



University of Kentucky
UKnowledge

Theses and Dissertations--Computer Science

Computer Science


2023

Machine Learning Framework for Real-World Electronic Health Records Regarding Missingness, Interpretability, and Fairness

Jing Lucas Liu

University of Kentucky, lucas.jingliu@gmail.com

Author ORCID Identifier:

 <https://orcid.org/0000-0003-2890-9673>

Digital Object Identifier: <https://doi.org/10.13023/etd.2023.156>

[Right click to open a feedback form in a new tab to let us know how this document benefits you.](#)

Recommended Citation

Liu, Jing Lucas, "Machine Learning Framework for Real-World Electronic Health Records Regarding Missingness, Interpretability, and Fairness" (2023). *Theses and Dissertations--Computer Science*. 131. https://uknowledge.uky.edu/cs_etds/131

This Doctoral Dissertation is brought to you for free and open access by the Computer Science at UKnowledge. It has been accepted for inclusion in Theses and Dissertations--Computer Science by an authorized administrator of UKnowledge. For more information, please contact UKnowledge@lsv.uky.edu.

STUDENT AGREEMENT:

I represent that my thesis or dissertation and abstract are my original work. Proper attribution has been given to all outside sources. I understand that I am solely responsible for obtaining any needed copyright permissions. I have obtained needed written permission statement(s) from the owner(s) of each third-party copyrighted matter to be included in my work, allowing electronic distribution (if such use is not permitted by the fair use doctrine) which will be submitted to UKnowledge as Additional File.

I hereby grant to The University of Kentucky and its agents the irrevocable, non-exclusive, and royalty-free license to archive and make accessible my work in whole or in part in all forms of media, now or hereafter known. I agree that the document mentioned above may be made available immediately for worldwide access unless an embargo applies.

I retain all other ownership rights to the copyright of my work. I also retain the right to use in future works (such as articles or books) all or part of my work. I understand that I am free to register the copyright to my work.

REVIEW, APPROVAL AND ACCEPTANCE

The document mentioned above has been reviewed and accepted by the student's advisor, on behalf of the advisory committee, and by the Director of Graduate Studies (DGS), on behalf of the program; we verify that this is the final, approved version of the student's thesis including all changes required by the advisory committee. The undersigned agree to abide by the statements above.

Jing Lucas Liu, Student

Dr. Jin Chen, Major Professor

Dr. Simone Silvestri, Director of Graduate Studies

MACHINE LEARNING FRAMEWORK FOR REAL-WORLD ELECTRONIC
HEALTH RECORDS REGARDING MISSINGNESS, INTERPRETABILITY,
AND FAIRNESS

DISSERTATION

A dissertation submitted in partial
fulfillment of the requirements for
the degree of Doctor of Philosophy
in the College of Engineering at the
University of Kentucky

By
Jing (Lucas) Liu
Lexington, Kentucky

Director: Dr. Jin Chen, Professor of Computer Science
Lexington, Kentucky 2023

Copyright© Jing (Lucas) Liu 2023

ABSTRACT OF DISSERTATION

MACHINE LEARNING FRAMEWORK FOR REAL-WORLD ELECTRONIC HEALTH RECORDS REGARDING MISSINGNESS, INTERPRETABILITY, AND FAIRNESS

Machine learning (ML) and deep learning (DL) techniques have shown promising results in healthcare applications using Electronic Health Records (EHRs) data. However, their adoption in real-world healthcare settings is hindered by three major challenges. Firstly, real-world EHR data typically contains numerous missing values. Secondly, traditional ML/DL models are typically considered black-boxes, whereas interpretability is required for real-world healthcare applications. Finally, differences in data distributions may lead to unfairness and performance disparities, particularly in subpopulations.

This dissertation proposes methods to address missing data, interpretability, and fairness issues. The first work proposes an ensemble prediction framework for EHR data with large missing rates using multiple subsets with lower missing rates. The second method introduces the integration of medical knowledge graphs and double attention mechanism with the long short-term memory (LSTM) model to enhance interpretability by providing knowledge-based model interpretation. The third method develops an LSTM variant that integrates medical knowledge graphs and additional time-aware gates to handle multi-variable temporal missing issues and interpretability concerns. Finally, a transformer-based model is proposed to learn unbiased and fair representations of diverse subpopulations using domain classifiers and three attention mechanisms.

KEYWORDS: Machine Learning, Deep Learning, Artificial intelligence, Electronic Health Records

Author's signature: Jing (Lucas) Liu

Date: May 3, 2023

MACHINE LEARNING FRAMEWORK FOR REAL-WORLD ELECTRONIC
HEALTH RECORDS REGARDING MISSINGNESS, INTERPRETABILITY,
AND FAIRNESS

By
Jing (Lucas) Liu

Director of Dissertation: Jin Chen

Director of Graduate Studies: Simone Silvestri

Date: May 3, 2023

Dedicated to the memory of my grandfather and grandmother,
who have always inspired me, stood by me, and provided support and strength
throughout my education and growth.

ACKNOWLEDGMENTS

Undertaking a PhD has been one of the best decisions I've taken, it has been a challenging yet fun journey.

I would like to express my sincerest gratitude to my advisor Dr. Jin Chen for his continuous support and guidance in helping me succeed in my studies. Dr.Chen has always been incredibly supportive from the very moment I joined the research group. He has given me lots of opportunities to explore various projects to shape my research interests. He has always motivated and encouraged me to rise and to exceed my expectations academically. The completion of my Ph.D. study could not be possible without his patience and guidance.

I would also like to thank Dr. Zongming Fei, Dr. Qiang Ye, and Dr. Mirek Truszczyński for being on my committee and for valuable feedback and time. Their expertise and suggestions helped me to refine my work.

I would also like to extend my deep gratitude to Dr. Javier Neyra, my mentor and a good friend, for his support and guidance. Dr. Neyra provided invaluable clinical insights into this dissertation and gave me a lot of advice for my research and career in general.

I am deeply thankful to my first-year advisor, Dr. Licong Cui, for taking me into the fields of biomedical informatics and computer science research. Dr. Cui has always been patient in guiding and inspiring me with insightful suggestions towards the right directions. Her support and encouragement have played a significant role in igniting my passion for research.

I would also like to thank Dr. Bin Huang, Dr. Hartmut Malluche, Dr. Barbara Nikolańczyk for their support and collaboration on various projects and for generously sharing their expertise and insights with me.

Special thanks to my former and current lab mates, including Rashmie Abeysinghe, Fengbo Zheng, Xufeng Qu, Yuanyuan Wu, Victor Ortiz-Soriano, Sajjad Fouladvand, Md Selim, Qi Qiao, and many other friends I might've missed. Working, chatting, and acting silly with you all was really fun.

Last but not least, I would like to thank my family, especially my grandpa and grandma for their endless and unconditional love. They have always believed in me and supported me every step of the way for pursuing my dreams. I am forever grateful. I would also like to thank my loving then-girlfriend and now-spouse, Min, for being the warmest sunshine in my life.

TABLE OF CONTENTS

Acknowledgments	iii
List of Tables	viii
List of Figures	x
CHAPTER 1 Introduction	1
1.1 Motivation	1
1.2 Contributions	3
1.3 Dissertation Outline	6
CHAPTER 2 Background	7
2.1 Electronic Health Record Systems (EHR)	7
2.2 Representative Machine Learning and Deep Learning Methods	8
2.2.1 XGBoost	8
2.2.2 Long short-term memory	8
2.3 Approaches for Missingness	9
2.4 Approaches for Interpretability	11
2.4.1 Self-Interpretable Machine Learning Methods	11
2.4.2 Post-hoc Interpretable Machine Learning Methods	12
2.4.3 Attention-based Interpretable Deep Learning Methods	13
2.4.4 Knowledge-Graph Guided Interpretable Deep Learning Methods	14
2.5 Approaches for Fairness	15
2.5.1 Fairness in Healthcare AI	15
2.5.2 Domain Adaptation	15
CHAPTER 3 ELMV: an Ensemble-Learning Approach for Analyzing Electronic Health Records with Significant Missing Values	18
3.1 Introduction	18
3.2 Methods	19
3.2.1 ELMV Stage 1. Predictive Model Generation	20
3.2.2 ELMV Stage 2. Ensemble Prediction	25
3.2.3 ELMV Stage 3. Critical Feature Identification	28
3.3 Experiments Settings	30
3.4 Results	33
3.5 Conclusion	39
CHAPTER 4 KGDAL: Knowledge Graph Guided Double Attention LSTM for Rolling Mortality Prediction for AKI-D Patients	40
4.1 Introduction	40
4.2 Method	42
4.2.1 Phase 1. Data Extraction	43

4.2.2	Phase 2. Knowledge-Graph Extraction	44
4.2.3	Phase 3. Knowledge-Graph Guided Double Attention	46
4.3	Experiment Settings	51
4.3.1	Data Preprocessing	51
4.4	Results	54
4.5	Conclusion	62
CHAPTER 5	KIT-LSTM: Knowledge-guided Time-aware LSTM for Continuous Risk Prediction for Acute Kidney Injury Patients Requiring Dialysis	63
5.1	Introduction	63
5.2	Method	67
5.2.1	Notations	67
5.2.2	KIT-LSTM Cell	68
5.2.3	Patient Outcome Prediction	71
5.3	Experiment Settings	72
5.3.1	Experiment Data	73
5.3.2	Baseline Algorithms	74
5.3.3	Model Robustness Evaluation Metric	75
5.4	Results	77
5.4.1	Ablation Study	79
5.4.2	Model Validation and Interpretation	79
5.5	Conclusion	81
CHAPTER 6	MTATE: Unbiased Representation of Electronic Health Records for Patient Outcome Prediction	82
6.1	Introduction	82
6.2	Background	86
6.3	Method	87
6.3.1	Notations	87
6.3.2	Temporal Relevance Attention	88
6.3.3	Domain-specific Feature Relevance Attention	89
6.3.4	Domain Classifier	90
6.3.5	Domain-focused Representation	91
6.3.6	Patient Outcome Prediction	92
6.4	Experiments Settings	93
6.4.1	Prediction Tasks	93
6.4.2	Experiment Data	94
6.4.3	Baseline Models	96
6.4.4	Performance Metrics	96
6.5	Results and Discussion	98
6.5.1	Performance Comparison	98
6.5.2	Ablation Study	101
6.5.3	Assessment of Data Representation	102
6.5.4	Effectiveness Assessment of RW-Attention	102

6.5.5	Impact of Masking Rate	104
6.6	Conclusion	105
CHAPTER 7	Conclusion and Future Direction	106
7.1	Conclusion	106
7.2	Future Direction	108
7.2.1	Transfer and Federated Learning for Transportable Healthcare AI	108
7.2.2	Topic Modeling for EHR Data Harmonization	109
7.2.3	Multi-modality Models for Personalized Medicine	109
	Bibliography	111
	Vita	126

LIST OF TABLES

3.1	Definition of hyperparameters used in ELMV.	29
3.2	Averaged accuracy of ELMV, XGBoost, and two imputation methods on the simulation data with low (above the horizontal line) or high missing rates (below the horizontal line).	33
3.3	Averaged precision of ELMV, XGBoost, and two imputation methods on the simulation data with low (above the horizontal line) or high missing rates (below the horizontal line).	35
3.4	Averaged recall of ELMV, XGBoost, and two imputation methods on the simulation data with low (above the horizontal line) or high missing rates (below the horizontal line).	36
3.5	Averaged F-1 of ELMV, XGBoost, and two imputation methods on the simulation data with low (above the horizontal line) or high missing rates (below the horizontal line).	36
3.6	Performance of qualified maximal subsets of the LRYGB data with different missing rates.	37
3.7	Average computational time comparison among ELMV, Mean imputation, MICE imputation (two iterations, and two multiple imputations), and XGboost.	38
3.8	Accuracy of ELMV and kNN on the LRYGB data.	39
4.1	Training, validation and testing data at the subsequence level.	54
4.2	Performance of morality prediction in the next 72 hours during RRT (UK data)	57
4.3	Performance of morality prediction in the next 48 hours during RRT (UK Data)	58
4.4	Performance of morality prediction in the next 24 hours during RRT (UK Data)	58
4.5	Performance of morality prediction in the next 72 hours during RRT (MIMIC-III)	58
4.6	Performance of morality prediction in the next 48 hours during RRT (MIMIC-III)	59
5.1	Training, Validation and Testing Data.	73
5.2	Overall Performance of KIT-LSTM and seven compared algorithms on balanced test sets	75
5.3	Overall Performance of KIT-LSTM and seven compared algorithms on imbalance test sets	76
5.4	Balanced performance of KIT-LSTM and seven compared algorithms on multiple subpopulation levels (pos:neg = 1:1).	76
5.5	Imbalanced performance of KIT-LSTM and compared algorithms on multiple subpopulation levels (pos:neg = 1:9).	76
5.6	Ablation study: Performance of KIT-LSTM variants on balanced data at level 1, 2 and 5 subpopulations.	78

6.1	Training, validation, and testing data. (Rolling mortality prediction on the proprietary dataset)	95
6.2	Training, validation, and testing data. (In-hospital mortality prediction on MIMIC3 dataset)	96
6.3	Performance comparison on rolling mortality prediction in the next 72 hours for proprietary imbalanced test data (pos:neg=1:4). DPD, EOD, and EQOD are the lower the better.	98
6.4	Balanced performance of MTATE and compared algorithms for rolling mortality prediction in the next 72 hours for the proprietary test data (pos:neg = 1:1).	99
6.5	Performance comparison on in-hospital mortality prediction for MIMIC3 imbalanced test data (pos:neg=1:6). DPD, EOD, and EQOD are the lower, the better.	100
6.6	Balanced performance of MTATE and compared algorithms for in-hospital mortality prediction for MIMIC3 dataset (pos:neg = 1:1).	101
6.7	Performance comparison of MTATE and its ablation components for rolling mortality prediction in the next 72 hours for the proprietary test data (pos:neg = 1:4). w/o. DC: remove all domain classifiers. w/o. RW-ATT: remove representation-wise attention. w/o. DC & RW-ATT: remove both domain classifier and representation-wise attention. w/o. Masking: remove masking layers in the FR-Attention module. w/o. L_c : remove contrastive loss.	102

LIST OF FIGURES

1.1	An overview of the methods introduced in the dissertation	4
3.1	Overall framework of ELMV. It includes three stages: predictive model generation, ensemble prediction, and critical feature identification.	20
3.2	The framework of critical feature identification using ELMV.	28
3.3	In the LRYGB follow-up study, the distribution of the missing values of all the 24 variables at six time points. In general, more values are missing towards the end of the follow-up study. Red indicates higher missing ratio towards 100%, green is for lower missing ratio towards 0%, and black indicates 50% missing ratio.	32
3.4	The moving average of accuracy of ELMV, XGBoost, and two imputation methods on the simulation data with missing rate increasing from 60% to 70%.	34
3.5	The moving average of F-1 Scores of ELMV, XGBoost, and two imputation methods on the simulation data with missing rate increasing from 60% to 70%.	35
3.6	The distribution of all the maximal subset of the original LRYGB data with 78 features and 202 T2DM patients. Every point represents a maximal subset with x number of patients and y number of features. Color indicates different missing rates.	37
4.1	The three phases of the Knowledge-Graph Guided Double Attention (KG-DAL) model for rolling mortality prediction for critically ill patients with AKI-D in real-world healthcare settings.	43
4.2	The detailed architecture of the Knowledge-Graph Guided Double Attention LSTM (KGDAL) model.	48
4.3	A: The partial hierarchical structure of the Human Phenotype Ontology (HPO) that includes the following concepts. Colors indicate different concept groups (Red: Acute Kidney injury (AKI); Orange: "Cardiovascular"; Green: "Metabolism"; Blue: "Blood"; Pink: "Respiratory"). B: The similarities of the same selected features in a projected space generated using t-SNE.	55
4.4	Two risk trajectory clusters with different endings.	60
4.5	The risk trajectory of a survival patient.	61
4.6	An example of the KG-adjusted 2-D attentions.	61
5.1	An example of real-world EHR data in the ICU. "SBP" stands for systolic blood pressure, "HCT" for Hematocrit, and "sCr" for serum creatinine. Arrows highlight irregular and asynchronous gaps between measurements.	64

5.2	The architecture of KIT-LSTM cell (left), orange represents the unique gates in KIT-LSTM, and green represents the original gates in LSTM. The prediction layers (right) combine all the hidden states learned from KIT-LSTM and the static features, such as patient demographics, for the final prediction.	68
5.3	Identified subpopulation using hierarchical clustering. In the clustering dendrogram (left), the horizontal lines show two different levels of subpopulations. The resulting subpopulations at level two (right) are shown on a three-dimensional t-SNE space.	75
5.4	Attention scores for two samples of one patient. RR: respiratory rate; DBP: diastolic blood pressure; ATT: represents attention scores.	78
6.1	Overall framework of masked triple attention transformer encoder (MTATE). HR, SBP, and sCr stand for heart rate, systolic blood pressure, and serum creatinine, respectively. x_t represents all clinical features at time t , f_i represents values of feature i at all time points. Z'_i represents the data representations learned from the i_{th} feature relevance attention module.	85
6.2	Network structure of the masked triple attention transformer (MTATE) algorithm. The TR-attention and domain-specific FR-Attention modules can be stacked N times.	85
6.3	A. The structure of FR-Attention module in MTATE. B. The multi-head attention module from the original Transformer used in MTATE.	91
6.4	Normalized equalized odds difference (EQOD) for every subpopulation domain on rolling mortality prediction in the next 72 hours for the proprietary test data. A low EQOD score indicates high model fairness. The value and color represent the normalized EQOD score (the lower/lighter, the better), and X-axis represents the subpopulation domains, where each domain consists of two subpopulations (e.g., young vs. old in age). CCI, DB, CVD, and CKD stand for Charlson comorbidity score, diabetes, hypertension, cardiovascular disease, and chronic kidney disease, respectively.	99
6.5	The comparison between MTATE with baseline methods for the percentage difference score in PRAUC for each domain. Y-axis represents the percentage difference. X-axis represents the subpopulation domain, each domain consists of two subpopulations (e.g., Young (< 65 y/o) vs. Old in Age domain, Sepsis vs. Non-Sepsis in Sepsis Domain). CCI stands for charlson comorbidity score, DB stands for diabetes, HT stands for hypertension, CVD stands for cardiovascular disease, CKD stands for chronic kidney disease.	100
6.6	Relationship between outcome loss, domain loss and representation-wise attention. Y-axis represents the outcome loss, X-axis represents the domain loss. The colored dots represent representation-wise attention, and the darker color represents higher attention.	103

6.7	Relationship between outcome loss, domain loss and representation-wise attention in all domains. Y-axis represents the outcome loss, x-axis represents the domain loss. The colored dots represent the representation-wise attention, and darker color represents higher attention.	104
6.8	Performance Score of ROCAUC, PRAUC, and Accuracy with different masking rate	105

CHAPTER 1. Introduction

1.1 Motivation

Electronic Health Record (EHR) data stores patients' information such as demographics and medical histories, including diagnoses, procedures, medication, and laboratory test results [1, 2]. Hospitals collect EHR data daily, which has played an essential role in improving patient care, and clinician experience [2, 3, 4].

Accurate and timely prediction of patient risk using EHR data in hospitals can assist clinicians in doing early interventions for high-risk patients and better-using hospital resources. For example, it is important for clinicians to make decisions for patients when intensifying therapies is needed or transitioning patients with a high risk of mortality to comfort care [5, 6].

Given the large volume and rich information stored in EHR systems, machine learning (ML) and deep learning (DL) methods have been explored greatly in many healthcare applications such as predicting patient outcomes or patient risk trajectories as well as identifying risk factors [1, 7, 8]. ML methods such as random forest [9], SVM [10] and gradient boost machine (GBM) [11, 12] are widely used for mortality prediction and length-of-stay (LOS) in hospital [7]. For example, Kong et al. successfully used the GBM model to predict mortality in ICU for sepsis patients [13]; Daghistani et al. used RF, which achieved good performance for the prediction of LOS in hospital [14]. DL methods, including Transformer [15], Long-Short Term Memory (LSTM) [16] and gated recurrent neural networks [17], have shown promising performance on patient outcome prediction using large-scale EHR data in recent years [1]. For example, Doctor AI [18] applies RNN on visit-based medical codes to predict the diagnosis and medication categories for the subsequent visit. Maragatham and Devi used LSTM Model to predict the diagnosis of heart failure using

time-stamped time series data [19].

However, some limitations hinder the usage of ML/DL methods in realistic healthcare settings. We will focus on three significant limitations in this dissertation. Firstly, EHR data in real healthcare settings are usually sparse and contain many missing values [7]. Moreover, the causes of missingness could be intentional. For example, patients might not need a specific type of laboratory test due to relatively healthier conditions [20]; or unintentional, for example, a urine test sample randomly broken, resulting in missing value [21]. Finally, the different measurements frequency of time-stamped data such as vital signs or laboratory variables can also result in missingness in terms of the irregularity of scales and asynchronous multi-variable inputs to the ML/DL model [1, 2].

Directly applying machine learning algorithms on EHR data with many missing values would downgrade the performance [7]. Many imputation methods have been developed to deal with the missing value issue. However, bias prediction and results will be introduced if unsuitable imputation methods are used for different causes of missingness or the pattern of missingness [8, 20]. Since no unique method would be good for data with different missingness patterns, if the missingness pattern between the training and the testing dataset is not considered in the machine learning algorithm, the prediction performance would also be impacted.

The accountability of the models in terms of the interpretation in healthcare practice is critical for clinicians to make the decision based on the model results and rationale [22]. Though ML/DL models have been proven to have outstanding performance, they are usually used as a black box because their algebraic complexity is complex for human comprehension [7]. A helpful model should give good predictions and provide the reason why it makes such predictions. Some traditional machine learning is interpretable such as regression models or decision trees. However, there is a trade-off between performance and interpretability. The interpretable methods

usually have lower prediction performance than the less interpretable methods (e.g., support vector machines, random forest, and neural networks) [22]. Thus, there is a call to develop interpretable ML/DL models with good prediction performance to be better adopted in routine uses in practical healthcare settings [23].

The model’s fairness is the third concern about using ML/DL methods in real healthcare settings. A fair model should not only favor certain groups while failing predictions for other groups. The assumption for getting good performance using ML/DL methods is that training and testing data distribution are the same or similar. Nevertheless, this is not always the case in realistic healthcare settings [7]. The performance would likely drop when the trained model is directly used on another data set [24]. This distribution discrepancy between training and testing data is called domain shifting. The reason causing domain shifting can vary. For example, different geographic locations in which the data set are collected result in different data distribution [25]. Different healthcare settings might use different laboratory feature measurement devices resulting in distribution discrepancies of the corresponding variable [26]. Moreover, how frequently the data are collected and the missingness patterns might also differ across healthcare settings, contributing to the domain shift issue in terms of missingness discrepancies as mentioned above.

1.2 Contributions

This dissertation develops robust ML and DL frameworks to address the concerns of sparsity, interpretability, and fairness. Specifically, the dissertation discusses four methods: 1) an ensemble-learning framework for EHR data with significant missing value; 2) a knowledge-graph guided double attention LSTM framework for the rolling mortality predictions of temporal EHR data. 3) a Knowledge guided Time-aware LSTM model for irregular and asynchronous temporal EHR data; 4) an adaptive multi-task learning algorithm called for learning unbiased and fair data representa-

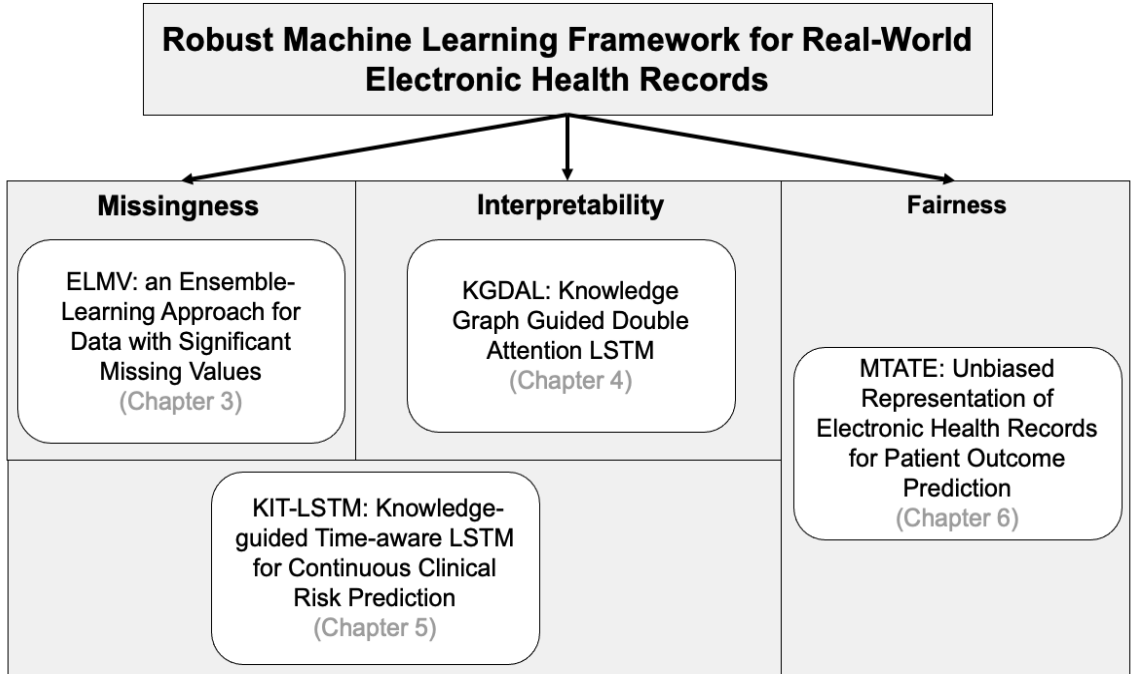


Figure 1.1: An overview of the methods introduced in the dissertation

tions of EHR data. An overview of methods is shown in Figure 1.1. In the following, the contribution of each method is discussed.

The first method (ELMV) addresses the missingness and domain shift issues regarding missingness and feature discrepancy between training and testing data. The general idea of ELMV is that it uses part of the training data to form a support set similar to the testing data and uses the support data to select the ensemble of models for the prediction on testing data. ELMV has the following advantages: 1) It considers the discrepancy between training and testing regarding missingness and critical feature recognition. 2) It is capable of handling substantial missing values. 3) It is adaptable to different datasets and predictive models.

The second method (KGDAL) introduces knowledge-graph guided attention mechanisms for better interpretations. In particular, KGDAL uses the knowledge graph in three ways: First, it uses knowledge graph to guide the group of features; Second, it uses knowledge graph to build the feature attention mechanism automatically on top

of the temporal attention mechanism; Third, it uses knowledge graph to regulate the loss function. KGDAL has the following advantages: 1) It obtains two-dimensional attention in both the time and feature spaces for improved prediction power and enhanced model interpretability. 2) The attention mechanism in the feature space is automatically derived based on the KG rather than manual curation. 3) It can model both continuous and discrete temporal EHR data types. 4) It can make precise rolling mortality predictions for AKI-D patients on two independent clinical datasets.

The third method (KITLSTM) introduces a Knowledge guided Time-aware LSTM model, which handles irregular and asynchronous time series EHR data. It uses medical ontology to guide the attention between multiple numerical clinical variables and provides knowledge-based model interpretation. In particular, KIT-LSTM extends LSTM with two time-aware gates and a knowledge-aware gate. The time-aware gates adjust the memory content according to two types of elapsed time, i.e., the elapsed time since the last visit for all variable streams and the elapsed time since the last measured values for each variable stream. The knowledge-aware gate uses medical ontology to guide attention between multiple numerical variables at each time step. As a result, the proposed model provides better attention and interpretation guidance and handles irregular and asynchronous problems simultaneously.

The fourth method (MTATE) introduces an adaptive multi-task learning algorithm (i.e., Masked Triple Attention Transformer Encoder) to learn and select the optimal and fair data representations automatically. The purpose of MTATE is to generate multiple masked representations of the same data that are attended by both time-wise attention and multiple feature-wise attentions in parallel, where each masked representation corresponds to a specific domain classification task. The learned EHR representations could be domain-specific, domain-invariant, or a mix of the two reflected by the classification loss values. A low loss value indicates the representation is domain-specific, and a high value indicates domain-invariant. The

model will compute the representation-wise attention for each individual testing case, leading to personalized data representation for downstream predictive tasks.

This dissertation uses materials from four papers (three published and one submitted) first authored by the author [27, 28, 29]. Chapter 3 uses materials from Reference [27]. Chapter 4 uses materials from Reference [28]. Chapter 5 uses materials from Reference [29]. Chapter 6 uses materials from an unpublished paper.

1.3 Dissertation Outline

The remainder of the dissertation is organized as follows: Chapter 2 introduces the fundamentals of Electronic Health Records (EHR), traditional machine learning, deep learning methods for longitudinal EHR data, domain adaption, and knowledge graphs. Chapter 3 introduces our first novel machine learning framework (ELMV) for addressing the missing value issues. Finally, chapter 4 introduces the second novel deep learning method (KGDAL) for patient outcome prediction, addressing interpretability concerns. Chapter 5 introduces the third novel deep learning model (KIT-LSTM) for handling irregular and asynchronous issues in temporal EHR data and interpretation concerns. Chapter 6 introduces the fourth novel deep learning method (MTATE) for handling bias and fairness issues in clinical/healthcare AI applications. Finally, Chapter 7 concludes this dissertation by discussing the limitation of the methods and future directions.

