Journal of the American College of Cardiology* 75.22 (June 2020), pp. 2769–2780. DOI: 10.1016/j.jacc.2020.04.027.

[18] S. H. Woolf, R. Grol, A. Hutchinson, M. Eccles and J. Grimshaw. 'Clinical guidelines: potential benefits, limitations, and harms of clinical guidelines'. In: *BMJ* 318.7182 (Feb. 1999), pp. 527–530. DOI: 10.1136/bmj.318.7182.527.

[19] M. A. F. Pimentel, O. C. Redfern, S. Gerry, G. S. Collins, J. Malycha, D. Prytherch, P. E. Schmidt, G. B. Smith and P. J. Watkinson. 'A Comparison of the Ability of the National Early Warning Score and the National Early Warning Score 2 to Identify Patients at Risk of In-Hospital Mortality: A Multi-centre Database Study'. In: *Resuscitation* 134 (2019), pp. 147–156. DOI: 10.1016/j.resuscitation.2018.09.026.

[20] M. E. B. Smith, J. C. Chiovaro, M. O'Neil, D. Kansagara, A. R. Quiñones, M. Freeman, M. Motu'apuaka and C. G. Slatore. *Early Warning System Scores: A Systematic Review*. VA Evidence-based Synthesis Program Reports. Washington, DC: Department of Veterans Affairs, Jan. 2014.

[21] J. M. Banda, M. Seneviratne, T. Hernandez-Boussard and N. H. Shah. 'Advances in Electronic Phenotyping: From Rule-Based Definitions to Machine Learning Models'. In: *Annual Review of Biomedical Data Science* 1 (July 2018), pp. 53–68. DOI: 10.1146/annurev-biodatasci-080917-013315.

[22] A. K. Jain. 'Data Clustering: 50 Years Beyond K-Means'. In: *Pattern Recognition Letters* 31 (8 June 2010), pp. 651–666. DOI: 10.1016/j.patrec.2009.09.011.

[23] V. Estivill-Castro. 'Why So Many Clustering Algorithms: A Position Paper'. In: *ACM SIGKDD Explorations Newsletter* 4.1 (2002), pp. 65–75. DOI: 10.1145/568574.568575.

[24] T. J. Loftus, B. Shickel, J. A. Balch, P. J. Tighe, K. L. Abbott, B. Fazzone, E. M. Anderson, J. Rozowsky, T. Ozrazgat-Baslanti, Y. Ren, S. A. Berceli, W. R. Hogan, P. A. Efron, J. R. Moorman, P. Rashidi, G. R. U. Jr and A. Bihorac. 'Phenotype Clustering in Health Care: A Narrative Review for Clinicians'. In: *Frontiers in Artificial Intelligence* 5 (Aug. 2022). DOI: 10.3389/frai.2022.842306.

[25] S. Aghabozorgi, A. S. Shirkhorshidi and T. Y. Wah. 'Time-Series Clustering - A Decade Review'. In: *Information Systems* 53 (May 2015), pp. 16–38. DOI: 10.1016/j.is.2015.04.007.

[26] M. Ghassemi, T. Naumann, P. Schulam, A. L. Beam, I. Y. Chen and R. Ranganath. 'A Review of Challenges and Opportunities in Machine Learning for Health'. In: *AMIA Joint Summits on Translational Science Proceedings* 2020 (2020), pp. 191–200.

[27] Y. Yang and H. Wang. 'Multi-view clustering: A survey'. In: *Big Data Mining and Analytics* 1 (2 2018), pp. 83–107. DOI: 10.26599/BDMA.2018.9020003.

[28] R. A. Hubbard, J. Xu, R. Siegel, Y. Chen and I. Eneli. 'Studying pediatric health outcomes with electronic health records using Bayesian clustering and trajectory analysis'. In: *Journal of Biomedical Informatics* 113 (Jan. 2021). Epub 2020 Dec 11, p. 103654. DOI: 10.1016/j.jbi.2020.103654.

[29] Y. Qin, M. van der Schaar and C. Lee. 'T-Phenotype: Discovering Phenotypes of Predictive Temporal Patterns in Disease Progression'. In: *Proceedings of the 26th International Conference on Artificial Intelligence and Statistics*. Vol. 206. Proceedings of Machine Learning Research. 2023, pp. 3466–3492.

[30] C. Shivade, P. Raghavan, E. Fosler-Lussier, P. J. Embi, N. Elhadad, S. B. Johnson and A. M. Lai. 'A Review of Approaches to Identifying Patient Phenotype Cohorts Using Electronic Health Records'. In: *Journal of the American Medical Informatics Association* 21.2 (Mar. 2014). Published online 07 November 2013, pp. 221–230. DOI: 10.1136/amiajnl-2013-001935.

[31] G. Preud'homme, K. Duarte, K. Dalleau, C. Lacomblez, E. Bresso, M. Smaïl-Tabbone, M. Couceiro, M.-D. Devignes, M. Kobayashi, O. Huttin, J. P. Ferreira, F. Zannad, P. Rossignol and N. Girerd. 'Head-to-Head Comparison of Clustering Methods for Heterogeneous Data: a Simulation-Driven Benchmark'. In: *Scientific Reports* 11 (1 Feb. 2021), p. 4202. DOI: 10.1038/s41598-021-83340-8.

[32] A. S. Desuky and S. Hussain. 'An Improved Hybrid Approach for Handling Class Imbalance Problem'. In: *Arabian Journal for Science and Engineering* 46.4 (Jan. 2021), pp. 3853–3864. DOI: 10.1007/s13369-021-05347-7.

[33] J. Adebayo, J. Gilmer, M. Muelly, I. Goodfellow, M. Hardt, B. Kim and G. Brain. 'Sanity Checks for Saliency Maps'. In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi and R. Garnett. Curran Associates, Inc., 2018, pp. 9505–9515.

[34] S. M. Lundberg and S.-I. Lee. 'A Unified Approach to Interpreting Model Predictions'. In: *Advances in Neural Information Processing Systems* 30 (2017).

[35] O. C. Redfern, M. A. F. Pimentel, D. Prytherch, P. Meredith, D. A. Clifton, L. Tarassenko, G. B. Smith and P. J. Watkinson. 'Predicting In-Hospital Mortality and Unanticipated Admissions to the Intensive Care Unit Using Routinely Collected Blood Tests and Vital Signs: Development and Validation of a Multivariable Model'. In: *Resuscitation* 133 (2018), pp. 75–81. DOI: 10.1016/j.resuscitation.2018.09.021.

[36] M. A. F. Pimentel, O. C. Redfern, J. Malycha, P. Meredith, D. Prytherch, J. Briggs, J. D. Young, D. A. Clifton, L. Tarassenko and P. J. Watkinson. 'Detecting Deteriorating Patients in the Hospital: Development and Validation of a Novel Scoring System'. In: *American Journal of Respiratory and Critical Care Medicine* 204.1 (2021). PMID: 33525997, pp. 44–52. DOI: 10.1164/rccm.202007-2700OC. eprint: https://doi.org/10.1164/rccm.202007-2700OC.

[37] A. E. Johnson, L. Bulgarelli, L. Shen, A. Gayles, A. Shammout, S. Horng, T. J. Pollard, B. Moody, B. Gow, L. wei H. Lehman, L. A. Celi and R. G. Mark. 'MIMIC-IV, a freely accessible electronic health record dataset'. In: *Scientific Data* 10 (1 Dec. 2023). DOI: 10.1038/s41597-022-01899-x.

[38] H. Aguiar, M. Santos, P. Watkinson and T. Zhu. 'Phenotyping Clusters of Patient Trajectories Suffering from Chronic Complex Disease'. In: *Proceedings of the Machine Learning for Health (ML4H) Workshop at NeurIPS 2020*. 2020.

[39] H. Aguiar, M. Santos, P. Watkinson and T. Zhu. 'Learning of Cluster-based Feature Importance for Electronic Health Record Time-series'. In: *Proceedings of the 39th International Conference on Machine Learning*. Ed. by K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu and S. Sabato. Vol. 162. Proceedings of Machine Learning Research. PMLR, July 2022, pp. 161–179.

[40] H. Aguiar, M. Santos, P. Watkinson and T. Zhu. 'Variational Modelling of Temporal EHRs for Phenotypic Clustering'. In: *ICLR 2023 Workshop on Time Series Representation Learning for Health*. 2023.

[41] L.-C. Nisip Avram, T. M. Porosnicu, P. Hogea, E. Tudorache, E. Hogea and C. Oancea. 'Phenotypes of Exacerbations in Chronic Obstructive Pulmonary Disease'. In: *Journal of Clinical Medicine* 14.9 (2025). DOI: 10.3390/jcm14093132.

[42] S. A. Christenson. 'COPD Phenotyping'. In: *Respiratory Care* 68.7 (July 2023), pp. 871–880. DOI: 10.4187/respcare.11035.

[43] S. P. Bhatt, A. Agusti, M. Bafadhel, S. A. Christenson, J. Bon, G. C. Donaldson, D. D. Sin, J. A. Wedzicha and F. J. Martinez. 'Phenotypes, Etiotypes, and Endotypes of Exacerbations of Chronic Obstructive Pulmonary Disease'. In: *American Journal of Respiratory and Critical Care Medicine* 208.10 (2023), pp. 1026–1041. DOI: 10.1164/rccm.202209-1748SO.

[44] W. Barker, W. Chang, J. Everson, M. Gabriel, V. Patel, C. Richwine and C. Strawley. 'The Evolution of Health Information Technology for Enhanced Patient-Centric Care in the United States: Data-Driven Descriptive Study'. In: *Journal of Medical Internet Research* 26 (2024), e59791. DOI: 10.2196/59791.

[45] J. X. Jiang, K. Qi, G. Bai and K. Schulman. 'Pre-Pandemic Assessment: A Decade of Progress in Electronic Health Record Adoption among U.S. Hospitals'. In: *Health Affairs Scholar* 1.5 (Oct. 2023), qxad056. DOI: 10.1093/haschl/qxad056.

[46] Y. Shen, J. Yu, J. Zhou and G. Hu. 'Twenty-Five Years of Evolution and Hurdles in Electronic Health Records and Interoperability in Medical Research: Comprehensive Review'. In: *Journal of Medical Internet Research* 27 (Jan. 2025), e59024. DOI: 10.2196/59024.

[47] N. S. Abul-Husn and E. E. Kenny. 'Personalized Medicine and the Power of Electronic Health Records'. In: *Cell* 177.1 (Mar. 2019), pp. 58–69. DOI: 10.1016/j.cell.2019.02.039.

[48] Z. Zeng, Y. Deng, X. Li, T. Naumann and Y. Luo. 'Natural Language Processing for EHR-Based Computational Phenotyping'. In: *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 16.1 (Jan. 2019), pp. 139–153. DOI: 10.1109/TCBB.2018.2849968.

[49] J. Abonyi and B. Feil. *Cluster Analysis for Data Mining and System Identification*. 1st ed. Birkhäuser Basel, June 2007, pp. 1–306. DOI: 10.1007/978-3-7643-7988-9.

[50] M. Piernik and T. Morzy. 'A Study on Using Data Clustering for Feature Extraction to Improve the Quality of Classification'. In: *Knowledge and Information Systems* 63 (7 May 2021), pp. 1771–1805. DOI: 10.1007/s10115-021-01572-6.

[51] X. Yan, I. Misra, A. Gupta, D. Ghadiyaram and D. Mahajan. 'Clusterfit: Improving generalization of visual representations'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 6509–6518.

[52] C. Xiao, E. Choi and J. Sun. 'Opportunities and Challenges in Developing Deep Learning Models Using Electronic Health Records Data: A Systematic Review'. In: *Journal of the American Medical Informatics Association* 25.10 (Oct. 2018), pp. 1419–1428. DOI: 10.1093/jamia/ocy068.

[53] B. H. Aubaidan, R. A. Kadir, M. T. Lajb et al. 'A Review of Intelligent Data Analysis: Machine Learning Approaches for Addressing Class Imbalance in Healthcare—Challenges and Perspectives'. In: *Intelligent Data Analysis* 29.3 (2025), pp. 699–719. DOI: 10.1177/1088467X241305509.

[54] P. Schulam, F. Wigley and S. Saria. 'Clustering Longitudinal Clinical Marker Trajectories from Electronic Health Data: Applications to Phenotyping and Endotype Discovery'. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. 2015. DOI: 10.1609/aaai.v29i1.9537.

[55] J. Qiu, Y. Hu, L. Li, A. M. Erzurumluoglu, I. Braenne, C. Whitehurst, J. Schmitz, J. Arora, B. A. Bartholdy, S. Gandhi, P. Khoueiry, S. Mueller, B. Noyvert, Z. Ding, J. N. Jensen and J. de Jong. 'Deep Representation Learning for Clustering Longitudinal Survival Data from Electronic Health Records'. In: *Nature Communications* 16.1 (2025), p. 2534. DOI: 10.1038/s41467-025-56625-z.

[56] D. Gotz, J. Sun, N. Cao and S. Ebadollahi. 'Visual Cluster Analysis in Support of Clinical Decision Intelligence'. In: *AMIA Annual Symposium Proceedings* 2011 (2011). Epub 2011 Oct 22, pp. 481–490.

[57] A. Mariam, H. Javidi, E. C. Zabor, R. Zhao, T. Radivoyevitch and D. M. Rotroff. 'Unsupervised Clustering of Longitudinal Clinical Measurements in Electronic Health Records'. In: *PLOS Digital Health* 3.10 (2024), e0000628. DOI: 10.1371/journal.pdig.0000628.

[58] R. B. Cattell. 'The Description of Personality: Basic Traits Resolved into Clusters.' In: *The Journal of Abnormal and Social Psychology* 38 (4 Oct. 1943), pp. 476–506. DOI: 10.1037/h0054116.

[59] J. Singh and D. Singh. 'A comprehensive review of clustering techniques in artificial intelligence for knowledge discovery: Taxonomy, challenges, applications and future prospects'. In: *Advanced Engineering Informatics* 62 (2024), p. 102799. DOI: https://doi.org/10.1016/j.aei.2024.102799.

[60] D. Pfitzner, R. Leibbrandt and D. Powers. 'Characterization and Evaluation of Similarity Measures for Pairs of Clusterings'. In: *Knowledge and Information Systems* 19 (June 2009), pp. 361–394. DOI: 10.1007/s10115-008-0150-6.

[61] T. Liu, H. Yu and R. H. Blair. 'Stability estimation for unsupervised clustering: A review'. In: *WIREs Computational Statistics* 14.6 (2022), e1575. DOI: https://doi.org/10.1002/wics.1575. eprint: https://wires.onlinelibrary.wiley.com/doi/pdf/10.1002/wics.1575.

[62] R. Feldman and J. Sanger. 'Clustering'. In: *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*. New York, NY, USA: Cambridge University Press, 2007, pp. 82–92.

[63] C. X. Gao, D. Dwyer, Y. Zhu, C. L. Smith, L. Du, K. M. Filia, J. Bayer, J. M. Menssink, T. Wang, C. Bergmeir, S. Wood and S. M. Cotton. 'An overview of clustering methods with guidelines for application in mental health research'. In: *Psychiatry Research* 327 (2023), p. 115265. DOI: https://doi.org/10.1016/j.psychres.2023.115265.

[64] A. A. Wani. 'Comprehensive analysis of clustering algorithms: exploring limitations and innovative solutions'. In: *PeerJ Computer Science* 10 (2024), e2286. DOI: 10.7717/peerj-cs.2286.

[65] L. Rokach and O. Maimon. 'Clustering Methods'. In: Springer, 2005, pp. 321–352. DOI: 10.1007/0-387-25465-X_15.

[66] S. Miyamoto, H. Ichihashi and K. Honda. *Algorithms for Fuzzy Clustering*. Vol. 10. Springer Berlin, Heidelberg, 2008, pp. 1–247. DOI: 10.1007/978-3-540-78737-2.

[67] S. Wenzel. 'Asthma phenotypes: the evolution from clinical to molecular approaches'. In: *Nature Medicine* 18.5 (2012), pp. 716–725. DOI: 10.1038/nm.2678.

[68] X. Fan, Y. Yue, P. Sarkar and Y. X. R. Wang. 'On Hyperparameter Tuning in General Clustering Problems'. In: *Proceedings of the 37th International Conference on Machine Learning*. 2020, pp. 2996–3007.

[69] U. von Luxburg. 'Clustering Stability: An Overview'. In: *Foundations and Trends in Machine Learning* 2 (3 2010), pp. 235–274. DOI: 10.1561/2200000008.

[70] R. Tibshirani, G. Walther and T. Hastie. 'Estimating the number of clusters in a data set via the gap statistic'. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 63.2 (2001), pp. 411–423. DOI: 10.1111/1467-9868.00293.

[71] R. L. Thorndike. 'Who belongs in the family?' In: *Psychometrika* 18 (4 Dec. 1953), pp. 267–276. DOI: 10.1007/BF02289263.