CHAPTER 2. Background

This chapter first introduces fundamental medical terminologies used in this dissertation: Electronic Health Records. In addition, two representative machine learning and deep learning methods for EHR data are introduced. Furthermore, this chapter discusses related works which have been done for addressing the three challenges (Missingness, interpretability, and fairness) of ML/DL methods in healthcare settings.

2.1 Electronic Health Record Systems (EHR)

Many hospitals have adopted electronic health records (EHR) to store patients' data such as demographic information, diagnoses, laboratory test [1]. Because of the rich information stored in EHR, they are used for building different clinical/medical applications such as mortality prediction, disease inference/diagnosis, patients' trajectory modeling, clinical decision support system, etc. [7, 2]. Different applications are built using different types of EHR data and various ML/DL algorithms. In general, EHRs can be classified into two categories based on their format. The first category is structured data or semi-structured data such as demographic information, laboratory tests results, diagnosis codes, medications, etc., where the data are stored in tables with fixed or semi-fixed schema; The second category is unstructured data such as clinical notes are stored as free text [2]. In this dissertation, we will focus on structured/semi-structured data types.

Above all, the initial and primary goal of designing EHR was to document patients' medical information and support care in clinical settings [8, 2], which is not designed for answering any specific research questions [25]. Thus, implementation of ML/DL methods using EHR data in real healthcare settings should take into account the nature of the EHR itself [2].

2.2 Representative Machine Learning and Deep Learning Methods

2.2.1 XGBoost

Many machine learning has been deployed in various healthcare applications such as Logistic Regression, Random Forest, Support Vector Machine [8]. A recent ensemble method called eXtreme Gradient Boosting (XGBoost) [12] has shown competitive results in many applications. XGBoost is built based on boosting algorithms, and it combines weak learners into a strong learner for the prediction task [30]. XGBoost is trained by iteratively adding weak learners to reduce the error between the target value and the prediction of the current ensembles of learners. XGBoost has superior performance than other traditional machine learning methods in many applications. However, one limitation is that it is not well-designed for temporal EHR data because it lacks consideration of temporal dependencies.

2.2.2 Long short-term memory

Long short-term memory (LSTM) [16] is a deep learning structure that is designed for sequence data. LSTM uses three gates (forget gate, input gate, and output gate) to control the flow of information along the time. Specifically, LSTM uses forget gate to determine how much of the previous information to discard, an input gate to determine how much of the current information to keep, and an output gate to determine how much of the current information to pass to the future.

The detailed mathematics information is described below: denote the forget gate as \mathbf{f}_t , the input gate as \mathbf{i}_t , and the output gate as \mathbf{o}_t , where $\mathbf{f}_t, \mathbf{i}_t, \mathbf{o}_t \in \mathbb{R}^m$, and m is the dimension of the hidden vectors. Using \mathbf{c}_t and \mathbf{h}_t to represent the cell state and the hidden state, ($\mathbf{c}_t, \mathbf{h}_t \in \mathbb{R}^m$). The LSTM cell is updated as follows:

$$\mathbf{f}_t = \sigma(\mathbf{W}_f \mathbf{h}_{t-1} + \mathbf{U}_f \mathbf{x}_t + \mathbf{b}_f) \quad (2.1)$$

$$\mathbf{i}_t = \sigma(\mathbf{W}_i \mathbf{h}_{t-1} + \mathbf{U}_i \mathbf{x}_t + \mathbf{b}_i) \quad (2.2)$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_o \mathbf{h}_{t-1} + \mathbf{U}_o \mathbf{x}_t + \mathbf{b}_o) \quad (2.3)$$

$$\tilde{\mathbf{c}}_t = \tanh(\mathbf{W}_c \mathbf{h}_{t-1} + \mathbf{U}_c \mathbf{x}_t + \mathbf{b}_c) \quad (2.4)$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{c}}_t \quad (2.5)$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t) \quad (2.6)$$

LSTM structure has provided an effective way of handling temporal information of data. Many variations of the LSTM deep learning model have been developed for different applications. We will introduce them through the rest of the sections.

2.3 Approaches for Missingness

In recent years, techniques have been developed for handling missing values in big data. The simplest and most common strategy is to conduct complete-case analysis (CCA), which refers to removing records with any missing values and focusing only on patients who have the complete records of all parameters [31]. However, in practice, eliminating patients with any missing values will inevitably introduce biases, given that there is often a huge difference between the true distribution of all patients and that of the patients with complete records [32]. In addition, the CCA strategy will significantly reduce the training size regarding inference model training, resulting in models being under-trained.

Another common strategy for handling missing values is data imputation. Imputation techniques can be categorized into two groups: single imputation and multiple imputation [33]. The single imputation refers to replacing a missing value with an estimated value [34]. An example of the simple imputation strategy is the mean imputation [21], where a missing value is replaced with the arithmetic mean. The problem of the simple imputation strategy is that it may significantly underestimate the variance of the data and ignores the complex relationships among explanatory

variables [31]. This problem can be addressed using more sophisticated single imputation methods such as regression imputation and the expectation-maximization (EM) algorithm, in which a missing value is assigned by studying the statistical relationships between the target variable and the rest variables in the same dataset [21]. In contrast, multiple imputation techniques estimate a missing value with multiple imputed data. One such technique is Multivariate Imputation by Chained Equations (MICE), where the statistical uncertainty of different imputed data is taken into account [35].

There is inevitably increasing variability of effect estimates with increased missingness; and results may not be reliable enough for hypothesis validation if more than 40% data are missing in important variables [36, 37, 38, 39, 40], indicating that data imputation is not a go-to solution when a significant portion of the values is missing. Missing data in clinical studies do not occur at random. Certain data points are missing because of patient dropout, treatment toxicity, or biomarkers that are difficult to measure [41]. Applying data imputation algorithms designed for missing-at-random to EHR data may lead to biases in model prediction [42]. Inference models that account for the missing data in real-world EHR data must consider the reasons for missingness [43]. Furthermore, none of the existing imputation method outperforms the others on every dataset, indicating that there are no universal model [21] for missing value imputation.

While most machine learning models can only be applied to complete data or will automatically conduct a complete-cases analysis [21], XGBoost [12], the recent implementation of the gradient boosting model can automatically handle missing values with its built-in mechanisms. Specifically, XGBoost handles the missing data problem by adding a default direction for missing values in each tree splitting. The optimal direction for a missing value in each particular explanatory variable at each tree node is learned during the model training process to minimize the regulated

loss [12]. XGBoost model chooses the default direction if there is no missing value in any particular explanatory variable in training data, but there are missing values in the external validation set. A potential problem in handling missing values in XGBoost is that XGBoost will always choose the default direction for model prediction on the validation set. Thus, the prediction could be a random guess if the missingness patterns in training and validation are entirely different. This could be the case when a large amount of missing value existing especially in the validation data.

Overall, the common problem of existing machine learning approaches is that they do not adapt to handling large missing values. In addition, the discrepancy between training and validation has not been well addressed regarding model inference.

2.4 Approaches for Interpretability

ML/DL methods have shown success in a variety range of prediction tasks. The applications of ML/DL models in clinical settings require a certain level of interpretability. However, the trade-off between performance and interpretability is still a concern in the machine learning community.

2.4.1 Self-Interpretable Machine Learning Methods

Traditional interpretable methods such as logistic/linear regression is still the most widely used method in practice though their performance might not be good as non-interpretable methods such as neural networks [1]. It is quite straightforward to use the coefficient of the regression model to explain the relationship between the outcome and the predictor features in terms of the effect on the outcome of one unit change in the features [22, 44]. However, linear/logistic regression methods are good at modeling linear relationships but insufficient on modeling more complex non-linear relationships. Decision trees are an example of interpretable methods which can better capture the non-linear relationship between outcome and the features.

Decision trees use a set of “if-then” rules along the tree to split the nodes, and it determines the best splits based on the impurity scores of features [45]. The “if-then” rule-based classification algorithm is easy to understand and interpret, one example illustrated in [46] shows that one study only uses seven “if-then” rules to classify 12,586 stroke risk patients with good accuracy.

Self-interpretable methods are easy to understand, but their prediction power is low compared to other complex models such as random forest, or support vector machines, neural networks [44]. Thus, model agnostic interpretable methods are developed for any machine learning models in a post-hoc manner.

2.4.2 Post-hoc Interpretable Machine Learning Methods

One of the popular used model agnostic methods is Local interpretable model-agnostic explanations (LIME) [47], the key idea of this method is that it tries to explain the prediction of each instance by building a simpler model (e.g., linear regression) locally using the samples that are near the instance of interest. LIME uses the simpler surrogate local model to explain why the global model makes such a prediction for the instance of interest. Another widely used model agnostic method is SHapley Additive exPlanations (SHAP) [48] which is based on LIME [47] and game theory model of Shapley values [49, 50, 51]. Shapley values determine the conditional contributions of individual feature to the outcomes by considering all possible combinations of features for one instance, which maximize the difference between the actual prediction and the average prediction of all instances [22]. On the other hand, SHAP explains the prediction of each instance by assigning each feature a Shapley value as an importance score in local space as its done in LIME for the prediction of the outcome.

Post-hoc interpretable methods are supportive for any type of ML/DL method. However, the importance scores provided are just an estimate of feature importance by mimicking how and why the actual ML/DL model makes a particular prediction.

Plus, it needs an extra and separate step from training the actual model. Thus, the recent deep learning models use attention mechanisms to help in terms of interpretation.

2.4.3 Attention-based Interpretable Deep Learning Methods

In this section, we will mainly focus on discussing one type of DL method: recurrent neural networks (RNNs) because of the temporal or sequence nature of EHR data, and RNNs have been particularly extensively used for applications of EHR data [44].

The attention mechanism was initially introduced to address the long-term dependency problem of recurrent neural network (RNN) models [52] on sequence-to-sequence modeling tasks (e.g., language translation task), where additive attention scores are computed for each hidden vector of input word to each target word using a fully connected layer. With the help of attention scores, the model learned which words to pay more attention to in the input sequence for the output sequence. In the following years, multiple other forms of attention mechanisms have been developed using different score formulas [53, 15, 54].

RETAIN [55] was the first study introducing attention mechanism in prediction tasks in healthcare. It uses two RNNs (one in original order, another in reverse order) to generate two-level attention scores: one for the visit level attention and the other for variable level attention. The visit-level attention finds the contributions of each visit (time steps) to the outcome, and the variable level attention finds the contributions of each variable to the outcome. [56] and Dipole [57] are the other two state-of-the-art methods inspired from RETAIN that employ attention mechanism on top of RNN models for prediction task in healthcare using EHR data. As in RETAIN, [56] also generates two attention scores, one for medical code level attention and hospital visit level attention. Instead of using two RNNs in RETAIN, they used one bidirectional Gated Recurrent Units (GRUs). Dipole introduces three attention scores: location-

based scores for only considering one individual hidden state information, general attention scores, and concatenation-based attention scores for considering all previous hidden states on top of the bidirectional GRU which further improve the prediction performance. Zhou et al. proposed an attention mechanism along the time-step dimension for relation classification [58].

2.4.4 Knowledge-Graph Guided Interpretable Deep Learning Methods

The adoption of ML/DL methods in healthcare would be more preferable if the methods take into account both the computational ability and clinical domain knowledge [25]. Even with the self-interpretable or post-hoc methods, the process of feature engineering to select interpretable and critical features with the help of clinical expertise is required. Otherwise, those interpretable methods would still be inexplicable.

Domain-specific knowledge is often encoded in medical or biomedical ontologies databases, which can be used as prior knowledge similar to knowledge from domain experts [59]. Ontology provides a standardized vocabulary of medical/biomedical concepts and their relations. The relations between concepts are usually denoted as "A relation B" meaning A has a particular relation with B (e.g., A is a subclass of B) [60]. Ontology databases usually are organized in hierarchies, where the node represents a concept in a particular domain, and the edge represents the relation between them.

There have been many algorithms developed to extract knowledge from a graph in general by learning the concept and relation embeddings [61, 62]. These algorithms have been used in applications such as medicine recommendation [63, 64], psychiatric disorders patients classification [65]. Recent studies have been introduced to obtain attention scores from concept and relation embeddings from medical knowledge graph (KG) and then use the attentions scores to adjust vector representations of medical codes in EHR for downstream prediction task, GRAM [66], DG-RNN [67]

and MMORE [68].

2.5 Approaches for Fairness

2.5.1 Fairness in Healthcare AI

Fairness is one of the newly emerging focuses for building trustworthy artificial intelligence (AI) models. One of the reasons resulting in an unfair model is the algorithm bias towards different groups of samples. A biased model may benefit certain groups but disfavor other ones. The learned representation by an unfair model could be solely based on protected attributes (e.g., race, gender, etc.). However, they may be biases rather than the essential factor to the outcome. As a result, leaving the bias unresolved might have a significant negative impact, especially in the context of healthcare applications.

Fairness in AI/DL refers to a model’s ability to make a prediction or decision without any bias against any individual or group [69]. The behaviors of a biased model often result in two facets: it performs significantly better in certain populations than others [70], and it makes inequities decisions towards different groups [71]. Clinical decision-making based upon biased predictions may cause delayed treatment plans for patients in minority groups or misspend healthcare resources where treatment is unnecessary [72].

The data distribution shift problem across different domains is one of the major reasons a model could be biased or unfair [73]. Domain shifting is a common issue in real healthcare settings due to different situations when data are collected (e.g., geographic location, healthcare setting or measurement devices, etc.) [25, 26].

2.5.2 Domain Adaptation

To address the fairness issue, the ML community is exploring ways to tackle the domain-shifting challenge. Domain adaptation is a general approach that refers to

learning a model from the source domain that also performs well on a target domain when the distribution of source (e.g., training) and target (e.g., testing) domain is different [24, 74].

Based on the availability of the labels in source and target domain, domain adaptation (DA) approaches can be generally categorized into three types [24]: 1) Unsupervised DA refers to all source data are labeled, and all target data are unlabeled; 2) Semi-supervised DA refers to all source data are labeled, and some of the target data are labeled; 3) Supervised DA refers to all source data are labeled, and all target data are labeled. Moreover, the DA approach can be further categorized into another two types based on the type of differences between source and target domain [24]: 1) Homogeneous DA refers to the domain features distribution are different, but the feature spaces remain the same cross domains; 2) Heterogeneous DA refers to the domain feature spaces differs across domains. In this dissertation, we will focus on Homogeneous and unsupervised domain adaptation approaches.

Many recent studies have been exploring ways to address the domain-shifting issue. The general idea of most recent domain adaptation approaches learns invariant hidden features cross domains [24], then the predictions made using the invariant hidden features will be more accurate for the target domain [74]. One of the earlier works [75] proposed a model which learns invariant hidden features by using a regularization loss to minimize the maximum mean discrepancy (MMD) between source and target distribution. Later a pioneer work [74] proposed a domain-adversarial training of neural Networks (DANN), which learns invariant hidden features by adding a second domain classifier for classifying source and target domain together with the original label classifier in the network. The general idea of this work is to minimize the classification loss of the label classifier and, at the same time, maximize the classification loss of the domain classifier. Thus, the network will be discriminative for label classification but indiscriminate for distribution shifts between domains. An-

other work [76] proposed a network named WDGRL, which is inspired by DANN and was published later on in which the author replaced the domain classifier with a network that estimates the Wasserstein distance between source and target domains distributions [76]. Also, they use the network to estimate the distribution distance instead of using the computed distance, which is different from the work mentioned above [75].

However, most of the recent domain adaptation work has been approved that has excellent performance improvement on the image applications. Few works have focused on temporal data [24]. These approaches might not work well on temporal EHR data since they do not consider temporal dependencies. The first deep learning network that addresses the domain shift issue for temporal data is called Variational Recurrent Adversarial Domain Adaptation (VARADA) [77]. VARADA uses a similar idea in DANN where they also have two classifiers in the network, one for the label and another for the domain. In contrast to DANN, they use the variational recurrent neural network (VRNN) [78] as the feature extractor to learn the hidden feature representations which capture the temporal information. Some similar networks have been proposed ever since on domain adaptation applications for temporal EHR data. For example, [79] proposed a network for disease progression modeling which uses a domain classifier at every time step. The proposed network considers domain shifting issues at every time step. As a result, the learned hidden features will be invariant for all time steps. Another recent example of temporal EHR data adaptations is the framework named VR-ADS [80]. VR-ADS introduced a framework that uses invariant globally shared features across domains and different variant local features to address the domain-shifting issues for early septic shock prediction.

CHAPTER 3. ELMV: an Ensemble-Learning Approach for Analyzing Electronic Health Records with Significant Missing Values

This chapter introduces our novel ensemble-learning approach for analysis electronic health records with significant missing values (ELMV).

3.1 Introduction

Many EHR data contain a significant proportion of missing values, which could be as high as 50%, leading to a substantially reduced sample size even in initially large cohorts if we restrict the analysis to individuals with complete data [81, 41]. On the other hand, leaving a big portion of missing information unaddressed usually cause bias, loss of efficiency, and finally leads to the inappropriate conclusion to be drawn [82].

Data imputation algorithms (e.g., the scikit-learn estimators [83]) attempt to replace missing data with meaningful values, including random values, the mean or median, the spatial-temporal regressed values, most frequent values in the same columns, or representative values identified using k-nearest neighbor [84]. Advanced data imputation algorithms, such as Multivariate Imputation by Chained Equation (MICE) [85], have been developed to fill missing values multiple times.

However, applying data imputation algorithms without considering the reasons for missingness and the distribution discrepancies between training and testing data may lead to significant biases in model prediction.

We observe that in the EHR data, important variables are likely to be retained by auxiliary variables. For example, hemoglobin A1c (HbA1c) is an important index for diabetes patients. By measuring HbA1c, clinicians can get an overall picture of the average blood sugar levels over a few months. Multiple clinical measurements, such as fasting blood glucose, are highly correlated with HbA1c [86] and are often found

in the EHR of diabetes studies. Hence, if HbA1c is missing, a well trained predictive model can still rely on the auxiliary features of HbA1c, thus maintaining relatively high performance.

In this chapter, we present a novel method called Ensemble-Learning for Missing Value (ELMV) to analyze EHR data with significant missing values, aiming to identify unbiased, precise predictive patterns from EHR data. Specifically, given an EHR dataset with a significant missing rate, ELMV first generates multiple qualified maximal subsets of the original EHR data using dynamic programming. These qualified maximal subsets have much lower missing rates than the original data. And then, ELMV trains predictive models using every qualified maximal subset and save the trained model for further use. Finally, for each record in the external validation data, ELMV selects multiple pre-trained models and employs ensemble learning for the final prediction. ELMV has the following advantages: 1) It is capable of handling substantial missing values without using data imputation, 2) By constructing multiple maximal subsets of the original EHR data, opportunities are that even if critical features are removed due to high missingness, the generated predictive models using auxiliary features may still maintain a relatively high performance, and 3) It introduces dedicated support data for ensemble learning where the discrepancy between training and validation are considered for the purpose of reducing the bias.

3.2 Methods

Specifically, given an EHR dataset I with significant missing values, ELMV first generates a set of subsets of I with low missing rates, denoted as S , using dynamic programming, and upon these, builds predictive models M so as to mitigate the overall bias in each dataset in S for a single predictive model. Second, for every record in the external validation data, ELMV selects the most suitable models from M for the final prediction using ensemble learning.

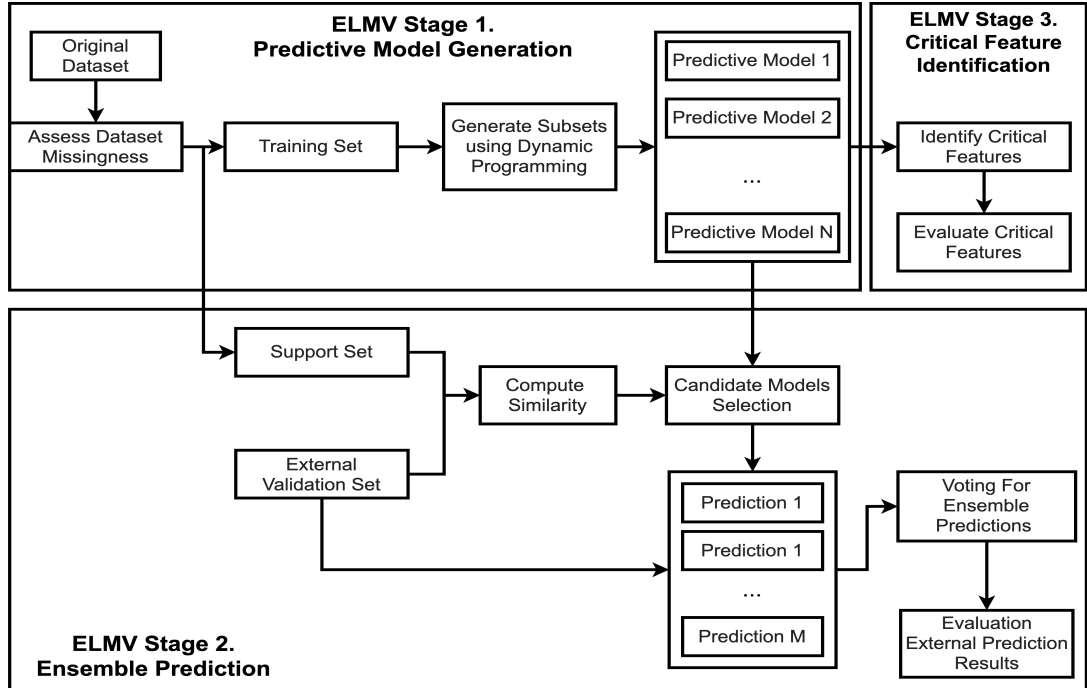


Figure 3.1: Overall framework of ELMV. It includes three stages: predictive model generation, ensemble prediction, and critical feature identification.

Since ELMV is a general machine learning framework for learning from EHR data with significant missing rates, any conventional machine learning model, such as XGBoost [12] and SVM [87], can be used in our framework. For the demonstration purpose, we used XGBoost in the paper. The framework of ELMV involves three stages, namely model generation, critical feature identification, and outcome prediction. The architecture of ELMV is illustrated in Figure 3.1.

3.2.1 ELMV Stage 1. Predictive Model Generation

In the predictive model generation stage, we first compute the data missingness of a given EHR dataset, assessing whether it is appropriate to use ELMV. And then, we generate multiple subsets of the original data with lower missing rates. Finally, a predictive model is trained on each subset.

3.2.1.1 Assessing Data Missingness

Given an EHR dataset I where rows are patients P , columns are features F in the EHR, N_p is the total number of patients, N_f is the total number of features, a missingness indicator for each patient p , denoted as $MissingI^p$, is defined as a binary vector with the length of N_f where a one in a specific entry represents that the corresponding feature is missing for patient p .

Specifically, for EHR data with temporal features TF^{N_f, T, N_p} , where T is the number of time points of a temporal feature, we define the missingness indicator as a two-dimensional matrix: for each temporal feature of patient p , denoted as $tf_j^p \in TF$, if a temporal trend-based feature is missing because of the missing data at time point t_i , let $MissingI^p(t_i, tf_j) = 1$.

A 2-dimensional binary matrix denoted as $MissingI \in \mathbb{R}^{N_p \times N_f}$ can then be generated to store the missingness information of all patients. In the 2D case, $MissingIM[i, j] = 1$ representing the patient i has a missing value in the j^{th} feature. In the case of a 3D temporary dataset where the third dimension represents time of records, $MissingIM[i, j] = 1$ representing the patient i at least have one data point missing in the time trajectory of the j^{th} feature.

Based on the definition of data missingness, we compute the missing rate of the entire dataset I , assessing whether ELMV or data imputation techniques should be used. Typically, if the data missingness is low, it is appropriate to impute missing data. However, if the missing rate is above 40%, data imputation may inevitably increase the variability of effect estimates. Instead of imputing missing values directly, ELMV relies on ensemble learning which aggregates predictive models built on multiple subsets with significantly lower missing rates.

Note that although ELMV is still applicable when the missing rate is low (e.g., under 10%), its performance is similar to other state-of-art models.

3.2.1.2 Generating Subsets with Low Data Missingness

Given an EHR dataset I and a user-defined data missing rate upper bound $T_{max-missing}$ (e.g., 20%), which is much lower than the missing rate of I , we generate a set of maximal subsets of I with its missing rate lower than or equal to $T_{max-missing}$, saved in S .

A subset s of I ($s \in S$) is a 2-dimensional matrix where rows are patients and columns are features in EHR. We say s is maximal if and only if its missing rate can only increase if its rows or columns are replaced by any new rows or columns in I .

Since the total number of possible subsets is $\binom{N_p}{x} \times \binom{N_f}{y}$, where x and y are the numbers of rows and columns of s , it is impractical to enumerate all the possibilities and then select the maximal ones. Thus, to identify all the qualified maximal subsets of I with missing rates lower than or equal to $T_{max-missing}$, we develop a two-step approach.

The approach for generating qualified maximal subsets consists of two steps: 1) to generate all the maximal subsets using dynamic programming, and 2) to filter the maximal subsets with nearly duplicated information. The pseudocode for maximal subsets generation is illustrated in Algorithm 1 and 2. In the following section, we explain the steps for generating the qualified maximal subsets.

In the first step, we track the missingness of all the subsets-to-generate using a 2-dimensional matrix $MissingC \in \mathbb{R}^{N_p \times N_f}$. The value in each entry of $MissingC(x, y)$ represents the minimum number of missing values of any subset of I with x patients and y features. For instance, $MissingC(100, 200) = 1300$ means that the minimum number of missing values is 1300 for any sub-matrix of I with 100 patients and 200 features. $MissingC$ can be used to select maximal subsets (see details in Algorithm 1 and 2).

We start to fill $MissingC$ and to generate the corresponding maximal subsets from the bottom right corner, $MissingC(N_p, N_f)$. Naturally, it represents the num-

Algorithm 1: Algorithm For Generating Maximal Subsets - Part 1

Input : 2D *DataMatrix* [$N_p \times N_f$] or
3D *Temporal DataMatrix* [$N_p \times N_f \times N_t$]

Intermediate: *MissingI*, *MissingI_List*, *MissingC*,

Output : *Max_S* # Maximal Subsets

Function *ConstructMissingI*(*DataMatrix*):

```
    for  $i = N_p$  to 1 do
        for  $j = N_f$  to 1 do
            if  $\sum_{t=1}^{N_t} \text{MissingI}^p[t, j] \geq 1$  then
                |  $\text{MissingI}_{i,j} = 1$ 
            else
                |  $\text{MissingI}_{i,j} = 0$ 
            end
        end
    end
    return MissingI
```

Function *Order*(*MissingI*):

```
    Order input by the missing percentage of patients and features
    ascendingly from left to right and from top to bottom
    return ordered MissingI
```

Function *CountMissings*(*MissingI*):

```
    Count the total number of ones in input
    return Total_Number_Of_Missing_Values
```

ber of missing values when all features and all patients are selected. Hence, the corresponding maximal subset is I itself. And then, we repeatedly remove one feature or one patient that has the maximum number of missing values at a time until the subset reaches the smallest required number of features and the smallest number of patients. By removing a feature or a patient with the maximum missing values at each time step, the generated subset is ensured to have the missing rate corresponding to the required number of features and patients. The whole process is achieved using dynamic programming [88].

The second step of subset generation is to identify and remove subsets conveying nearly identical information. For all the subset with a similar missing rate, we keep the

Algorithm 2: Algorithm For Generating Maximal Subsets - Part 2

```
1 Initialization;
2  $MissingI\_List_{N_p, N_f} = \text{ConstructMissingI}(DataMatrix)$  ;
3  $MissingC_{N_p, N_f} = \text{CountMissings}(MissingI\_List_{N_p, N_f})$  ;
4  $MissingI\_List_{N_p, N_f} = \text{Order}(MissingI\_List_{N_p, N_f})$  ;
5 for  $i = N_p$  to 1 do
6   for  $j = N_f$  to 1 do
7     if  $i \neq N_p$  and  $j \neq N_f$  then
8       if  $MissingC_{i, j+1} < MissingC_{i+1, j}$  or  $MissingC_{i+1, j}$  is empty
9         then
10           $MissingI\_List_{i, j+1} = \text{Order}(MissingI\_List_{i, j+1})$  ;
11           $last\_step = MissingI\_List_{i, j+1}$ ;
12          /* then remove the last feature */
13           $MissingI\_List_{i, j} = last\_step[-last\ column]$  ;
14        else if  $MissingC_{i, j+1} \geq MissingC_{i+1, j}$  or  $MissingC_{i, j+1}$  is empty
15          then
16           $MissingI\_List_{i+1, j} = \text{Order}(MissingI\_List_{i+1, j})$  ;
17           $last\_step = MissingI\_List_{i+1, j}$ ;
18          /* then remove the last patient */
19           $MissingI\_List_{i, j} = last\_step[-last\ row]$ ;
20         $MissingC_{i, j} = \text{CountMissings}(MissingI\_List_{i, j})$ ;
21         $Max\_S_{i, j} = \text{Patient and Features in } MissingC_{i, j}$ ;
22      end
23    end
24  end
```

subsets with the maximum number of features if the number of patients is identical, or keep the subsets with the maximum number of patients if the number of features is identical.

The final outcome of this step is a set of maximal subsets of the original EHR dataset with missing ratio smaller than or equal to a user-defined data missing rate upper bound $T_{max-missing}$.

3.2.1.3 Training Predictive Models

Using every qualified maximal subset of the original data I , we train a traditional classification model and save all the trained models in model set M . Since ELMV is a general framework for learning predictive patterns from data with significant

missingness, any classification model, such as support vector machine and gradient boosting, can be used in this step. We expect that the classification model deployed here is capable of handling a few missing values. Otherwise, we recommend to employ a data imputation method before calling a classification model.