[72] A. E. Ezugwu, A. K. Shukla, M. B. Agbaje, O. N. Oyelade, A. José-García and J. O. Agushaka. 'Automatic Clustering Algorithms: a Systematic Review and Bibliometric Analysis of Relevant Literature'. In: *Neural Computing and Applications* 33 (Oct. 2021), pp. 6247–6306. DOI: 10.1007/s00521-020-05395-4.

[73] J. C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Advanced Applications in Pattern Recognition. New York, NY: Springer, 1981, p. 272. DOI: 10.1007/978-1-4757-0450-1.

[74] J. C. Bezdek, R. Ehrlich and W. Full. 'FCM: The fuzzy c-means clustering algorithm'. In: *Computers & Geosciences* 10.2 (1984), pp. 191–203. DOI: 10.1016/0098-300484-90020-7.

[75] G. J. McLachlan and D. Peel. *Finite Mixture Models*. John Wiley & Sons, Ltd., 2000.

[76] J. C. Dunn. 'A Fuzzy Relative of the ISODATA Process and its Use in Detecting Compact Well-Separated Clusters'. In: *Journal of Cybernetics* 3 (3 Sept. 1973), pp. 32–57. DOI: 10.1080/01969727308546046.

[77] F. Murtagh and P. Contreras. 'Algorithms for Hierarchical Clustering: An Overview'. In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 2 (1 2012), pp. 86–97. DOI: 10.1002/widm.53.

[78] R. Vidal. 'Subspace Clustering'. In: *IEEE Signal Processing Magazine* 28 (2 Feb. 2011), pp. 52–68. DOI: 10.1109/MSP.2010.939739.

[79] A. S. Shirkhorshidi, S. Aghabozorgi and T. Y. Wah. 'A Comparison Study on Similarity and Dissimilarity Measures in Clustering Continuous Data'. In: *PLOS ONE* 10 (12 Dec. 2015), pp. 1–20. DOI: 10.1371/journal.pone.0144059.

[80] A. Saxena, M. Prasad, A. Gupta, N. Bharill, O. P. Patel, A. Tiwari, E. M. Joo, D. Weiping and L. Chin-Teng. 'A Review of Clustering Techniques and Developments'. In: *Neurocomputing* 267 (Aug. 2017), pp. 664–681. DOI: 10.1016/j.neucom.2017.06.053.

[81] J. MacQueen. 'Some Methods for Classification and Analysis of Multivariate Observations'. In: *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*. University of California Press, 1967, pp. 281–297.

[82] V. Kumar, J. K. Chhabra and D. Kumar. 'Performance Evaluation of Distance Metrics in the Clustering Algorithms'. In: *INFOCOMP Journal of Computer Science* 13 (1 Sept. 2014), pp. 38–52.

[83] T. Strauss and M. J. V. Maltitz. 'Generalising Ward's method for Use with Manhattan Distances'. In: *PLOS ONE* 12 (1 Jan. 2017), pp. 1–21. DOI: 10.1371/journal.pone.0168288.

[84] S. Surono and R. D. A. Putri. 'Optimization of Fuzzy C-Means Clustering Algorithm with Combination of Minkowski and Chebyshev Distance Using Principal Component Analysis'. In: *International Journal of Fuzzy Systems* 23 (Feb. 2021), pp. 139–144. DOI: 10.1007/s40815-020-00997-5.

[85] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. New York: McGraw-Hill, 1983.

[86] A. Huang. 'Similarity Measures for Text Document Clustering'. In: *Proceedings of the New Zealand Computer Science Research Student Conference*. 2008, pp. 9–56.

[87] P. Jaccard. 'The Distribution of the Flora in the Alpine Zone. 1'. In: *New Phytologist* 11 (2 1912), pp. 37–50. DOI: 10.1111/j.1469-8137.1912.tb05611.x.

[88] N. C. Chung, B. Miasojedow, M. Startek and A. Gambin. 'Jaccard/Tanimoto Similarity Test and Estimation Methods for Biological Presence-Absence Data'. In: *BMC Bioinformatics* 20 (Suppl. 15 Dec. 2019). DOI: 10.1186/s12859-019-3118-5.

[89] P. C. Mahalanobis. 'On the generalized distance in statistics'. In: *Proceedings of the National Institute of Sciences of India* 2.1 (1936), pp. 49–55.

[90] S. Xiang, F. Nie and C. Zhang. 'Learning a Mahalanobis Distance Metric for Data Clustering and Classification'. In: *Pattern Recognition* 41 (12 May 2008), pp. 3600–3612. DOI: 10.1016/j.patcog.2008.05.018.

[91] H. Sakoe and S. Chiba. 'Dynamic Programming Algorithm Optimization for Spoken Word Recognition'. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 26 (1 Feb. 1978), pp. 43–49. DOI: 10.1109/TASSP.1978.1163055.

[92] F. Petitjean, A. Ketterlin and P. Gançarski. 'A Global Averaging Method for Dynamic Time Warping, with Applications to Clustering'. In: *Pattern Recognition* 44 (3 2011), pp. 678–693. DOI: 10.1016/j.patcog.2010.09.013.

[93] T. Caliński and J. Harabasz. 'A Dendrite Method For Cluster Analysis'. In: *Communications in Statistics* 3 (1 1974), pp. 1–27. DOI: 10.1080/03610927408827101.

[94] D. Schlee. 'Reviews'. In: *Systematic Zoology* 24.2 (1975), pp. 263–268.

[95] R. Xu and D. W. II. 'Survey of clustering algorithms'. In: *IEEE Transactions on Neural Networks* 16.3 (May 2005), pp. 645–678. DOI: 10.1109/TNN.2005.845141.

[96] R. Souvenir and R. Pless. 'Manifold Clustering'. In: *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*. 2005, pp. 648–653. DOI: 10.1109/ICCV.2005.149.

[97] B. Nguyen and B. D. Baets. 'Kernel-Based Distance Metric Learning for Supervised-Means Clustering'. In: *IEEE Transactions on Neural Networks and Learning Systems* 30 (10 Jan. 2019), pp. 3084–3095. DOI: 10.1109/TNNLS.2018.2890021.

[98] Z. Kang, C. Peng and Q. Cheng. 'Kernel-driven similarity learning'. In: *Neurocomputing* 267 (Dec. 2017), pp. 210–219. DOI: 10.1016/j.neucom.2017.06.005.

[99] E. Min, X. Guo, Q. Liu, G. Zhang, J. Cui and J. Long. 'A Survey of Clustering with Deep Learning: From the Perspective of Network Architecture'. In: *IEEE Access* 6 (July 2018), pp. 39501–39514. DOI: 10.1109/ACCESS.2018.2855437.

[100] B. Yang, X. Fu, N. D. Sidiropoulos and M. Hong. 'Towards K-Means-Friendly Spaces: Simultaneous Deep Learning and Clustering'. In: *Proceedings of the 34th International Conference on Machine Learning*. 2017, pp. 3861–3870.

[101] D. P. Kingma and M. Welling. 'Auto-Encoding Variational Bayes'. In: *Proceedings of the 2nd International Conference on Learning Representations (ICLR)*. 2014.

[102] F. Ros, R. Riad and S. Guillaume. 'Deep Clustering Framework Review using Multicriteria Evaluation'. In: *Knowledge-Based Systems* 285 (Feb. 2024), p. 111315. DOI: 10.1016/j.knosys.2023.111315.

[103] D. Aloise, A. Deshpande, P. Hansen and P. Popat. 'NP-Hardness of Euclidean Sum-of-Squares Clustering'. In: *Machine Learning* 75 (May 2009), pp. 245–248. DOI: 10.1007/s10994-009-5103-0.

[104] S. P. Lloyd. 'Least Squares Quantization in PCM'. In: *IEEE Transactions on Information Theory* 28 (2 Mar. 1982), pp. 129–137. DOI: 10.1109/TIT.1982.1056489.

[105] J. A. Hartigan and M. A. Wong. 'Algorithm AS 136: A K-Means Clustering Algorithm'. In: *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 28 (1 1979), pp. 100–108.

[106] S. Har-Peled and B. Sadri. 'How Fast Is the K-Means Method?' In: *Algorithmica* 41 (Mar. 2005), pp. 185–202. DOI: 10.1007/s00453-004-1127-9.

[107] M. Jianliang, S. Haikun and B. Ling. 'The Application on Intrusion Detection Based on K-Means Cluster Algorithm'. In: *2009 International Forum on Information Technology and Applications*. Vol. 1. IEEE, 2009, pp. 150–152. DOI: 10.1109/IFITA.2009.34.

[108] Y. Lu, S. Lu, F. Fotouhi, Y. Deng and S. J. Brown. 'Incremental Genetic K-means Algorithm and its Application in Gene Expression Data Analysis'. In: *BMC Bioinformatics* 5 (172 Oct. 2004), pp. 1–10. DOI: 10.1186/1471-2105-5-172.

[109] P. Govender and V. Sivakumar. 'Application of K-Means and Hierarchical Clustering Techniques for Analysis of Air Pollution: A Review (1980–2019)'. In: *Atmospheric Pollution Research* 11 (1 Jan. 2020), pp. 40–56. DOI: 10.1016/j.apr.2019.09.009.

[110] M. Ahmed, R. Seraj and S. M. S. Islam. 'The K-Means Algorithm: A Comprehensive Survey and Performance Evaluation'. In: *Electronics* 9 (8 Aug. 2020). DOI: 10.3390/electronics9081295.

[111] P. Bradley, O. Mangasarian and W. Street. 'Clustering via Concave Minimization'. In: *Advances in Neural Information Processing Systems 9 (NIPS 1996)*. 1996.

[112] L. Kaufman and P. J. Rousseeuw. 'Partitioning Around Medoids (Program PAM)'. In: *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley & Sons, 1990, pp. 68–125. DOI: 10.1002/9780470316801.ch2.

[113] D. Arthur and S. Vassilvitskii. 'K-Means++: The Advantages of Careful Seeding'. In: *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*. Society for Industrial and Applied Mathematics, 2007, pp. 1027–1035.

[114] A. P. Dempster, N. M. Laird and D. B. Rubin. 'Maximum Likelihood from Incomplete Data via the EM Algorithm'. In: *Journal of the Royal Statistical Society: Series B (Methodological)* 39 (1 1977), pp. 1–22. DOI: 10.1111/j.2517-6161.1977.tb01600.x.

[115] D. A. Reynolds. 'Gaussian Mixture Models'. In: *Encyclopedia of Biometrics* 741 (2009), pp. 659–663.

[116] Y. Zhang, M. Li, S. Wang, S. Dai, L. Luo, E. Zhu, H. Xu, X. Zhu, C. Yao and H. Zhou. 'Gaussian Mixture Model Clustering with Incomplete Data'. In: *ACM Transactions on Multimedia Computing, Communications and Applications (TOMM)* 17 (1s Apr. 2021), pp. 1–14. DOI: 10.1145/3408318.

[117] K. Li, Z. Ma, D. Robinson and J. Ma. 'Identification of Typical Building Daily Electricity Usage Profiles using Gaussian Mixture Model-based Clustering and Hierarchical Clustering'. In: *Applied Energy* 231 (Dec. 2018), pp. 331–342. DOI: 10.1016/j.apenergy.2018.09.050.

[118] P. D. McNicholas and T. B. Murphy. 'Model-based Clustering of Microarray Expression Data via Latent Gaussian Mixture Models'. In: *Bioinformatics* 26 (21 Nov. 2010), pp. 2705–2712. DOI: 10.1093/bioinformatics/btq498.

[119] L. A. García-Escudero, A. Gordaliza, F. Greselin, S. Ingrassia and A. Mayo-Iscar. 'Eigenvalues and Constraints in Mixture Modeling: Geometric and Computational Issues'. In: *Advances in Data Analysis and Classification* 12 (2 June 2018), pp. 203–233. DOI: 10.1007/s11634-017-0293-y.

[120] H. Zhou, W. Pan and X. Shen. 'Penalized Model-based Clustering with Unconstrained Covariance Matrices'. In: *Electronic Journal of Statistics* 3 (Jan. 2009), pp. 1473–1496. DOI: 10.1214/09-EJS487.

[121] J. Baek and G. J. McLachlan. 'Mixtures of Common t-Factor Analyzers for Clustering High-Dimensional Microarray Data'. In: *Bioinformatics* 27 (9 Mar. 2011), pp. 1269–1276. DOI: 10.1093/bioinformatics/btr112.

[122] B. C. Franczak, R. P. Browne and P. D. McNicholas. 'Mixtures of Shifted Asymmetric Laplace Distributions'. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36 (6 June 2014), pp. 1149–1157. DOI: 10.1109/TPAMI.2013.216.

[123] D. Qiu. 'A Comparative Study of the K-Means Algorithm and the Normal Mixture Model for Clustering: Bivariate Homoscedastic Case'. In: *Journal of Statistical Planning and Inference* 140 (7 July 2010), pp. 1701–1711. DOI: 10.1016/j.jspi.2009.12.025.

[124] E. Patel and D. S. Kushwaha. 'Clustering Cloud Workloads: K-Means vs Gaussian Mixture Model'. In: *Procedia Computer Science* 171 (June 2020), pp. 158–167. DOI: 10.1016/j.procs.2020.04.017.

[125] D. Steinley and M. J. Brusco. 'Evaluating Mixture Modeling for Clustering: Recommendations and Cautions'. In: *Psychological Methods* 16 (1 Mar. 2011), pp. 63–79. DOI: 10.1037/a0022673.

[126] A. M. Ikotun, A. E. Ezugwu, L. Abualigah, B. Abuhaija and J. Heming. 'K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data'. In: *Information Sciences* 622 (2023), pp. 178–210. DOI: https://doi.org/10.1016/j.ins.2022.11.139.

[127] Y. Sheng, R. Bond, R. Jaiswal, J. Dinsmore and J. Doyle. 'Augmenting K-Means Clustering With Qualitative Data to Discover the Engagement Patterns of Older Adults With Multimorbidity When Using Digital Health Technologies: Proof-of-Concept Trial'. In: *J Med Internet Res* 26 (Mar. 2024), e46287. DOI: 10.2196/46287.

[128] S. R. Kasa, S. Bhattacharya and V. Rajan. 'Gaussian mixture copulas for high-dimensional clustering and dependency-based subtyping'. In: *Bioinformatics* 36.2 (Aug. 2019), pp. 621–628. DOI: 10.1093/bioinformatics/btz599. eprint: https://academic.oup.com/bioinformatics/article-pdf/36/2/621/48990951/btz599.pdf.

[129] E. Keogh and S. Kasetty. 'On the Need for Time Series Data Mining Benchmarks: A Survey and Empirical Demonstration'. In: *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Association for Computing Machinery, July 2013, pp. 102–111. DOI: 10.1145/775047.775062.

[130] P. Esling and C. Agon. 'Time-series Data Mining'. In: *ACM Computing Surveys* 45 (1 Dec. 2012), pp. 1–34. DOI: 10.1145/2379776.2379788.

[131] T. W. Liao. 'Clustering of Time Series Data - A Survey'. In: *Pattern Recognition* 38 (11 Nov. 2005), pp. 1857–1874. DOI: 10.1016/j.patcog.2005.01.025.

[132] B. F. Crabtree, S. C. Ray, P. M. Schmidt, P. J. O'Connor and D. D. Schmidt. 'The Individual Over Time: Time Series Applications in Healthcare Research'. In: *Journal of Clinical Epidemiology* 43 (3 1990), pp. 241–260. DOI: 10.1016/0895-4356(90)90005-A.

[133] H. Song, D. Rajan, J. J. Thiagarajan and A. Spanias. 'Attend and Diagnose: Clinical Time Series Analysis Using Attention Models'. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 32 (1 Apr. 2018). DOI: 10.1609/aaai.v32i1.11635.

[134] M. Ambigavathi and D. Sridharan. 'Analysis of Clustering Algorithms in Machine Learning for Healthcare Data'. In: *Advances in Computing and Data Sciences, ICACDS 2020.* Springer, Singapore., 2020, pp. 117–128. DOI: 10.1007/978-981-15-6634-9_12.

[135] E. Keogh and J. Lin. 'Clustering of Time-Series Subsequences is Meaningless: Implications for Previous and Future Research'. In: *Knowledge and Information Systems* 8 (Aug. 2005), pp. 154–177. DOI: 10.1007/s10115-004-0172-7.

[136] C. Cassisi, P. Montalto, M. Aliotta, A. Cannata and A. Pulvirenti. 'Similarity Measures and Dimensionality Reduction Techniques for Time Series Data Mining'. In: *Advances in Data Mining Knowledge Discovery and Applications*. InTech, 2012, pp. 71–96. DOI: 10.5772/49941.

[137] A. Kianimajd, M. G. Ruano, P. Carvalho, J. Henriques, T. Rocha, S. Paredes and A. E. Ruano. 'Comparison of Different Methods of Measuring Similarity in Physiologic Time Series'. In: *IFAC-PapersOnLine* 50 (1 July 2017), pp. 11005–11010. DOI: 10.1016/j.ifacol.2017.08.2479.

[138] D. Kotsakos, G. Trajcevski, D. Gunopulos and C. C. Aggarwal. 'Time-Series Data Clustering'. In: *Data Clustering: Algorithms and Applications*. Ed. by C. C. Aggarwal and C. K. Reddy. Chapman and Hall/CRC, 2014, pp. 355–374.

[139] M. Ji, F. Xie and Y. Ping. 'A Dynamic Fuzzy Cluster Algorithm for Time Series'. In: *Abstract and Applied Analysis* 2013 (1 2013), p. 183410. DOI: 10.1155/2013/183410.

[140] K. Kalpakis, D. Gada and V. Puttagunta. 'Distance Measures for Effective Clustering of ARIMA Time-Series'. In: *Proceedings 2001 IEEE International Conference on Data Mining*. IEEE Computer Society Press, 2001, pp. 273–280. DOI: 10.1109/ICDM.2001.989529.

[141] J. Batóg and B. Batóg. 'Synchronization of Business Cycles in the EU: Time Series Clustering'. In: *WSEAS Transactions on Business and Economics* 16 (2019), pp. 298–305.

[142] V. Niennattrakul and C. A. Ratanamahatana. 'Clustering Multimedia Data Using Time Series'. In: *20006 International Conference on Hybrid Information Technology*. IEEE, Nov. 2006. DOI: 10.1109/ICHIT.2006.253514.

[143] F. Gullo, G. Ponti, A. Tagarelli, G. Tradigo and P. Veltri. 'A Time Series Approach for Clustering Mass Spectrometry Data'. In: *Journal of Computational Science* 3 (5 Sept. 2012), pp. 344–355. DOI: 10.1016/j.jocs.2011.06.008.

[144] M. Guéguin, E. Roux, A. Hernández, F. Porée, P. Mabo, L. Graindorge and G. Carrault. 'Clustering follow-up time-series recorded by cardiac implantable devices'. In: *Proceedings of the 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS)*. 2007, pp. 3564–3567. DOI: 10.1109/IEMBS.2007.4353172.

[145] M. Herrmann, C. W. Tan and G. I. Webb. 'Parameterizing the Cost Function of Dynamic Time Warping with Application to Time Series Classification'. In: *Data Mining and Knowledge Discovery* 37 (Apr. 2023), pp. 2024–2045. DOI: 10.1007/s10618-023-00926-8.

[146] O. Gold and M. Sharir. 'Dynamic Time Warping and Geometric Edit Distance: Breaking the Quadratic Barrier'. In: *ACM Transactions on Algorithms* 14 (4 Aug. 2018), pp. 1–17. DOI: 10.1145/3230734.

[147] K. Bringmann and M. Kunnemann. 'Quadratic Conditional Lower Bounds for String Problems and Dynamic Time Warping'. In: *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*. Vol. 2015-December. IEEE Computer Society, 2015, pp. 79–97. DOI: 10.1109/FOCS.2015.15.

[148] D. F. Silva and G. E. A. P. A. Batista. 'Speeding Up All-Pairwise Dynamic Time Warping Matrix Calculation'. In: *Proceedings of the 2016 SIAM International Conference on Data Mining*. 2016, pp. 837–845. DOI: 10.1137/1.9781611974348.94.

[149] S. Salvador and P. Chan. 'Toward Accurate Dynamic Time Warping in Linear Time and Space'. In: *Intelligent Data Analysis* 11 (5 Oct. 2007), pp. 561–580. DOI: 10.3233/IDA-2007-11508.

[150] M. Cuturi and M. Blondel. 'Soft-DTW: a Differentiable Loss Function for Time-Series'. In: *Proceedings of the 34th International Conference on Machine Learning*. Ed. by D. Precup and Y. W. Teh. PMLR, Aug. 2017, pp. 894–903.

[151] R. Tavenard, J. Faouzi, G. Vandewiele, F. Divo, G. Androz, C. Holtz, M. Payne, R. Yurchak and M. Rußwurm. 'Tslearn, A Machine Learning Toolkit for Time Series Data'. In: *Journal of Machine Learning Research* 21 (118 Apr. 2020), pp. 1–6.

[152] M. Vlachos, G. Kollios and D. Gunopulos. 'Discovering Similar Multidimensional Trajectories'. In: *Proceedings 18th International Conference on Data Engineering*. IEEE, Feb. 2002, pp. 673–684. DOI: 10.1109/ICDE.2002.994784.

[153] S. R. Aghabozorgi, T. Y. Wah, A. Amini and M. R. Saybani. 'A New Approach to Present Prototypes in Clustering of Time Series'. In: *Proceedings of the 2011 International Conference on Data Mining*. 2011.

[154] S. Aghabozorgi and T. Y. Wah. 'Effective Clustering of Time-Series Data Using FCM'. In: *International Journal of Machine Learning and Computing* 4 (2 2014), pp. 170–176. DOI: 10.7763/IJMLC.2014.V4.407.

[155] L. Bonomi, L. Fan and X. Jiang. 'Noise-tolerant Similarity Search in Temporal Medical Data'. In: *Journal of Biomedical Informatics* 113 (2020), p. 103625. DOI: 10.1016/j.jbi.2020.103625.

[156] L. Chen and R. Ng. 'On The Marriage of Lp-norms and Edit Distance'. In: *Proceedings of the Thirtieth International Conference on Very Large Data Bases - Volume 30*. VLDB Endowment, 2004, pp. 792–803.

[157] C. S. Möller-Levet, F. Klawonn, K.-H. Cho and O. Wolkenhauer. 'Fuzzy Clustering of Short Time-Series and Unevenly Distributed Sampling Points'. In: *Advances in Intelligent Data Analysis V*. Springer Berlin Heidelberg, 2003, pp. 330–340. DOI: 10.1007/978-3-540-45231-7_31.

[158] C. A. Ratanamahatana and E. Keogh. 'Three Myths about Dynamic Time Warping Data Mining'. In: *Proceedings of the 2005 SIAM International Conference on Data Mining*. 2005, pp. 506–510. DOI: 10.1137/1.9781611972757.50.

[159] F. Iglesias and W. Kastner. 'Analysis of Similarity Measures in Times Series Clustering for the Discovery of Building Energy Patterns'. In: *Energies* 6 (2 2013), pp. 579–597. DOI: 10.3390/en6020579.

[160] M. Kljun, M. Teršek and E. E. Erikštrumbelj. 'A Review and Comparison of Time Series Similarity Measures'. In: *29th International Electrotechnical and Computer Science Conference (ERK 2020)*. 2020, pp. 367–370.

[161] T. Górecki and P. Piasecki. 'A Comprehensive Comparison of Distance Measures for Time Series Classification'. In: *Stochastic Models, Statistics and Their Applications*. Vol. 294. Springer International Publishing, 2019, pp. 409–428. DOI: 10.1007/978-3-030-28665-1_31.

[162] S. Yoshida and B. Chakraborty. 'A Comparative Study of Similarity Measures for Time Series Classification'. In: *New Frontiers in Artificial Intelligence*. Ed. by M. Otake, S. Kurahashi, Y. Ota, K. Satoh and D. Bekki. Vol. 10091. Springer, Cham, 2017, pp. 397–408. DOI: 10.1007/978-3-319-50953-2.

[163] A. Nanopoulos, R. Alcock and Y. Manolopoulos. 'Feature-based Classification of Time-Series Data'. In: *Information Processing and Technology* (2001), pp. 49–61.

[164] F. Mörchen. *Time Series Feature Extraction for Data Mining using DWT and DFT*. 2003.

[165] X. Wang, K. Smith and R. Hyndman. 'Characteristic-Based Clustering for Time Series Data'. In: *Data Mining and Knowledge Discovery* 13 (Sept. 2006), pp. 335–364. DOI: 10.1007/s10618-005-0039-x.

[166] Q. Ma, J. Zheng, S. Li and G. W. Cottrell. 'Learning Representations for Time Series Clustering'. In: *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2019.

[167] S. J. Wilson. 'Data Representation for Time Series Data Mining: Time Domain Approaches'. In: *WIREs Computational Statistics* 9 (1 Jan. 2017), e1392. DOI: 10.1002/wics.1392.

[168] Y. Morinaka, M. Yoshikawa, T. Amagasa and S. Uemura. 'The L-index: An Indexing Structure for Efficient Subsequence Matching in Time Sequence Databases'. In: *Proceedings of the 5th PacificAsia Conference on Knowledge Discovery and Data Mining*. Citeseer, 2001, pp. 51–60.

[169] E. Keogh, K. Chakrabarti, M. Pazzani and S. Mehrotra. 'Locally Adaptive Dimensionality Reduction for Indexing Large Time Series Databases'. In: *Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data*. Association for Computing Machinery, 2001, pp. 151–162. DOI: 10.1145/375663.375680.

[170] F. Korn, H. V. Jagadish and C. Faloutsos. 'Efficiently Supporting Ad Hoc Queries in Large Datasets of Time Sequences'. In: *ACM SIGMOD Record* 26 (2 1997), pp. 289–300. DOI: 10.1145/253260.253332.

[171] R. H. Lesch, Y. Caille and D. Lowe. 'Component Analysis in Financial Time Series'. In: *Proceedings of the IEEE/IAFE 1999 Conference on Computational Intelligence for Financial Engineering (CIFEr) (IEEE Cat. No.99TH8408)*. IEEE, 199, pp. 183–190. DOI: 10.1109/CIFER.1999.771118.

[172] J. Lin, E. Keogh, L. Wei and S. Lonardi. 'Experiencing SAX: A Novel Symbolic Representation of Time Series'. In: *Data Mining and Knowledge Discovery* 15 (2 Apr. 2007), pp. 107–144. DOI: 10.1007/s10618-007-0064-z.

[173] F. Morchen and A. Ultsch. 'Optimizing Time Series Discretization for Knowledge Discovery'. In: *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*. Association for Computing Machinery, 2005, pp. 660–665. DOI: 10.1145/1081870.1081953.

[174] R. Agrawal, C. Faloutsos and A. Swami. 'Efficient Similarity Search In Sequence Databases'. In: *Foundations of Data Organization and Algorithms*. Springer Berlin Heidelberg, 1993, pp. 69–84. DOI: 10.1007/3-540-57301-1_5.

[175] G. J. Janacek, A. J. Bagnall and M. Powell. 'A Likelihood Ratio Distance Measure for the Similarity Between the Fourier Transform of Time Series'. In: *Advances in Knowledge Discovery and Data Mining*. Springer Berlin Heidelberg, 2005, pp. 737–743. DOI: 10.1007/11430919_85.

[176] H. Zhang, T. B. Ho, Y. Zhang and M.-S. Lin. 'Unsupervised Feature Extraction for Time Series Clustering Using Orthogonal Wavelet Transform'. In: *Informatica* 30 (3 2006), pp. 305–319.

[177] E. A. Maharaj, P. D'Urso and D. U. A. Galagedera. 'Wavelet-based Fuzzy Clustering of Time Series'. In: *Journal of Classification* 27 (2010), pp. 231–275. DOI: 10.1007/s00357-010-9058-4.

[178] K.-P. Chan and A. W.-C. Fu. 'Efficient Time Series Matching by Wavelets'. In: *Proceedings of the 15th International Conference on Data Engineering (Cat. No. 99CB36337)*. IEEE, 1999, pp. 126–133. DOI: 10.1109/ICDE.1999.754915.

[179] X. Wang, A. Mueen, H. Ding, G. Trajcevski, P. Scheuermann and E. Keogh. 'Experimental Comparison of Representation Methods and Distance Measures for Time Series Data'. In: *Data Mining and Knowledge Discovery* 26 (2 Mar. 2013), pp. 275–309. DOI: 10.1007/s10618-012-0250-5.

[180] H. Ding, G. Trajcevski, P. Scheuermann, X. Wang and E. Keogh. 'Querying and Mining of Time Series Data: Experimental Comparison of Representations and Distance Measures'. In: *Proceedings of the VLDB Endowment*. VLDB Endowment, Aug. 2008, pp. 1542–1552. DOI: 10.14778/1454159.1454226.

[181] G. S. Chadha, I. Islam, A. Schwung and S. X. Ding. 'Deep Convolutional Clustering-Based Time Series Anomaly Detection'. In: *Sensors* 21 (16 2021). DOI: 10.3390/s21165488.

[182] A. Alqahtani, M. Ali, X. Xie and M. W. Jones. 'Deep Time-Series Clustering, A Review'. In: *Electronics* 10 (3001 Dec. 2021), pp. 1–29. DOI: 10.3390/electronics10233001.

[183] G. E. P. Box, G. M. Jenkins, G. C. Reinsel and G. M. Ljung. *Time Series Analysis: Forecasting and Control*. 5th ed. Hoboken, NJ: Wiley, 2015.

[184] L. R. Rabiner. 'A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition'. In: *Proceedings of the IEEE* 77.2 (1989), pp. 257–286.

[185] I. E. Leaning and colleagues. 'Uncovering Social States in Healthy and Clinical Populations Using Digital Phenotyping Time Series: A Hidden Markov Model Approach'. In: *Journal of Medical Internet Research* 27 (2025), e64007. DOI: 10.2196/64007.

[186] T. Mitsa. 'Temporal Data Similarity Computation, Representation, and Summarization'. In: 1st ed. Chapman and Hall/CRC, 2010, pp. 21–67.

[187] C. Fraley and A. E. Raftery. 'Model-based Clustering, Discriminant analysis, and Density Estimation'. In: *Journal of the American Statistical Association* 97 (458 2002), pp. 611–631. DOI: 10.1198/016214502760047131.

[188] J. Abonyi, B. Feil, S. Nemeth and P. Arva. 'Principal Component Analysis Based Time Series Segmentation'. In: *IEEE International Conference on Computational Cybernetics*. 2005.

[189] C. Ratanamahatana, E. Keogh, A. J. Bagnall and S. Lonardi. 'A Novel Bit Level Time Series Representation with Implication of Similarity Search and Clustering'. In: *Advances in Knowledge Discovery and Data Mining. PAKDD 2005*. Springer Berlin, Heidelberg, 2005, pp. 771–777. DOI: 10.1007/11430919_90.

[190] W. Liu and L. Shao. 'Research of SAX in Distance Measuring for Financial Time Series Data'. In: *2009 First International Conference on Information Science and Engineering*. IEEE, 2009, pp. 935–937. DOI: 10.1109/ICISE.2009.924.

[191] D. E. Rumelhart, G. E. Hinton and R. J. Williams. *Learning Internal Representations by Error Propagation*. Tech. rep. ICS 8504. San Diego, CA: Institute for Cognitive Science, University of California, Sept. 1985.

[192] A. Sherstinsky. 'Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network'. In: *Physica D: Nonlinear Phenomena* 404 (2020), p. 132306. DOI: 10.1016/j.physd.2019.132306.

[193] D. J. Trosten, A. S. Strauman, M. Kampffmeyer and R. Jenssen. 'Recurrent Deep Divergence-based Clustering for Simultaneous Feature Learning and Clustering of Variable Length Time Series'. In: *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2019, pp. 3257–3261. DOI: 10.1109/ICASSP.2019.8682365.

[194] N. Tavakoli, S. Siami-Namini, M. Adl Khanghah and A. S. Namin. 'An Autoencoder-based Deep Learning Approach for Clustering Time Series Data'. In: *SN Applied Sciences* 2.6 (2020), p. 937. DOI: 10.1007/s42452-020-2584-8.

[195] R. Pascanu, T. Mikolov and Y. Bengio. 'On the Difficulty of Training Recurrent Neural Networks'. In: *Proceedings of the 30th International Conference on Machine Learning*. Vol. 28. Proceedings of Machine Learning Research 3. PMLR, 2013, pp. 1310–1318.

[196] K. Cho, B. van Merriënboer, D. Bahdanau, F. Bougares, H. Schwenk and Y. Bengio. 'Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation'. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2014.

[197] J. C. Heck and F. M. Salem. 'Simplified Minimal Gated Unit Variations for Recurrent Neural Networks'. In: *2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)*. IEEE, Aug. 2017, pp. 1593–1596. DOI: 10.1109/MWSCAS.2017.8053242.

[198] S. Hochreiter and J. Schmidhuber. 'Long Short-Term Memory'. In: *Neural Computation* 9 (8 Nov. 1997), pp. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735.

[199] F. A. Gers, J. Schmidhuber and F. Cummins. 'Learning to Forget: Continual Prediction with LSTM'. In: *Neural Computation* 12 (10 Oct. 2000), pp. 2451–2471. DOI: 10.1162/089976600300015015.

[200] E. Doğan. 'LSTM Training Set Analysis and Clustering Model Development for Short-Term Traffic Flow Prediction'. In: *Neural Computing and Applications* 33 (Jan. 2021), pp. 11175–11188. DOI: 10.1007/s00521-020-05564-5.

[201] N. S. Madiraju. *Deep Temporal Clustering: Fully Unsupervised Learning of Time-Domain Features*. Tech. rep. Arizona State University, 2018.

[202] Y. Fan, J. Liu, H. Ye and Z. Lyu. 'TA-LSTM: A Time and Attribute Aware LSTM for Deep Flight Track Clustering'. In: *IEEE Transactions on Aerospace and Electronic Systems* 59 (5 June 2023), pp. 7047–7060. DOI: 10.1109/TAES.2023.3285203.