For the demonstration purpose, the XGBoost implementation [12] "xgboost" in R library is used in this step. Specifically, we choose a tree-based model called "gbtree" booster with a softmax objective "multi:softprob" for relatively easier classification tasks. Also, we choose a linear model called "gblinear" with a logistic objective "binary:logistic" for relative harder classification tasks, such that a multi-class task can be converted into binary classification using the one vs. rest approach [89]. Finally, each trained predictive model is evaluated using leave-one-out cross validation (LOOCV) [90] approach. Model validation performance is saved for later use.

3.2.2 ELMV Stage 2. Ensemble Prediction

In the ensemble prediction stage, ELMV aggregates multiple selected predictive models trained in stage one to make predictions for records in an external validation set.

Here, each predictive model is trained with a qualified maximal subset with its missing rate smaller than or equal to $T_{max-missing}$. If $T_{max-missing}$ is significantly smaller than the missing rate of the original data I , the qualified maximal subsets could be much smaller subsets of the original data. Therefore, a predictive model can successfully capture the local but not the global properties of the original data. Directly using these predictive models individually may not result in optimal results. Meanwhile, for the records in the external validation set, they may differ regarding which distributions the records are drawn from, indicating that we may not obtain the best performance by aggregating all the pre-trained models. Hence, in the ensemble prediction stage, we develop a novel strategy to select pre-trained predictive models according to data representation and ensemble them for external validation.

3.2.2.1 Constructing Support Set

To estimate the distribution of the external validation records, a support set is generated. Mathematically, the support set SS^{N_{ss}, N_f} is generated by randomly select N_{ss} rows from the original dataset I . Similar to I , SS may have a significant missingness. For SS , a binary missingness matrix $MissingSS$ is obtained using the same method described in Section 3.2.1.1.

3.2.2.2 Measuring Patients Similarity

For each external validation record, we measure the similarities between it and all the records in the support set SS pair-wisely. Top k_1 similar records in SS are selected. ELMV assigns a set of dedicated pre-trained models to each external validation record by selecting all the pre-trained models that can successfully predict at least k_2 top records ($k_2 \leq k_1$). Both k_1 and k_2 is a user-defined parameter.

Formally, the similarity between a external validation record and all records in the support set SS is defined in Equation 3.1:

$$Sim = W_F * Softmax(-Dist_F) + W_M * Softmax(-Dist_M) \quad (3.1)$$

where $Sim \in \mathbb{R}^{(1 \times N_{ss})}$ represents the similarity between each individual validation record and all the records in the support set, $Dist_F \in \mathbb{R}^{(1 \times N_{ss})}$ represents the Euclidean distance of the corresponding feature vectors, $Dist_M \in \mathbb{R}^{(1 \times N_{ss})}$ represents the Hamming distance of the missingness indicator vectors $MissingI^p$, N_{ss} represents the number of records in support set SS , and the overall similarity score is a weighted sum of the two distances normalized using softmax. Here, weights W_F and W_M are user-adjustable parameters. Larger W_F indicates ELMV pays more attention to feature vectors similarity, and likewise larger W_M indicates the missingness vectors similarity is more important.

3.2.2.3 Ensemble Prediction

Finally, we select multiple pre-trained predictive models and aggregate them by adopting the ensemble prediction approach. The model selection procedure can be described as a multi-objective optimization problem that considers the following objectives: the model prediction performance on support records similar to the target external validation records, the model performance on all records in the support set, the model cross-validation performance such as accuracy, precision, recall, and F1, as well as the characteristics of the subset that is used to train the model including the number of features, the number of patients, and the missing rate.

Given a list of model selection criterion $\{C^1, C^2, \dots, C^n\}$ and a list of candidate models $\{M_1, M_2, \dots, M_m\} \in M$, let $TBest_M^C$ be a binary vector indicating whether model M performs the best under criteria C . Mathematically,

$$TBest_M^C = \begin{cases} 1, & \text{if } C_{M_j}^i = MAX(C^i) \\ 0, & \text{otherwise} \end{cases} \quad (3.2)$$

A pre-trained model is selected if and only if it performs the best on at least one criterion formulated in Equation 3.3 or the overall performance in all criterion is the highest (see Equation 3.4). The number/type of the objectives K_{obj} are user adjustable.

$$\exists C : \in TBest_M^C = 1 \quad (3.3)$$

$$\arg \max_M \sum_{i=1}^n TBest_M^{C_i} \quad (3.4)$$

In the last step, the final prediction for each record in the external validation set can be obtained by integrating all the selected models. For the demonstration purpose, a majority voting of all the selected models is used here, which can be replaced with other ensemble learning approaches with a simple modification.

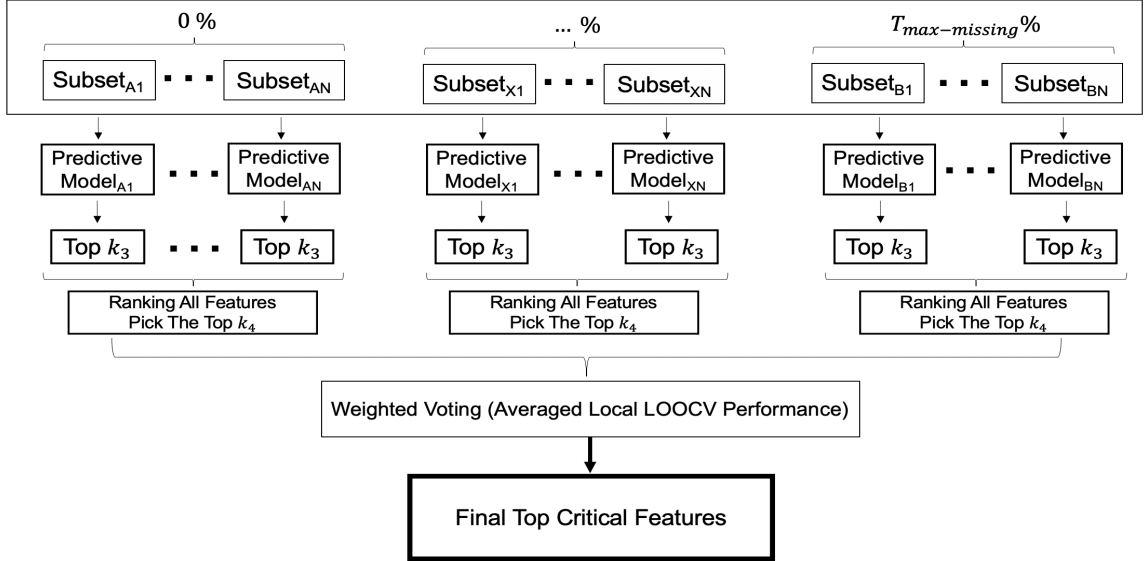


Figure 3.2: The framework of critical feature identification using ELMV.

3.2.3 ELMV Stage 3. Critical Feature Identification

Each predictive model trained with a qualified maximal subset produces its own critical features in its local context. In order to identify the critical features of the entire data, we repeatedly apply the leave-one-out cross validation (LOOCV) [90] on each qualified maximal subset. Finally, we aggregate the most critical features of each predictive model using a weighted voting mechanism. The critical feature identification process is shown in Figure 3.2. Through this process, domain experts can examine the validity and reliability of ELMV by checking whether the critical features found is reasonable under both the local and global context.

In the weighted voting process, the weight of a critical feature is determined by three factors, i.e. the local LOOCV performance of the pre-trained predictive model, missing rate of the qualified maximal subset used to train the predictive model, and local feature importance.

Generally speaking, the higher the local LOOCV performance, the more weight is put on the features found by that predictive model. Specifically, for each predictive model, the top- k_3 local critical features are determined by model feature importance.

Table 3.1: Definition of hyperparameters used in ELMV.

Parameter	Definition	Suggested Value
$T_{max-missing}$	Subset missing rate upper bound	\leq missing rate of original dataset I
k_1	Number of similar records in Support Set (SS)	k_1 and k_2 can be chosen according to the distribution of similarity scores (e.g., top 20% similarity scores)
k_2	Number of similar records that were predicted correctly by each generated model	
k_3	Number of local critical features identified by each qualified maximal subset	k_3 and k_4 can be chosen according to the total number of features (e.g., 10% of the total number of features)
k_4	Number of top ranked critical features identified by group of qualified maximal subsets	
k_{obj}	Number/type of model selection criterion	$k_{obj} \geq 1$
W_F	Similarity weights for feature vector	$0 \leq W_F \leq 1$ Larger W_F paying more attention to feature similarity
W_M	Similarity weights for missingness vector	$0 \leq W_M \leq 1$ Larger W_M paying more attention to missingness similarity

And then, all the top- k_3 critical features of every predictive model with a similar missing rate are sorted and ranked. The feature ranking is based on the ratio between the number of times a given feature being selected as a critical feature by individual predictive models and the number of times it is available. Given the ranked feature list, we select top- k_4 critical features using weighted voting where weights are determined by the averaged local LOOCV model performance.

The description of all the user-defined hyperparameters is provided in Table 3.1.

The source code of ELMV is available at:

<https://github.com/lucasliu0928/ELMV>.

3.3 Experiments Settings

Multiple experiments were carried out on both simulation datasets and a real-world EHR dataset to validate the usefulness of ELMV. For performance comparison, XGBoost was used as the base predictive model. We compared ELMV with three models: 1) to impute missing values with the mean imputation and to train XGBoost with the imputed data, 2) to impute missing values with MICE [85] and to train XGBoost with the imputed data, and 3) to train XGBoost directly without using any data imputation method.

3.3.0.1 Simulation Data

To simulate EHR data with a significantly high missing rate, we selected a complete data and constructed multiple simulation datasets with a wide range of missing rates. On the simulation data, we test whether adopting ELMV can achieve performance comparable to that of a predictive model trained on the complete dataset. Specifically, the complete dataset obtained was the IRIS dataset widely used in machine learning education from the UCI repository [91]. The IRIS data consists of four features, 150 records, and three outcome labels. The LOOCV accuracy of XGBoost on the IRIS data is as high as 0.97.

In total, 18 simulation datasets were generated using the IRIS data, each having 40 features and 150 records, while the missing rate varying from 5% to 70%. All the simulation datasets were constructed similarly, except for the missing rates. First, using each of the original features in the IRIS data, we generated nine additional features with their correlation coefficient to the original feature ranging from 0.1 to 0.9. The purpose was to test whether the model performance can be retained using auxiliary (highly correlated) features when original features are missing. In addition, the additional features were used to test whether the model can identify and retain high-quality features while discarding low-quality features. Finally, we randomly

removed 5% to 70% entries from every simulation dataset.

3.3.0.2 Real World Healthcare Data

The real-world EHR data we used was collected in a follow-up study of 240 type 2 diabetes (T2DM) patients who went through the Laparoscopic Roux-en-Y Gastric Bypass (LRYGB) surgery [92] in the Shanghai Jiaotong University Affiliated 6th People’s Hospital. The data have been de-identified before use.

The LRYGB dataset consists of 79 variables including HbA1c and the other 78 biomedical variables collected at six different time points, i.e. before the LRYGB surgery, 3-month, 6-month, 12-month, 24-month, and 36-month after the surgery. In total, 240 T2DM patients participated the study. 24 out of the 78 biomedical variables, such as CysC, weight index, and direct bilirubin, were pre-selected based on domain knowledge for further studies.

The purpose of the study is to predict the HbA1c trajectories that are defined as follows. The types of HbA1c trajectories were determined using clustering, followed by manual curation. Specifically, we adopted the reversed K-nearest neighbor (rKNN) [93] to remove outliers and adopted the agglomerative hierarchical clustering with Ward’s method [94] to separate all the patients into nine clusters. The Elbow method was then used to determine the optimal number of clusters, on which the decreasing rate of With-in-Sum-of-Squares (WSS) was the slowest. Two clinicians examined the obtained clusters independently and defined six types of HbA1c trajectory. In summary, after semi-automatic labeling, the LRYGB data consists of 214 patients, 24 features, and six labels. The missingness of all the features of the LRYGB data is shown in Figure 3.3. The missing ratio at every time point is 3%, 33%, 18%, 18%, 37%, and 56% respectively. Clearly, patient dropout is a main issue that resulted in high missing rates at later time points. Using this real-world data, we aim to test ELMV at the non-random missing data situation. Specifically, we

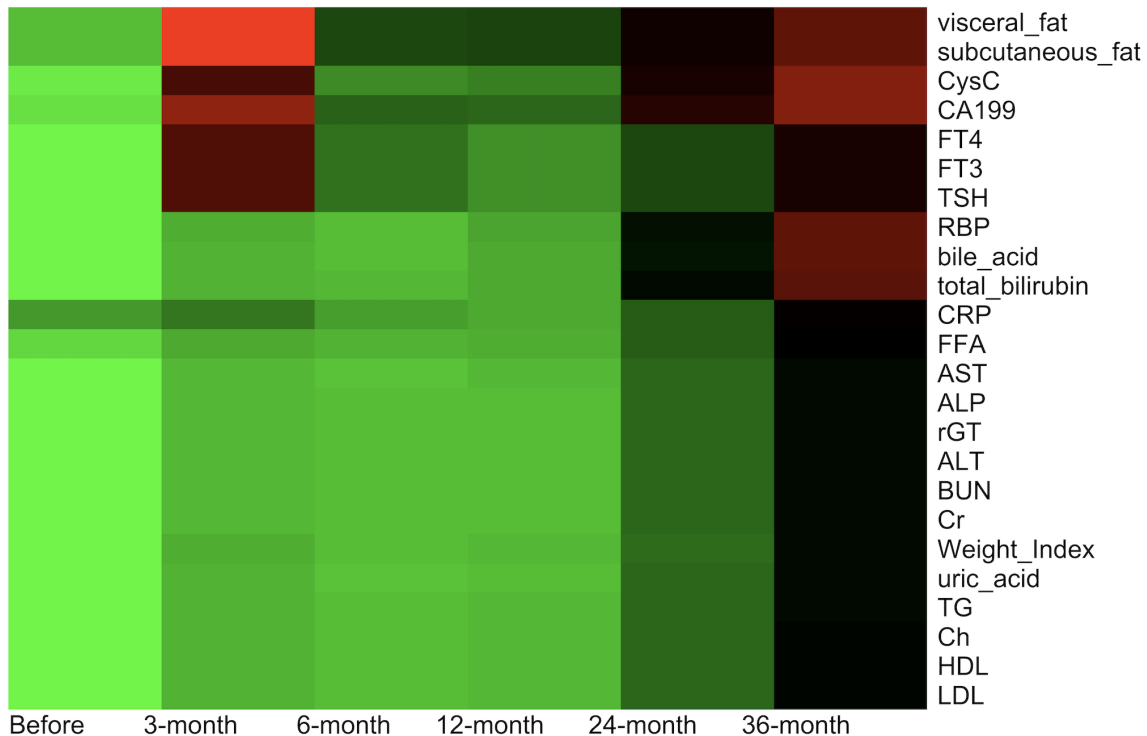


Figure 3.3: In the LRYGB follow-up study, the distribution of the missing values of all the 24 variables at six time points. In general, more values are missing towards the end of the follow-up study. Red indicates higher missing ratio towards 100%, green is for lower missing ratio towards 0%, and black indicates 50% missing ratio.

evaluated ELMV by testing whether it can identify critical features for predicting the trajectory of HbA1c.

As part of the data preprocessing, we imputed a small portion of the missing values using domain knowledge and simple statistics such as linear interpolation. Also, we copied the 6th month values to the 3rd month, if the 3rd month values were missing. We removed patients whose HbA1c values at both 3rd month and 6th month are missing. After this step, the LRYGB follow-up data consists of 202 patients, 24 features, and the overall missing rate of the LRYGB data was reduced. For example, the missing rates at 24-month and 36-month have been effectively reduced from 37% to 25% and from 56% to 48%, respectively. But still, the high missing rate towards the end of the T2DM follow-up study prevents us from using any predictive models

Table 3.2: Averaged accuracy of ELMV, XGBoost, and two imputation methods on the simulation data with low (above the horizontal line) or high missing rates (below the horizontal line).

Missing Rate	XGBoost	Mean Imputation	MICE Imputation	ELMV
5%	0.97	0.97	0.97	0.97
10%	1.00	0.97	0.93	1.00
20%	0.97	0.97	0.97	0.97
60%	0.67	0.63	0.73	0.80
65%	0.70	0.73	0.70	0.77
70%	0.70	0.67	0.63	0.77

directly.

3.4 Results

We applied ELMV, as well as three baseline algorithms, i.e., mean imputation, MICE, and XGBoost without data imputation, on both the simulation data and the LRYGB data. For performance comparison, conventional classification metrics were used, including accuracy, precision, recall, and F-1. Additionally, domain experts manually reviewed the critical features selected by ELMV, assessing whether they are clinically reasonable for predicting the HbA1c trajectory.

3.4.0.1 Prediction Performance on Simulation Data

On all the simulation datasets with their missing rates ranging from 5% to 70%, the performance of ELMV, mean imputation, MICE, and XGBoost without data imputation were systematically compared. Table 3.2 compares model prediction accuracy of the four methods on the simulation datasets. When the missing rate was low (5% to 20%), all the models can achieve nearly perfect performance (accuracy ≥ 0.93). However, if the missing rate was in the range of 60% and 70%, the accuracy of all other methods was reduced significantly below 75% no matter how the missing values were handled while ELMV still can maintain its accuracy above 75%.

A moving average of accuracy and F-1 on the finer granularity of missing rates shown in Figure 3.4 and Figure 3.5 reveal that ELMV is not affected by the high missing rates as bad as the other models. The performance trends suggest that ELMV achieved the best performance towards larger missing rates steadily, and XGBoost had the best performance if the missing rate was relatively low. MICE had the overall lowest accuracy and its accuracy trend dropped steadily when the missing rate was increased. Surprisingly, the mean imputation had a relatively stable performance, probably because the missingness was generated completely randomly. Both mean imputation and MICE have lower accuracy than XGBoost, indicating that the two imputation methods tested failed to reinforce XGBoost to handle missing values. The averaged precision, recall, and F-1 are reported in Table 3.3, Table 3.4, and Table 3.5, respectively. Similarly, ELMV achieved the best performance in all but one case when the missing rate was high.

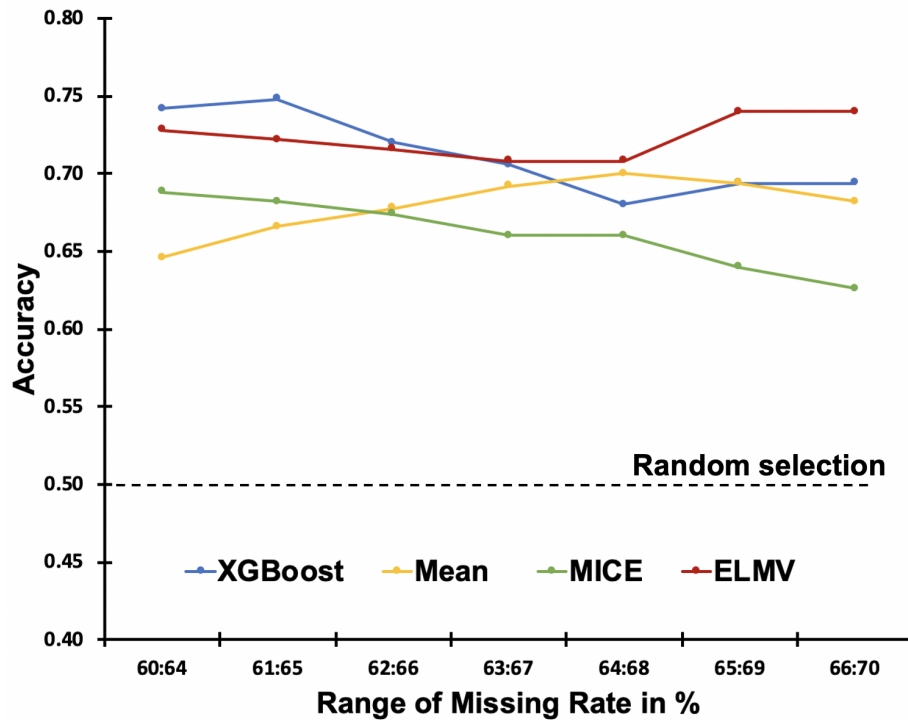


Figure 3.4: The moving average of accuracy of ELMV, XGBoost, and two imputation methods on the simulation data with missing rate increasing from 60% to 70%.

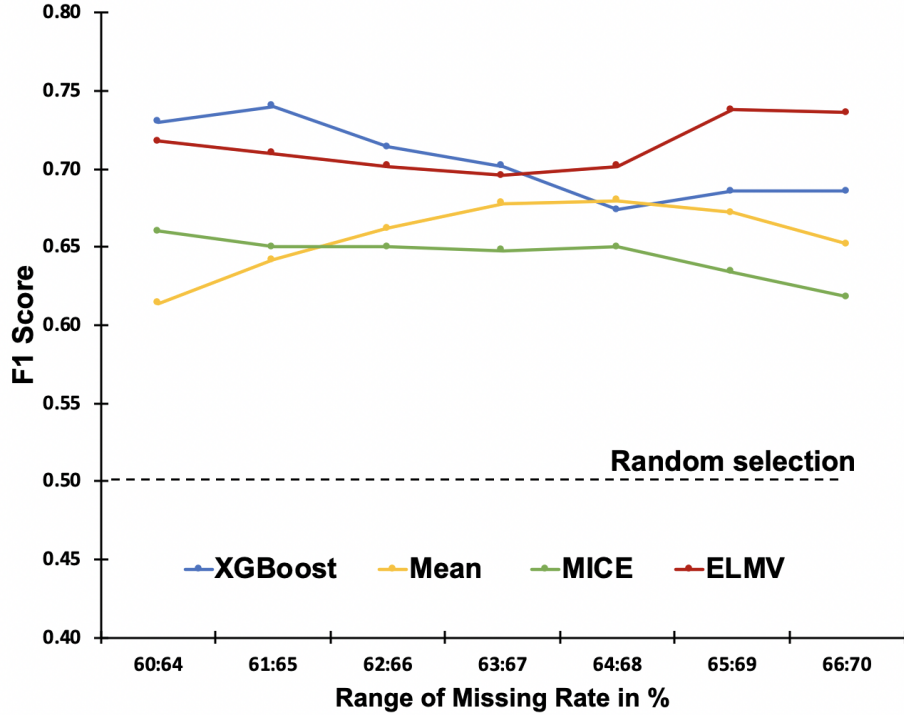


Figure 3.5: The moving average of F-1 Scores of ELMV, XGBoost, and two imputation methods on the simulation data with missing rate increasing from 60% to 70%.

3.4.0.2 Feature Selection on Real World EHR Data

We applied ELMV on the LRYGB data (78 features and 202 T2DM patients), aiming at identifying critical features for the HbA1c trajectory prediction. All the qualified maximal subsets of the LRYGB data generated by ELMV are shown in Figure 3.6. Every point in the figure represents a qualified maximal subset of the LRYGB dataset.

Table 3.3: Averaged precision of ELMV, XGBoost, and two imputation methods on the simulation data with low (above the horizontal line) or high missing rates (below the horizontal line).

Missing Rate	XGBoost	Mean Imputation	MICE Imputation	ELMV
5%	0.97	0.97	0.97	0.97
10%	1.00	0.97	0.95	1.00
20%	0.97	0.97	0.97	0.97
60%	0.65	0.64	0.75	0.83
65%	0.71	0.80	0.72	0.78
70%	0.71	0.69	0.62	0.76

Table 3.4: Averaged recall of ELMV, XGBoost, and two imputation methods on the simulation data with low (above the horizontal line) or high missing rates (below the horizontal line).

Missing Rate	XGBoost	Mean Imputation	MICE Imputation	ELMV
5%	0.95	0.95	0.95	0.95
10%	1.00	0.95	0.90	1.00
20%	0.95	0.95	0.95	0.95
60%	0.64	0.59	0.73	0.83
65%	0.70	0.71	0.69	0.76
70%	0.70	0.63	0.61	0.74

Table 3.5: Averaged F-1 of ELMV, XGBoost, and two imputation methods on the simulation data with low (above the horizontal line) or high missing rates (below the horizontal line).

Missing Rate	XGBoost	Mean Imputation	MICE Imputation	ELMV
5%	0.96	0.96	0.96	0.96
10%	1.00	0.96	0.92	1.00
20%	0.96	0.96	0.96	0.96
60%	0.64	0.59	0.74	0.80
65%	0.69	0.73	0.69	0.76
70%	0.69	0.63	0.61	0.75

The X-axis indicates the number of patients, and the y-axis indicates the number of features of the qualified maximal subset.

In Figure 3.6, the points with the same color have a similar missing rate. We generated all the qualified maximal subsets of the LRYGB data so that any combinations of features of interest can be evaluated in the critical feature identification stage of ELMV. Note that since the goal of this experiment is to identify the critical features among 24 pre-selected features, we only used the qualified maximal subsets of the pre-selected features in the following analysis. Several early-stage biomarkers, such as serum Ca²⁺ and cholesterol level measured at 3-month found by ELMV were supported well by the literature [95, 96] to be critical for predicting HbA1c trajectory in the first three years after the LRYGB surgery.

In addition, the overall accuracy of ELMV on the LRYGB data is 0.93, signif-

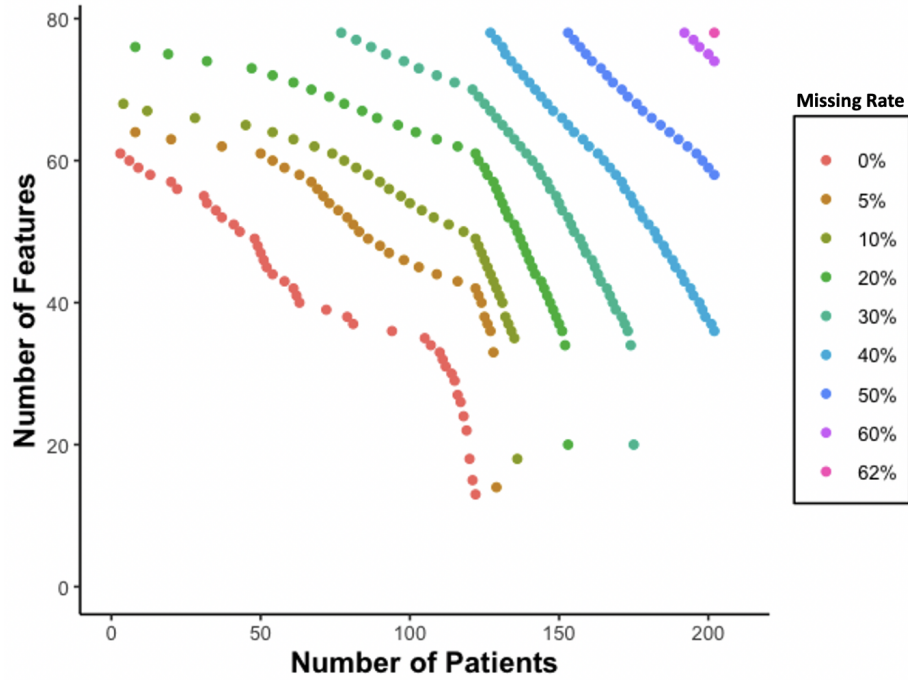


Figure 3.6: The distribution of all the maximal subset of the original LRYGB data with 78 features and 202 T2DM patients. Every point represents a maximal subset with x number of patients and y number of features. Color indicates different missing rates.

Table 3.6: Performance of qualified maximal subsets of the LRYGB data with different missing rates.

Missing Rate	Accuracy
0%	0.90
5%	0.95
10%	0.94
20%	0.94
30%	0.93
Average	0.93

icantly higher than that of XGBoost (0.63), Mean imputation (0.30), and MICE (0.28). The performance of ELMV on all the qualified maximal subsets with the missing rate ranging from 0% to 30% is shown in Table 3.6. It indicates that ELMV can maintain its accuracy above 90% and is not significantly affected by the high missing rates.

Table 3.7: Average computational time comparison among ELMV, Mean imputation, MICE imputation (two iterations, and two multiple imputations), and XGboost.

Data	Methods	Training and Validation
Simulation Data (150x40)	ELMV	101.40 secs
	Mean	1.66 secs
	MICE	31.20 secs
	XGBoost	1.92 secs
Healthcare Data (202x78x6)	ELMV	18.34 mins
	Mean	1.04 mins
	MICE	354.00 mins
	XGBoost	1.36 mins

Since extra steps have been taken in ELMV, an interesting question is whether ELMV is significantly slower than the other models. We compared the computational time between ELMV with the baseline methods on both the simulation data and the healthcare data. As shown in Table 3.7, the mean imputation was the fastest on both datasets, while ELMV was the slowest (101 secs) on the simulation data when the number of features was relatively small. On the real healthcare data where the number of features was relatively large, MICE took more than 300 minutes while ELMV spent only around 18 minutes, and most of its time (70%) was spent on generating the maximal subsets using dynamic programming. This issue could be further addressed by clustering patients with similar missingness.

In ELMV, a novel approach is introduced to estimate the distribution of external validation data and to guide the ensemble learning using a support set. An interesting question is to what extent the support set can contribute to the ensemble learning since it is useful only when the external validation data are known. To this end, we compared ELMV with the k-nearest neighbor (kNN) model, which simply assigns each external validation record to the label of most similar records in the support set. The results shown in Table 3.8 indicate that the kNN-based voting approach is unlikely to provide the correct prediction most of the time. This experiment further

Table 3.8: Accuracy of ELMV and kNN on the LRYGB data.