[203] M. Zaheer, A. Ahmed and A. J. Smola. 'Latent LSTM Allocation Joint Clustering and Non-Linear Dynamic Modeling of Sequential Data'. In: *Proceedings of the 34th International Conference on Machine Learning*. Ed. by D. Precup and Y. W. Teh. PMLR, Aug. 2017, pp. 3967–3976.

[204] Y. LeCun, L. Bottou, Y. Bengio and P. Haffner. 'Gradient-Based Learning Applied to Document Recognition'. In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324. DOI: 10.1109/5.726791.

[205] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj and D. J. Inman. '1D Convolutional Neural Networks and Applications: A Survey'. In: *Mechanical Systems and Signal Processing* 151 (2021), p. 107398. DOI: 10.1016/j.ymssp.2020.107398.

[206] S. Yi, J. Ju, M.-K. Yoon and J. Choi. 'Grouped convolutional neural networks for multivariate time series'. In: *arXiv preprint arXiv:1703.09938* (2017).

[207] J. Wu, Z. Zhang, R. Tong, Y. Zhou, Z. Hu and K. Liu. 'Imaging Feature-Based Clustering of Financial Time Series'. In: *PLOS ONE* 18.7 (2023), e0288836. DOI: 10.1371/journal.pone.0288836.

[208] B. Lafabregue, J. Weber, P. Gançarski and G. Forestier. 'End-to-End Deep Representation Learning for Time Series Clustering: A Comparative Study'. In: *Data Mining and Knowledge Discovery* 36 (2022), pp. 29–81. DOI: 10.1007/s10618-021-00796-y.

[209] M. Cai, Y. Zheng, Z. Peng, C. Huang and H. Jiang. 'Research on Load Clustering Algorithm Based on Variational Autoencoder and Hierarchical Clustering'. In: *PLOS ONE* 19.6 (2024), e0303977. DOI: 10.1371/journal.pone.0303977.

[210] T. Kohonen. 'The Self-organizing Map'. In: *Proceedings of the IEEE* 78 (9 Mar. 1990), pp. 1464–1480. DOI: 10.1109/5.58325.

[211] V. Fortuin, M. Hüser, F. Locatello, H. Strathmann and G. Rätsch. 'SOM-VAE: Interpretable Discrete Representation Learning on Time Series'. In: *Proceedings of the International Conference on Learning Representations 2019*. June 2019.

[212] L. Manduchi, M. Hüser, M. Faltys, J. Vogt, G. Rätsch and V. Fortuin. 'T-DPSOM – An Interpretable Clustering Method for Unsupervised Learning of Patient Health States'. In: *Proceedings of the ACM Conference on Health, Inference, and Learning (CHIL '21)*. New York, NY, USA: ACM, 2021, 10 pages. DOI: 10.1145/3450439.3451872.

[213] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser and I. Polosukhin. 'Attention Is All You Need'. In: *Proceedings of the 31st Conference on Neural Information Processing Systems*. Ed. by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan and R. Garnett. Curran Associates, Inc., 2017, pp. 5998–6008.

[214] G. Zerveas, S. Jayaraman, D. Patel, A. Bhamidipaty and C. Eickhoff. 'A Transformer-based Framework for Multivariate Time Series Representation Learning'. In: *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. Association for Computing Machinery, 2021, pp. 2114–2124. DOI: 10.1145/3447548.3467401.

[215] Q. Liu, Y. Fang, P. Jiang and G. Li. *DGCformer: Deep Graph Clustering Transformer for Multivariate Time Series Forecasting*. 2024. arXiv: 2405.08440 [cs.LG].

[216] S. Tipirneni and C. K. Reddy. 'Self-Supervised Transformer for Sparse and Irregularly Sampled Multivariate Clinical Time-Series'. In: *ACM Transactions on Knowledge Discovery from Data* 16.6 (2022), 105:1–105:17. DOI: 10.1145/3516367.

[217] Z. Yang, A. Mitra, W. Liu, D. Berlowitz and H. Yu. 'TransformEHR: Transformer-Based Encoder–Decoder Generative Model to Enhance Prediction of Disease Outcomes Using Electronic Health Records'. In: *Nature Communications* 14.1 (2023), p. 7857. DOI: 10.1038/s41467-023-43715-z.

[218] A. A. Toye, A. Celik and S. Kleinberg. 'Benchmarking Missing Data Imputation Methods for Time Series Using Real-World Test Cases'. In: *Proceedings of the Machine Learning for Health Conference*. Vol. 287. Proceedings of Machine Learning Research. June 2025, pp. 480–501.

[219] H. Ghaderi, B. Foreman, A. Nayebi, S. Tipirneni, C. K. Reddy and V. Subbian. 'A Self-Supervised Learning-Based Approach to Clustering Multivariate Time-Series Data with Missing Values (SLAC-Time): An Application to TBI Phenotyping'. In: *Journal of Biomedical Informatics* 143 (2023), p. 104401. DOI: 10.1016/j.jbi.2023.104401.

[220] Z. Che, S. Purushotham, K. Cho, D. Sontag and Y. Liu. 'Recurrent Neural Networks for Multivariate Time Series with Missing Values'. In: *Scientific Reports* 8 (2018), p. 6085. DOI: 10.1038/s41598-018-24271-9.

[221] E. D. Brouwer, J. Simm, A. Arany and Y. Moreau. 'GRU-ODE-Bayes: Continuous modeling of sporadically-observed time series'. In: *Neural Information Processing Systems*. 2019.

[222] S. Shan, Y. Li and J. B. Oliva. 'Nrtsi: Non-recurrent time series imputation'. In: *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2023, pp. 1–5.

[223] L. Deng. 'A tutorial survey of architectures, algorithms, and applications for deep learning'. In: *APSIPA Transactions on Signal and Information Processing* 3 (2014), e2. DOI: DOI:10.1017/atsip.2013.9.

[224] A. Khamparia and K. M. Singh. 'A systematic review on deep learning architectures and applications'. In: *Expert Systems* 36 (3 2019). e12400 EXSY-Jul-18-241.R3, e12400. DOI: https://doi.org/10.1111/exsy.12400.

[225] I. M. Baytas, C. Xiao, X. Zhang, F. Wang, A. K. Jain and J. Zhou. 'Patient Subtyping via Time-Aware LSTM Networks'. In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Association for Computing Machinery, 2017, pp. 65–74. DOI: 10.1145/3097983.3097997.

[226] D. Zhang, C. Yin, J. Zeng, X. Yuan and P. Zhang. 'Combining structured and unstructured data for predictive models: a deep learning approach'. In: *BMC Medical Informatics and Decision Making* 20 (1 2020), p. 280. DOI: 10.1186/s12911-020-01297-6.

[227] T. Pham, T. Tran, D. Phung and S. Venkatesh. 'DeepCare: A Deep Dynamic Memory Model for Predictive Medicine'. In: *Advances in Knowledge Discovery and Data Mining*. Ed. by J. Bailey, L. Khan, T. Washio, G. Dobbie, J. Z. Huang and R. Wang. Springer International Publishing, 2016, pp. 30–41.

[228] R. Miotto, F. Wang, S. Wang, X. Jiang and J. T. Dudley. 'Deep Learning for Healthcare: Review, Opportunities and Challenges'. In: *Briefings in Bioinformatics* 19.6 (2018), pp. 1236–1246. DOI: 10.1093/bib/bbx044.

[229] Y. Lu, H. I. Salmons, J. P. Mickley, N. A. Bedard, M. J. Taunton and C. C. Wyles. 'Defining Clinically Meaningful Subgroups for Risk Stratification in Patients Undergoing Revision Total Hip Arthroplasty: A Combined Unsupervised and Supervised Machine Learning Approach'. In: *The Journal of Arthroplasty* 38.10 (2023), pp. 2023–2031. DOI: 10.1016/j.arth.2023.06.020.

[230] P. E. Bretos-Azcona, E. Sánchez-Iriso and J. M. Cabasés Hita. 'Tailoring Integrated Care Services for High-Risk Patients with Multiple Chronic Conditions: A Risk Stratification Approach Using Cluster Analysis'. In: *BMC Health Services Research* 20.1 (2020), p. 806. DOI: 10.1186/s12913-020-05668-7.

[231] J.-J. Nunez, Y. S. Liu, B. Cao, B. N. Frey, K. Ho, R. Milev, D. J. Müller, S. Rotzinger, C. N. Soares, V. H. Taylor, R. Uher, S. H. Kennedy and R. W. Lam. 'Response Trajectories During Escitalopram Treatment of Patients with Major Depressive Disorder'. In: *Psychiatry Research* 326 (2023), p. 115298. DOI: 10.1016/j.psychres.2023.115298.

[232] D. Moggia, W. Lutz, E.-L. Brakemeier and L. Bickman. 'Treatment Personalization and Precision Mental Health Care: Where Are We and Where Do We Want to Go?' In: *Administration and Policy in Mental Health and Mental Health Services Research* 51.5 (2024), pp. 611–616. DOI: 10.1007/s10488-024-01407-w.

[233] H. Estiri, Z. H. Strasser and S. N. Murphy. 'High-Throughput Phenotyping with Temporal Sequences'. In: *Journal of the American Medical Informatics Association* 28.4 (2021), pp. 772–781. DOI: 10.1093/jamia/ocaa288.

[234] E. Werner, J. N. Clark, A. Hepburn, R. S. Bhamber, M. Ambler, C. P. Bourdeaux, C. J. McWilliams and R. Santos-Rodriguez. 'Explainable Hierarchical Clustering for Patient Subtyping and Risk Prediction'. In: *Experimental Biology and Medicine* 248.24 (2023), pp. 2547–2559. DOI: 10.1177/15353702231214253.

[235] Y. Liu, Z. Zhang, S. Qin and J. Bian. 'Multi-Task Deep Neural Networks for Irregularly Sampled Multivariate Clinical Time Series'. In: *Proceedings of the IEEE International Conference on Healthcare Informatics (ICHI)*. 2024, pp. 135–140. DOI: 10.1109/ICHI61247.2024.00025.

[236] T. Sarwar, S. Seifollahi, J. Chan, X. Zhang, V. Aksakalli, I. Hudson, K. Verspoor and L. Cavedon. 'The Secondary Use of Electronic Health Records for Data Mining: Data Characteristics and Challenges'. In: *ACM Comput. Surv.* 55.2 (Jan. 2022). DOI: 10.1145/3490234.

[237] O. Ciobanu-Caraus, A. Aicher, J. M. Kernbach, L. Regli, C. Serra and V. E. Staartjes. 'A Critical Moment in Machine Learning in Medicine: On Reproducible and Interpretable Learning'. In: *Acta Neurochirurgica* 166.1 (2024), p. 14. DOI: 10.1007/s00701-024-05892-8.

[238] Y. Singh, Q. A. Hathaway, V. Keishing, S. Salehi, Y. Wei, N. Horvat, D. V. Vera-Garcia, A. Choudhary, A. Mula Kh, E. Quaia and J. B. Andersen. 'Beyond Post hoc Explanations: A Comprehensive Framework for Accountable AI in Medical Imaging Through Transparency, Interpretability, and Explainability'. In: *Bioengineering* 12.8 (2025), p. 879. DOI: 10.3390/bioengineering12080879.

[239] Y. Ren, Y. Li, T. J. Loftus, J. Balch, K. L. Abbott, M. M. Ruppert, Z. Guan, B. Shickel, P. Rashidi, T. Ozrazgat-Baslanti and A. Bihorac. 'Identifying Acute Illness Phenotypes via Deep Temporal Interpolation and Clustering Network on Physiologic Signatures'. In: *Scientific Reports* 14.1 (2024), p. 8442. DOI: 10.1038/s41598-024-59047-x.

[240] S. N. Shukla and B. M. Marlin. 'Interpolation-Prediction Networks for Irregularly Sampled Time Series'. In: *International Conference on Learning Representations*. 2019.

[241] J. Xie, R. Girshick and A. Farhadi. 'Unsupervised Deep Embedding for Clustering Analysis'. In: *Proceedings of the 33rd International Conference on Machine Learning*. 2016, pp. 478–487.

[242] X. Ruan, S. Lu, L. Wang, A. Wen, S. Murali and H. Liu. 'Deep Phenotyping of Obesity: Electronic Health Record–Based Temporal Modeling Study'. In: *Journal of Medical Internet Research* 27 (2025), e70140. DOI: 10.2196/70140.

[243] K. Merkelbach, S. Schaper, C. Diedrich and et al. 'Novel Architecture for Gated Recurrent Unit Autoencoder Trained on Time Series from Electronic Health Records Enables Detection of ICU Patient Subgroups'. In: *Scientific Reports* 13 (2023), p. 4053. DOI: 10.1038/s41598-023-30986-1.

[244] Y. Qu, R. Lyu, D. Wang, Y. Dai, A. Turcan, S. Yu, J. Xie, J. Roach, C. Butler, P.-T. Yap, H. Zhu, S. Dashper, A. A. Ribeiro, D. Li, K. Divaris and D. Wu. 'Deep-Learning-Based Interpolation of Longitudinal Microbiome Data Powers Biologically Informative Discovery'. In: *bioRxiv* (2025). DOI: 10.1101/2025.02.17.638709.

[245] J. W. T. M. de Kok, F. van Rosmalen, J. Koeze and et al. 'Deep Embedded Clustering Generalisability and Adaptation for Integrating Mixed Datatypes: Two Critical Care Cohorts'. In: *Scientific Reports* 14 (2024), p. 1045. DOI: 10.1038/s41598-024-51699-z.

[246] P.-J. Tsai, K.-F. Chen, C. Limbud and Y.-J. Tseng. 'naviDCN: Navigator-Guided Multi-Modal Deep Clustering for Sepsis Phenotyping in Early ICU Admission'. In: *Proceedings of the 47th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. 2025, pp. 1–7. DOI: 10.1109/EMBC58623.2025.11254005.

[247] J. Castela Forte, G. Yeshmagambetova, M. L. van der Grinten, B. Hiemstra, T. Kaufmann, R. J. Eck, F. Keus, A. H. Epema, M. A. Wiering and I. C. C. van der Horst. 'Identifying and Characterizing High-Risk Clusters in a Heterogeneous ICU Population with Deep Embedded Clustering'. In: *Scientific Reports* 11.1 (2021), p. 12109. DOI: 10.1038/s41598-021-91297-x.

[248] X. Zheng, M. Schürch, X. Chen, M. A. Komninou, R. Schüpbach, A. Allam, J. Bartussek and M. Krauthammer. 'Clustering of Disease Trajectories with Explainable Machine Learning: A Case Study on Postoperative Delirium Phenotypes'. In: *arXiv preprint arXiv:2405.03327* (2024).

[249] D. T. A. Luong, P. Singh, M. Ramezani and V. Chandola. 'longSil: An Evaluation Metric to Assess Quality of Clustering Longitudinal Clinical Data'. In: *Journal of Healthcare Informatics Research* 3.4 (2019), pp. 441–459. DOI: 10.1007/s41666-019-00058-z.

[250] A. Dagliati, N. Geifman, N. Peek, J. H. Holmes, L. Sacchi, R. Bellazzi, S. E. Sajjadi and A. Tucker. 'Using Topological Data Analysis and Pseudo Time Series to Infer Temporal Phenotypes from Electronic Health Records'. In: *Artificial Intelligence in Medicine* 108 (2020), p. 101930. DOI: 10.1016/j.artmed.2020.101930.

[251] X. Wang, D. Sontag and F. Wang. 'Unsupervised learning of disease progression models'. In: *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Association for Computing Machinery, 2014, pp. 85–94. DOI: 10.1145/2623330.2623754.

[252] C. Lee, J. Rashbass and M. van der Schaar. 'Outcome-Oriented Deep Temporal Phenotyping of Disease Progression'. In: *IEEE Transactions on Biomedical Engineering* 68.8 (2021), pp. 2423–2434. DOI: 10.1109/TBME.2020.3041815.

[253] O. Mentes, D. Celik, M. Yildiz, K. Ensarioglu, M. O. Cirik, T. T. Peker, F. Canbay, G. E. Doganay and A. Kahraman. 'A Retrospective Evaluation of the Cardiometabolic Profile of Patients with COPD-Related Type 2 Respiratory Failure in the Intensive Care Unit'. In: *Medicina* 61 (2025), p. 705. DOI: 10.3390/medicina61040705.

[254] M. Juniper. 'NEWS2, patient safety and hypercapnic respiratory failure'. In: *Clinical Medicine* 22.6 (Nov. 2022), pp. 518–521. DOI: 10.7861/clinmed.2022-0352.

[255] V. Nikolaou, S. Massaro, M. Fakhimi, L. Stergioulas and D. Price. 'COPD phenotypes and machine learning cluster analysis: A systematic review and future research agenda'. In: *Respiratory Medicine* 171 (2020), p. 106093. DOI: 10.1016/j.rmed.2020.106093.

[256] C. C. Hardin. 'Novel Phenotypes in Respiratory Failure: Same As It Ever Was'. In: *American Journal of Respiratory and Critical Care Medicine* 202.9 (2020), pp. 1207–1209. DOI: 10.1164/rccm.202008-3322ED.