Missing Rate	ELMV	kNN
60%	0.80	0.53
65%	0.77	0.43
70%	0.77	0.40
Average	0.78	0.45

confirms that it is critical to integrate the support set with ensemble learning rather than simple voting.

3.5 Conclusion

This chapter presented a novel ensemble learning model called ELMV to predict patient outcomes using EHR data with substantial missing values. In our experimental results, ELMV outperformed two widely used data imputation methods and an ensemble learning method on patient outcome prediction and critical feature identification. We also demonstrated that ELMV is novel on model selection, which considers data and missingness distributions in training and validation.

CHAPTER 4. KGDAL: Knowledge Graph Guided Double Attention LSTM for Rolling Mortality Prediction for AKI-D Patients

This chapter introduces our novel knowledge-graph guided double attention LSTM model named KGDAL for rolling mortality prediction to address interpretability concerns.

4.1 Introduction

Acute kidney injury (AKI) is a common complication of hospitalized patients and the incidence increase in patients admitted to the intensive care unit (ICU) [97, 98]. AKI that results in the need for dialysis (AKI-D) is associated with a high risk of hospital mortality [99], and for survivors a risk of incident or progressive chronic kidney disease (CKD) [100, 101, 102, 103], cardiovascular disease [104, 105, 106] or end-stage renal disease (ESRD) [107, 108, 109]. By identifying mortality risk factors from patient individual and population data, providers can implement early intervention strategies leading to better health care and substantially reducing the cost of care.

Numerous factors may influence in-hospital mortality including acute anemia, respiratory failure, electrolytes disarrangements, hemodynamic instability, and demographic information. There is a critical need to identify and correlate these patients and dialysis-specific parameters with inpatient mortality in this specific population. Moreover, accurate prediction of mortality over time (i.e., rolling prediction) in real-world healthcare settings for critically ill patients with AKI-D is needed for better utilization of hospital resources, such as intensifying therapies when is needed, or transitioning patients with a high risk of mortality to comfort care [5, 6]. Two general approaches have been used for mortality predictions in the ICU. The first approach uses clinical scores, including Acute Physiology and Chronic Health Evaluation (APACHE) and Sequential Organ Failure Assessment (SOFA), to identify

at-risk patients at any time point [110]. The second approach employs machine learning (ML) methods, such as random forest [9] and SVM [10], to predict mortality risks using electronic health record (EHR) data. With the rapid development of ML techniques, ML-based mortality prediction attracts much attention recently. Nevertheless, ML-based methods mainly focus on mortality prediction at the end of the treatment [111], and the clinical needs for rolling mortality prediction is often overlooked [112, 113]. Traditional ML models rely heavily on feature engineering, requiring not only a deep understanding of the domain knowledge but also tremendous efforts on manual feature extraction and model tuning [114].

In recent years, deep learning (DL) models, including Transformer [15], Long-Short Term Memory (LSTM) [16] and gated recurrent neural networks [17], have shown promising performance on end-to-end patient risk prediction using large-scale EHR data [1].

In this study, we propose a Knowledge-Graph Guided Double Attention LSTM (KGDAL) model, aiming to make precise rolling mortality predictions in a real-world healthcare setting for critically ill patients with AKI-D. To our knowledge, KGDAL is the first KG-guided model that extracts both time and feature attention on continuous temporal data. KGDAL has the following advantages:

- KGDAL obtains two-dimensional attention in both the time and feature spaces for improved prediction power and enhanced model interpretability.
- The attention mechanism in the feature space is automatically derived based on the KG rather than manual curation.
- KGDAL can model both continuous and discrete temporal EHR data types.
- KGDAL can make precise rolling mortality predictions for AKI-D patients on two independent clinical datasets.

4.2 Method

The goal of this study is to conduct rolling mortality prediction to assist clinicians in making timely decisions and actions [111]. Mathematically, for each particular subsequence of a patient’s EHR data, KGDAL predicts the patient’s outcome in the next K hours, where K is the time granularity (e.g., 72 hours). The outcome could be mortality or survival. KGDAL’s overall architecture, as shown in Figure 4.1, contains three phases:

Phase 1. EHR Data Extraction. Patient clinical features are extracted from patient’s EHR. Each feature is matched to a set of concepts in a Knowledge-Graph (KG), followed by manual validation by clinicians. Patient subsequences with random starting time points and varying lengths are generated. The label of each subsequence is “mortality” or “survival” in the next K hours. Static features of each patient are consistent for all the subsequences of the same patient.

Phase 2. Knowledge-Graph Extraction. The entire KG is used to learn the concept embeddings for every concept identified in Phase 1. The concept embeddings are grouped based on the KG’s hierarchical structure, resulting in multiple concept embedding groups. Subsequently, the KG-embedding distances (including both pairwise distances between concept embedding groups and the distances from a concept-of-interest to every concept embedding group) are computed.

Phase 3. Knowledge-Graph Guided Double Attention. All the temporal features are grouped based on their corresponding concept embedding groups. An LSTM is assigned to each temporal feature group. All the LSTM models are trained simultaneously to minimize the overall loss. Both feature and time attentions are learned using fully connected layers followed by softmax. Double attention is formed using both feature and time attention. The double attention is adopted to adjust feature embeddings for the final prediction and to regularize the prediction loss that minimizes the discrepancy between the attention-based distance and the

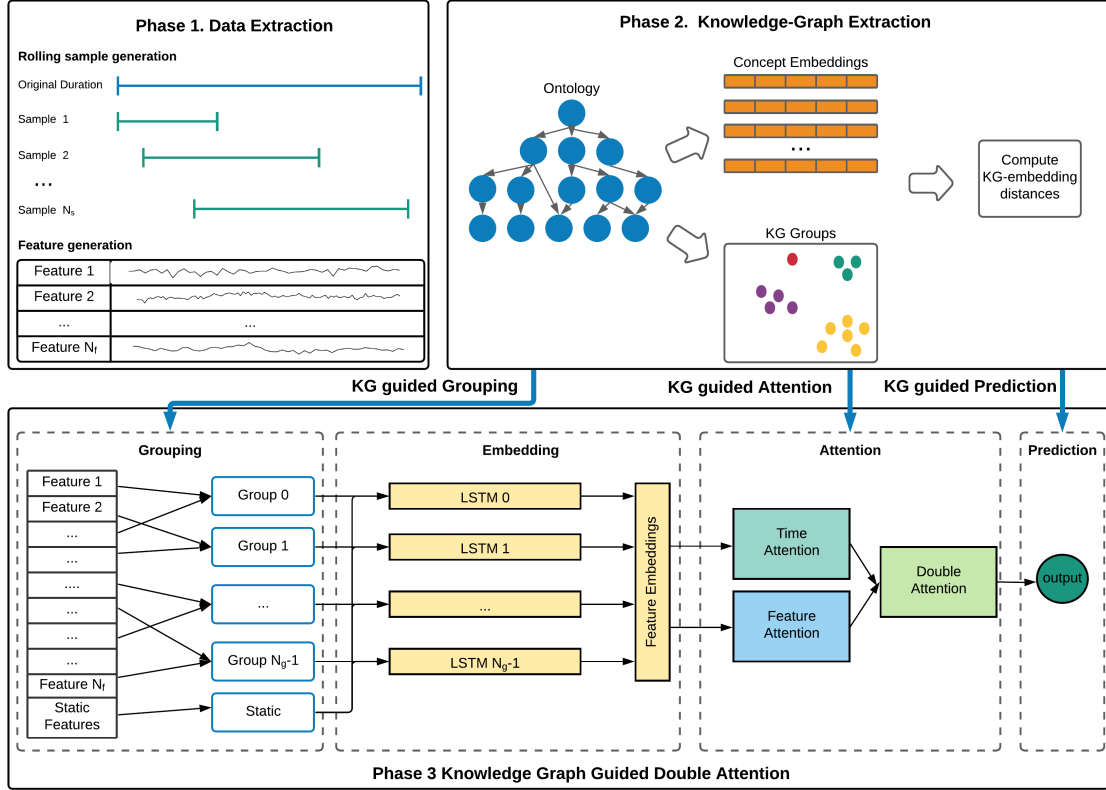


Figure 4.1: The three phases of the Knowledge-Graph Guided Double Attention (KGDAL) model for rolling mortality prediction for critically ill patients with AKI-D in real-world healthcare settings.

KG-embedding distance.

4.2.1 Phase 1. Data Extraction

Given a patient’s EHR data $\{S_1, S_2, \dots, S_t, \dots, S_{N_t}\}$, where $S_t \in \mathbb{R}^{N_f}$ is a set of features (e.g. clinical measurements) at time point $t \in \{1, 2, \dots, N_t\}$, N_f is the total number of features, and N_t is the total number of time points. For each patient, we generate N_s subsequences with randomly starting time and random length between 24 and 72 hours (see Figure 4.1 Phase 1). The outcome label of each subsequence is whether the patient dies or is alive in the next K hours.

Patient clinical features are matched to a set of corresponding concepts in a KG. For features that can be directly matched to a concept in a KG (e.g., a diagnosis

code), the corresponding concept will be used. Otherwise, all the associated concepts in the same KG are extracted and then filtered by clinicians. For example, the corresponding concepts of “systolic blood pressure” in HPO [115] are “elevated systolic blood pressure (HP:0004421)” and “decreased systolic blood pressure (HP:0500105)”, both of which are valid and kept for later use.

4.2.2 Phase 2. Knowledge-Graph Extraction

In phase 2, KGDAL generates concept embeddings and computes the group-wise KG-embedding distances in a KG.

4.2.2.1 Concept Embedding

Let E_c be the set of concepts and E_r be all possible relationship types in a KG, a concept relation O can be represented using a triplet denoted as (C_h, r_i, C_t) , where $C_h, C_t \in E_c$ are the head and tail concepts respectively, and $r_i \in E_r$ represents the relationship from C_h to C_t . For example, triplet (“Diabetes mellitus type 1”, “is-a”, “diabetes mellitus”) in SNOMED-CT represents that the Diabetes mellitus type 1 is a subtype of diabetes mellitus.

All the triplets in a KG are used to learn the concept embeddings. The basic idea is to make the learned embeddings of tail concept C_t be close to the sum of the embeddings of head concept C_h and the embeddings of relation r_i . Here, we use TransE [61], one of the most representative translational distance model, to learn the concept embeddings by formulating the problem as follows: given a concept triplet $O = (C_h, r_i, C_t)$ in a KG, we learn the embedding triplets denoted as $G = (\mathbf{h}, \mathbf{l}, \mathbf{t})$, where $\mathbf{h}, \mathbf{t} \in \mathbb{R}^{d_e}$ represents the head and tail concepts embeddings respectively, $\mathbf{l} \in \mathbb{R}^{d_l}$ is the relation vector between \mathbf{h} and \mathbf{t} . TransE is trained with negative

sampling to learn the embeddings that minimize a margin-based ranking loss function:

$$L_{KG} = \sum_{(h,l,t) \in O} \sum_{(h',l',t') \in O'} \max(0, \gamma + d(h + l, t) - d(h' + l', t')) \quad (4.1)$$

where O represents the positive samples and O' represents the negative samples that were randomly generated by replacing the head or tail concepts of positive samples. d represents any distance metrics, $\gamma > 0$ is a margin hyper-parameter. Equation 4.1 shows that the distances in positive samples are minimized where the distances in negative samples are maximized. In this step, the concept embedding denoted as $E_{concept}$ are obtained for each concept associated with the patient clinical features.

4.2.2.2 Embedding Grouping

The hierarchical level in a KG represents classes of concepts holding similar characteristics. To capture the commonality within a class and the difference between classes, embedding group E is formed by taking the sum of concept embeddings $E_{concept}$ in each group based on the hierarchical structure of a KG. The number of groups N_g is determined by the number of classes in the user-specified KG level. In general, using a higher level will form more general concept groups, and using a lower level will form more specific concept groups.

4.2.2.3 Embedding Group-wise Distance

In order to measure the difference between concept embedding groups as well as how much each group are related to the outcome (i.e., mortality) and to let a DL model pay more attention to the closely related concept embedding groups, two KG-embedding distances are computed. The pairwise distance between concept embedding groups is computed using Eq 4.2.

$$KG_{dist}(i, j) = dist(E_i, E_j) \quad (4.2)$$

where E represents a set of concept embedding groups, the subscript represents the group indexes, and $dist$ denotes a distance metric, such as the Euclidean distance.

The distance between each concept embedding group and the concept-of-interest are computed using Eq 4.3.

$$KG_Target_{dist}(E_i) = dist(E_i, E_{target}) \quad (4.3)$$

where $KG_Target_{dist}(i)$ represents the distance from the target embedding E_{target} to the i th concept embedding group.

4.2.3 Phase 3. Knowledge-Graph Guided Double Attention

4.2.3.1 LSTM Embedding

By assigning each temporal feature to its corresponding concept embedding group in a KG, a patient’s subsequence can be denoted as $\{\mathbf{x}_1^i, \mathbf{x}_2^i, \dots, \mathbf{x}_t^i\}$, where $\mathbf{x}_t^i \in \mathbb{R}^{n_i}$ is a list of feature values in the corresponding concept embedding group i at time step t , where i is the group index ($0 \leq i < N_g$), $t \in \{1, 2, \dots, T\}$ is the time step, and n_i is the number of features at each time step for the i th group.

KGDAL consists of N_g LSTM [16] models, each for a feature group. For simplicity, we assume all the feature groups have the equal number of features, the subscript i of n is omitted in the following text. LSTM has three gates, namely the forget gate \mathbf{f}_t , the input gate \mathbf{i}_t , and the output gate \mathbf{o}_t , where $\mathbf{f}_t, \mathbf{i}_t, \mathbf{o}_t \in \mathbb{R}^m$, and m is the dimension of the hidden vectors. Using \mathbf{c}_t and \mathbf{h}_t to represent the cell state vector and the hidden state vector, ($\mathbf{c}_t, \mathbf{h}_t \in \mathbb{R}^m$), the updated LSTM cell in KGDAL can be represented as follows:

$$\mathbf{f}_t = \sigma(\mathbf{W}_f \mathbf{h}_{t-1} + \mathbf{U}_f \mathbf{x}_t + \mathbf{b}_f) \quad (4.4)$$

$$\mathbf{i}_t = \sigma(\mathbf{W}_i \mathbf{h}_{t-1} + \mathbf{U}_i \mathbf{x}_t + \mathbf{b}_i) \quad (4.5)$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_o \mathbf{h}_{t-1} + \mathbf{U}_o \mathbf{x}_t + \mathbf{b}_o) \quad (4.6)$$

$$\tilde{\mathbf{c}}_t = \tanh(\mathbf{W}_c \mathbf{h}_{t-1} + \mathbf{U}_c \mathbf{x}_t + \mathbf{b}_c) \quad (4.7)$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{c}}_t \quad (4.8)$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t) \quad (4.9)$$

where $\mathbf{W}_f, \mathbf{W}_i, \mathbf{W}_o, \mathbf{W}_c \in \mathbb{R}^{m \times m}$, $\mathbf{U}_f, \mathbf{U}_i, \mathbf{U}_o, \mathbf{U}_c \in \mathbb{R}^{m \times n}$, $\mathbf{b}_f, \mathbf{b}_i, \mathbf{b}_o, \mathbf{b}_c \in \mathbb{R}^m$ are learnable parameters, σ is a sigmoid function, and \odot is the Hadamard product.

As a result, each LSTM layer outputs a hidden state matrix for each feature group denoted as $\mathbf{H}^{(i)} = [\mathbf{h}_1^{(i)}, \mathbf{h}_2^{(i)}, \dots, \mathbf{h}_T^{(i)}]$, where i is the group index. The hidden state metrics for each feature group is called the feature embedding matrix. Then the feature embedding matrix learned using multiple LSTM layers can be denoted as $\{\mathbf{H}^{(0)}, \mathbf{H}^{(2)}, \dots, \mathbf{H}^{(N_g-1)}\}$, where $\mathbf{H}^{(i)} \in \mathbb{R}^{m_i \times T}$ represents the feature embedding matrix of group i , and m_i is the dimension of hidden state vector from the i th LSTM for group i .

4.2.3.2 KG-Guided Double Attention

To model the latent dependencies between different feature groups and at different time steps, KGDAL learns the attentions in both time and feature spaces guided by a KG. The detailed architecture of KGDAL is in Figure 4.2.

Time Attention. All the N_g feature embedding matrices are concatenated into one matrix denoted as \mathbf{H}^C with the dimension of $\mathbb{R}^{(m_1+m_2+\dots+m_{N_g}) \times T}$. For simplicity, we assume all LSTM layers have equal dimensions of hidden vectors. Hence, the superscript or subscript i of m is omitted in the following text. and the dimension of \mathbf{H}^C is now $\mathbb{R}^{(N_g m) \times T}$. The time attention is computed as follows:

$$\mathbf{M}_\alpha = \tanh(\mathbf{H}^C) \quad (4.10)$$

$$\boldsymbol{\alpha} = \text{softmax}(\mathbf{M}_\alpha^T \mathbf{w}_\alpha) \quad (4.11)$$

where $\boldsymbol{\alpha} \in \mathbb{R}^T$ is the time attention, $\mathbf{w}_\alpha \in \mathbb{R}^{N_g m}$ is the learnable parameter, and \mathbf{M}_α^T is the transpose of $\mathbf{M}_\alpha \in \mathbb{R}^{(N_g m) \times T}$.

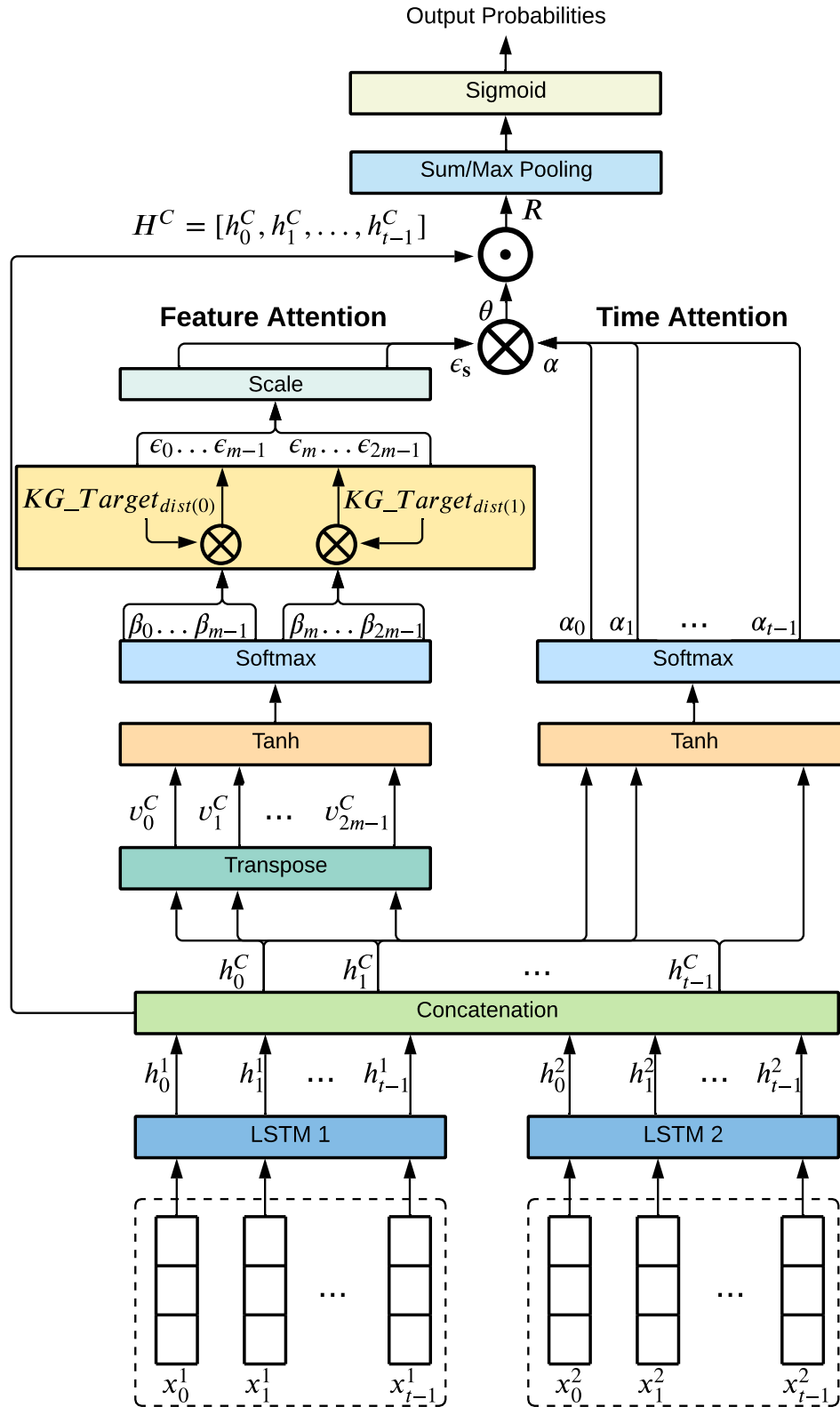


Figure 4.2: The detailed architecture of the Knowledge-Graph Guided Double Attention LSTM (KGDAL) model.

The time attention mechanism is inspired from and is similar to the work by Zhou et al. [58], but there are two key differences: 1) the input to LSTM is the grouped feature sequences and 2) the time attention mechanism is applied on the output of multiple LSTMs.

Feature Attention. A similar attention mechanism is used to compute the feature attentions. However, there are two changes. Firstly, we transpose \mathbf{H}^C and use it as the input so that the attention mechanism will be applied on the feature space instead of the time space. Secondly, the KG-embedding distances (KG_Target_{dist}) between the concept embedding groups and the concept-of-interest are used to weight the raw feature attentions. Mathematically, the raw feature attentions is computed using:

$$\mathbf{M}_\beta = \tanh((\mathbf{H}^C)^T) \quad (4.12)$$

$$\boldsymbol{\beta} = \text{softmax}(\mathbf{M}_\beta^T \mathbf{w}_\beta) \quad (4.13)$$

where $\boldsymbol{\beta} \in \mathbb{R}^{N_g m}$ is the raw feature attention, $\mathbf{w}_\beta \in \mathbb{R}^T$ is the learnable parameter, and \mathbf{M}_β^T is the transpose of $\mathbf{M}_\beta \in \mathbb{R}^{(T) \times N_g m}$.

Then the KG-adjusted feature attention $\boldsymbol{\epsilon}$ is equals to $\boldsymbol{\beta}$ weighted by the KG-embedding distances at corresponding position, which is computed using:

$$\boldsymbol{\epsilon}_{(pos,i)} = \boldsymbol{\beta}_{(pos,i)} \otimes KG_Target_{dist}(i) \quad (4.14)$$

where $\boldsymbol{\epsilon} \in \mathbb{R}^{N_g m}$, \otimes represents the outer product. pos represents the corresponding positions of feature attentions for each group, where $pos = [im : (i+1)m - 1]$, i is the group index, m is the dimension of the hidden vectors. For example, if the first m raw feature attentions $\boldsymbol{\beta}$ are obtained for the first feature group, then these m values are weighted by the distance from the first KG-embedding group to the target embedding.

4.2.3.3 Double Attended Representations.

We combine the attention on both the time and feature spaces by taking the outer product of the time attentions α and a scaled version of KG-adjusted feature attention ϵ to obtain the double attention. In addition, the feature embeddings are then adjusted by the obtained double attention. The double attention Θ and the adjusted feature embeddings \mathbf{R} are computed as follows:

$$\Theta = \epsilon_s \otimes \alpha \quad (4.15)$$

$$\mathbf{R} = \mathbf{H}^C \odot \Theta \quad (4.16)$$

where Θ and $\mathbf{R} \in \mathbb{R}^{N_{gm} \times T}$. $\epsilon_s \in \mathbb{R}^{N_{gm}}$ is the scaled feature attention which is computed by taking the ratio between each KG-adjusted feature attention to the first KG-adjusted feature attention ϵ_0 so that the proportion of attentions are maintained, as shown in Equation 4.17.

$$\epsilon_s = \epsilon / \epsilon_0 \quad (4.17)$$

Finally, the double-attention adjusted feature embeddings \mathbf{R} are passed into the pooling layer for taking the sum/max over each time step, followed with a dense layer with sigmoid function for final predictions.

4.2.3.4 Loss Function

To consolidate the concept relations from KG in attentions, a regularization term is added to the original prediction loss function. The new regularization term minimizes the discrepancy between the pairwise attention-based distances and the pairwise KG-embedding distances.

Let the ground truth label be y and the predicted label be \hat{y} , we use the cross-entropy for the original prediction loss denoted as L_{pred} , and the regularization term

is denoted as L_{reg} , the final loss L is defined as:

$$L = L_{pred} + L_{reg} \quad (4.18)$$

$$L_{pred} = \sum_{k=1}^{N_s} -(y_k \log(\hat{y}_k) + (1 - y_k) \log(1 - \hat{y}_k)) \quad (4.19)$$

$$L_{reg} = \sum_{i=1}^{N_g-1} \sum_{j=i+1}^{N_g} (dist(\Theta_i, \Theta_j) - KG_{dist}(i, j)) \quad (4.20)$$

where Θ represents the double attentions, KG_{dist} is the KG-embedding distance discussed in 4.2.2.3, and i, j are the group indexes, N_s is the number of samples, k is the sample index.

4.3 Experiment Settings

The performance of KGDAL was evaluated using two (proprietary and public) AKI-D datasets.

4.3.1 Data Preprocessing

4.3.1.1 Proprietary EHR Data

The proprietary EHR data include 608 AKI-D patients who were admitted to the University of Kentucky (UK) HealthCare from January 2009 to October 2019. Among them, 247 (41%) died in the hospital and 361 (59%) survived. This cohort excluded patients who were less than 18 years old, or were diagnosed with end-stage kidney disease (ESKD) at the time of index hospital admission, or were recipients of kidney transplant. The EHR records during renal replacement therapy (RRT), including both haemodialysis (HD) and continuous renal replacement therapy (CRRT), were extracted. The duration of RRT was limited from 72 hours to 2,000 hours. Any records beyond this range were excluded.

Twelve types of temporal features were collected from EHR, which were systolic blood pressure, diastolic blood pressure, creatinine, bicarbonate, hematocrit, potas-

sium, bilirubin, sodium, temperature, white blood cells (WBC) count, heart rate, and respiratory rate. Six types of static features were also collected, which were demographics (age, race, and gender), admission weight, body mass index (BMI), and Charlson comorbidity score. Three status flags were constructed, which indicated the on or off of CRRT or HD, and the status of being in the ICU or not. In total, 21 features were included in the UK data and the average missing rate of the temporal features was 58.7%.

For each feature, outlier values greater than 97.5 percentile or below 2.5 percentile were both excluded. The temporal granularity of the temporal features was set to one value (median) per hour. Linear interpolation was employed to fill the gaps between two actual measurements if needed. The only exception is creatinine, for which we only kept one value every six hours to maintain the in-practice frequency.

4.3.1.2 Public EHR Data

The public data were extracted from the MIMIC-III [116]. We first identified all the AKI patients by the presence of ICD-9 codes of 584.5 to 584.9, then we identified AKI-D patients with the additional presence of ICD-9 procedure codes of 3995 as well as diagnosis codes of V45.11 and V561 [117]. Applying the same cohort exclusion criteria resulted in the MIMIC-III data with 170 AKI-D patients. Among them, 66 (39%) died in the hospital and 104 (61%) survived. The temporal features of MIMIC-III were the same as the UK data except for WBC and temperature, since neither of them was available in the RRT duration. The average missing rate of the temporal features was 49.4%. The same static features and status flags as those in the UK data were included in the MIMIC-III data. In total, 19 features were included in the MIMIC-III data. The same data extraction and outlier detection procedures were conducted.

4.3.1.3 Knowledge Map

The Human phenotype ontology (HPO) was used as the clinical knowledge map to learn the concept embeddings. HPO, a widely used biomedical ontology, provides a standardized vocabulary of phenotypic abnormalities encountered in human disease [118]. Concepts in HPO are organized in hierarchies. Among the six sub-hierarchies on the top level, we focused on the “Phenotypic abnormality” sub-hierarchy since it includes most of the concepts of abnormalities related to the selected features in this experiment. To represent the strength of the relations between any two concepts, we counted the number of the common ancestors of the two concepts. Due to the large number of unique descendants in the “Phenotypic abnormality” sub-hierarchy (15,560), we computed the relations between every concept and “Acute Kidney Injury” (HP:0001919) and 100 random selected concepts. Finally, we obtained 1,566,363 identical concept triplets (see definition in Section 4.2.2) from HPO, where the number of common ancestors was considered as the relation strength between two concepts.

4.3.1.4 Experimental Data Generation

From each patient’s EHR sequence during RRT, we randomly generated at most 30 subsequences with their lengths varying from 48 to 96 hours. For each subsequence in the UK data, the possible class labels are whether the patient died (positive) or survived (negative) 24, 48, or 72 hours after the end of that subsequence. For the MIMIC-III data, the possible class labels of each subsequence are whether the patient died (positive) or survived (negative) 48 or 72 hours after the end of that subsequence. Note that due to the small study cohort size, the “24 hours” label was absent from the MIMIC-III data. In summary, the UK data include 14,757, 15,468, and 16,660 subsequences labeled as negative (alive) and 3,455, 2,744, and 1,552 subsequences labeled as positive (mortality) in the next 72, 48 and 24 hours respectively. For the

Table 4.1: Training, validation and testing data at the subsequence level.

	Proprietary EHR dataset (UK data)						Public EHR data (MIMIC-III)			
	In Next 72h		In Next 48h		In Next 24h		In Next 72h		In Next 48h	
	Alive	Death	Alive	Death	Alive	Death	Alive	Death	Alive	Death
Train	12275	2643	12817	2101	13717	1201	2725	391	2831	285
Validation	1077	419	1147	349	1305	191	1281	211	1337	155
Test	1405	393	1504	294	1638	160	385	125	406	104

MIMIC-III data, 4,391 and 4,574 subsequences were labeled as negative (alive); 727 and 544 subsequences were labeled as positive (mortality) in the next 72 and 48, respectively. The ratio between positive and negative ranged between 10% and 25%.

Table 4.1 showed the data used for training, validation, and testing. All the subsequences of 50 randomly selected patients (25 died vs. 25 alive) were used for validation. All the rest patient data were randomly split into training (90%) and testing (10%).

The data split was patient-wise so that the subsequences from the same patient only appeared in one of the three datasets.

4.4 Results

4.4.0.1 Concepts Embedding and Concepts Grouping

Figure 4.3 shows the HPO concepts matched to the temporal features in both patient cohorts. These concepts were clustered into four groups based on the structure of the “phenotypic abnormality” sub-hierarchy in HPO, which were “Abnormality of the cardiovascular system (Cardiovascular)”, “Abnormality of metabolism/homeostasis (Metabolism)”, “Abnormality of blood and blood-forming tissues (Blood)”, and “Abnormality of the respiratory system (Respiratory)”. Figure 4.3A shows the partial hierarchical structure of HPO where colors indicate concept groups.

The concept embeddings for all the selected concepts were obtained by training the

TransE model using all the HPO concept triplets. The resulting concept embeddings were used to compute two types of pairwise distances, i.e., distances between any two concept embedding groups and distances from the concept-of-interest “AKI” to each concept embedding group. The concept-wise distances were visualized using the t-SNE plot in Figure 4.3B. “Respiratory” is mostly related to “AKI” (averaged distance 0.02), while “Cardiovascular” is the most distant from “AKI” (averaged distance 0.98).

4.4.0.2 Performance Comparison

We compared KGDAL with various mortality rolling prediction models on both the UK data and the MIMIC-III data. In addition, an ablation study was conducted to test whether KGDAL’s KG-adjusted feature attentions were critical in mortality rolling prediction by 1) only using the time attention mechanism, or 2) removing from the loss the KG adjustment that minimizes the discrepancy between the pairwise attention based distance and the pairwise KG-embedding distances. We compared all

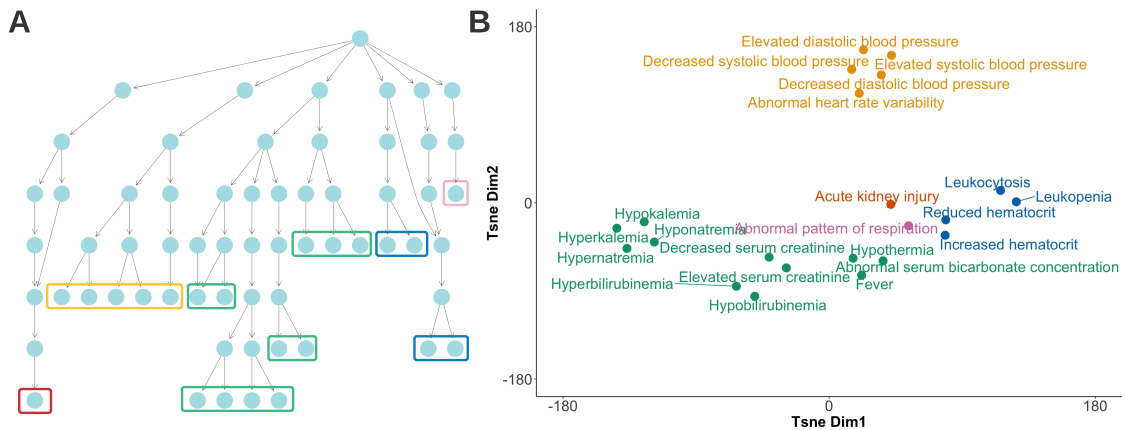


Figure 4.3: A: The partial hierarchical structure of the Human Phenotype Ontology (HPO) that includes the following concepts. Colors indicate different concept groups (Red: Acute Kidney injury (AKI); Orange: ”Cardiovascular”; Green: ”Metabolism”; Blue: ”Blood”; Pink: ”Respiratory”). B: The similarities of the same selected features in a projected space generated using t-SNE.

the models’ performance on both the balanced and imbalanced test sets, the ratio of positive samples (died) to negative samples (survived) are 1:1 and 1: 2, respectively. By randomly sampling five times for each case, we reported the averaged performance on all the evaluation metrics. All the compared models and their inputs are described as following:

- **Random Forest:** The input to this model is the un-grouped temporal features where each temporal feature at each time step is appended as a column, and each static feature is a column. Column-wise mean imputation is used to fill the missing values.
- **Boosted Tree:** We use Extreme Gradient Boosting (XGBoost) [12] as the second baseline model. The input to this model is the same as Random Forest.
- **LSTM:** The input to a LSTM is the un-grouped temporal features and static features concatenated at each time point.
- **Transformer:** The input to a Transformer is the same as LSTM. We use the encoder part of the original transformer with a dense layer for the prediction task.
- **KGDAL $_{\alpha}$:** This KGDAL model only uses the time attention mechanism. By removing feature attention, it explores the usefulness of the feature attention mechanism. The inputs are the grouped (KG-guided) features.
- **KGDAL $_{\alpha\beta}$:** This KGDAL model uses both time attention and feature attention, but it removes the KG-adjusted attention and the KG-adjusted loss from KGDAL. It explores the usefulness of the KG-adjusted attention mechanism. The inputs are the grouped (KG-guided) features.

The performance of mortality rolling prediction in the next 72, 48, and 24 hours on the UK data are listed in Table 4.2, Table 4.3, and Table 4.4 respectively. The

performance of mortality rolling prediction in the next 72 and 48 hours on MIMIC-III data are listed in Table 4.5 and Table 4.6 respectively. In all the tables, precision (PREC), recall (REC), and F1 scores are for the positive (died) class.

As a baseline, a random model achieved ROCAUC 0.52, accuracy 0.52, precision 0.52, recall 0.51, and F1 0.52. The random forest model predicted that almost every patient survived in all experiments. Its average performance on all experiments are ROCAUC 0.50, accuracy 0.58, precision 0.25, recall 0.01 and F1 0.02. The XGBoost model performed better than random forest. However, it is only slightly better than the random model. The performance of deep learning models, including LSTM and Transformer, performed significantly better than the compared traditional machine learning models, indicating that for temporal data based rolling prediction, deep learning models can better capture the critical temporal patterns, resulting in better performance than that of traditional machine learning methods.

Table 4.2 and Table 4.3 show that KGDAL has the best performance on 72 and 48-hour rolling prediction on the UK data on almost all evaluation metrics on both balanced and imbalanced test data. The second-best model is KGDAL $_{\alpha}$ followed by KGDAL $_{\alpha\beta}$ and Transformer. However, Table 4.4 shows that KGDAL $_{\alpha}$ outperformed KGDAL on both the balanced and imbalanced test data for the 24-hour mortality rolling prediction, even though KGDAL has the best recall. It suggests that for

Table 4.2: Performance of morality prediction in the next 72 hours during RRT (UK data)

	$N_{pos} : N_{neg} = 1 : 1$					$N_{pos} : N_{neg} = 1 : 2$				
	AUC	ACC	PREC	REC	F1	AUC	ACC	PREC	REC	F1
XGBoost	0.50	0.50	0.51	0.35	0.42	0.50	0.55	0.33	0.35	0.34
LSTM	0.62	0.61	0.60	0.67	0.63	0.63	0.60	0.44	0.67	0.53
Transformer	0.70	0.64	0.64	0.64	0.64	0.69	0.63	0.46	0.64	0.53
KGDAL $_{\alpha}$	0.75	0.66	0.64	0.77	0.69	0.75	0.64	0.47	0.77	0.59
KGDAL $_{\alpha\beta}$	0.70	0.63	0.63	0.63	0.63	0.70	0.63	0.46	0.63	0.53
KGDAL	0.76	0.71	0.66	0.87	0.75	0.74	0.64	0.48	0.87	0.62

Table 4.3: Performance of morality prediction in the next 48 hours during RRT (UK Data)

	$N_{pos} : N_{neg} = 1 : 1$					$N_{pos} : N_{neg} = 1 : 2$				
	AUC	ACC	PREC	REC	F1	AUC	ACC	PREC	REC	F1
XGBoost	0.50	0.49	0.49	0.35	0.41	0.50	0.55	0.33	0.35	0.34
LSTM	0.61	0.60	0.59	0.67	0.63	0.62	0.59	0.43	0.67	0.52
Transformer	0.69	0.63	0.62	0.66	0.64	0.70	0.63	0.46	0.66	0.54
KGDAL $_{\alpha}$	0.74	0.67	0.63	0.80	0.70	0.75	0.63	0.47	0.80	0.59
KGDAL $_{\alpha\beta}$	0.70	0.64	0.63	0.66	0.64	0.72	0.64	0.47	0.66	0.55
KGDAL	0.73	0.68	0.63	0.88	0.74	0.74	0.63	0.47	0.88	0.61

Table 4.4: Performance of morality prediction in the next 24 hours during RRT (UK Data)

	$N_{pos} : N_{neg} = 1 : 1$					$N_{pos} : N_{neg} = 1 : 2$				
	AUC	ACC	PREC	REC	F1	AUC	ACC	PREC	REC	F1
XGBoost	0.50	0.51	0.51	0.37	0.43	0.50	0.55	0.34	0.37	0.35
LSTM	0.62	0.62	0.61	0.69	0.65	0.61	0.58	0.42	0.69	0.53
Transformer	0.72	0.65	0.64	0.71	0.67	0.71	0.63	0.46	0.71	0.56
KGDAL $_{\alpha}$	0.78	0.71	0.66	0.89	0.76	0.78	0.65	0.49	0.89	0.63
KGDAL $_{\alpha\beta}$	0.75	0.68	0.66	0.76	0.71	0.75	0.65	0.48	0.76	0.59
KGDAL	0.75	0.70	0.64	0.94	0.76	0.75	0.62	0.47	0.94	0.62

Table 4.5: Performance of morality prediction in the next 72 hours during RRT (MIMIC-III)

	$N_{pos} : N_{neg} = 1 : 1$					$N_{pos} : N_{neg} = 1 : 2$				
	AUC	ACC	PREC	REC	F1	AUC	ACC	PREC	REC	F1
XGBoost	0.51	0.49	0.47	0.15	0.23	0.50	0.59	0.28	0.15	0.20
LSTM	0.70	0.64	0.80	0.38	0.52	0.69	0.72	0.65	0.38	0.48
Transformer	0.64	0.60	0.69	0.38	0.49	0.62	0.67	0.50	0.38	0.44
KGDAL $_{\alpha}$	0.57	0.38	0.08	0.02	0.04	0.56	0.49	0.04	0.02	0.03
KGDAL $_{\alpha\beta}$	0.69	0.72	0.97	0.46	0.62	0.68	0.81	0.97	0.46	0.62
KGDAL	0.65	0.59	0.58	0.62	0.60	0.63	0.56	0.40	0.62	0.48

Table 4.6: Performance of morality prediction in the next 48 hours during RRT (MIMIC-III)

	$N_{pos} : N_{neg} = 1 : 1$					$N_{pos} : N_{neg} = 1 : 2$				
	AUC	ACC	PREC	REC	F1	AUC	ACC	PREC	REC	F1
XGBoost	0.50	0.50	0.49	0.17	0.25	0.50	0.60	0.32	0.17	0.22
LSTM	0.71	0.64	0.79	0.38	0.52	0.70	0.71	0.61	0.38	0.47
Transformer	0.66	0.59	0.67	0.37	0.47	0.65	0.65	0.48	0.37	0.41
KGDAL $_{\alpha}$	0.59	0.39	0.08	0.02	0.03	0.57	0.49	0.03	0.02	0.02
KGDAL $_{\alpha\beta}$	0.67	0.70	0.92	0.44	0.60	0.66	0.79	0.84	0.44	0.58
KGDAL	0.62	0.56	0.56	0.59	0.57	0.61	0.53	0.37	0.59	0.46

shorter prediction windows, the time attention mechanism has more contribution than the feature attention mechanism with/without KG adjustment, and KG-adjusted feature attention can improve the overall model performance slightly. In summary, all the experiments on UK data show that the attention-based models including Transformer, KGDAL $_{\alpha}$, KGDAL $_{\alpha\beta}$, have better overall performance than LSTM, indicating the attention per se is critical for rolling prediction tasks.

On the MIMIC-III data, KGDAL $_{\alpha\beta}$ has the best overall performance for both 72 and 48-hour rolling mortality prediction. While LSTM has the highest ROCAUC and KGDAL achieves better recall, KGDAL $_{\alpha\beta}$ has higher scores on accuracy, precision, and F1. Surprisingly, KGDAL $_{\alpha}$, which is the second-best model on the UK data, has the lowest performance among the three in all the experiments on the MIMIC-III data. The performance of KGDAL $_{\alpha}$ on this experiment suggests that it is feature attentions rather than time attentions that play the critical role in MIMIC-III rolling mortality prediction.

In summary, we found that KGDAL’s time and feature attentions are essential in rolling mortality prediction. While on one data, time attention is more important; feature attention could dominate the model performance on another data. The KG-guided grouping is crucial for all experiments and datasets. Nevertheless, other factors, such as sample size or patient distribution, may also affect performance.

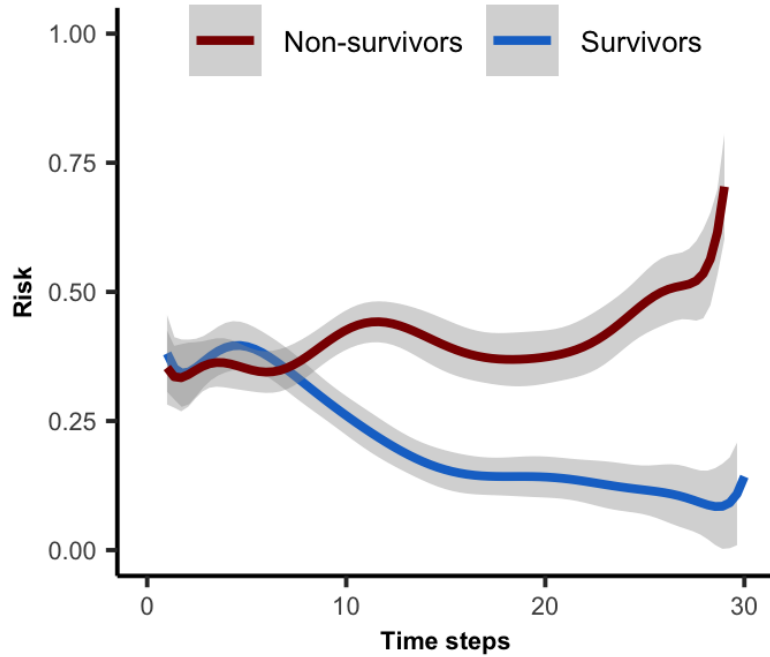


Figure 4.4: Two risk trajectory clusters with different endings.

4.4.0.3 Mortality Risk Trajectory Analysis

A patient’s mortality risk trajectory is a series of predicted risks of all the subsequences of that patient ordered by time. An example trajectory is shown in Figure 4.5. Given all the mortality risk trajectories, we computed the pairwise trajectory distances using dynamic time wrapping [119], and used hierarchical clustering to identify similar risk trends among all the correctly predicted patients. Two trajectory clusters shown in Figure 4.4 revealed multiple episodes of increasing risks for non-survivors and quick decreasing risks for survivors. The trend-based analysis may assist healthcare providers in making early decisions before the risk increases.

As a case study, we visualized the mortality risk trajectory of an AKI-D patient in Figure 4.5. In the figure, the x-axis is the days before outcome event (survival (end of follow-up) / mortality), the y-axis is the predicted risk of death, and the risk scores range from 0 (survived) and 1 (died). Every blue point is the predicted mortality risk from a subsequence of the same patient. All the points were fitted

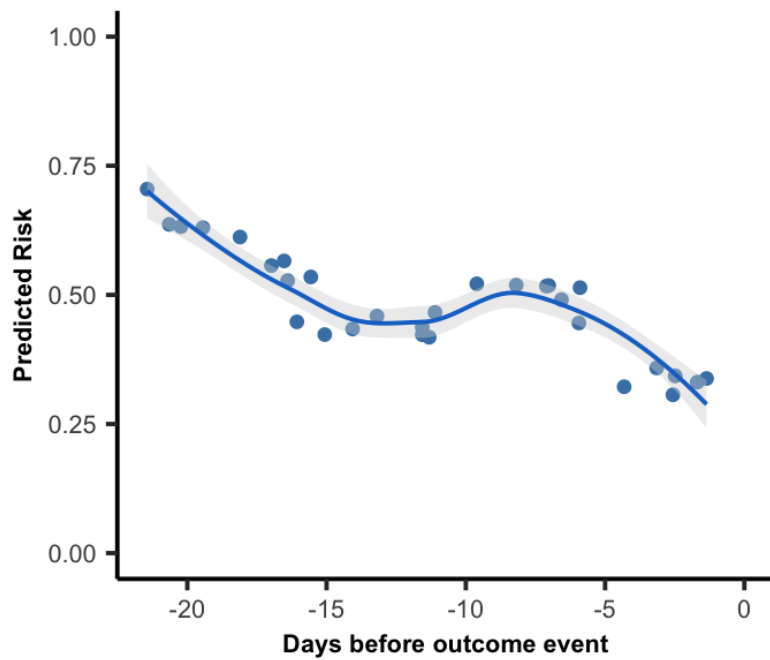


Figure 4.5: The risk trajectory of a survival patient.

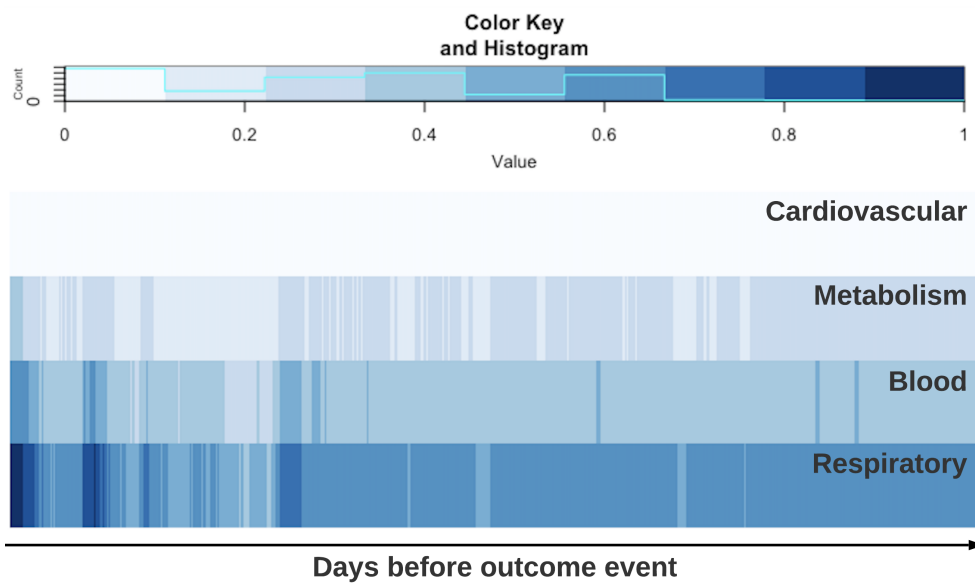


Figure 4.6: An example of the KG-adjusted 2-D attentions.

to a smooth curve using polynomial regression showing the trajectory of predicted mortality risks. In the case study, the patient finally survived. However, the risk score was not monotonically decreasing. Starting with a high risk, KGDAL predicted that the risk was gradually decreased for 5 days. The risk trajectory then stayed roughly at 50% for 10 days with mild fluctuations. Finally, the trajectory decreased quickly in the last 5 days of RRT.

The corresponding KG-adjusted attentions are shown in Figure 4.6, where each row represents the attentions of a feature group, each column is a time point (the darker the color, the high the score). The risk trajectory aligned well with the KG-adjusted attention. The attention hit three times the maximum in the early RRT duration, indicating a high mortality risk at that time. The figure also reveals that “Respiratory” and “Blood” had overall higher attentions than the other concept groups. This is well-aligned with the observations in the knowledge graph that “Respiratory”, “Blood”, and “AKI” are closely related concepts. The high agreement between the risk trajectory and the time and feature attentions suggests that the attentions obtained from KGDAL may be useful to explain to clinicians the potential risks and why the risk is high or low so that interventions can be taken in place timely.

4.5 Conclusion

In this chapter, we presented a novel model called KGDAL for rolling mortality prediction for AKI-D patients. KGDAL uses a knowledge graph to guide the generation of 2D attention in both time and feature spaces. KGDAL and its variations achieved the best performance on both the UK data and the MIMIC-III data. Using a case study, we demonstrated the interpretability of KGDAL and the capability of using KGDAL for assisting timely decisions for clinicians.

CHAPTER 5. KIT-LSTM: Knowledge-guided Time-aware LSTM for Continuous Risk Prediction for Acute Kidney Injury Patients Requiring Dialysis

This chapter introduces our novel deep learning architecture to address the irregularity, asynchronous, and interpretability concerns for temporal EHR data.

5.1 Introduction

Clinical risk prediction using Electronic Health Record (EHR) data provides accurate and timely individualized patient outcomes, allowing early interventions for high-risk patients and better-allocating hospital resources [120, 6]. It is particularly critical to predicting risks for patients with Acute Kidney Injury requiring Dialysis (AKI-D), a severe complication associated with a very high mortality rate for critically ill patients [5, 121].

Artificial intelligence (AI), esp. deep learning (DL) models, have drawn increasing attention to patients' outcome predictions using temporal EHR data [25, 122]. However, due to complicated data collection procedures and strict data management, EHR data are not generally AI-ready, which hinders the adaption of AI tools directly in the clinical settings [123, 124, 125].

Firstly, EHR data are collected daily in hospitals for efficient patient care delivery but are usually not in ideal shape for ML/DL models [25], with temporal irregularity and asynchrony being the most common problems encountered when building ML/DL applications in clinical settings [2, 126]. Irregularity refers to the uneven time gaps between measurements of a single feature. Asynchrony refers to the unaligned measurements across multiple features. Figure 5.1 shows an example of EHR data with three clinical variables (SBP, HCT, and sCr) collected in the ICU. The green and blue lines in HCT show irregular time gaps between the measurements of a single

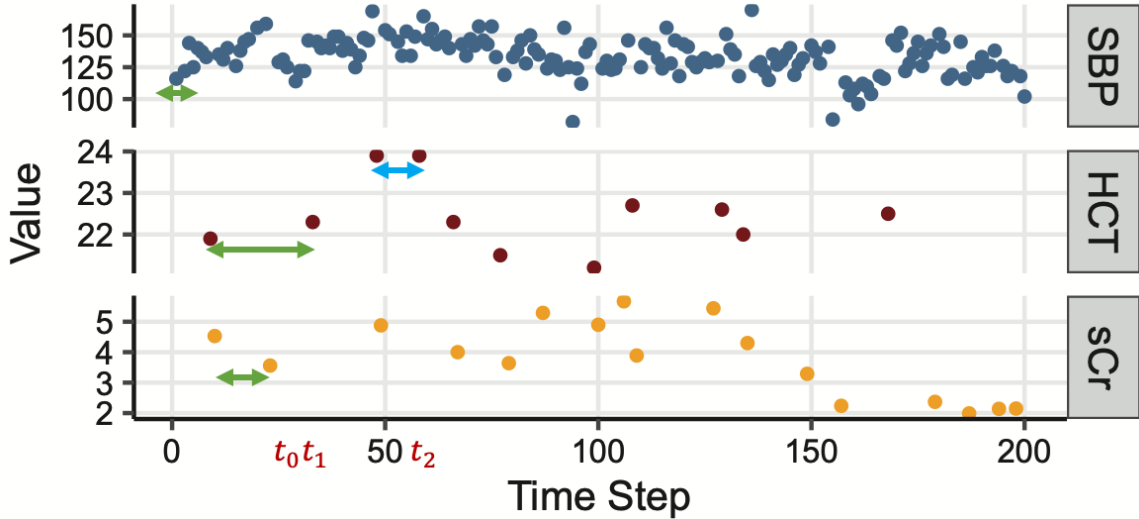


Figure 5.1: An example of real-world EHR data in the ICU. “SBP” stands for systolic blood pressure, “HCT” for Hematocrit, and “sCr” for serum creatinine. Arrows highlight irregular and asynchronous gaps between measurements.

clinical parameter, while the three green lines in SBP, HCT, and sCr show unaligned observations across the three clinical variables with different measurements frequency.

Early DL methods ignore the irregularity and asynchrony problem. For example, the standard LSTM [16] assumes equal temporal gaps between time steps, and the original Transformer [15] uses absolute positions encoding. Recent LSTM variants have been focused on addressing the irregularity problem. T-LSTM [127] considered time elapse between patients’ visits. Phased-LSTM [128] introduced a time gate in the LSTM cell to update cell states and hidden states only if the time gate is open. Nevertheless, most of the existing LSTM models still ignore the asynchrony problem, as illustrated in Figure 5.1. With recurrent neural networks (RNN), the irregularity and asynchrony problems have been addressed using missingness patterns and time elapsed between measurements. The GRU-D model updates GRU cells using missing patterns and decayed input/hidden state according to the elapsed time [129]. The BRITS model estimates missing values by introducing a complement input variable in the RNN unit when the variable is missing and also uses the hidden time decayed state

of the RNN unit [130]. However, the missingness patterns are not always informative and are challenging to interpret.

Secondly, the accountability of DL models in terms of model interpretation in healthcare practice is critical for clinicians to make decisions based on the model results and rationale [22]. Model-agnostic methods, such as LIME [47] and SHAP [48], support the interpretation of any ML/DL method in a post-hoc fashion. Nevertheless, using these methods needs an extra and separate step to training the actual ML/DL models. Self-interpretable DL models such as RETAIN [55] use two-level attention scores for model interpretation, but cannot address the irregularity and asynchrony problem in EHR. ATTAIN [131] builds time-wise attention based on all/some previous cell states of LSTM plus a time-aware decay function for resolving the irregular time gaps issues. The trade-off between interpretability and prediction power invokes the development of self-interpretable ML/DL models without sacrificing prediction power, promoting a better adoption in routine uses in practical healthcare settings [23].

Another model interpretation approach uses domain-specific knowledge encoded in medical or biological ontologies databases as prior knowledge [59]. A knowledge-driven ML model that utilizes ontologies databases may gain better interpretation and potentially higher prediction power [66, 25]. Recent studies [64, 65, 63] have incorporated medical knowledge graphs into medical applications using translation-based graph embeddings methods [61, 132]. Moreover, medical knowledge-graph-based attention models such as GRAM [66], DG-RNN [67], and KGDAL [133] have demonstrated comparable performance as well as the power of result interpretation. Nevertheless, these methods do not embed knowledge for numerical features and lack a mechanism for handling irregular and asynchronous EHR data.

In this article, we present a Knowledge guided Time-aware LSTM model (KIT-LSTM), which handles irregular and asynchronous time series EHR data, and uses medical ontology to guide the attention between multiple numerical clinical variables,

and provides knowledge-based model interpretation.

KIT-LSTM extends LSTM with two time-aware gates and a knowledge-aware gate. The time-aware gates adjust the memory content according to two types of elapsed time, i.e., the elapsed time since the last visit for all variable streams and the elapsed time since the last measured values for each variable stream. The knowledge-aware gate uses medical ontology to guide attention between multiple numerical variables at each time step. To the best of our knowledge, KIT-LSTM is the first LSTM variant that incorporates medical ontology with the addition of two time-aware gates to guide the attention mechanism inside the LSTM cell. As a result, the proposed model provides better guidance for attention and interpretation and handles both irregular and asynchronous problems simultaneously. Our contributions are summarized as follows:

- 1) KIT-LSTM adds to the original LSTM cell two unique time-aware gates. The time-aware gates adjust different proportions of the LSTM cell memory contents, which address irregularity and asynchrony.

- 2) KIT-LSTM adds to the original LSTM cell a knowledge-aware gate. It uses the relationship between concepts learned from medical ontology to guide attention between multiple variables at each time step, and the loss function enforces the learned attention aligned with the medical ontology, enabling knowledge-based model interpretation.

- 3) Using EHR data, KIT-LSTM continuously and accurately predicts mortality risks in the next 24 hours for critically ill AKI-D patients in the ICU.

- 4) KIT-LSTM shows high robustness in subpopulation distribution shift.

5.2 Method

5.2.1 Notations

EHR features: A patient’s EHR data at time step t can be represented as a vector of clinical parameters (e.g., heart rate) denoted as $\mathbf{x}_t \in \mathbb{R}^{N_f}$, where N_f is the number of features.

Time-related features: $\Delta_t \in \mathbb{R}$ denotes the time elapsed since the last time step, and $\Delta'_t \in \mathbb{R}^{N_f}$ denotes the time elapsed since the last measured value of the same feature. The value of Δ'_t for each feature can be different since features are possibly measured at different frequencies.