[257] D. Wong, T. Bonnici, J. Knight, L. Morgan, P. Coombes and P. Watkinson. 'SEND: A System for Electronic Notification and Documentation of Vital Sign Observations'. In: *BMC Medical Information and Decision Making* 15 (Aug. 2015), pp. 1–12. DOI: 10.1186/s12911-015-0186-y.

[258] C. A. Kelly, A. Upex and D. Bateman. 'Comparison of consciousness level assessment in the poisoned patient using the alert/verbal/painful/unresponsive scale and the glasgow coma scale'. In: *Annals of Emergency Medicine* 44 (2 2004), pp. 108–113. DOI: https://doi.org/10.1016/j.annemergmed.2004.03.028.

[259] D. Bell, J. Baker, C. Williams and L. Bassin. 'A Trend-Based Early Warning Score Can Be Implemented in a Hospital Electronic Medical Record to Effectively Predict Inpatient Deterioration'. In: *Critical Care Medicine* 49.10 (2021), e961–e967. DOI: 10.1097/CCM.0000000000005064.

[260] J. Briggs, I. Kostakis, P. Meredith, C. Dall'Ora, J. Darbyshire, S. Gerry et al. 'Safer and More Efficient Vital Signs Monitoring Protocols to Identify the Deteriorating Patients in the General Hospital Ward: An Observational Study'. In: *Health Services and Delivery Research* 12.6 (2024). DOI: 10.3310/HYTR4612.

[261] National Institute for Health and Care Excellence. *National Early Warning Score Systems That Alert to Deteriorating Adult Patients in Hospital*. Medtech Innovation Briefing MIB205. Accessed 11 February 2026. London, UK: National Institute for Health and Care Excellence, 2020.

[262] A. E. W. Johnson, T. J. Pollard, L. Shen, L.-w. H. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. A. Celi and R. G. Mark. 'MIMIC-III, a freely accessible critical care database'. In: *Scientific Data* 3 (2016), p. 160035. DOI: 10.1038/sdata.2016.35.

[263] A. L. Goldberger, L. A. N. Amaral, L. Glass, H. Jeffrey M, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng and H. E. Stanley. 'PhysioBank, PhysioToolkit, and PhysioNet Components of a New Research Resource for Complex Physiologic Signals'. In: *Circulation* 101 (23 June 2000), pp. 215–220.

[264] D. Kim and B. T. Jin. 'Development and Comparative Performance of Physiologic Monitoring Strategies in the Emergency Department'. In: *JAMA Network Open* 5.9 (2022), e2233712. DOI: 10.1001/jamanetworkopen.2022.33712.

[265] J. Wei, C. Ouyang, B. Wickramanayake, Z. He, K. Perera and C. Moreira. 'Curation and Analysis of MIMICEL - An Event Log for MIMIC-IV Emergency Department'. In: *arXiv preprint arXiv:2506.12808* (2025).

[266] L. Wolf, K. Ceci, D. McCallum and D. Brecher. *Emergency Severity Index Handbook (5th ed.)* 2023.

[267] M. M. Churpek, T. C. Yuen and D. P. Edelson. 'Risk Stratification of Hospitalized Patients on the Wards'. In: *Chest* 143.6 (June 2013), pp. 1758–1765. DOI: 10.1378/chest.12-1605.

[268] V. Thevrekandy, A. Sreekumar, P. Aggarwal, J. Nayer and K. R. Sanith. 'Emergency Department Boarding Time and In-Hospital Mortality: A Prospective Observational Study'. In: *Turkish Journal of Emergency Medicine* 25 (2025), pp. 297–304.

[269] G. Z. Gabayan, M. K. Gould, R. E. Weiss, V. Y. Chiu and C. A. Sarkisian. 'A Risk Score to Predict Short-term Outcomes Following Emergency Department Discharge'. In: *Western Journal of Emergency Medicine* 19.5 (Sept. 2018), pp. 842–848. DOI: 10.5811/westjem.2018.7.37945.

[270] I. Olaronke and O. Oluwaseun. 'Big Data in Healthcare: Prospects, Challenges and Resolutions'. In: *2016 Future Technologies Conference (FTC)*. 2016, pp. 1152–1157. DOI: 10.1109/FTC.2016.7821747.

[271] A. L. Jeannic, C. Quelen, C. Alberti and I. Durand-Zaleski. 'Comparison of two data collection processes in clinical studies: electronic and paper case report forms'. In: *BMC Medical Research Methodology* 14 (7 2014), pp. 1–10. DOI: 10.1186/1471-2288-14-7.

[272] S. Dash, S. K. Shakyawar, M. Sharma and S. Kaushik. 'Big data in healthcare: management, analysis and future prospects'. In: *Journal of Big Data* 6 (1 Dec. 2019), pp. 1–25. DOI: 10.1186/s40537-019-0217-0.

[273] M. Gupta, B. Gallamoza, N. Cutrona, P. Dhakal, R. Poulain and R. Beheshti. 'An Extensive Data Processing Pipeline for MIMIC-IV'. In: *Proceedings of the Machine Learning for Health Conference*. Vol. 193. Proceedings of Machine Learning Research. Nov. 2022, pp. 311–325.

[274] S. Wang, M. B. A. McDermott, G. Chauhan, M. Ghassemi, M. C. Hughes and T. Naumann. 'MIMIC-Extract: a data extraction, preprocessing, and representation pipeline for MIMIC-III'. In: *Proceedings of the ACM Conference on Health, Inference, and Learning*. CHIL '20. Toronto, Ontario, Canada: Association for Computing Machinery, 2020, pp. 222–235. DOI: 10.1145/3368555.3384469.

[275] A. Tawakuli, B. Havers, V. Gulisano, D. Kaiser and T. Engel. 'Survey: Time-series Data Preprocessing: A Survey and an Empirical Analysis'. In: *Journal of Engineering Research* 13.2 (2025), pp. 674–711. DOI: 10.1016/j.jer.2024.02.018.

[276] P. C. Austin, I. R. White, D. S. Lee and S. van Buuren. 'Missing Data in Clinical Research: A Tutorial on Multiple Imputation'. In: *Canadian Journal of Cardiology* 37.9 (Sept. 2021), pp. 1322–1331. DOI: 10.1016/j.cjca.2020.11.010.

[277] J. Digitale, D. Franzon, M. J. Pletcher, C. E. McCulloch and E. D. Gennatas. 'Methods for Addressing Missingness in Electronic Health Record Data for Clinical Prediction Models: Comparative Evaluation'. In: *JMIR Medical Informatics* 13 (Nov. 2025), e79307. DOI: 10.2196/79307.

[278] L. Grigoroff, R. Masuda, J. Lindon, J. Kadyrov, J. K. Nicholson, E. Holmes and J. Wist. 'Evaluation of Imputation Strategies for Multi-centre Studies: Application to a Large Clinical Pathology Dataset'. In: *PLOS ONE* 20.11 (Nov. 2025), e0335852. DOI: 10.1371/journal.pone.0335852.

[279] P. J. Rousseeuw. 'Silhouettes: A graphical aid to the interpretation and validation of cluster analysis'. In: *Journal of Computational and Applied Mathematics* 20 (C Nov. 1987), pp. 53–65. DOI: 10.1016/0377-0427(87)90125-7.

[280] J. Bezdek and N. Pal. 'Cluster validation with generalized Dunn's indices'. In: *Proceedings 1995 Second New Zealand International Two-Stream Conference on Artificial Neural Networks and Expert Systems*. 1995, pp. 190–193. DOI: 10.1109/ANNES.1995.499469.

[281] D. L. Davies and D. W. Bouldin. 'A Cluster Separation Measure'. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-1 (2 Feb. 1979), pp. 224–227. DOI: 10.1109/TPAMI.1979.4766909.

[282] G. Tsoumakas and I. Katakis. 'Multi-Label Classification: An Overview'. In: *International Journal of Data Warehousing and Mining* 3.3 (2007), pp. 1–13. DOI: 10.4018/jdwm.2007070101.

[283] T. Fawcett. 'An introduction to ROC analysis'. In: *Pattern Recognition Letters* 27 (8 June 2006), pp. 861–874. DOI: 10.1016/j.patrec.2005.10.010.

[284] E. A. Freeman and G. G. Moisen. 'A Comparison of the Performance of Threshold Criteria for Binary Classification in terms of Predicted Prevalence and Kappa'. In: *Ecological Modelling* 217 (1-2 Sept. 2008), pp. 48–58. DOI: 10.1016/j.ecolmodel.2008.05.015.

[285] J. Opitz. 'From Bias and Prevalence to Macro F1, Kappa, and MCC: A Structured Overview of Metrics for Multi-Class Evaluation'. In: *arXiv preprint arXiv:2209.05355* (2022). arXiv: 2209.05355 [cs.LG].

[286] J. Opitz. 'A Closer Look at Classification Evaluation Metrics and a Critical Reflection of Common Evaluation Practice'. In: *Transactions of the Association for Computational Linguistics* 12 (2024), pp. 820–836. DOI: 10.1162/tacl_a_00675.

[287] W. M. Rand. 'Objective Criteria for the Evaluation of Clustering Methods'. In: *Journal of the American Statistical Association* 66 (336 Dec. 1971), pp. 846–850. DOI: 10.1080/01621459.1971.10482356.

[288] L. Hubert and P. Arabie. 'Comparing Partitions'. In: *Journal of Classification* 2.1 (Dec. 1985), pp. 193–218. DOI: 10.1007/BF01908075.

[289] D. Steinley. 'Properties of the Hubert–Arabie Adjusted Rand Index'. In: *Psychological Methods* 9.3 (Sept. 2004), pp. 386–396. DOI: 10.1037/1082-989X.9.3.386.

[290] S. Romano, N. X. Vinh, J. Bailey and K. Verspoor. 'Adjusting for Chance Clustering Comparison Measures'. In: *Journal of Machine Learning Research* 17.134 (2016), pp. 1–32.

[291] C. E. Shannon. 'A Mathematical Theory of Communication'. In: *The Bell System Technical Journal* 27 (3 1948), pp. 379–423. DOI: 10.1002/j.1538-7305.1948.tb01338.x.

[292] N. X. Vinh, J. Epps and J. Bailey. 'Information Theoretic Measures for Clusterings Comparison: Variants, Properties, Normalization and Correction for Chance'. In: *Journal of Machine Learning Research* 11.95 (2010), pp. 2837–2854.

[293] E. H. Yamga, L. Mullie, M. Durand, A. Cadrin-Chenevert, A. Tang, E. Montagnon, C. Chartrand-Lefebvre and M. Chassé. 'Interpretable Clinical Phenotypes among Patients Hospitalized with COVID-19 Using Cluster Analysis'. In: *Frontiers in Digital Health* 5 (Apr. 2023), p. 1142822. DOI: 10.3389/fdgth.2023.1142822.

[294] P. Kent, M. J. Stochkendahl, H. W. Christensen and A. Kongsted. 'Could the Clinical Interpretability of Subgroups Detected Using Clustering Methods Be Improved by Using a Novel Two-Stage Approach?' In: *Chiropractic & Manual Therapies* 23 (July 2015), p. 20. DOI: 10.1186/s12998-015-0064-9.

[295] Royal College of Physicians. *National Early Warning Score (NEWS) 2: Standardising the Assessment of Acute-Illness Severity in the NHS. Updated Report of a Working Party*. Tech. rep. Accessed 25 March 2020. London: Royal College of Physicians, 2017.

[296] C. Holder, M. Middlehurst and A. Bagnall. 'A review and evaluation of elastic distance functions for time series clustering'. In: *Knowl. Inf. Syst.* 66.2 (Sept. 2023), pp. 765–809. DOI: 10.1007/s10115-023-01952-0.

[297] V. R. Konda and J. N. Tsitsiklis. 'Actor-Critic Algorithms'. In: *Advances in Neural Information Processing Systems*. MIT Press, 1999, pp. 1008–1014.

[298] H. Lv, X. Yang, B. Wang, S. Wang, X. Du, Q. Tan, Z. Hao, Y. Liu, J. Yan and Y. Xia. 'Machine Learning-Driven Models to Predict Prognostic Outcomes in Patients Hospitalized With Heart Failure Using Electronic Health Records: Retrospective Study'. In: *Journal of Medical Internet Research* 23.4 (2021), e24996. DOI: 10.2196/24996.

[299] N. Bhaskhar, W. Ip, J. H. Chen and D. L. Rubin. 'Clinical Outcome Prediction Using Observational Supervision With Electronic Health Records and Audit Logs'. In: *Journal of Biomedical Informatics* 147 (2023), p. 104522. DOI: 10.1016/j.jbi.2023.104522.

[300] R. Karasneh, S. Al-Azzam, K. H. Alzoubi, M. Araydah, D. Rahhal, Y. Al-Azzam, Z. Kharaba, S. Kabbaha and M. A. Aldeyab. 'ICU Mortality Prediction Using XGBoost-Based Scoring Systems: A Study from a Developing Country'. In: *Recent Patents on Clinical Trials* (2025). Epub ahead of print. DOI: 10.2174/0115748871348585250604065542.

[301] X. Zhou, X. Li, Z. Zhang, Q. Han, H. Deng, Y. Jiang, C. Tang and L. Yang. 'Support Vector Machine Deep Mining of Electronic Medical Records to Predict the Prognosis of Severe Acute Myocardial Infarction'. In: *Frontiers in Physiology* 13 (2022), p. 991990. DOI: 10.3389/fphys.2022.991990.

[302] C. Cortes and V. Vapnik. 'Support-vector networks'. In: *Machine Learning* 20.3 (1995), pp. 273–297. DOI: 10.1007/BF00994018.

[303] T. Chen and C. Guestrin. 'XGBoost: A Scalable Tree Boosting System'. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '16. San Francisco, California, USA: Association for Computing Machinery, 2016, pp. 785–794. DOI: 10.1145/2939672.2939785.

[304] J. H. Friedman. 'Greedy function approximation: A gradient boosting machine'. In: *Annals of Statistics* 29.5 (2001), pp. 1189–1232. DOI: 10.1214/aos/1013203451.

[305] T. Sugino, T. Kawase, S. Onogi, T. Kin, N. Saito and Y. Nakajima. 'Loss Weightings for Improving Imbalanced Brain Structure Segmentation Using Fully Convolutional Networks'. In: *Healthcare* 9.8 (2021), p. 938. DOI: 10.3390/healthcare9080938.

[306] D. P. Kingma and J. L. Ba. 'Adam: A Method for Stochastic Optimization'. In: *3rd International Conference on Learning Representations (ICLR) 2015*. Ed. by Y. Bengio and Y. LeCun. May 2015.

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

[308] J. Demšar. 'Statistical Comparisons of Classifiers over Multiple Data Sets'. In: *J. Mach. Learn. Res.* 7 (Dec. 2006), pp. 1–30.

[309] T. J. VanderWeele and M. B. Mathur. 'Some Desirable Properties of the Bonferroni Correction: Is the Bonferroni Correction Really So Bad?' In: *American Journal of Epidemiology* 188.3 (2019), pp. 617–618. DOI: 10.1093/aje/kwy250.

[310] A. Şenol. 'VIASCKDE Index: A Novel Internal Cluster Validity Index for Arbitrary-Shaped Clusters Based on the Kernel Density Estimation'. In: *Computational Intelligence and Neuroscience* 2022 (2022), p. 4059302. DOI: 10.1155/2022/4059302.

[311] F. Xie, H. Yuan, Y. Ning, M. E. H. Ong, M. Feng, W. Hsu, B. Chakraborty and N. Liu. 'Deep Learning for Temporal Data Representation in Electronic Health Records: A Systematic Review of Challenges and Methodologies'. In: *Journal of Biomedical Informatics* 126 (Jan. 2022), p. 103980. DOI: 10.1016/j.jbi.2021.103980.

[312] A. Amirahmadi, M. Ohlsson and K. Etminani. 'Deep Learning Prediction Models based on EHR Trajectories: A Systematic Review'. In: *Journal of Biomedical Informatics* 144 (July 2023), p. 104430. DOI: 10.1016/j.jbi.2023.104430.

[313] E. Choi, M. T. Bahadori, A. Schuetz, W. F. Stewart and J. Sun. 'Doctor AI: Predicting Clinical Events via Recurrent Neural Networks'. In: *Proceedings of the 1st Machine Learning for Healthcare Conference*. Ed. by F. Doshi-Velez, J. Fackler, D. Kale, B. Wallace and J. Wiens. PMLR, Aug. 2016, pp. 301–318.

[314] T. Pham, T. Tran, D. Phung and S. Venkatesh. 'Predicting Healthcare Trajectories from Medical Records: A Deep Learning Approach'. In: *Journal of Biomedical Informatics* 69 (Apr. 2017), pp. 218–229. DOI: 10.1016/j.jbi.2017.04.001.

[315] S. M. Lauritsen, M. Kristensen, M. V. Olsen, M. S. Larsen, K. M. Lauritsen, M. J. Jørgensen, J. Lange and B. Thiesson. 'Explainable artificial intelligence model to predict acute critical illness from electronic health records'. In: *Nature Communications* 11 (3852 July 2020). DOI: 10.1038/s41467-020-17431-x.