Knowledge-related features: We use Human Phenotype Ontology (HPO) [115] as the prior knowledge to guide the model learning process. We extract ontology concepts related to the selected clinical features and call them feature concepts (e.g., “elevated systolic blood pressure” (HP:0004421) is a concept related to systolic blood pressure in HPO). In addition, we extract a concept related to the study population, i.e., “acute kidney injury” (HP:0001919), which is called the target concept. The total number of ontology concepts is $N_o + 1$, where N_o is the number of feature concepts. Note that N_o is not the same as the number of features N_f because some features can be mapped to more than one related ontology concept, and some can only be mapped to one. All concepts are encoded as one-hot vectors as the initial embeddings denoted as $\mathbf{O} \in \mathbb{R}^{(N_o+1) \times (N_o+1)}$.

Based on the feature values and the ontology concepts at time step t , we extract the physiological status denoted as $\mathbf{p}_t \in \mathbb{R}^{N_o}$. The value of p_t for each concept is either 0 or 1. For example, $p_t = 1$ for the concept “high systolic blood pressure” means patient systolic blood pressure is greater than 130. Thresholds of all features are defined based on clinical practice and are validated by clinicians.

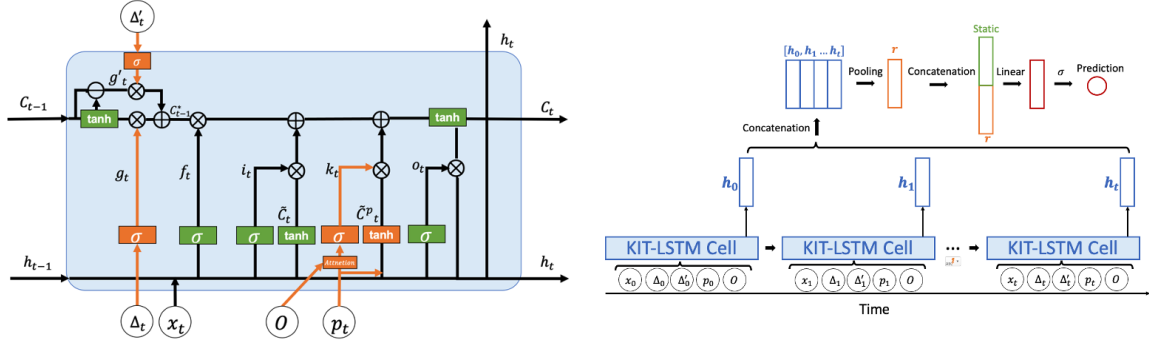


Figure 5.2: The architecture of KIT-LSTM cell (left), orange represents the unique gates in KIT-LSTM, and green represents the original gates in LSTM. The prediction layers (right) combine all the hidden states learned from KIT-LSTM and the static features, such as patient demographics, for the final prediction.

5.2.2 KIT-LSTM Cell

The architecture of KIT-cell is illustrated in Figure 5.2. The input to a KIT-LSTM cell consists of five components: clinical feature \mathbf{x}_t , elapsed time Δ_t since last time step, elapsed time Δ'_t since the last measured values, physiological status \mathbf{p}_t , and initial concept embedding \mathbf{O} .

KIT-LSTM kept the original three gates (forget, input, output) from LSTM and added three additional gates (two time-aware gates and one knowledge-aware gate).

The first time gate, the long-term time-aware gate, is a time decay function that adjusts long-term memory by using the elapsed time since the last measured value of the same clinical parameter. For example, the green and blue arrows in Figure 5.1 for the ‘‘HCT’’ indicate that the previous memory will be more likely to be discounted when the green arrow ends (t_1) than the time when the blue arrow ends (t_2).

The second time gate, the short-term time-aware gate, is a time decay function to adjust short-term memory. It measures the elapsed time since the last time step (e.g, $t_1 - t_0$ in Figure 5.1), similar to the time decay function in T-LSTM [127].

The long and short time-aware gates control how the previous short or long-term memories can be passed into the current memory. Intuitively, the longer the elapsed

times, the less likely the long or short-term memory gate will open.

The knowledge-aware gate uses concepts embeddings and physiological features to control which feature should be paid more attention to at each time step. Intuitively, if the physiological status is abnormal and there is a strong relationship between a feature concept and the target concept, more attention will be paid to the corresponding feature.

Gate update: We denoted the forget gate as \mathbf{f}_t , the input gate as \mathbf{i}_t , the output gate as \mathbf{o}_t , the time-aware gates as \mathbf{g}_t and \mathbf{g}'_t , and the knowledge-aware gate as \mathbf{k}_t , where $\mathbf{f}_t, \mathbf{i}_t, \mathbf{o}_t, \mathbf{g}_t, \mathbf{g}'_t, \mathbf{k}_t \in \mathbb{R}^m$, m is the dimension of the hidden vectors, and t is the time step. The short-term time-aware gate \mathbf{g}_t is updated by the elapsed time since the last time step, and the long-term time-aware gate \mathbf{g}'_t is updated by the elapsed time since last measured value for each feature. The knowledge-aware gate \mathbf{k}_t is updated by the attention scores $\boldsymbol{\alpha}_t \in \mathbb{R}^{N_o}$ learned from the concept embeddings as well as physiological status \mathbf{p}_t . Gates at time step t are: ¹

$$\mathbf{g}_t = \sigma(1/\Delta_t) \quad (5.1)$$

$$\mathbf{g}'_t = \sigma(\mathbf{W}_g(1/\sigma(\Delta'_t)) + \mathbf{b}_g) \quad (5.2)$$

$$\mathbf{k}_t = \sigma(\mathbf{W}_k\boldsymbol{\alpha}_t + \mathbf{b}_k) \quad (5.3)$$

$$\mathbf{f}_t = \sigma(\mathbf{W}_f\mathbf{x}_t + \mathbf{U}_f\mathbf{h}_{t-1} + \mathbf{b}_f) \quad (5.4)$$

$$\mathbf{i}_t = \sigma(\mathbf{W}_i\mathbf{x}_t + \mathbf{U}_i\mathbf{h}_{t-1} + \mathbf{b}_i) \quad (5.5)$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_o\mathbf{x}_t + \mathbf{U}_o\mathbf{h}_{t-1} + \mathbf{b}_o) \quad (5.6)$$

where $\mathbf{W}_g \in \mathbb{R}^{m \times N_f}$, $\mathbf{W}_k \in \mathbb{R}^{m \times N_o}$, $\mathbf{W}_f, \mathbf{W}_i, \mathbf{W}_o \in \mathbb{R}^{m \times N_f}$, $\mathbf{U}_f, \mathbf{U}_i, \mathbf{U}_o \in \mathbb{R}^{m \times m}$, and $\mathbf{b}_g, \mathbf{b}_k, \mathbf{b}_f, \mathbf{b}_i, \mathbf{b}_o \in \mathbb{R}^m$ are the learnable parameters. σ is a sigmoid function.

¹ Δ_t is repeated for every hidden dimension, thus $\boldsymbol{\Delta}_t \in \mathbb{R}^m$

The attention score α_t for each physiological feature at time t is computed using:

$$\alpha_t = \text{softmax}(\beta_t) \quad (5.7)$$

$$\beta_t = \mathbf{e}_t \odot \mathbf{p}_t \quad (5.8)$$

$$\mathbf{e}'_t = \mathbf{O}\mathbf{W}_e + \mathbf{b}_e \quad (5.9)$$

where $\mathbf{W}_e \in \mathbb{R}^{(N_o+1)}$ and $\mathbf{b}_e \in \mathbb{R}^{(N_o+1)}$ are the learnable parameters; and $\mathbf{e}'_t \in \mathbb{R}^{(N_o+1)}$ is the learned concepts embedding from initial embeddings \mathbf{O} using a linear function including the feature concepts embeddings $\mathbf{e}_t \in \mathbb{R}^{N_o}$ and the target concept embedding denoted as $e_{target} \in \mathbb{R}$. Note that e_{target} is not shown in above equation, but will be used in loss regularization described in Section 5.2.3.

Memory cell update: Following the definition in T-LSTM [127], we extracted the short and long-term memory from the previous memory cell $\mathbf{C}_{t-1} \in \mathbb{R}^m$, denoted as $\mathbf{C}_{t-1}^S \in \mathbb{R}^m$ and $\mathbf{C}_{t-1}^L \in \mathbb{R}^m$ respectively. Then, the short- and long-term memory are adjusted separately by their corresponding time-aware gates \mathbf{g}_t and \mathbf{g}'_t . In particular, the short-term memory is discounted by the elapsed time since last time step, and the long-term memory is discounted by the elapsed time since last measured values. We denote the discounted short-term and long-term memory cell as $\mathbf{C}_{t-1}^{DS} \in \mathbb{R}^m$ and $\mathbf{C}_{t-1}^{DL} \in \mathbb{R}^m$ respectively. Finally, the total adjusted previous memory cell denoted as $\mathbf{C}_{t-1}^* \in \mathbb{R}^m$ is the sum of all the discounted short- and long-term memories. Memory cells are computed as:

$$\mathbf{C}_{t-1}^S = \tanh(\mathbf{W}_d \mathbf{C}_{t-1} + \mathbf{b}_d) \quad (5.10)$$

$$\mathbf{C}_{t-1}^L = \mathbf{C}_{t-1} - \mathbf{C}_{t-1}^S \quad (5.11)$$

$$\mathbf{C}_{t-1}^{DS} = \mathbf{C}_{t-1}^S \odot \mathbf{g}_t \quad (5.12)$$

$$\mathbf{C}_{t-1}^{DL} = \mathbf{C}_{t-1}^L \odot \mathbf{g}'_t \quad (5.13)$$

$$\mathbf{C}_{t-1}^* = \mathbf{C}_{t-1}^{DL} + \mathbf{C}_{t-1}^{DS} \quad (5.14)$$

where $\mathbf{W}_d \in \mathbb{R}^{m \times m}$, $\mathbf{b}_d \in \mathbb{R}^m$ are the learnable parameters, and \odot is the Hadamard product.

Candidates memory cells: While the original feature candidate cell, denoted as $\tilde{\mathbf{C}}_t$, computed from the feature value \mathbf{x}_t and the previous hidden state \mathbf{h}_t , a new candidate cell is added, denoted as $\tilde{\mathbf{C}}_t^{\mathbf{p}}$, which also considers the physiological status \mathbf{p}_t . Candidate memory cells can be computed using:

$$\tilde{\mathbf{C}}_t = \tanh(\mathbf{W}_c \mathbf{x}_t + \mathbf{U}_c \mathbf{h}_{t-1} + \mathbf{b}_c) \quad (5.15)$$

$$\tilde{\mathbf{C}}_t^{\mathbf{p}} = \tanh(\mathbf{V}_p \mathbf{p}_t + \mathbf{W}_p \mathbf{x}_t + \mathbf{U}_p \mathbf{h}_{t-1} + \mathbf{b}_p) \quad (5.16)$$

where $\mathbf{W}_c, \mathbf{W}_p \in \mathbb{R}^{m \times N_f}$, $\mathbf{V}_p \in \mathbb{R}^{m \times N_o}$, $\mathbf{U}_c, \mathbf{U}_p \in \mathbb{R}^{m \times m}$, and $\mathbf{b}_c, \mathbf{b}_p \in \mathbb{R}^m$ are the learnable parameters.

Current memory cell and hidden state: $\mathbf{C}_t \in \mathbb{R}^m$ and $\mathbf{h}_t \in \mathbb{R}^m$ represent the current memory cell and its corresponding hidden state. \mathbf{C}_t is a combination of adjusted previous memory \mathbf{C}_{t-1}^* multiplied by the forget gate, the feature candidate memory multiplied by the input gate, and the physiological candidates memory multiplied by the knowledge-aware gate. The current cell and hidden state are computed using:

$$\mathbf{C}_t = \mathbf{f}_t \odot \mathbf{C}_{t-1}^* + \mathbf{i}_t \odot \tilde{\mathbf{C}}_t + \mathbf{k}_t \odot \tilde{\mathbf{C}}_t^{\mathbf{p}} \quad (5.17)$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t) \quad (5.18)$$

5.2.3 Patient Outcome Prediction

Prediction layer: All the hidden states \mathbf{h}_t are concatenated and passed into a pooling layer for taking the sum/max along time steps. The resulting hidden representation is denoted as $\mathbf{r} \in \mathbb{R}^m$. Then the static features (e.g, demographics) are concatenated with \mathbf{r} followed by a fully connected layer with a sigmoid function for the binary prediction. The process of final prediction is shown in Figure 5.2.

Loss function: Let the ground truth label be y and the predicted label be \hat{y} , we use the binary cross entropy as the part of the final prediction loss denoted as L_{pred} . Inspired by a knowledge graph guided model KGDAL [133], where a regularization

term is employed to consolidate the concept relations from medical ontology into attentions, we add a similar regularization term to ensure the relationship between the learned feature concept embeddings \mathbf{e} and the target concept embedding \mathbf{e}_{target} aligns to the observed relations in a medical ontology. Thus, the regularization term counts the discrepancy at the knowledge level, i.e. the difference between the learned concept embedding distance and the corresponding concept distance in a medical ontology. The regularization term L_{reg} , cross entropy loss L_{pred} , and final prediction loss L are:

$$L = L_{pred} + L_{reg} \quad (5.19)$$

$$L_{pred} = \sum_{i=1}^{N_s} -(y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)) \quad (5.20)$$

$$L_{reg} = \sum_{i=1}^{N_o} (dist_i^O - dist_i^L)^2 \quad (5.21)$$

$$dist_i^L = \sqrt{(e_i - e_{target})^2} \quad (5.22)$$

where N_s is the total number of samples. $dist_i^L$ represents the distance between feature concept embedding i and the target embedding; similarly, $dist_i^O$ represents the distance between feature concepts i and the target concept in the medical ontology. The observed distance can be obtained directly from the ontology graph by computing node-based distances, or it can be obtained from the pre-trained initial embedding using graph embedding methods [61].

The source code of KIT-LSTM is available at:

<https://github.com/lucasliu0928/KITLSTM>.

5.3 Experiment Settings

The experiment aims to continuously predict AKI-D patients' mortality risk in their dialysis/renal replacement therapy (RRT) duration. More specifically, given any pe-

Table 5.1: Training, Validation and Testing Data.

	Total N Samples (Patient)	Alive N Samples (Patient)	Death in the next 24h N Samples (Patient)	Negative to Positive Ratio (Patient)
Train (75% of patients)	9979 (432)	8843 (424)	1136 (125)	8:1 (3:1)
Validation (5% of patients)	642 (24)	586 (24)	56 (6)	10:1 (6:1)
Test (20 % of patients)	2712 (114)	2448 (113)	264 (25)	9 :1 (5:1)

riod of EHR in dialysis duration before time T , we will continuously predict the mortality risk at $T + 24$, i.e., 24 hours after T .

5.3.1 Experiment Data

Patient cohort: The study population consists of 570 AKI-D adult patients admitted to ICU at the University of Kentucky Albert B. Chandler Hospital from January 2009 to October 2019. Among them, 237 (41.6%) died in hospital, and 333 (58.4%) survived. Patients were excluded if they were diagnosed with end-stage kidney disease (ESKD) before or at the time of hospital admission, were recipients of a kidney transplant, or had RRT less than 72 or greater than 2,000 hours.

EHR data: Data features include twelve temporal features (systolic blood pressure, diastolic blood pressure, serum creatinine, bicarbonate, hematocrit, potassium, bilirubin, sodium, temperature, white blood cells (WBC) count, heart rate, and respiratory rate) and six static features (age, race, gender, admission weight, body mass index (BMI), and Charlson comorbidity score). All outliers greater than 97.5 or lower than 2.5 percentile were excluded. Measurement frequencies vary dramatically, ranging from 0.2 to 21.7 observations per day.

Sample generation: To continuously predict patient’s mortality risks, we generate 30 samples from each patient’s EHR data with a random start and end time as long as the length of the sample is greater than 10 time steps where a time step refers to the time when any feature has a value. The class label of a sample is whether the

patient died (positive) or survived (negative) in the next 24 hours from the end of the sample.

Obvious negative sample exclusion: The negative to the positive ratio in the data is 10:1 at the sample level. In such cases, ML models may be biased to the negative, resulting in so-called “good” performance. Hence, we excluded “obvious negative samples” from EHR, allowing models to focus on the more difficult cases and better balance positives and negatives in model training. This process is applied to all the compared methods to ensure fair performance comparison. The process is described as follows: 1) apply PCA [134] on the average, minimum or maximum values of all temporal and static features; 2) For each sample, compute the weighted sum of the top seven features using the squared correlation (contribution score) to the first principal component as weights; 3) determine obvious negative samples using the distribution of the weighted sum values. In total, 4,563 (or 25%) obvious negative samples were identified and excluded.

Training, validation and testing data: From 570 AKI-D patients, 13,333 EHR samples were extracted, including 1,456 positives and 11,877 negatives. As shown in Table 5.1, the data were split into training (75%), validation (5%), and testing data (20%) patient-wise to ensure that the samples from the same patients only appeared in one of the three datasets.

5.3.2 Baseline Algorithms

We compared KIT-LSTM with eight existing algorithms, including two traditional ML algorithms (XGBoost and SVM) and six DL models (LSTM, T-LSTM, Phased-LSTM, RETAIN, ATTAIN, and Transformer). For all DL models, static features are concatenated with the hidden states before the prediction layer, as described in Section 5.2.3. Moreover, all missing temporal features are imputed with the last observation carried forward (LOCF) method.

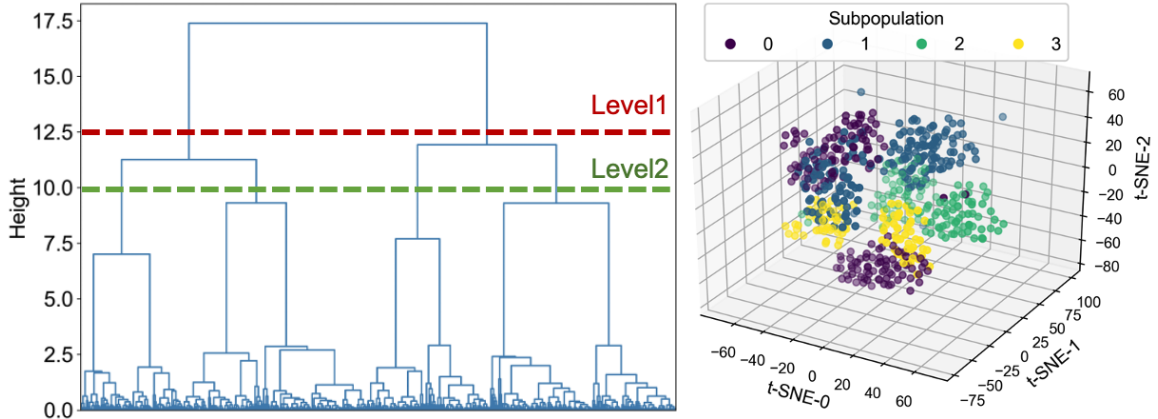


Figure 5.3: Identified subpopulation using hierarchical clustering. In the clustering dendrogram (left), the horizontal lines show two different levels of subpopulations. The resulting subpopulations at level two (right) are shown on a three-dimensional t-SNE space.

Table 5.2: Overall Performance of KIT-LSTM and seven compared algorithms on balanced test sets

	Balanced Test (Pos:Neg = 1:1)					
	ROCAUC	ACC	F-1	F-3	Recall	Precision
XGBoost	0.55	0.55	0.18	0.11	0.10	0.96
SVM	0.63	0.59	0.58	0.57	0.56	0.59
LSTM	0.66	0.64	0.56	0.48	0.46	0.72
Transformer	0.70	0.59	0.39	0.28	0.26	0.79
T-LSTM	0.70	0.63	0.55	0.47	0.46	0.70
Phased-LSTM	0.65	0.65	0.56	0.47	0.45	0.74
RETAIN	0.68	0.64	0.54	0.44	0.42	0.74
ATTAIN	0.60	0.55	0.39	0.30	0.28	0.61
KIT-LSTM (Ours)	0.72	0.64	0.62	0.58	0.58	0.67

5.3.3 Model Robustness Evaluation Metric

Robustness is one of the most important performance metrics for clinical applications, which can be assessed using the subpopulation distribution shift approach [135, 136]. First, patient subpopulations were identified using demographics and comorbidity in EHR. Second, subpopulations at different levels of granularity were obtained using hierarchical clustering and applying thresholds at the dendrogram. Third, the clustering dendrogram and the corresponding t-SNE plot of subpopulations were visualized to identify distinct subpopulations. Figure 5.3 shows four subpopulations at

Table 5.3: Overall Performance of KIT-LSTM and seven compared algorithms on imbalance test sets

	Imbalanced Test (Pos:Neg = 1:9)					
	ROCAUC	ACC	F-1	F-3	Recall	Precision
XGBoost	0.55	0.91	0.17	0.11	0.10	0.65
SVM	0.66	0.67	0.25	0.45	0.56	0.16
LSTM	0.67	0.80	0.31	0.42	0.46	0.24
Transformer	0.70	0.87	0.28	0.26	0.26	0.30
T-LSTM	0.70	0.77	0.28	0.41	0.46	0.20
Phased-LSTM	0.66	0.82	0.33	0.42	0.45	0.26
RETAIN	0.67	0.80	0.29	0.39	0.42	0.23
ATTAIN	0.58	0.76	0.19	0.26	0.28	0.14
KIT-LSTM (Ours)	0.71	0.71	0.28	0.47	0.58	0.18

Table 5.4: Balanced performance of KIT-LSTM and seven compared algorithms on multiple subpopulation levels (pos:neg = 1:1).

	Subpopulation level 1			Subpopulation level 2			Subpopulation level 5		
	ROCAUC	ACC	F-3	ROCAUC	ACC	F-3	ROCAUC	ACC	F-3
XGBoost	0.54(0.04)	0.55(0.07)	0.09(0.09)	0.54(0.06)	0.56(0.07)	0.09(0.13)	0.52(0.08)	0.47(0.17)	0.06(0.16)
SVM	0.61(0.04)	0.59(0.03)	0.54(0.15)	0.61(0.16)	0.59(0.12)	0.53(0.20)	0.57(0.30)	0.56(0.22)	0.56(0.36)
LSTM	0.63(0.15)	0.64(0.06)	0.43(0.24)	0.63(0.14)	0.64(0.06)	0.42(0.26)	0.65(0.19)	0.62(0.18)	0.43(0.37)
Transformer	0.67(0.11)	0.59(0.01)	0.24(0.18)	0.68(0.08)	0.59(0.07)	0.23(0.18)	0.65(0.18)	0.55(0.18)	0.23(0.29)
T-LSTM	0.67(0.14)	0.63(0.07)	0.41(0.28)	0.66(0.13)	0.62(0.07)	0.41(0.28)	0.64(0.22)	0.62(0.19)	0.47(0.34)
Phased-LSTM	0.61(0.20)	0.64(0.08)	0.41(0.30)	0.60(0.23)	0.64(0.08)	0.40(0.30)	0.63(0.25)	0.61(0.18)	0.42(0.36)
RETAIN	0.64(0.18)	0.63(0.05)	0.39(0.23)	0.64(0.19)	0.63(0.08)	0.39(0.26)	0.64(0.24)	0.60(0.14)	0.38(0.30)
ATTAIN	0.58(0.11)	0.55(0.01)	0.27(0.15)	0.58(0.15)	0.55(0.03)	0.27(0.15)	0.60(0.20)	0.53(0.21)	0.29(0.32)
KIT-LSTM (Ours)	0.70(0.10)	0.64(0.08)	0.54(0.23)	0.69(0.13)	0.64(0.09)	0.53(0.23)	0.66(0.15)	0.62(0.16)	0.55(0.28)

the second level (green) of dendrogram have distinctly different distributions.

Table 5.5: Imbalanced performance of KIT-LSTM and compared algorithms on multiple subpopulation levels (pos:neg = 1:9).

	Subpopulation level 1		Subpopulation level 2		Subpopulation level 5	
	ROCAUC	F-3	ROCAUC	F-3	ROCAUC	F-3
XGBoost	0.54(0.04)	0.09(0.08)	0.54(0.06)	0.09(0.13)	0.52(0.07)	0.06(0.16)
SVM	0.64(0.04)	0.43(0.14)	0.63(0.15)	0.42(0.18)	0.63(0.26)	0.46(0.32)
LSTM	0.64(0.14)	0.38(0.22)	0.63(0.13)	0.37(0.24)	0.65(0.22)	0.38(0.33)
Transformer	0.68(0.09)	0.22(0.17)	0.69(0.08)	0.22(0.17)	0.63(0.22)	0.22(0.29)
T-LSTM	0.67(0.12)	0.36(0.25)	0.67(0.12)	0.35(0.24)	0.64(0.23)	0.42(0.31)
Phased-LSTM	0.62(0.20)	0.37(0.27)	0.61(0.22)	0.35(0.28)	0.63(0.28)	0.38(0.34)
RETAIN	0.63(0.18)	0.35(0.21)	0.63(0.19)	0.34(0.23)	0.63(0.26)	0.35(0.28)
ATTAIN	0.56(0.11)	0.24(0.14)	0.55(0.13)	0.23(0.14)	0.60(0.20)	0.26(0.31)
KIT-LSTM (Ours)	0.69(0.08)	0.44(0.20)	0.69(0.12)	0.43(0.20)	0.64(0.17)	0.47(0.26)

5.4 Results

Table 5.2, 5.3 show the overall performance on both balanced and imbalanced test data (pos: neg ratio being 1:1 and 1:9) of our proposed model KIT-LSTM compared with other baseline models. KIT-LSTM has the best overall performance with the highest ROCAUC, F-3, and recall on both test data and the highest F1 on balanced data. The traditional ML method XGBoost has the highest precision on both test data and the highest accuracy on the imbalanced test data, but the scores on all other metrics are the lowest.

To assess model robustness using subpopulation shift, we measured performance variability across subpopulations at different levels of granularity. The performance on both balanced and imbalanced test data are shown in Table 5.4 and Table 5.5 respectively, the scores are shown in average and standard deviation (in parentheses).

Table 5.4 and Table 5.5 show that overall KIT-LSTM outperforms all the compared methods on almost all tested subpopulations on all evaluation metrics. SVM has the same highest F-3 as KIT-LSTM on the balanced test data at level 1 and 2, but the ROCAUC of SVM is at the lower end at all levels. LSTM has the highest ROCAUC on the unbalanced data at level 5, but its F-3 on the same data is moderate.

Table 5.4 and Table 5.5 show that the tested DL methods (except for ATTAIN) outperformed the traditional ML methods XGBoost and SVM on almost all metrics. ATTAIN has the second worst performance on balanced and unbalanced data. Phased-LSTM and RETAIN have moderate performance. Phased-LSTM has comparable accuracy on the balanced data, but its ROCAUC is on the lower end. Transformer and T-LSTM are the most competitive methods as they performed the second or third to the best on ROCAUC for all different subpopulations on both balanced and imbalanced test data. Table 5.5 shows that Transformer has the same highest ROCAUC score as KIT-LSTM at level 2 subpopulation on the imbalanced data, but its F-3 score is at the lower end. T-LSTM maintained comparable performance on

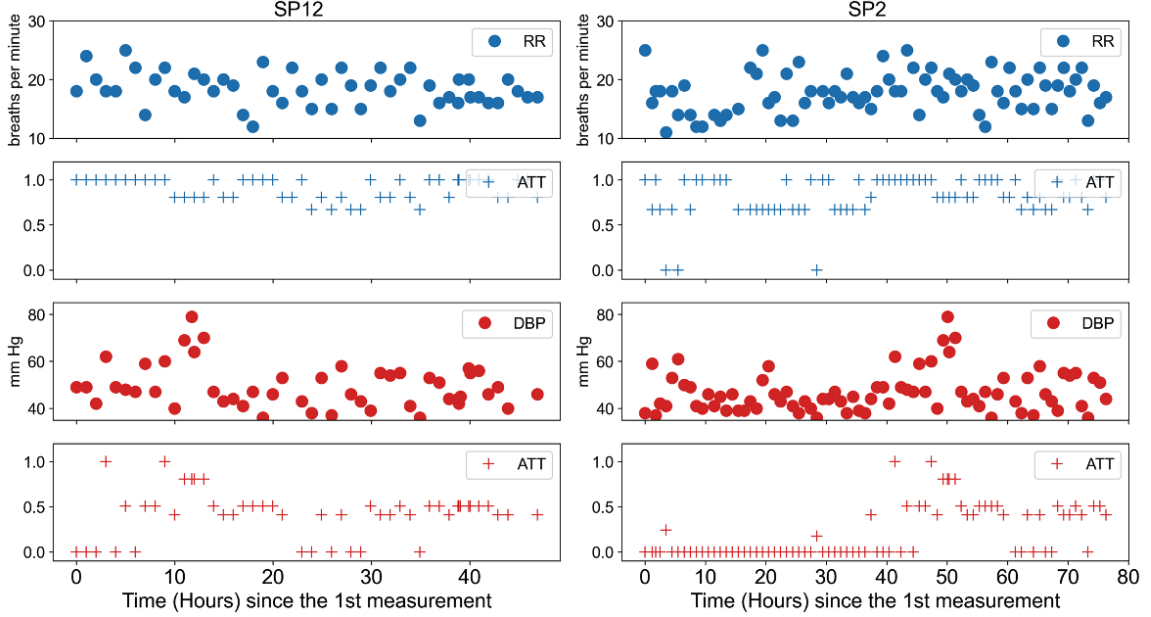


Figure 5.4: Attention scores for two samples of one patient. RR: respiratory rate; DBP: diastolic blood pressure; ATT: represents attention scores.

all datasets, which we considered the second-best model after KIT-LSTM.