[316] L. Franz, Y. Shrestha and B. Paudel. 'A Deep Learning Pipeline for Patient Diagnosis Prediction using Electronic Health Records'. In: *19th International Workshop on Data Mining in Bioinformatics - at the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM, New York, NY, USA, Aug. 2020. DOI: 10.1145/1122445.1122457.

[317] J. F. Silva and S. Matos. 'Patient Trajectory Modelling in Longitudinal Data: A Review on Existing Solutions'. In: *2021 IEEE 34th International Symposium on Computer-Based Medical Systems (CBMS)*. Vol. 2021-June. Institute of Electrical and Electronics Engineers Inc., June 2021, pp. 480–485. DOI: 10.1109/CBMS52027.2021.00057.

[318] Z. Yang, M. Dehmer, O. Yli-Harja and F. Emmert-Streib. 'Combining Deep Learning with Token Selection for Patient Phenotyping from Electronic Health Records'. In: *Scientific Reports* 10 (1 Jan. 2020), p. 1432. DOI: https://doi.org/10.1038/s41598-020-58178-1.

[319] B. Xu and G. Yang. 'Interpretability research of deep learning: A literature survey'. In: *Information Fusion* 115 (2025), p. 102721. DOI: https://doi.org/10.1016/j.inffus.2024.102721.

[320] A. Shrikumar, P. Greenside and A. Kundaje. 'Learning Important Features Through Propagating Activation Differences'. In: *Proceedings of Machine Learning Research*. Ed. by D. Precup and Y. W. Teh. PMLR, Aug. 2017, pp. 3145–3153.

[321] K. Simonyan, A. Vedaldi and A. Zisserman. 'Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps'. In: *arXiv preprint arXiv:1312.6034* (2013). DOI: 10.48550/arXiv.1312.6034.

[322] P. Linardatos, V. Papastefanopoulos and S. Kotsiantis. 'Explainable AI: A Review of Machine Learning Interpretability Methods'. In: *Entropy* 23 (1 2021), pp. 1–45. DOI: 10.3390/e23010018.

[323] S. Lei, W. Yi, C. Ying and W. Ruibin. 'Review of Attention Mechanism in Natural Language Processing'. In: *Data Analysis and Knowledge Discovery* 4 (5 2020), pp. 1–14. DOI: 10.11925/infotech.2096-3467.2019.1317.

[324] T. Gonçalves, I. Rio-Torto, L. F. Teixeira and J. S. Cardoso. 'A Survey on Attention Mechanisms for Medical Applications: Are we Moving Toward Better Algorithms?' In: *IEEE Access* 10 (2022), pp. 98909–98935. DOI: 10.1109/ACCESS.2022.3206449.

[325] T. Lin, Y. Wang, X. Liu and X. Qiu. 'A Survey of Transformers'. In: *AI Open* 3 (Jan. 2022), pp. 111–132. DOI: 10.1016/j.aiopen.2022.10.001.

[326] Z. Liu, Y. Hu, X. Wu, G. Mertes, Y. Yang and D. A. Clifton. 'Patient Clustering for Vital Organ Failure Using ICD Code With Graph Attention'. In: *IEEE Transactions on Biomedical Engineering* 70 (8 Aug. 2023), pp. 2329–2337. DOI: 10.1109/TBME.2023.3243311.

[327] F. E. Shamout, T. Zhu, P. Sharma, P. J. Watkinson and D. A. Clifton. 'Deep Interpretable Early Warning System for the Detection of Clinical Deterioration'. In: *IEEE Journal of Biomedical and Health Informatics* 24 (2 Feb. 2020), pp. 437–446. DOI: 10.1109/JBHI.2019.2937803.

[328] D. A. Kaji, J. R. Zech, J. S. Kim, S. K. Cho, N. S. Dangayach, A. B. Costa and E. K. Oermann. 'An Attention based Deep Learning Model of Clinical Events in the Intensive Care Unit'. In: *PloS one* 14 (2 Feb. 2019). DOI: 10.1371/JOURNAL.PONE.0211057.

[329] I. Gandin, A. Scagnetto, S. Romani and G. Barbati. 'Interpretability of Time-Series Deep Learning Models: A Study in Cardiovascular Patients Admitted to Intensive Care Unit'. In: *Journal of Biomedical Informatics* 121 (1532 Oct. 2021), p. 103876. DOI: 10.1016/J.JBI.2021.103876.

[330] scikit-learn developers. *LinearRegression Class. scikit-learn User Guide*. 2024. URL: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html (visited on 23/04/2026).

[331] T. M. Cover and J. A. Thomas. 'Entropy, Relative Entropy, and Mutual Information'. In: *Elements of Information Theory*. Hoboken, NJ: John Wiley & Sons, 2005, pp. 13–55. DOI: 10.1002/047174882X.ch2.

[332] X. Glorot and Y. Bengio. 'Understanding the Difficulty of Training Deep Feedforward Neural Networks'. In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. Ed. by Y. W. Tee and M. Titterington. PMLR, May 2010, pp. 249–256.

[333] E. Jang, S. Gu and B. Poole. 'Categorical Reparameterization with Gumbel-Softmax'. In: *5th International Conference on Learning Representations (ICLR) 2017*. 2017.

[334] C. J. Maddison, A. Mnih and Y. W. Teh. 'The Concrete Distribution: A Continuous Relaxation of Discrete Random Variables'. In: *Proceedings of the 5th International Conference on Learning Representations*. Toulon, France, 2017.

[335] Y. Bengio. 'Estimating or Propagating Gradients Through Stochastic Neurons'. In: *arXiv e-prints*, arXiv:1305.2982 (May 2013), arXiv:1305.2982. DOI: 10.48550/arXiv.1305.2982. arXiv: 1305.2982 [cs.LG].

[336] D. J. Rezende, S. Mohamed and D. Wierstra. 'Stochastic Backpropagation and Approximate Inference in Deep Generative Models'. In: *Proceedings of the 31st International Conference on Machine Learning*. Ed. by E. P. Xing and T. Jebara. Vol. 32. Proceedings of Machine Learning Research 2. Bejing, China: PMLR, June 2014, pp. 1278–1286.

[337] D. M. Blei, A. Kucukelbir and J. D. McAuliffe. 'Variational Inference: A Review for Statisticians'. In: *Journal of the American Statistical Association* 112.518 (2017), pp. 859–877.

[338] J. Chung, K. Kastner, L. Dinh, K. Goel, A. C. Courville and Y. Bengio. 'A Recurrent Latent Variable Model for Sequential Data'. In: *Advances in Neural Information Processing Systems 28*. Ed. by C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama and R. Garnett. Curran Associates, Inc., 2015, pp. 2980–2988.

[339] J. Bayer and C. Osendorfer. 'Learning Stochastic Recurrent Networks'. In: *arXiv e-prints*, arXiv:1411.7610 (Nov. 2014), arXiv:1411.7610. DOI: 10.48550/arXiv.1411.7610. arXiv: 1411.7610 [stat.ML].

[340] J. Durbin and S. J. Koopman. *Time Series Analysis by State Space Methods*. Oxford University Press, May 2012. DOI: 10.1093/acprof:oso/9780199641178.001.0001.

[341] Y. Xue, E. Choi, M. Zitnik, B. Wang and J. Sun. 'Deep State-Space Generative Model for Correlated Time-to-Event Predictions'. In: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2020, X–Y. DOI: 10.1145/3394486.3403206.

[342] R. E. Kalman. 'A New Approach to Linear Filtering and Prediction Problems'. In: *Journal of Basic Engineering* 82.1 (Mar. 1960), pp. 35–45. DOI: 10.1115/1.3662552. eprint: https://asmedigitalcollection.asme.org/fluidsengineering/article-pdf/82/1/35/5518977/35_1.pdf.

[343] R. G. Krishnan, U. Shalit and D. Sontag. 'Deep Kalman Filters'. In: *arXiv preprint arXiv:1511.05121* (2015).

[344] M. J. Johnson, D. Duvenaud, A. B. Wiltschko, S. R. Datta and R. P. Adams. 'Composing graphical models with neural networks for structured representations and fast inference'. In: *Proceedings of the 30th International Conference on Neural Information Processing Systems*. Barcelona, Spain: Curran Associates Inc., 2016, pp. 2954–2962.

[345] Y. Okamoto, A. Nakamura, R. Kojima, E. Uchino, Y. Mineharu, Y. Harada, M. Kamada, M. Sakuragi, M. Muto, M. Yanagita and Y. Okuno. 'A Deep State-Space Analysis Framework for Cancer Patient Latent State Estimation and Classification from EHR Time-Series Data'. In: *PLOS ONE* 21.1 (2026), e0341003. DOI: 10.1371/journal.pone.0341003.

[346] S. J. Gershman and D. M. Blei. 'A tutorial on Bayesian nonparametric models'. In: *Journal of Mathematical Psychology* 56.1 (2012), pp. 1–12. DOI: https://doi.org/10.1016/j.jmp.2011.08.004.

[347] N. L. Hjort, C. Holmes, P. Müller and S. G. Walker, eds. *Bayesian Nonparametrics*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge: Cambridge University Press, 2010.

[348] T. S. Ferguson. 'A Bayesian Analysis of Some Nonparametric Problems'. In: *Annals of Statistics* 1 (1973), pp. 209–230.

[349] C. E. Rasmussen. 'The Infinite Gaussian Mixture Model'. In: *Advances in Neural Information Processing Systems 12*. Ed. by S. A. Solla, T. K. Leen and K.-R. Müller. MIT Press, 2000.

[350] D. Görür and C. E. Rasmussen. 'Dirichlet Process Gaussian Mixture Models: Choice of the Base Distribution'. In: *Journal of Computer Science and Technology* 25.4 (2010), pp. 653–664.

[351] Y. W. Teh, M. I. Jordan, M. J. Beal and D. M. Blei. 'Hierarchical Dirichlet Processes'. In: *Journal of the American Statistical Association* 101.476 (2006), pp. 1566–1581.

[352] E. Fox, E. Sudderth, M. Jordan and A. Willsky. 'An HDP-HMM for systems with state persistence'. In: *Proceedings of the 25th International Conference on Machine Learning*. Jan. 2008, pp. 312–319.

[353] A. Pérez-Herrero, P. Félix, J. Presedo and C. H. Ek. 'Bayesian Nonparametric Dynamical Clustering of Time Series'. In: *arXiv preprint arXiv:2510.06919* (2025). DOI: 10.48550/arXiv.2510.06919. arXiv: 2510.06919 [stat.ML].

[354] A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari and D. B. Rubin. *Bayesian Data Analysis*. 3rd ed. CRC Press, 2013.

[355] L. Devroye. 'Sample-based non-uniform random variate generation'. In: *Proceedings of the 18th Conference on Winter Simulation*. Washington, D.C., USA: Association for Computing Machinery, 1986, pp. 260–265. DOI: 10.1145/318242.318443.

[356] O. Sener and V. Koltun. 'Multi-Task Learning as Multi-Objective Optimization'. In: *Advances in Neural Information Processing Systems 31*. Ed. by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi and R. Garnett. Curran Associates, Inc., 2018, pp. 527–538.

[357] P. Chapfuwa, C. Li, N. Mehta, L. Carin and R. Henao. 'Survival cluster analysis'. In: *Proceedings of the ACM Conference on Health, Inference, and Learning*. Toronto, Ontario, Canada: Association for Computing Machinery, 2020, pp. 60–68. DOI: 10.1145/3368555.3384465.

[358] K. H. Eng and B. M. Hanlon. 'Discrete mixture modeling to address genetic heterogeneity in time-to-event regression'. In: *Bioinformatics* 30.12 (June 2014), pp. 1690–1697. DOI: 10.1093/bioinformatics/btu065. eprint: https://academic.oup.com/bioinformatics/article-pdf/30/12/1690/48926918/bioinformatics_30_12_1690.pdf.

[359] M. Shankar-Hari, S. Santhakumaran, A. T. Prevost, J. K. Ward, T. Marshall, C. Bradley and et al. 'Defining phenotypes and treatment effect heterogeneity to inform acute respiratory distress syndrome and sepsis trials: secondary analyses of three RCTs'. In: *Efficacy and Mechanism Evaluation* 8.10 (2021). DOI: 10.3310/eme08100.

[360] D. C. Kale, Z. Che, M. T. Bahadori, W. Li, Y. Liu and R. Wetzel. 'Causal Phenotype Discovery via Deep Networks'. In: *AMIA Annual Symposium Proceedings*. San Francisco, CA, USA: American Medical Informatics Association, Nov. 2015, pp. 677–686.

A Hyperparameter Grids

This appendix reports the values used for running hyperparameter grid search to derive the results referenced in the main text. Detailed interpretation and analysis are described in the corresponding chapters.

A.1 Experimental Setup for Chapter 4

Table A.1 summarizes the hyperparameter grids explored for all models evaluated in Chapter 4. For each model, the set of candidate values considered during grid search is reported. All experiments were repeated across five random seeds $\{1717, 2323, 4343, 4747, 1111\}$ to ensure robustness. The best-performing configuration per model, selected according to the procedure described in Section 4.4.2, is indicated with an asterisk *.

Model	Hyperparameter	Values Explored
Classical and Clinical Methods		
TSKM	Distance function d	L_2, DTW*, Soft-DTW
	Initialization strategy	Random, $K\text{-}means++$*
	Number of clusters K	2*, 3, 4, 5, 6, 8, 10
	Maximum iterations	100, 200, 500*
	Convergence threshold	1e-5*
SVM	Regularization C_{SVM}	0.01, 0.1*, 1, 2, 5
	Kernel κ	Linear, RBF*, Polynomial
	RBF bandwidth γ	1e-4, 1e-3, 1e-2*
	Temporal adaptation	Flattening*, Per-feature ensemble
XGBoost	Number of estimators	50, 100, 200*, 300
	Maximum tree depth	3, 5*, 7, 10
	Learning rate	0.01, 0.05*, 0.1, 0.3
	Minimum child weight	1*, 3, 5
	Temporal adaptation	Flattening*, Per-feature ensemble
NEWS2	—	No free parameters

Table A.1: Hyperparameter grids relevant to the classical and clinical models evaluated in Chapter 4. For each model, the full set of values explored via grid search is listed. The best-performing configuration per model is indicated with an asterisk (*), selected according to the procedure described in Section 4.4.2. For NEWS2, no hyperparameters are optimized as it is a fixed clinical protocol; it is included for completeness.

Model	Hyperparameter	Values Explored
Deep Learning Methods		
SOM-VAE	Latent dimension d	16, 32*, 64
	SOM grid dimensions	2×2*, 3×3, 4×4
	Commitment loss weight α_{SV}	0.001, 0.01*, 0.1, 1.0
	SOM loss weight β_{SV}	0.001, 0.01*, 0.1, 1.0
	Transition loss weight γ_{SV}	0.001*, 0.01, 0.1, 1.0
	Smoothness loss weight τ_{SV}	0.001, 0.01, 0.1*, 1.0
	SOM learning rate ηSV	0.001, 0.01*, 0.1, 1.0
	Learning rate	1e-4, 1e-3*, 1e-2
	Batch size	32*, 64, 128
AC-TPC	Latent dimension d	16, 32, 64*
	LSTM hidden dimension	32, 64*, 128
	Number of clusters K	2, 3, 4*, 5, 6, 8, 10
	Confidence loss weight α_{AC}	0.001, 0.01, 0.1*, 1.0
	Separation loss weight β_{AC}	0.01, 0.1*, 1.0
	Learning rate	$10^{-4}, 10^{-3}, 10^{-2}$*
	Batch size	32, 64*, 128
AC-TPC$_{\text{w}}$	Latent dimension d	16, 32*, 64
	LSTM hidden dimension	32, 64*, 128
	Number of clusters K	2, 3, 4*, 5, 6
	Confidence loss weight α_{ACw}	0.001, 0.01, 0.1*, 1.0
	Separation loss weight β_{ACw}	0.001*, 0.01, 0.1, 1.0
	Learning rate	$10^{-4}, 5\times10^{-4}$*, $10^{-3}, 10^{-2}$
	Batch size	32, 64*, 128

Table A.1: (continued) Hyperparameter grids for the deep learning models evaluated in Chapter 4. For each model, the full set of values explored via grid search is listed. The configuration with the best performance per model is indicated with an asterisk (*), selected according to the procedure described in Section 4.4.2.

A.2 Experimental Setup for Chapter 5

Table A.2 summarizes the hyperparameter grids explored for all models evaluated in Chapter 5. The format follows that of Table A.1. The benchmark models (TSKM, SOM-VAE, AC-TPC, SVM, and XGBoost) retain the same grids as in Table A.1. The ablation variants (ABL1–ABL6) share the full CAMELOT grid, with structural modifications fixed as described in Section 5.4.2. The set of seeds used was $\{1001, 1012, 1134, 2475, 6138, 7415, 1663, 7205, 9253, 1782\}$. The configurations with the best performance are determined according to Section 5.4.1 and are reported separately for each dataset: † denotes the optimal configuration in HAVEN-T2RF and ‡ denotes the optimal configuration in MIMIC-IV-ED.

Model	Hyperparameter	Values Explored
Classical and Clinical Methods		
SVM	Same grid as Table A.1	
XGBoost	Same grid as Table A.1	
ESI	—	No free parameters
NEWS2	—	No free parameters
Deep Learning Benchmarks		
TSKM	Same grid as Table A.1	
SOM-VAE	Same grid as Table A.1	
AC-TPC	Same grid as Table A.1	
CAMELOT and Ablation Variants		
CAMELOT	Number of clusters K	$2, 3, 4, 5, 6^{\dagger,\ddagger}, 8, 10$
	Latent dimension l	$16, 32, 64, 128$
	Number of LSTM layers	$1^{\ddagger}, 2^{\dagger}, 3$
	LSTM hidden dimension	$32, 64^{\ddagger}, 128^{\dagger}$
	Number of MLP layers (Iden/Pred)	$1^{\dagger}, 2^{\ddagger}, 3$
	MLP hidden dimension (Iden/Pred)	$32, 64^{\ddagger}, 128^{\dagger}$
	Dropout rate	$0.0, 0.1^{\ddagger}, 0.2^{\dagger}, 0.3$
	Distribution loss weight λ_1	$0.001, 0.01^{\dagger,\ddagger}, 0.1, 1.0$
	Separation loss weight λ_2	$0.001^{\dagger}, 0.01, 0.1^{\ddagger}, 1.0$
	Learning rate	$10^{-4\dagger}, 10^{-3\ddagger}, 10^{-2}$
	Batch size b_s	$32, 64^{\dagger,\ddagger}, 128$
	Maximum epochs	$50^{\dagger}, 100^{\ddagger}, 200$
ABL1–ABL6	Same grid as CAMELOT; structural modifications fixed as described in Section 5.4.2.	

Table A.2: Hyperparameter grids for models evaluated in Chapter 5. For each model, the full set of values explored via grid search is listed. The best-performing configuration per model was selected according to the procedure described in Section 5.4.1. Benchmark models retained the same grid as Table A.1; see that table for full details. ESI has no free parameters and is included for completeness. Best-performing configurations are indicated per dataset: † for HAVEN-T2RF and ‡ for MIMIC-IV-ED.

A.3 Experimental Setup for Chapter 6

Table A.3 summarizes the hyperparameter grids explored for all models evaluated in Chapter 6. The format follows that of Table A.2 and the same 10 seeds are used as Chapter 5. Benchmark models retain identical grids; only the DeepGC configuration is reported in full.

Model	Hyperparameter	Values Explored
Classical and Clinical Methods		
SVM	Same grid as Table A.1	
XGBoost	Same grid as Table A.1	
NEWS2	—	No free parameters
ESI	—	No free parameters
Deep Learning Benchmarks		
TSKM	Same grid as Table A.1	
SOM-VAE	Same grid as Table A.1	
AC-TPC	Same grid as Table A.1	
CAMELOT	Same grid as Table A.2	
DeepGC		
DeepGC	Number of clusters K	$2, 3, 4, 5, 6^{\dagger,\ddagger}, 8, 10$
	Latent dimension l	$16, 32^{\dagger}, 64^{\ddagger}, 128$
	Hidden state dimension s	$32, 64^{\dagger,\ddagger}, 128$
	Gate function hidden dimension	$16^{\dagger,\ddagger}, 32, 64$
	Gate function layers	$1^{\dagger,\ddagger}, 2$
	Dropout rate	$0.0, 0.1^{\dagger}, 0.2^{\ddagger}, 0.3$
	Learning rate	$10^{-4}, 10^{-3\ddagger}, 10^{-2\dagger}$
	Batch size b_s	$32^{\dagger}, 64^{\ddagger}, 128$
	Maximum epochs	$50, 100^{\dagger}, 200^{\ddagger}$

Table A.3: Hyperparameter grids for models evaluated in Chapter 6. For each model, the full set of values explored via grid search is listed. Benchmark models retain the same grids as Table A.2; see that table for details. ESI has no free parameters and is included for completeness. Best-performing configurations are indicated per dataset: † for HAVEN-T2RF and ‡ for MIMIC-IV-ED.

B

Supplementary Results Chapter 5

This appendix provides additional material supporting the results in Chapter 5, including empirical cluster outcome counts, descriptive statistics of learned phenotypes, and trajectory visualizations for key clinical variables in HAVEN-T2RF. Detailed interpretation and statistical analysis are presented in the main text.

B.1 HAVEN-T2RF

B.1.1 Empirical Cluster Outcome Counts

Table B.1 reports the empirical outcome counts per outcome class within each cluster learned by CAMELOT in HAVEN-T2RF, under the inference assignment rule $\phi(\mathbf{X}_n)$ defined in Equation (5.2). These counts complement the learned outcome characterizations $\mathbf{o}_k^{\text{CAM,clus}}$ displayed in Figure 5.3, providing an empirical summary of the outcome frequencies within each cluster. Cluster sizes vary; proportions and statistical comparisons are discussed in Section 5.5.

B.1.2 Descriptive Statistics of Learned Cluster Phenotypes

Table B.2 presents summary statistics of all input features stratified by the clusters learned by CAMELOT in the HAVEN-T2RF dataset. Features are grouped according to clinical category — vital signs, static variables, serum laboratory tests,

Cluster	NoEvent	Death	ICU	Cardiac	Total
Cluster 1	149	44	10	7	210
Cluster 2	739	28	5	5	777
Cluster 3	579	15	6	2	602
Cluster 4	93	92	12	5	202
Cluster 5	373	2	2	0	377
Cluster 6	288	84	10	10	392

Table B.1: Empirical outcome frequencies within each cluster identified by CAMELOT in HAVEN-T2RF. Cluster assignments follow Equation (5.2), applied to the combined training and validation data. Outcome label definitions follow Section 3.2.1.

and hematological laboratory tests — consistent with the feature descriptions in Section 3.2.1. Summary statistics are reported per cluster, enabling a direct comparison of the temporal and static characteristics of each learned phenotype.

Variable (unit)	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6
n	210	777	602	202	377	392
Vital signs						
HR (bpm)	82.0 (71.0–94.0)	81.0 (72.0–91.0)	80.0 (70.4–90.0)	93.0 (83.0–103.0)	82.0 (72.0–92.0)	85.5 (74.0–97.0)
RR (breaths min^{-1})	18.0 (16.5–19.0)	18.0 (16.0–18.3)	17.0 (16.0–18.0)	19.3 (18.0–24.0)	17.0 (16.0–18.0)	18.0 (17.0–20.0)
SBP (mmHg)	123.0 (107.9–142.0)	136.0 (123.0–149.0)	117.0 (106.3–130.0)	130.0 (115.0–143.0)	127.0 (114.0–141.0)	123.0 (110.0–138.0)
DBP (mmHg)	65.0 (57.0–74.0)	70.0 (63.0–78.7)	65.0 (58.0–72.0)	68.0 (60.0–78.0)	67.0 (60.0–75.0)	67.0 (59.0–75.0)
SpO_2 (%)	95.5 (94.0–97.0)	95.0 (94.0–97.0)	95.0 (94.0–97.0)	94.0 (91.0–96.0)	96.0 (94.0–97.0)	94.0 (91.0–95.5)
FiO_2 (%)	21.0 (21.0–21.0)	21.0 (21.0–21.0)	21.0 (21.0–21.0)	29.0 (21.0–54.0)	21.0 (21.0–21.0)	35.0 (21.0–49.0)
TEMP (°C)	36.3 (36.0–36.7)	36.4 (36.1–36.8)	36.3 (36.0–36.7)	36.4 (36.1–36.9)	36.5 (36.2–36.9)	36.3 (36.0–36.7)
AVPU	1.0 (1.0–1.0)	1.0 (1.0–1.0)	1.0 (1.0–1.0)	1.0 (1.0–1.0)	1.0 (1.0–1.0)	1.0 (1.0–1.0)
Static						
Age (years)	78.0 (69.2–84.0)	73.0 (63.0–81.0)	71.0 (58.0–80.0)	76.5 (69.0–85.0)	70.0 (58.0–78.0)	72.5 (64.0–81.0)
Male, n (%)	145 (69.05%)	356 (45.82%)	301 (50.00%)	114 (56.44%)	196 (51.99%)	186 (47.45%)
Elective, n (%)	30 (14.29%)	210 (27.03%)	178 (29.57%)	8 (3.96%)	124 (32.89%)	110 (28.06%)
Surgical, n (%)	40 (19.05%)	245 (31.53%)	201 (33.39%)	31 (15.35%)	99 (26.26%)	65 (16.58%)
Hematological						
HGB (g/L)	9.9 (8.8–11.4)	11.4 (10.0–13.0)	11.4 (9.7–12.9)	11.0 (9.5–12.7)	11.1 (9.8–12.7)	11.9 (10.4–13.2)
WBC ($\times 10^9$/L)	9.9 (7.4–13.5)	10.0 (7.8–12.9)	9.4 (7.1–12.4)	12.5 (8.8–17.3)	10.1 (7.8–12.9)	10.5 (8.1–13.8)
EOS ($\times 10^9$/L)	0.1 (0.0–0.2)	0.1 (0.0–0.2)	0.1 (0.0–0.2)	0.0 (0.0–0.1)	0.2 (0.1–0.4)	0.1 (0.0–0.2)
BAS ($\times 10^9$/L)	0.0 (0.0–0.1)	0.0 (0.0–0.1)	0.0 (0.0–0.1)	0.0 (0.0–0.1)	0.1 (0.0–0.1)	0.0 (0.0–0.1)
EBR	2.8 (0.8–5.8)	2.3 (0.8–4.6)	2.5 (0.9–5.4)	0.8 (0.0–3.5)	4.5 (2.3–7.6)	2.0 (0.5–4.5)
NEU ($\times 10^9$/L)	7.6 (5.5–10.8)	7.6 (5.4–10.5)	6.9 (4.8–9.7)	10.4 (6.6–14.6)	7.2 (5.1–9.9)	8.2 (6.0–11.5)
LYM ($\times 10^9$/L)	0.9 (0.6–1.4)	1.2 (0.8–1.6)	1.3 (0.9–1.8)	0.9 (0.5–1.4)	1.4 (1.0–2.0)	1.0 (0.7–1.5)
NLR	8.7 (4.7–16.3)	6.4 (3.8–10.6)	5.3 (3.2–9.1)	12.9 (6.4–22.3)	4.8 (3.1–8.1)	7.7 (4.8–13.4)
Serum						
ALB (g/L)	23.0 (18.0–28.0)	27.0 (23.0–32.0)	26.0 (22.0–30.0)	24.0 (19.0–28.0)	27.0 (23.8–31.0)	26.0 (21.0–30.0)
CR (μmol/L)	210.0 (138.0–289.0)	71.0 (58.0–91.0)	72.0 (57.0–92.0)	72.0 (48.0–105.0)	71.0 (56.0–87.0)	75.0 (56.0–107.0)
CRP (mg/L)	71.8 (31.0–149.0)	56.0 (17.1–136.4)	47.0 (12.8–104.8)	126.7 (57.8–213.7)	63.5 (20.9–137.4)	62.5 (25.6–138.1)
POT (mmol/L)	4.3 (3.9–5.0)	4.0 (3.6–4.4)	4.0 (3.6–4.3)	3.9 (3.5–4.3)	4.0 (3.6–4.3)	4.1 (3.7–4.6)
SOD (mmol/L)	136.0 (133.0–140.0)	137.0 (134.0–140.0)	137.0 (134.0–140.0)	138.0 (135.0–142.0)	138.0 (135.0–139.0)	138.0 (134.0–140.0)
UR (mmol/L)	16.1 (10.7–23.6)	5.8 (4.2–8.3)	5.9 (4.2–8.7)	7.3 (4.5–11.3)	5.1 (3.7–6.8)	6.9 (4.8–11.1)

Table B.2: Summary statistics of vital-sign, laboratory and static input variables stratified by clusters identified by CAMELOT in the joint training and validation data of HAVEN-T2RF. Continuous variables are reported as median (interquartile range, IQR). Binary/count variables are reported as a count (%). Cluster sizes are shown in the first row.

B.1.3 Mean Trajectory Visualizations

Figure B.1 displays the mean RR and SpO_2 trajectories for each cluster identified by CAMELOT in HAVEN-T2RF (training and validation data). These variables were identified as potential primary drivers of cluster assignment in Figure 5.4. Trajectories are aligned relative to outcome time and displayed over a window of up to 72 hours prior to the outcome time, with the shaded region indicating one standard error around the cluster mean.

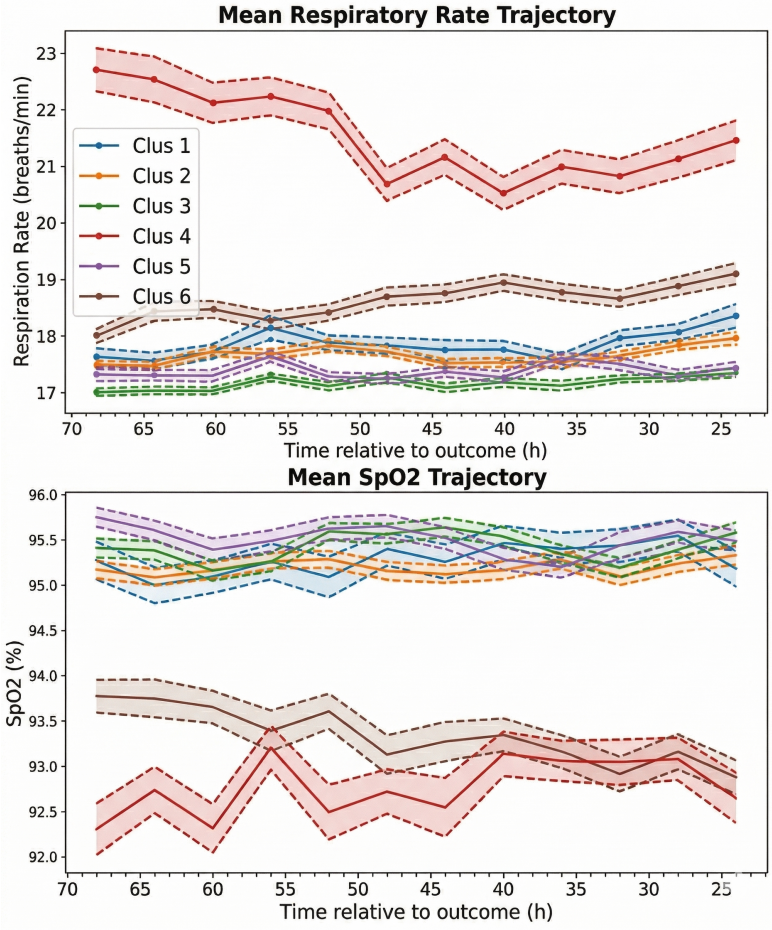


Figure B.1: Mean trajectories of respiratory rate (RR; breaths min^{-1}) and blood oxygen saturation (SpO_2; %) for each cluster identified by CAMELOT in the joint training and validation data of HAVEN-T2RF. Solid lines denote cluster means; dashed lines indicate ± 1 standard error. Trajectories are shown for up to 72 hours prior to outcome time.

B.2 MIMIC-IV-ED

Although HAVEN-T2RF was the primary dataset used to analyze the clinical relevance of CAMELOT, results for MIMIC-IV-ED are included for completeness. Specifically, empirical cluster outcome counts and feature–time relevance maps are reported. As discussed in the main text, HAVEN-T2RF is more suitable for detailed phenotypic analysis, so the translation of other phenotypic outputs to clinical interpretation is less relevant.

Table B.3 reports the empirical number of admissions per outcome class within each cluster (5 in total) learned by CAMELOT in MIMIC-IV-ED. Furthermore, Figure B.2 displays the corresponding attention maps learned by CAMELOT on this dataset.

Cluster	De	I	W	Di	Total
Cluster 1	2	48	141	3	194
Cluster 2	1	161	475	7	644
Cluster 3	14	173	242	1	430
Cluster 4	0	97	280	2	379
Cluster 5	1	89	539	22	651

Table B.3: Empirical outcome frequencies within each cluster identified by CAMELOT in the MIMIC-IV-ED dataset. Outcome labels are abbreviated as follows: De (Death), I (ICU admission), W (Ward admission), and Di (Discharge). Cluster assignments follow the inference rule of Equation (5.2), applied to the combined training and validation data.