Regarding the model robustness, we compared the variation of ROCAUC scores on subpopulations within a single level. XGBoost has the lowest standard deviation at all three subpopulation levels. Nevertheless, the average ROCAUC scores are lower than KIT-LSTM. On the other hand, KIT-LSTM has the best performance where its average ROCAUC scores are the highest, and the standard deviation of ROCAUC is the second- or the third-smallest at all three subpopulations levels.

Table 5.6: Ablation study: Performance of KIT-LSTM variants on balanced data at level 1, 2 and 5 subpopulations.

	Level 1			Level 2			Level 5		
	ROCAUC	ACC	F-3	ROCAUC	ACC	F-3	ROCAUC	ACC	F-3
KIT- kp	0.65(0.15)	0.63(0.06)	0.44(0.26)	0.65(0.16)	0.63(0.06)	0.43(0.28)	0.62(0.24)	0.60(0.17)	0.40(0.37)
KIT- kt	0.60(0.16)	0.59(0.05)	0.48(0.22)	0.59(0.21)	0.58(0.07)	0.47(0.22)	0.56(0.25)	0.56(0.19)	0.45(0.34)
KIT- k	0.65(0.17)	0.62(0.06)	0.37(0.27)	0.65(0.17)	0.61(0.08)	0.36(0.27)	0.64(0.19)	0.59(0.16)	0.40(0.33)
KIT-LSTM	0.70(0.10)	0.64(0.08)	0.54(0.23)	0.69(0.13)	0.64(0.09)	0.53(0.23)	0.66(0.15)	0.62(0.16)	0.55(0.28)

5.4.1 Ablation Study

We conducted an ablation study to test how each component of KIT-LSTM performed by removing several components of KIT-LSTM. The variants are:

KIT_{-k}: remove knowledge-aware gate \mathbf{k}_t while keeping two time-aware gates and physiological status \mathbf{p}_t .

KIT_{-kp}: remove knowledge-aware gate \mathbf{k}_t and physiological status \mathbf{p}_t while keeping the two time-aware gates.

KIT_{-kt}: remove knowledge-aware gate \mathbf{k}_t and two time aware gates while keeping physiological status features.

Table 5.6 shows that KIT_{-kt} without any time-aware gates or knowledge-aware gates has the lowest performance, whereas models maintained time-aware gates (KIT_{-kp} and KIT_{-k}) had better performance. The lower performance of KIT_{-kt} and KIT_{-k} indicates that the physiological status itself is not adequate for enhancing model prediction power, no matter whether it is added alone (KIT_{-kt}) or added with time-aware gates (KIT_{-k}).

KIT-LSTM has the best performance for all subpopulations. Especially, the F-3 scores are 6-17% higher than the models in the ablation study, indicating that the two time-aware gates and the knowledge-aware gate substantially improve the prediction power. In particular, when comparing KIT-LSTM with the latest baseline methods T-LSTM and ATTAIN that also used a time-aware gate, KIT-LSTM has better performance, indicating that both the long-term time-aware gate and the knowledge-aware gate play an essential role.

5.4.2 Model Validation and Interpretation

Four survived and three dead cases were randomly selected for clinical validation. Two experienced clinicians reviewed the patient records independently while the predicted risk and true label were blinded in the validation process. For the dead cases, KIT-

LSTM predicts them all with high risk, whereas the clinicians underestimate the risk for one case. For the survived cases, KIT-LSTM overestimate three case, while the clinicians overestimate one case. The results suggest that KIT-LSTM has a higher recall than precision. In clinical settings, higher recall is more important than higher precision. In this case, the high-risk patient correctly predicted by KIT-LSTM but underestimated by clinicians could cause severe clinical problems.

Attention scores α_t obtained at each time step are used to interpret the behavior of KIT-LSTM. Figure 5.4 illustrates the attention scores obtained from two adjacent samples of one patient. Case “SP12” with a low risk is predicted correctly by KIT-LSTM and clinician. Another case “SP2” has a relatively higher risk predicted by both KIT-LSTM and clinician. The figure shows that the attention scores for respiratory rate (RR) are high for almost all time for both SP12 and SP2, but the interpretation of attention should be different considering the opposite predicted outcome. For example, for the low-risk patient SP12, most of the RR values are steady within a mid-high normal range of 15-22 breaths per minute. Thus, the corresponding attention scores suggests that these values highly contributed to the low-risk prediction. For high-risk patient SP2, the RR values are relatively less steady than SP12. There are more lower-end values (before 10 hours) and more higher-end values (between 20 and 40 hours). Thus, the attention scores suggest these values contributed more towards high risk. For diastolic blood pressure (DBP), both attention scores and feature values show a bump along the time. However, the bump that happened earlier (far way from the prediction window) for SP12 contributes to lower risk prediction, and the bump that happened later (Closer to the prediction window) for SP2 contributions to a higher risk prediction. With both the trajectories and the attention scores, KIT-LSTM can assist clinicians in making better and timely decisions.

5.5 Conclusion

In this chapter, we presented KIT-LSTM, a new LSTM variant that uses two time-aware gates to address irregular and asynchronous problems in multi-variable temporal EHR data and uses a knowledge-aware gate to infuse medical knowledge for better prediction and interpretations. Experiments on real-world healthcare data demonstrated that KIT-LSTM outperforms the state-of-art ML methods on continuous mortality risk prediction for critically ill AKI-D patients.

CHAPTER 6. MTATE: Unbiased Representation of Electronic Health Records for Patient Outcome Prediction

This chapter introduces our novel unbiased deep learning model named MTATE for fairness concerns for mortality prediction using EHR data.

6.1 Introduction

The focus on building trustworthy artificial intelligence (AI) models has increased emphasis on fairness, as it has become a crucial problem in various applications, especially healthcare [137]. The lack of attention to fairness can result in severely negative consequences, where certain patient groups are given unfair advantages while others are inequitably overlooked. The failure to address the fairness problem not only undermines the trust and integrity of AI models but also perpetuates societal biases and inequalities [138, 139].

Fairness in healthcare AI refers to a model’s ability to make a prediction or decision without bias against any individual or patient group [69]. Bias in a model manifests in two forms: performance disparities (performing significantly better in certain populations than others) [70], and inequitable decisions (making inequities decisions towards different groups) [71]. Clinical decision-making based upon biased predictions may cause delayed treatment plans for patients in minority groups or mispend healthcare resources where treatment is unnecessary [72]. Several recent studies have suggested that the implementation of a fair clinical risk prediction tool based on temporal Electronic Health Records (EHR) could significantly improve clinical decision-making and optimize hospital resource allocation, particularly for patients who are classified as high-risk [140, 141]. By leveraging the power of machine learning (ML) and deep learning (DL) to analyze temporal EHR data, clinicians can identify high-risk patients and provide timely interventions to improve their outcomes.

Healthcare AI models could exhibit bias due to the data distribution shift problem, in which the model performance varies across different domains [73]. Domain adaptation methods seek to address this issue by learning hidden features that are invariant across domains. Pioneering models [74, 142, 78] use a domain classifier and a gradient reversal layer to induce domain-invariant feature representations. More recent work [143] aims to consolidate invariant globally-shared representations across domains. However, aligning large and complex domain shifts remains challenging. Another approach to addressing the data distribution shift problem is domain-specific bias correction. Recent research suggests that certain features may be subpopulation-specific and, therefore, unique to each domain [144]. Multi-task learning, double-prioritized bias correction, and clustering algorithms have also been leveraged to generate patient representations with similar backgrounds [145, 146, 147, 148]. Both approaches, domain adaptation and domain-specific bias correction, rely on different assumptions about the relationship between latent representation and prediction outcome. The effectiveness of domain-invariant versus domain-specific representations for a given prediction task is currently unclear. See details in Background Section.

To address the fairness issue in healthcare AI, we propose an adaptive multi-task learning algorithm named MTATE (Masked Triple Attention Transformer Encoder) that can automatically learn and select the optimal and fair data representations. Unlike other approaches that require the explicit selection of either domain adaptation or domain-specific bias correction, MTATE generates common representations where invariant and domain-specific representations are special cases where one of the approaches dominates the data representation. The purpose of MTATE is to generate multiple masked representations of the same data that are attended by both time-wise attention and multiple feature-wise attention. Each masked representation corresponds to a specific domain classification task, for instance, breaking the patient cohort into subpopulations based on race or gender. The learned EHR representa-

tions could be domain-specific, domain-invariant, or a mix of the two, as reflected by the domain classification loss values. Specifically, a low loss value indicates the representation is domain-specific, and a high loss value indicates that it is domain-invariant. The model computes the representation-wise attention for each individual testing case, leading to personalized data representation for downstream predictive tasks. The overall framework of MTATE is shown in Figure 6.1. Our primary objective is to learn an unbiased representation that can facilitate fair and accurate patient outcome predictions in real-world healthcare settings.

To demonstrate the effectiveness of MTATE, we will focus on two challenging risk prediction tasks, i.e., rolling mortality prediction and in-hospital mortality prediction. The first task is rolling mortality prediction for patients with Acute Kidney Injury requiring Dialysis (AKI-D), a severe complication for critically ill patients with a high in-hospital mortality rate [149]. This task is particularly challenging due to the complexity of subphenotypes and treatment exposures [150, 151]. There is an urgent need to develop actionable approaches to account for patient backgrounds and subpopulations for personalized medicine and improve patient-centered outcomes [152]. The second task is in-hospital mortality prediction for general ICU patients, which is one of the primary clinical outcomes of interest for ICU patients and is commonly used to evaluate machine learning model performances [153, 145, 154]. Both tasks are challenging due to the complexity of patient data, the diversity of patient subpopulations, and the importance of unbiased and fair data representation.

The contributions of our work are three-fold. Firstly, to our knowledge, MTATE is the first model to seamlessly integrate domain-specific and domain-invariant features in one model, making it possible to train fair representations and predict downstream tasks simultaneously. Secondly, MTATE employs time-wise, feature-wise, and representation-wise attention mechanisms to compose data representations for downstream prediction tasks dynamically. Finally, we demonstrated that MTATE effec-

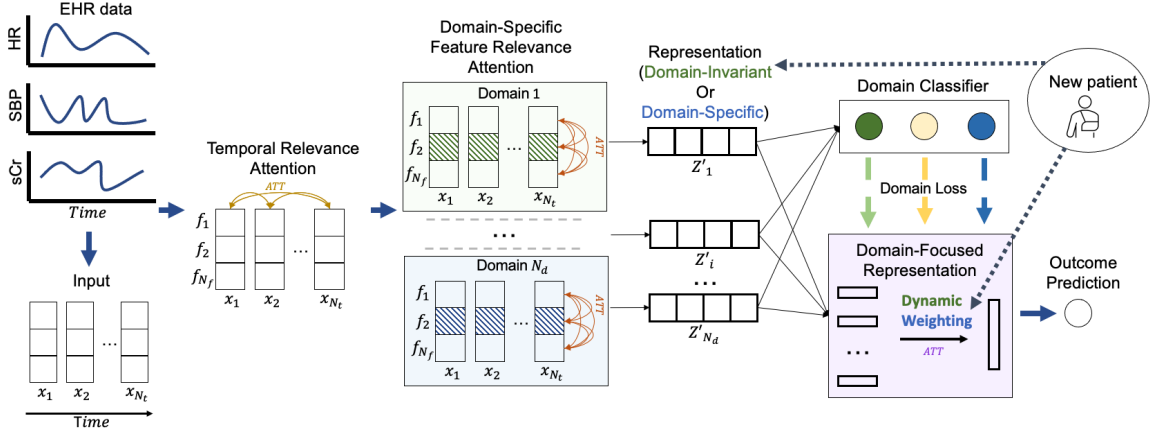


Figure 6.1: Overall framework of masked triple attention transformer encoder (MTATE). HR, SBP, and sCr stand for heart rate, systolic blood pressure, and serum creatinine, respectively. x_t represents all clinical features at time t , f_i represents values of feature i at all time points. Z'_i represents the data representations learned from the i_{th} feature relevance attention module.

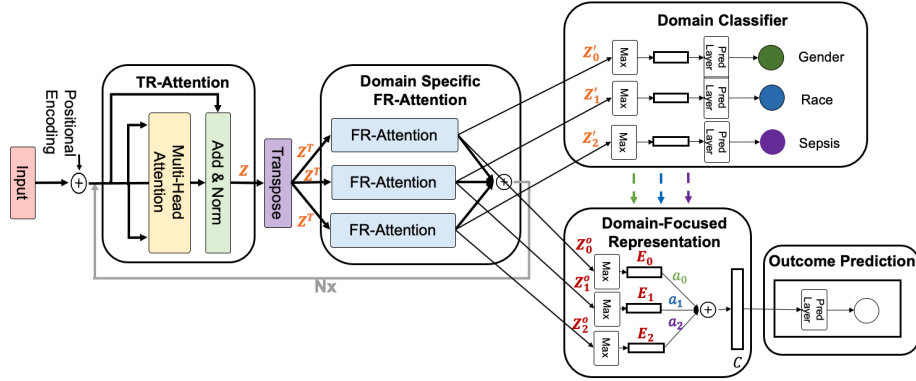


Figure 6.2: Network structure of the masked triple attention transformer (MTATE) algorithm. The TR-attention and domain-specific FR-Attention modules can be stacked N times.

tively mitigated bias towards different subpopulations in the risk prediction tasks and achieved the best overall performance compared to baselines. Overall, our work demonstrates the potential of MTATE to improve fairness and accuracy in healthcare AI and facilitate personalized medicine for diverse patient populations.

6.2 Background

One of the primary reasons an AI model could be biased is the data distribution shift problem [73]. Domain adaptation methods have been proposed to address this issue. The central concept behind domain adaptation is to learn hidden features invariant across different domains so that the model performs consistently irrespective of the domain to which the test cases belong. Pioneer domain adaptation models, including DANN [74], VARADA [142], and VRNN [78], aim to learn invariant hidden features by incorporating a domain classifier and employing a gradient reversal layer to maximize the domain classifier’s loss, thereby inducing domain-invariant feature representations. A recent work called MS-ADS [143] has demonstrated remarkable performance across minority racial groups by maximizing the distance between the globally-shared presentations with individual local representations of each domain. This effectively consolidates the invariant globally-shared representations across domains. Nevertheless, aligning large domain shifts and complex domain shifts across multiple overlapping domains remains a challenging task.

To address the data distribution shift problem, another approach that has gained attention is domain-specific bias correction. Recent research has shown that features highly associated with the outcome of interest can be subpopulation-specific [144]. It indicates that amalgamating features from patients with different backgrounds may conceal unique domain-specific characteristics. To overcome this challenge, [145] adopted a multi-task learning framework, where each subpopulation is considered a separate task, to enhance patient outcome prediction. Similarly, [146] used double prioritized bias correction to train multiple customized candidate models for different demographic groups. Similarly, AC-TPC [147] and CAMELOT [148] leveraged clustering algorithms to generate patient representations with similar backgrounds and use cluster-specific representations to predict outcomes.

To summarize, two main approaches, i.e., domain adaptation and domain-specific

bias correction, have been proposed to address the AI model fairness problem. These approaches rely on different assumptions about the relationship between latent representation and the prediction outcome: the former assumes that performance variations across domains can be mitigated by learning invariant feature representations; the latter contends that domain-specific representations can enhance prediction outcomes. It is currently unclear which type of data representation, domain-invariant or domain-specific, should be utilized for a given prediction task.

6.3 Method

MTATE consists of five components, and the detailed architecture is depicted in Figure 6.2. The first component is temporal-relevance attention (TR-Attention) which associates all the time steps, resulting in time-attended representation. The second is domain-specific feature-relevance attention which associates all the features, resulting in multiple feature-attended representations, one for each domain. The third is a set of domain classifiers, each classifies a feature-attended representation into a predefined domain. The fourth component is a unified data representation module that uses representation-wise attention to aggregate feature-attended representations (either domain-invariant or domain-specific) into a final representation. The last component is an outcome prediction module that utilizes the final representation to make patient outcome predictions.

6.3.1 Notations

A patient EHR data can be represented as $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_t, \dots, \mathbf{x}_{N_t}\}$, $\mathbf{X} \in \mathbb{R}^{N_t \times N_f}$, where N_f is the number of features and N_t is the number of time steps. $\mathbf{x}_t \in \mathbb{R}^{1 \times N_f}$ represents a vector of clinical parameters (e.g., heart rate, blood pressure, etc.) at time step t . We consider a binary outcome and domain classification problem in this study. The patient domain class labels are denoted as $\mathbf{dy} \in \mathbb{R}^{N_d}$, where N_d represents

the total number of domains, $dy^i \in \{0, 1\}$ represents the label for the i -th domain, 1 and 0 represent whether a given patient falls in the target domain or not, respectively. The patient outcome label is denoted as $y \in \{0, 1\}$, where 1 and 0 represent death and alive before hospital discharge.

6.3.2 Temporal Relevance Attention

The temporal-relevance attention (TR-Attention) module is the first component of MTATE. It enables each time step to attend to different time steps in patient EHR and capture complex temporal dependencies between different time steps, considering all input features, in \mathbf{X} . To achieve this, we encode relative position information of \mathbf{X} using the position encoding and learn TR-Attention using the multi-head attention mechanism from the Transformer [15] model. Specifically, query, key, and value vectors ($\mathbf{Q}, \mathbf{K}, \mathbf{V}$) are the linear projections of all features at every time step in \mathbf{X} . The attention weights computed from the query and key represent how much focus the features at one single time step are associated with themselves at other time steps. Then, the output of each head \mathbf{Z}^h is the multiplication of value vectors and time-wise attention \mathbf{A}^{TR} . The final output of TR-Attention $\mathbf{Z} \in \mathbf{R}^{N_t \times N_f}$ is the linear transformation of the concatenation of the output of every head. Lastly, the residual connection and layer normalization are applied to \mathbf{Z} , denoted as $\mathbf{Z} = \text{LayerNorm}(\mathbf{Z} + \mathbf{X})$. The TR-attention of each head is represented as:

$$\mathbf{Q}, \mathbf{K}, \mathbf{V} = \mathbf{X}\mathbf{W}_{\mathbf{Q}}, \mathbf{X}\mathbf{W}_{\mathbf{K}}, \mathbf{X}\mathbf{W}_{\mathbf{V}} \quad (6.1)$$

$$\mathbf{A}^{\text{TR}} = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right) \quad (6.2)$$

$$\mathbf{Z}^h = \mathbf{A}^{\text{TR}}\mathbf{V} \quad (6.3)$$

$$\mathbf{Z} = \text{Concat}(Z_1^h, \dots, Z_i^h, \dots, Z_{N_h}^h)\mathbf{W}_O \quad (6.4)$$

For simplicity, we assume all projection matrices $\mathbf{W}_{\mathbf{Q}}, \mathbf{W}_{\mathbf{K}}, \mathbf{W}_{\mathbf{V}}$ have the same dimension d_k . Thus, $\mathbf{W}_{\mathbf{Q}}, \mathbf{W}_{\mathbf{K}}, \mathbf{W}_{\mathbf{V}} \in \mathbf{R}^{N_f \times d_k}$, $\mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbf{R}^{N_t \times d_k}$, the temporal

relevance attention is $\mathbf{A}^{\text{TR}} \in \mathbb{R}^{N_t \times N_t}$, the output of each head is $Z^h \in \mathbb{R}^{N_t \times d_k}$. The projection matrix for the final output is $\mathbf{W}_O \in \mathbb{R}^{(N_h d_k) \times N_f}$, where N_h represents the number of heads.

6.3.3 Domain-specific Feature Relevance Attention

The domain-specific feature relevance attention (FR-Attention) module is another key component of MTATE, which aims to learn each domain’s unique latent data representation. This module includes a set of parallel FR-Attention sub-modules, where each sub-module focuses on the representation of a specific domain considering all time steps. Since features may not be equally important for different domains, we randomly masked a certain percentage of latent features in each FR-Attention module. This masking process enables each sub-module to attend to a unique subset of features, allowing it to learn domain-specific representations effectively. In other words, each sub-module learns to attend to the most relevant features for a given domain while ignoring features that may be irrelevant or even harmful to that particular domain. The feature-wise attention in the FR-Attention is computed using the multi-head attention mechanism similar to the TR-Attention. The output of each FR-Attention sub-module is a feature-attended representation that captures the important and unique information for each domain. Using this approach, MTATE can learn both domain-invariant and domain-specific representations, which are essential for accurate and fair patient outcome predictions.

The input of each FR-Attention sub-module is $\mathbf{Z}^T \in \mathbb{R}^{N_f \times N_t}$, which is the transposed output of TR-Attention \mathbf{Z} . \mathbf{Z}^T is passed through a masking layer, where $MR \times N_f$ number of latent features are randomly selected and removed from \mathbf{Z}^T , and MR is a masking rate. We denote the masked input of each sub-module as $\mathbf{M} \in \mathbb{R}^{N_l \times N_t}$, where N_l is the number of features after masking. \mathbf{M} is passed through the multi-head attention block as well as the residual connection and layer normal-

ization. Finally, \mathbf{M} is transposed back to the original form and passed through a point-wise feedforward neural network (FNN) [155] as well as the residual connection and layer normalization to get the final output, denoted as $\mathbf{Z}' \in \mathbf{R}^{N_t \times N_t}$. The architecture (6.3) and formula for FR-Attention are shown below:

The FR-Attention sub-module for each head is computed as:

$$\mathbf{Q}', \mathbf{K}', \mathbf{V}' = \mathbf{M}\mathbf{U}_{\mathbf{Q}}, \mathbf{M}\mathbf{U}_{\mathbf{K}}, \mathbf{M}\mathbf{U}_{\mathbf{V}} \quad (6.5)$$

$$\mathbf{A}^{\text{FR}} = \text{softmax}\left(\frac{\mathbf{Q}'\mathbf{K}'^T}{\sqrt{d'_k}}\right) \quad (6.6)$$

$$\mathbf{M}^{\text{th}} = \mathbf{A}^{\text{FR}}\mathbf{V}' \quad (6.7)$$

$$\mathbf{M}' = \text{Concat}(M_1^{\text{th}}, \dots, M_i^{\text{th}}, \dots, M_{N'_h}^{\text{th}})\mathbf{U}_{\mathbf{O}} \quad (6.8)$$

$$\mathbf{Z}' = \mathbf{max}(\mathbf{0}, (\mathbf{M}')^T\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2 \quad (6.9)$$

Similar to TR-Attention module, we assume all projection matrix $\mathbf{U}_{\mathbf{Q}}, \mathbf{U}_{\mathbf{K}}, \mathbf{U}_{\mathbf{V}}$ have the same dimension d'_k . Thus, $\mathbf{U}_{\mathbf{Q}}, \mathbf{U}_{\mathbf{K}}, \mathbf{U}_{\mathbf{V}} \in \mathbf{R}^{N_t \times d'_k}$ and $\mathbf{Q}', \mathbf{K}', \mathbf{V}' \in \mathbf{R}^{N_t \times d'_k}$. The feature-relevance attention $\mathbf{A}^{\text{FR}} \in \mathbf{R}^{N_t \times N_t}$. The output of each head is $\mathbf{M}^{\text{th}} \in \mathbf{R}^{N_t \times d'_k}$. Similar to the TR-Attention module, all outputs from all heads are concatenated to form $\mathbf{M}' \in \mathbf{R}^{N_t \times N_t}$, and linear transformation are applied with the projection matrix $\mathbf{U}_{\mathbf{O}} \in \mathbf{R}^{(N'_h d'_k) \times N_t}$, where N'_h represents the number of head.

6.3.4 Domain Classifier

The third component of MTATE comprises a set of domain classifiers, where each classifier is responsible for classifying feature-attended representations into a predefined domain, thereby assisting in learning the latent representation for each domain. The input of each domain classifier is $\mathbf{Z}'_i \in \mathbf{R}^{N_t \times N_t}$, where i denotes the index of a given sub-module or domain. \mathbf{Z}'_i is flattened by taking the max along the time dimensions, which produces a feature representation of size \mathbf{R}^{N_t} . This feature representation is fed into a linear layer with a sigmoid activation function for binary classification. We use binary cross-entropy as the loss for domain classification, de-

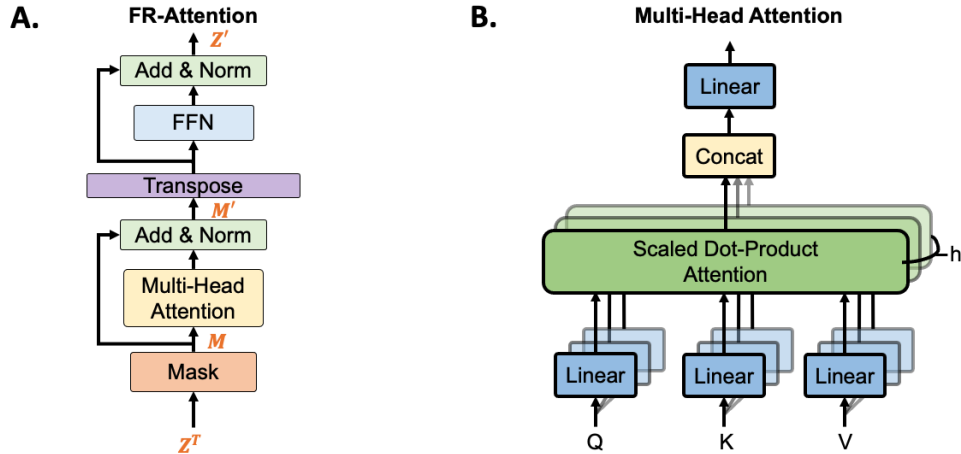


Figure 6.3: A. The structure of FR-Attention module in MTATE. B. The multi-head attention module from the original Transformer used in MTATE.

noted as L_{d_i} . The domain loss is then used to generate representation-wise attention in the next module, which is critical for learning fair and accurate data representations. Note that although the previous domain-specific FR-attention modules are focused on representing their target domains, the resulting representations can be domain-specific or domain-invariant, depending on the loss values from the domain classifier.

6.3.5 Domain-focused Representation

Not all data representations are equally important for patient outcome predictions. This domain-focused representation module in MTATE aims to generate the final data representation for the outcome prediction by considering both domain-specific and domain-invariant representations and assigns weights to each representation based on their corresponding domain classification loss values. We call this a domain-focused representation module, as it enables MTATE to dynamically generate different masked representations and select the optimal data representation that is both fair and accurate.

The inputs to the domain-focused representation module are transformed latent representations generated by each FR-Attention sub-module \mathbf{Z}'_i . To align all latent representations in the feature space, we transform \mathbf{Z}'_i to its original dimension by filling the masked (removed) features with 0s, resulting in the form of \mathbf{Z}°_i . To generate the final representation $\mathbf{C} \in \mathbb{R}^{N_f \times 1}$, we first flatten all \mathbf{Z}°_i representations by taking the maximum value along the time dimension, resulting in a matrix $\mathbf{E} \in \mathbb{R}^{N_d \times N_f}$. Next, we compute the representation-wise attention (RW-Attention) vector $\mathbf{a} \in \mathbb{R}^{N_d \times 1}$ based on \mathbf{E} and the domain prediction loss $\mathbf{L}_d \in \mathbb{R}^{N_d \times 1}$. Lastly, The final representation $\mathbf{C} \in \mathbb{R}^{N_f \times 1}$ is computed as the weighted sum of \mathbf{E} :

$$\mathbf{a} = \text{softmax}(\tanh(\text{Concat}(\mathbf{E}, \mathbf{L}_d)\mathbf{U}_A)\mathbf{W}_A) \quad (6.10)$$

$$C_j = \sum_{i=1}^{N_d} a_i E_{i,j} \quad (6.11)$$

where $U_A \in \mathbb{R}^{(N_f+1) \times d_a}$ and $W_A \in \mathbb{R}^{d_a \times 1}$ are the projection matrices, i represents the domain index, j represents the feature index.

6.3.6 Patient Outcome Prediction

To make fair and accurate patient outcome predictions, the final patient EHR representation \mathbf{C} generated by the domain-focused representation module is concatenated with all static features, such as demographics and comorbidity. This combined feature representation is then fed into an FNN following a sigmoid activation function to predict the patient outcome. Here, using a sigmoid activation function allows for the output to be interpreted as a probability, enabling the model to provide interpretable predictions. The prediction loss includes the binary cross entropy denoted as L_p and the supervised contrastive loss [156] to mitigate further the model bias as another

part of the final loss, denoted as L_c . The final prediction loss L is:

$$L = L_p + L_c \quad (6.12)$$

$$L_p = \sum_{i=1}^{N_s} -(y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)) \quad (6.13)$$

$$L_c = \sum_{j=1}^{N_s} \frac{-1}{N_p} \sum_{p=1}^{N_p} \log \frac{\exp(h_j * h_p / \tau)}{\sum_{a=1}^{N_a} \exp(h_j * h_a / \tau)} \quad (6.14)$$

where y is the patient outcome label; \hat{y} is the predicted label; N_s, N_p, N_a are the number of all samples, the number of samples having the same labels as the anchor samples (j), and the number of samples having the opposite label to the anchor samples (j); h represents the concatenation of the learned representation \mathbf{C} and the static features; and τ is a scale parameter.

6.4 Experiments Settings

This section describes the prediction task, experiment data, baseline methods, and evaluation metrics.

6.4.1 Prediction Tasks

We evaluate the performance of MTATE and all baselines using two independent prediction tasks:

Rolling mortality prediction for AKI-D patients: Continuously predict patients' mortality risk in their dialysis/renal replacement therapy (RRT) duration. Given a period of EHR right before time T , we continuously predict the mortality risk between T and $T + 72hours$.