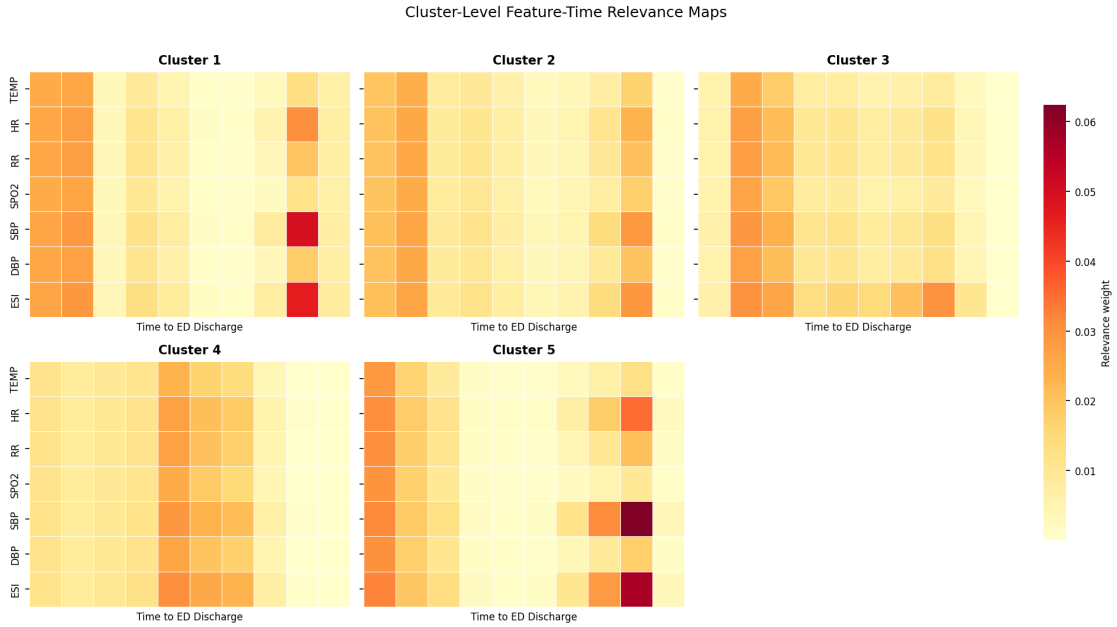


Figure B.2: Cluster-level feature–time relevance maps $\mathbf{M}^k$ learned by CAMELOT in MIMIC-IV-ED, shown as normalized heatmaps. Columns correspond to time steps and rows to vital-sign features. Warmer colors indicate higher relevance for cluster assignment. Maps are L_1-normalized to sum to one across all feature–time entries.

C

Supplementary Results Chapter 6

This appendix provides supplementary material for Chapter 6, including a restatement and proof of the DeepGC training objective and additional results.

C.1 Proof of the DeepGC ELBO

Lemma 7 (ELBO for DeepGC, restated) *The log-likelihood* $\log p(\mathbf{X}_n, \mathbf{y}_n^{mc})$ *is lower-bounded by*

$$\mathcal{L}_{DGC} \coloneqq \mathbb{E}_{\boldsymbol{\pi}^{DGC}_{n,\leq T'} \sim Q} \left[\log p\big(\mathbf{y}_n^{mc} \mid \mathbf{z}_{n,T'}^{DGC}\big) + \sum_{t=1}^{T'} \Big[\log f_g(t) - D_{KL}(q(t) \| f_p(t)) \Big] \right]$$

Proof 6 *For a fixed patient* n*, define the augmented sequence*

$$\mathbf{X}'_n \coloneqq \big\{ \mathbf{x}'_{n,t} \big\}_{t=1}^{T'},$$

where

$$\mathbf{x}'_{n,t} \coloneqq \begin{cases} \mathbf{x}_{n,t}, & t \leq T, \\ \hat{\mathbf{x}}_{n,t}^{DGC}, & T < t < T', \\ [\hat{\mathbf{x}}_{n,T'}^{DGC},\ \mathbf{y}_n^{mc}], & t = T'. \end{cases}$$

, that is, the observed sequence is extended with generated observations beyond time T*, and the outcome is absorbed into the final time step.*

Using this augmented representation, the joint likelihood of the observed trajectory and outcome, under DeepGC's augmented autoregressive model extending to time T', can be written as:

$$p(\mathbf{X}_n, \mathbf{y}_n^{mc}) = \int p\big(\mathbf{X}'_n \mid \boldsymbol{\pi}^{DGC}_{n,\leq T'}\big)\ \mathrm{d}\boldsymbol{\pi}^{DGC}_{n,\leq T'} \tag{C.1}$$

Now, let q' denote the variational family defined over the augmented sequence $\mathbf{X}'_n$, and write

$$Q' \coloneqq q'\big(\boldsymbol{\pi}^{DGC}_{n,\leq T'} \mid \mathbf{X}'_n\big)$$

By applying the standard variational lower-bound argument used for VRNNs to the augmented sequence representation, one obtains

$$\log p(\mathbf{X}_n, \mathbf{y}_n^{mc}) \geq \mathbb{E}_{\boldsymbol{\pi}^{DGC}_{n,\leq T'} \sim Q'}\left[\sum_{t=1}^{T'}\Big[\log p\big(\mathbf{x}'_{n,t} \mid \boldsymbol{\pi}^{DGC}_{n,t}, \mathbf{h}^{DGC}_{n,t-1}\big) - D_{KL}(q'(t)\|f_p(t))\Big]\right]$$

For all $t < T'$, the augmented sequence coincides with the original observed or generated sequence, so the corresponding reconstruction terms retain the same form. Therefore, it remains only to analyze the final term at $t = T'$. By construction,

$$\begin{aligned}
&\log p\big(\mathbf{x}'_{n,T'} \mid \boldsymbol{\pi}^{DGC}_{n,T'}, \mathbf{h}^{DGC}_{n,T'-1}\big) \\
&\quad = \log p\big(\mathbf{x}_{n,T'}, \mathbf{y}_n^{mc} \mid \boldsymbol{\pi}^{DGC}_{n,T'}, \mathbf{h}^{DGC}_{n,T'-1}\big) \\
&\quad = \log p\big(\mathbf{x}_{n,T'} \mid \boldsymbol{\pi}^{DGC}_{n,T'}, \mathbf{h}^{DGC}_{n,T'-1}\big) + \log p\big(\mathbf{y}_n^{mc} \mid \boldsymbol{\pi}^{DGC}_{n,T'}, \mathbf{h}^{DGC}_{n,T'-1}\big)
\end{aligned} \tag{C.2}$$

where the factorization follows from the modeling assumption that, conditional on $\boldsymbol{\pi}^{DGC}_{n,T'}$ and $\mathbf{h}^{DGC}_{n,T'-1}$, the final observation $\mathbf{x}_{n,T'}$ and outcome $\mathbf{y}_n^{mc}$ are independent.

By the architecture of DeepGC, the second term in Equation (C.2) can be modeled, with a slight abuse of notation, as

$$\log p\big(\mathbf{y}_n^{mc} \mid \mathbf{z}^{DGC}_{n,T'}\big)\,,$$

where dependence on the cluster representation parameters is implicit in the model definition. Under the restriction that the variational family conditions only on the trajectory component of the augmented sequence (as the DeepGC architecture assumes), the KL

term retains the same analytical form. Thus, substituting the above decomposition yields the following lower-bound:

$$\mathcal{L}_{DGC} = \mathbb{E}_{\boldsymbol{\pi}^{DGC}_{n,\leq T'} \sim Q}\left[\log p(\mathbf{y}^{mc}_n \mid \mathbf{z}^{DGC}_{n,T'}) + \sum_{t=1}^{T'}\Big[\log f_g(t) - D_{KL}(q(t)\|f_p(t))\Big]\right],$$

which coincides with the ELBO expression in Equation (6.20).

C.2 Cluster Transition Structure in HAVEN-T2RF

Figure C.1 presents a Sankey diagram that illustrates the trajectories of cluster transitions learned by DeepGC in HAVEN-T2RF. Five representative time points are shown: admission start ($t = 0$), $t = 16$ hours, $t = 32$ hours, $t = T = 48$ hours, and $t = T' = T + 24$ hours, where the final time point is obtained through the autoregressive generation procedure of DeepGC.

At each time point, patients are assigned to clusters based on the maximum posterior assignment probability, extracted from the outputs of DeepGC. The diagram provides a summary of how cluster memberships evolve over time and complements the trajectory-level results presented in Figure 6.3.

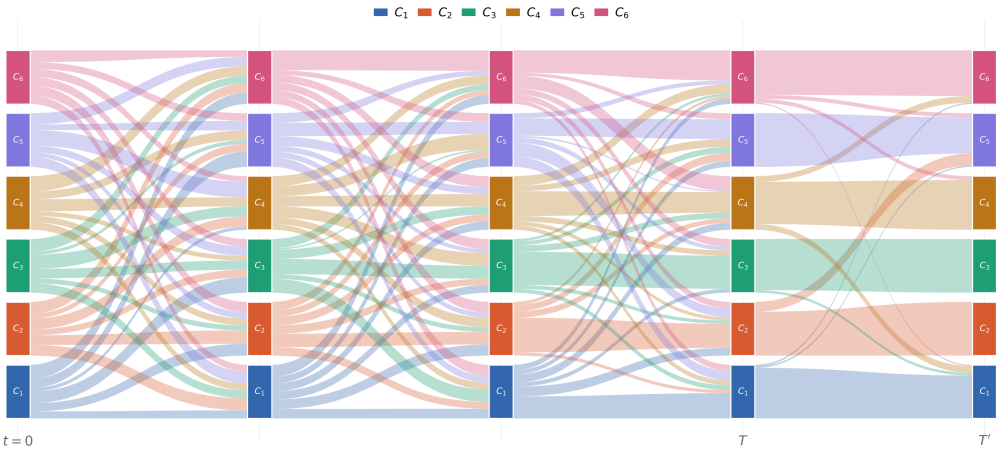


Figure C.1: Sankey diagram of cluster transitions learned by DeepGC in the HAVEN-T2RF dataset. Five representative time points are shown: $t = 0$ (admission start), $t = 16$ hours, $t = 32$ hours, $t = T$, and $t = T'$. At each time point, each patient is assigned to the cluster with highest posterior probability. Flows represent changes in hard cluster assignment across the selected time points.

C.3 DeepGC results in MIMIC-IV-ED

Similarly to Section B.2, the clinical interpretation of DeepGC is mainly conducted using the HAVEN-T2RF dataset, which provides a richer and more consistent phenotypic structure. For completeness, the corresponding results for MIMIC-IV-ED are reported here.

Cluster	De	I	W	Di	Total
Cluster 1	1	39	71	3	114
Cluster 2	15	251	505	3	774
Cluster 3	0	11	162	5	178
Cluster 4	0	134	610	17	761
Cluster 5	2	133	329	7	471

Table C.1: Empirical outcome counts within each cluster identified by DeepGC. Outcome labels are abbreviated as follows: De (Death), I (ICU admission), W (Ward admission), and Di (Discharge). Cluster assignments are determined using the maximum posterior assignment probability. The final column reports the total number of admissions per cluster.

Table C.1 summarizes the empirical number of admissions per outcome class within each cluster, where cluster assignments are defined at the final time step. Consequently, Figure C.2 shows the mean cluster assignment trajectory (Section 6.5.2) of the learned clusters over time, while Figure C.3 displays a Sankey diagram of cluster transitions at $t = 0, 3, 6, 9$, where cluster assignments are estimated at each time step.

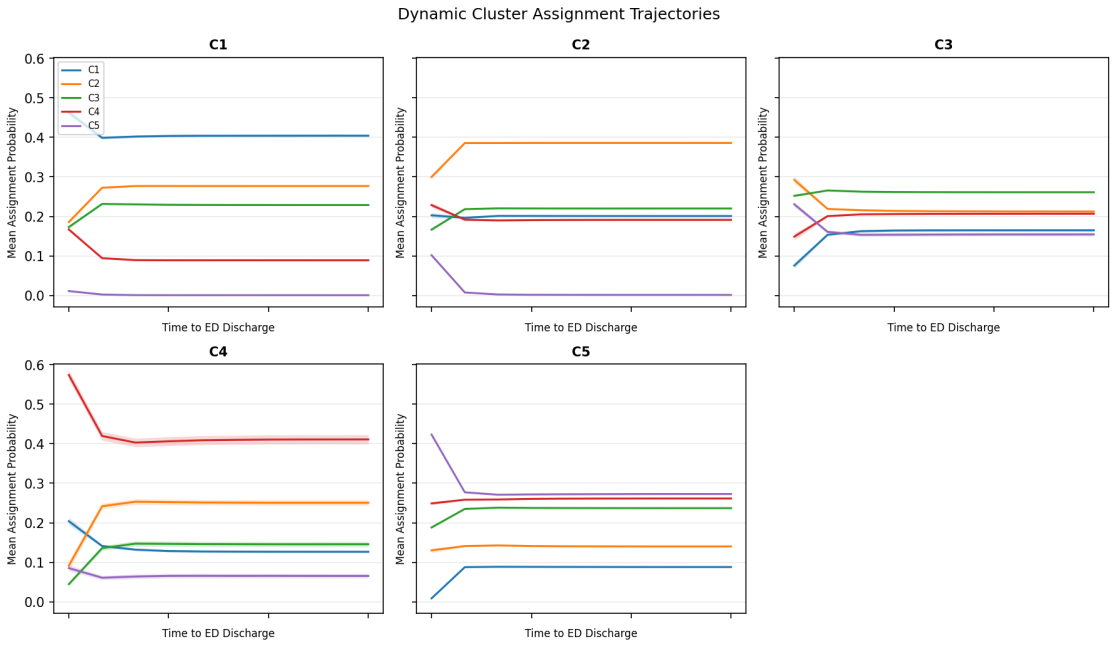


Figure C.2: The mean cluster assignment trajectories learned by DeepGC in the MIMIC-IV-ED dataset. $K = 5$ subplots are shown, each corresponding to one cluster, defined according to the final-time-step assignment ($t = 10$). Within each subplot, trajectories represent the average posterior assignment probability to the corresponding cluster over time. Shaded regions indicate ± 1 standard error.

Compared to HAVEN-T2RF, cluster assignments in MIMIC-IV-ED exhibit reduced temporal variability, with most changes occurring in the initial time steps. This behavior is consistent with expectations. First, the shorter temporal window in MIMIC-IV-ED (maximum length of 10 hours in the processed cohort) limits the extent to which distinct temporal patterns can emerge. Second, the outcome definitions in MIMIC-IV-ED are less directly reflective of the underlying physiological trajectories, which can reduce the separability of cluster-specific phenotypes and their temporal transitions.

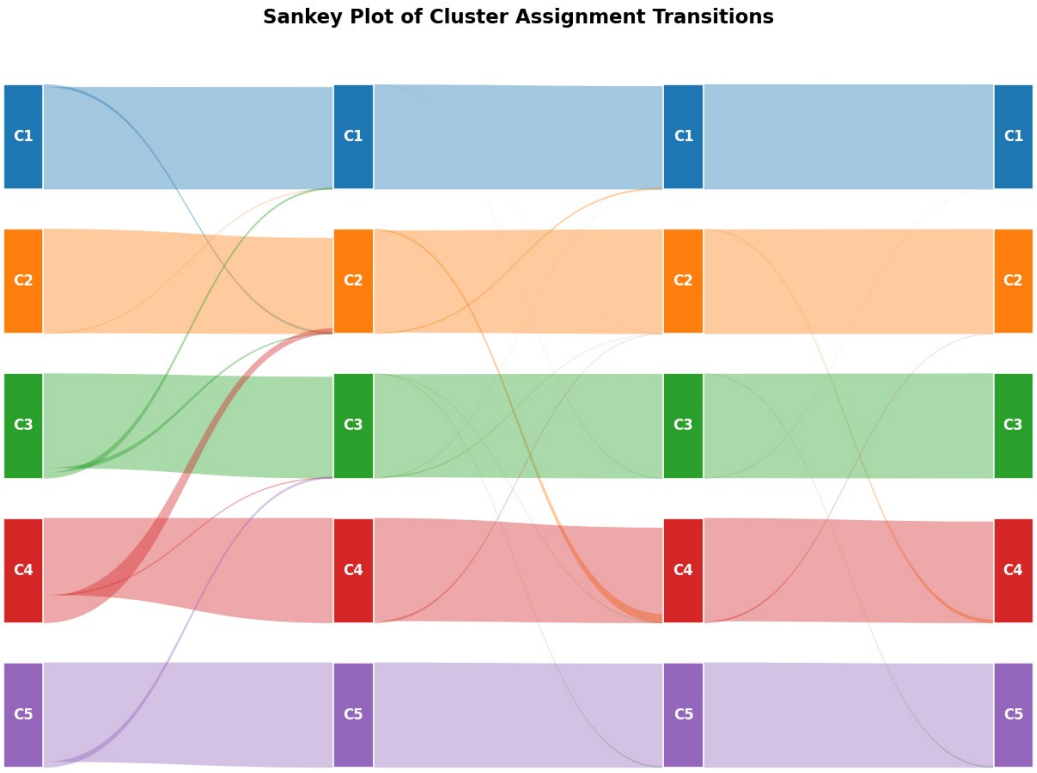


Figure C.3: Sankey diagram illustrating cluster transitions inferred by DeepGC in the MIMIC-IV-ED dataset. 4 representative time points are shown: $t = 0$ (admission start), $t = 3$ hours, $t = 6$ hours, and $t = 9$. At each time point, each patient is assigned to the cluster with highest posterior probability. Flows represent changes in hard cluster assignment across the selected time points.