In-hospital mortality prediction for ICU patients: Predict whether the patient dies during the hospital stay using the data from the first 48 hours after admission to ICU.

6.4.2 Experiment Data

The performance of MTATE and all baselines are comprehensively evaluated using proprietary and public data.

Proprietary EHR Data. For rolling mortality prediction, we use proprietary data where the study population comprises 570 AKI-D adult patients (13,333 sampled trajectories) admitted to ICU at XXX Hospital from January 2009 to October 2019. Among them, 237 (41.6 %) died before discharge, and 333 (58.4%) survived. Patients are excluded if they were diagnosed with end-stage kidney disease (ESKD) before or at the time of hospital admission, are recipients of a kidney transplant, or have RRT less than 72 or greater than 2,000 hours.

Data features include 12 temporal features (systolic blood pressure, diastolic blood pressure, serum creatinine, bicarbonate, hematocrit, potassium, bilirubin, sodium, temperature, white blood cells (WBC) count, heart rate, and respiratory rate) and 11 static features including demographics and comorbidities (age, race, gender, admission weight, body mass index (BMI), Charlson comorbidity score, diabetes, hypertension, cardiovascular disease, Chronic Kidney Disease, and Sepsis). All outliers ($> 97.5\%$ or $< 2.5\%$) are excluded after a manual review, and missing values are imputed with the last observation carried forward (LOCF) method.

To continuously predict mortality risks, we generate 30 samples from each patient’s EHR data with random start and end times as long as the duration exceeds 10 time steps. The class label of a sample is whether the patient died (positive) or survived (negative) in the next 72 hours. From 570 AKI-D patients, 13,333 EHR samples are extracted, including 2,975 positive and 10,358 negative samples. The samples are split into training (75%), validation (5%), and testing data (20%) patient-wise, which ensures that samples from the same patient only appear in one of the three sets (see detailed numbers of samples in Table 6.1). Eighteen subpopulations were considered in this study based on nine domains according to patient demographics

Table 6.1: Training, validation, and testing data. (Rolling mortality prediction on the proprietary dataset)

	Total N Samples (Patient)	Alive N Samples (Patient)	Death in the next 72 hours N Samples (Patient)	Negative to Positive Ratio (Patient)
Train	9979 (432)	7652 (388)	2327 (149)	3:1 (3:1)
Valid	642 (24)	519 (22)	123 (9)	4:1 (2:1)
Test	2712 (114)	2187 (104)	525 (31)	4:1 (3:1)
All	13333 (570)	10358 (514)	2975 (189)	3:1 (3:1)

(i.e., age, gender, race) and commodities (i.e., Charlson score, diabetes, hypertension, cardiovascular disease, chronic kidney disease, and sepsis).

Public EHR Data. Publicly available data from the Medical Information Mart for Intensive Care III (MIMIC-III) [116] are used for in-hospital mortality prediction. We consider the first ICU admission for all adult patients and exclude the patients with ICU length of stay of fewer than 48 hours. The data include 20,308 ICU admissions, and the median ICU length of stay is 3.9 days [Q1-Q3: 2.8-7.1]. Among them, 2,708 (13%) died in the hospital, and 17,600 (87%) survived.

Data features include 20 temporal features (bicarbonate, bilirubin, hematocrit, potassium, sodium, serum creatinine, pH, blood urea nitrogen, chloride, glucose, hemoglobin, lactate, magnesium, oxygen saturation, systolic blood pressure, diastolic blood pressure, respiratory rate, heart rate, temperature, and Glasgow coma scale) in the first 48 hours of each ICU stay. The values are averaged if multiple observations are present in the same hour. We consider three demographic variables (age, gender, and race) as static features and the subpopulation domains. All outliers ($> 99\%$ or $< 1\%$) are excluded after a manual review, and missing values are imputed with the LOCF method. The training, testing, and validation data sizes are shown in Table 6.2.

Table 6.2: Training, validation, and testing data. (In-hospital mortality prediction on MIMIC3 dataset)

	Total	Alive	Hospital Death	Neg to Pos Ratio
Train	15231	13213	2018	7:1
Validation	1015	884	131	7:1
Test	4062	3503	559	6:1
All	20308	17600	2708	7:1

6.4.3 Baseline Models

We compare MTATE with six baselines: two widely used sequence DL methods (LSTM and Transformer), two well-known EHR-specific representations methods (RETAIN [55] and ConvAE [157]), a pioneer domain-adaptation method (DANN* [74]) and one multi-task learning framework (MTL [145]). For the Transformer, the encoder part of the original Transformer is used. For ConvAE, we use the primary model to train patient EHR representation, followed by multiple dense layers for outcome predictions. For DANN*, we use the gradient reversal layer from the original DANN to get domain-invariant representation with all other structures the same as MTATE. For all models, the input data are the temporal features in EHR, and the static features are concatenated with latent representation before the prediction layer, as described in Section 6.3.6.

6.4.4 Performance Metrics

We evaluate the performance of all the compared models using supervised-learning performance metrics: Area under the ROC Curve (ROCAUC), Accuracy (ACC), Area under the Precision-Recall Curve (PRAUC), as well as three fairness metrics: Demographic Parity Difference (DPD), Equality of Opportunity Difference (EOD) and Equalized Odds Difference (EQOD) [158, 159, 146] (see fairness equations and explanation 6.15, 6.16, 6.17 below). All models are tested using imbalanced and balanced

sets, where the positive (died) and negative (survived) ratios for the imbalanced tests are 1:4 for the proprietary data and 1:6 for the public MIMIC3 data.

6.4.4.1 Fairness Metrics

Demographic parity suggests that a predictor is unbiased if the prediction is independent of the protected attribute (e.g., Age, Gender, and etc.). We denote protected attribute as $A \in a, b$, and A only take two groups a, b (e.g., Young vs Old for Age) for simplicity. Thus, the Demographic parity difference (DPD) is the difference between the two group a and b . The formula for Demographic parity difference (DPD) is shown in below:

$$DPD = P(\hat{y} = 1|A = a) - P(\hat{y} = 1|A = b) \quad (6.15)$$

Equality of opportunity suggests that a predictor is unbiased if the true-positive rate between two groups are equal. Similarly, the Equality of opportunity difference (EOD) is the difference between the two group a and b . The formula for EOD is shown in below:

$$EOD = P(\hat{y} = 1|y = 1, A = a) - P(\hat{y} = 1|y = 1, A = b) \quad (6.16)$$

Equalized odds suggests that a predictor is unbiased if both the true-positive rate (TPR) and false-positive rate (FPR) between two groups are equal. We compute the Equalized odds Difference (EQOD) as the average of the difference in both TPR and FPR. The formula for EQOD is shown in below:

$$EQOD = (TPR_D + FPR_D)/2 \quad (6.17)$$

$$TPR_D = P(\hat{y} = 1|y = 1, A = a) - P(\hat{y} = 1|y = 1, A = b) \quad (6.18)$$

$$FPR_D = P(\hat{y} = 1|y = 0, A = a) - P(\hat{y} = 1|y = 0, A = b) \quad (6.19)$$

We compare all models on both imbalanced and balanced sets. The positive samples (died) and negative samples (survived) are 1:4 and 1:1, respectively.

Table 6.3: Performance comparison on rolling mortality prediction in the next 72 hours for proprietary imbalanced test data (pos:neg=1:4). DPD, EOD, and EQOD are the lower the better.

Method	ROCAUC	ACC	PRAUC	DPD	EOD	EQOD
Transformer	0.71(0.08)	0.69(0.09)	0.39(0.17)	0.18(0.10)	0.08(0.08)	0.10(0.05)
LSTM	0.72(0.11)	0.77(0.07)	0.55(0.20)	0.12(0.08)	0.10(0.07)	0.07(0.04)
RETAIN	0.69(0.12)	0.78(0.07)	0.45(0.20)	0.13(0.12)	0.10(0.07)	0.07(0.06)
DANN*	0.60(0.12)	0.64(0.14)	0.31(0.16)	0.23(0.18)	0.08(0.06)	0.12(0.08)
MTL	0.67(0.13)	0.64(0.13)	0.44(0.21)	0.25(0.21)	0.10(0.08)	0.13(0.10)
ConvAE	0.65(0.08)	0.64(0.06)	0.34(0.16)	0.09(0.04)	0.07(0.06)	0.07(0.03)
MTATE (Ours)	0.72(0.09)	0.81(0.07)	0.49(0.18)	0.09(0.07)	0.07(0.06)	0.05(0.03)

6.5 Results and Discussion

6.5.1 Performance Comparison

Rolling Mortality Prediction The overall performance of rolling mortality prediction in the next 72 hours is shown in Table 6.3. The results show that MTATE outperforms all the compared baselines in almost all metrics. LSTM is the most competitive model since it has the same highest ROCAUC as MTATE and the highest PRAUC. Nevertheless, the fairness scores of LSTM are worse than MTATE. On the other hand, ConvAE has the best or second-best fairness scores as MTATE on fairness metrics, but its ROCAUC, ACC, and PRAUC are at the lower end. RETAIN and Transformer have moderate performance. In addition, the performance comparison on the balanced test data shows that MTATE has the overall best predictive power and best fairness scores (see details in Table 6.4).

We compare the performance of MTATE with all baselines within each subpopulation domain. Figure 6.4 shows that MTATE has the best (lowest) averaged normalized EQOD score and has the best or second best for almost every subpopulation domain. We also compare MTATE with all baselines regarding the difference in PRAUC between paired subpopulation domains (e.g., the difference in PRAUC between females and males). The percentage difference in PRAUC for each domain pair and the averaged score across all domains are listed in Figure 6.5. It shows that MTATE has

Table 6.4: Balanced performance of MTATE and compared algorithms for rolling mortality prediction in the next 72 hours for the proprietary test data (pos:neg = 1:1).

Method	ROCAUC	ACC	PRAUC	DPD	EOD	EQOD
Transformer	0.70(0.09)	0.67(0.09)	0.69(0.11)	0.17(0.13)	0.15(0.12)	0.10(0.06)
LSTM	0.71(0.11)	0.67(0.11)	0.76(0.12)	0.19(0.12)	0.20(0.12)	0.12(0.05)
RETAIN	0.68(0.12)	0.67(0.11)	0.72(0.13)	0.22(0.13)	0.21(0.12)	0.12(0.06)
DANN*	0.58(0.12)	0.54(0.14)	0.59(0.12)	0.21(0.19)	0.16(0.13)	0.12(0.08)
MTL	0.66(0.14)	0.65(0.10)	0.70(0.14)	0.24(0.17)	0.19(0.13)	0.13(0.08)
ConvAE	0.66(0.08)	0.62(0.07)	0.65(0.11)	0.10(0.07)	0.12(0.10)	0.09(0.05)
MTATE (Ours)	0.73(0.09)	0.65(0.11)	0.75(0.11)	0.15(0.10)	0.14(0.11)	0.08(0.05)

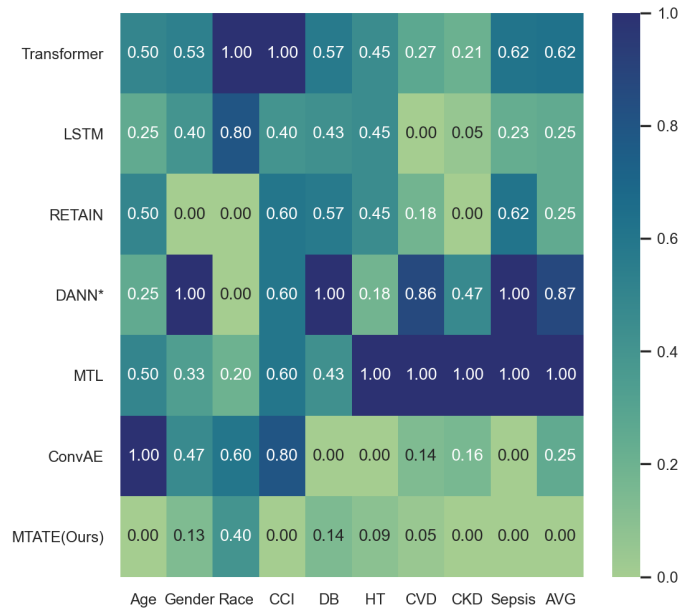


Figure 6.4: Normalized equalized odds difference (EQOD) for every subpopulation domain on rolling mortality prediction in the next 72 hours for the proprietary test data. A low EQOD score indicates high model fairness. The value and color represent the normalized EQOD score (the lower/lighter, the better), and X-axis represents the subpopulation domains, where each domain consists of two subpopulations (e.g., young vs. old in age). CCI, DB, CVD, and CKD stand for Charlson comorbidity score, diabetes, hypertension, cardiovascular disease, and chronic kidney disease, respectively.

the lowest percentage difference in PRAUC between subpopulations in Age, Gender, and Hypertension domains, and MTATE has the second-lowest overall percentage difference in PRAUC.

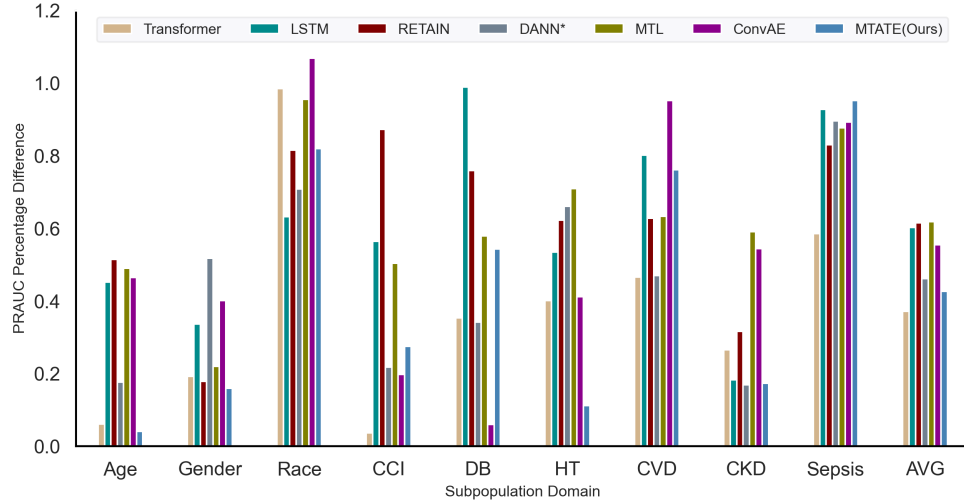


Figure 6.5: The comparison between MTATE with baseline methods for the percentage difference score in PRAUC for each domain. Y-axis represents the percentage difference. X-axis represents the subpopulation domain, each domain consists of two subpopulations (e.g., Young (< 65 y/o) vs. Old in Age domain, Sepsis vs. Non-Sepsis in Sepsis Domain). CCI stands for charlson comorbidity score, DB stands for diabetes, HT stands for hypertension, CVD stands for cardiovascular disease, CKD stands for chronic kidney disease.

Table 6.5: Performance comparison on in-hospital mortality prediction for MIMIC3 imbalanced test data (pos:neg=1:6). DPD, EOD, and EQOD are the lower, the better.

Method	ROCAUC	ACC	PRAUC	DPD	EOD	EQOD
Transformer	0.79(0.02)	0.67(0.04)	0.37(0.06)	0.11(0.07)	0.05(0.03)	0.05(0.04)
LSTM	0.80(0.02)	0.87(0.02)	0.41(0.06)	0.06(0.01)	0.03(0.01)	0.03(0.01)
RETAIN	0.80(0.02)	0.72(0.05)	0.38(0.05)	0.13(0.09)	0.04(0.02)	0.06(0.04)
DANN*	0.60(0.04)	0.69(0.12)	0.20(0.04)	0.29(0.17)	0.04(0.04)	0.14(0.08)
MTL	0.75(0.05)	0.61(0.05)	0.37(0.07)	0.12(0.09)	0.04(0.04)	0.07(0.04)
ConvAE	0.68(0.02)	0.74(0.01)	0.27(0.04)	0.03(0.01)	0.02(0.01)	0.02(0.01)
MTATE (Ours)	0.80(0.02)	0.83(0.02)	0.41(0.06)	0.07(0.03)	0.03(0.02)	0.03(0.01)

In-hospital Mortality Prediction The overall performance of in-hospital mortality prediction on the MIMIC3 test data with an imbalanced positive-to-negative ratio is shown in Table 6.5. The results show that MTATE has the best overall performance compared to baselines regarding in-hospital mortality prediction. MTATE has the highest ROCAUC, PRAUC, and the second-best score on ACC and fairness

Table 6.6: Balanced performance of MTATE and compared algorithms for in-hospital mortality prediction for MIMIC3 dataset (pos:neg = 1:1).

Method	ROCAUC	ACC	PRAUC	DPD	EOD	EQOD
Transformer	0.80(0.02)	0.73(0.01)	0.78(0.04)	0.13(0.08)	0.12(0.06)	0.07(0.03)
LSTM	0.80(0.02)	0.63(0.01)	0.80(0.04)	0.11(0.04)	0.10(0.03)	0.05(0.02)
RETAIN	0.80(0.01)	0.73(0.01)	0.79(0.03)	0.15(0.09)	0.12(0.06)	0.08(0.04)
DANN*	0.60(0.04)	0.56(0.02)	0.60(0.06)	0.28(0.19)	0.13(0.13)	0.14(0.09)
MTL	0.76(0.05)	0.69(0.01)	0.77(0.06)	0.16(0.10)	0.09(0.11)	0.08(0.05)
ConvAE	0.69(0.02)	0.63(0.03)	0.69(0.04)	0.03(0.03)	0.04(0.02)	0.03(0.02)
MTATE (Ours)	0.80(0.01)	0.67(0.01)	0.79(0.05)	0.09(0.06)	0.09(0.06)	0.05(0.03)

metrics EOD and EQOD, as well as the third-best DPD score. LSTM is the most competitive method since it has the same highest ROCAUC, and PRAUC as MTATE and the highest ACC and the best or second-best fairness scores. On the other hand, ConvAE has the best fairness scores on all fairness metrics, but its ROCAUC, ACC, and PRAUC are lower than the others. The performance of all the compared algorithms on the balanced test data shows that MTATE has the best or second-best scores on almost all metrics (see details in Table 6.6).

6.5.2 Ablation Study

We conduct an ablation study to test how each component of MTATE contributes to the model performance. Table 6.7 shows that the complete MTATE has the best performance for almost all metrics. Comparing MTATE with the two ablation models ("w/o RW-ATT" and "w/o. DC & RW-ATT"), which has the lowest performance, all performances of MTATE are boosted (the improvement ranges from 4% to 16%). This comparison indicates that RW-ATT is the most effective component since the performance drops the most in the two ablations without RW-ATT. Moreover, the ablation model "w/o. L_c " has a similar performance to MTATE, but all metrics are 1 – 5% lower than MTATE, indicating that the contrastive loss component (L_c) did improve the performance a little, but it is not a major factor.

Table 6.7: Performance comparison of MTATE and its ablation components for rolling mortality prediction in the next 72 hours for the proprietary test data (pos: neg = 1:4). **w/o. DC**: remove all domain classifiers. **w/o. RW-ATT**: remove representation-wise attention. **w/o. DC & RW-ATT**: remove both domain classifier and representation-wise attention. **w/o. Masking**: remove masking layers in the FR-Attention module. **w/o. L_c** : remove contrastive loss.

Method	ROCAUC	ACC	PRAUC	DPD	EOD	EQOD
w/o. DC	0.70(0.09)	0.70(0.07)	0.47(0.18)	0.15(0.12)	0.09(0.07)	0.09(0.06)
w/o. RW-ATT	0.64(0.13)	0.72(0.10)	0.37(0.21)	0.23(0.20)	0.12(0.09)	0.13(0.09)
w/o. DC & RW-ATT	0.63(0.13)	0.70(0.11)	0.37(0.21)	0.25(0.20)	0.11(0.10)	0.13(0.09)
w/o. Masking	0.73(0.09)	0.63(0.06)	0.48(0.18)	0.15(0.12)	0.09(0.08)	0.09(0.06)
w/o. L_c	0.71(0.09)	0.77(0.07)	0.44(0.17)	0.11(0.11)	0.08(0.06)	0.07(0.05)
MTATE	0.72(0.09)	0.81(0.07)	0.49(0.18)	0.09(0.07)	0.07(0.06)	0.05(0.03)
XGBoost w/ MTATE	0.69(0.09)	0.70(0.07)	0.43(0.17)	0.09(0.08)	0.07(0.07)	0.07(0.03)
SVM w/ MTATE	0.71(0.10)	0.81(0.06)	0.48(0.19)	0.10(0.08)	0.07(0.06)	0.06(0.04)
RF w/ MTATE	0.72(0.09)	0.80(0.06)	0.49(0.17)	0.07(0.07)	0.06(0.06)	0.05(0.03)

6.5.3 Assessment of Data Representation

A primary goal of MTATE is to learn fair representations that can be utilized by a wide spectrum of downstream predictive models. To evaluate this, we investigate whether the learned representations from MTATE can be directly used by traditional machine learning methods. The last three lines in Table 6.7 show that all three traditional methods (XGboost, SVM, and Random Forest) have achieved similar performance as MTATE and outperform some of the complex deep learning models. This comparison provides strong evidence that MTATE can serve as a pre-trained EHR data representation generator. The learned representations can be utilized by downstream prediction tasks implemented with traditional classifiers. Using MTATE-generated representations with different classifiers provides increased flexibility and adaptability in predicting patient outcomes in real-world clinical settings.

6.5.4 Effectiveness Assessment of RW-Attention

Further model performance analysis investigates the behavior of RW-Attention with respect to the changes in outcome prediction loss L_p and domain loss L_d . In Fig-

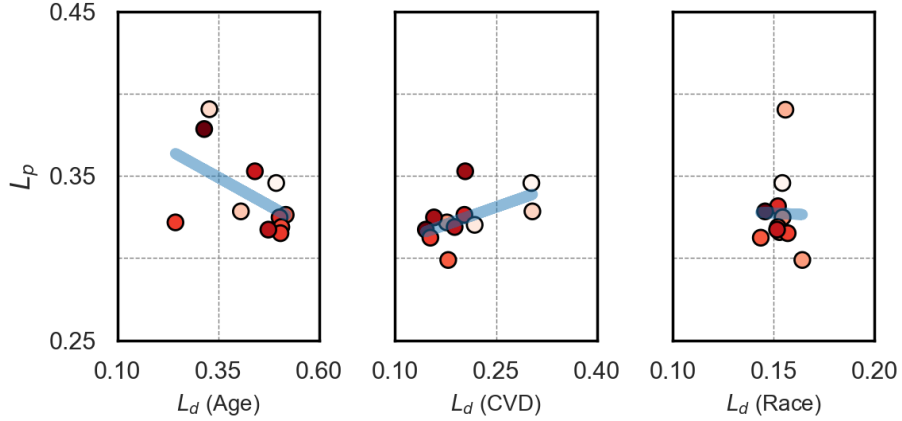


Figure 6.6: Relationship between outcome loss, domain loss and representation-wise attention. Y-axis represents the outcome loss, X-axis represents the domain loss. The colored dots represent representation-wise attention, and the darker color represents higher attention.

Figure 6.6, each dot presents the average value of all samples from the same subpopulation. The figure highlights three example domains in two separate facets. First, the correlation between outcome prediction loss L_p and domain loss L_d is not constant. It could be either negative, positive, or mixed, depending on the specific domain. In the case of a negative correlation, a higher domain loss is associated with a lower outcome prediction loss. This suggests that RW-Attention assigns more weight to the representations with larger domain loss, which are domain-invariant representations. On the other hand, in the case of a positive correlation, a decrease in domain loss is associated with a decrease in outcome prediction loss, indicating that RW-Attention places more emphasis on the representations with smaller domain loss, which are domain-specific representations. The mixed correlation scenario indicates that the relationship between L_p and L_d is complex and varies based on the specific subpopulation domain.

Attention weights, represented by the color of the dots in Figure 6.6, provide additional insight into the relationship between RW-Attention and the outcome pre-

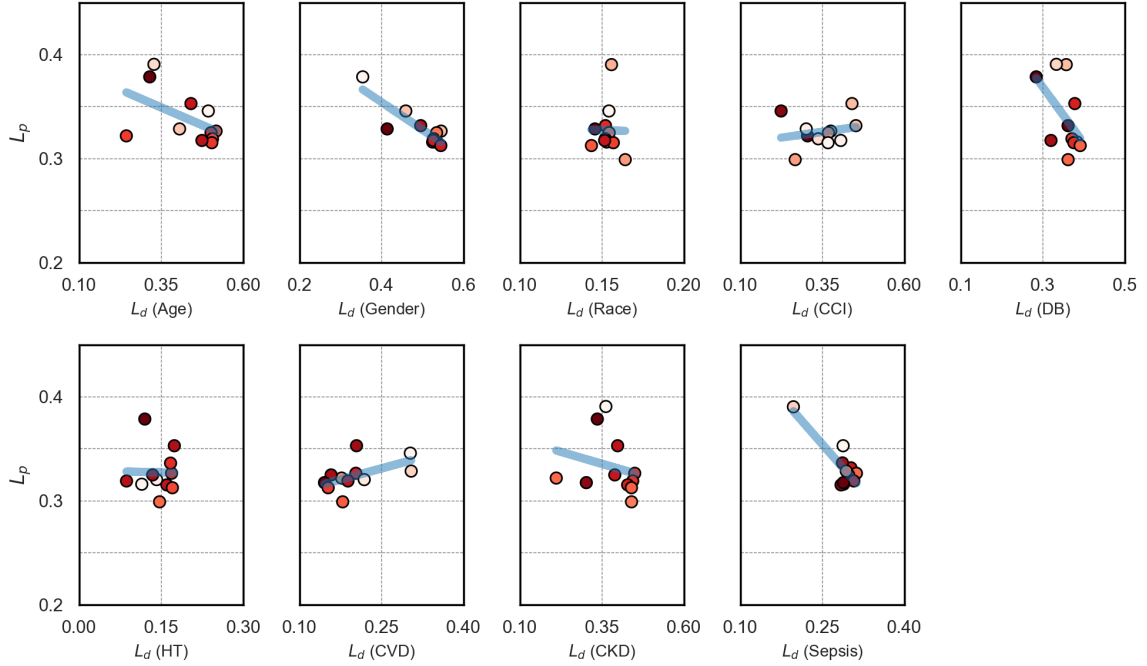


Figure 6.7: Relationship between outcome loss, domain loss and representation-wise attention in all domains. Y-axis represents the outcome loss, x-axis represents the domain loss. The colored dots represent the representation-wise attention, and darker color represents higher attention.

diction loss, where darker color indicates greater attention. It shows that darker dots are consistently associated with a lower outcome prediction loss, regardless of the correlation between the outcome prediction loss and domain loss. This suggests that RW-Attention can weigh domain-specific and domain-invariant representations toward more precise outcome prediction. The findings are consistent across all other domains, as demonstrated in Figure 6.7.

6.5.5 Impact of Masking Rate

We analyze the impact of different masking ratios on model performance. Figure 6.8 depicts ROCAUC, PRAUC, and accuracy performance metrics as a function of the masking ratio, ranging from 0.0 to 0.9. Our findings indicate that the appropriate

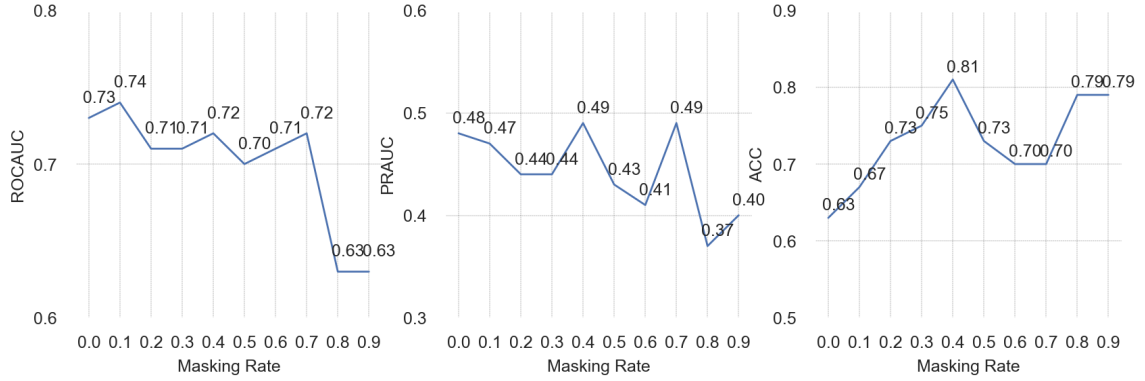


Figure 6.8: Performance Score of ROCAUC, PRAUC, and Accuracy with different masking rate

masking rate is highly dependent on the training data and that there is no universal masking rate that is optimal across all data. For our experimental data, the masking rate of 0.4 was found to be one of the most effective, yielding the highest accuracy and PRAUC and the third-best ROCAUC. These results suggest carefully selecting the masking rate is essential for maximizing model performance in different settings.

6.6 Conclusion

It is crucial to prioritize fairness and equity when designing and implementing healthcare AI models to ensure they serve all individuals and groups equitably. In this chapter, we present MTATE, an attention-based encoder for EHR data, which uses three different attention mechanisms to learn unbiased data representations. Our experiments on real-world healthcare data demonstrate that MTATE outperforms the compared state-of-the-art baselines and demonstrates the potential of MTATE to improve fairness and accuracy in healthcare AI and facilitate personalized medicine for diverse patient populations.

CHAPTER 7. Conclusion and Future Direction

7.1 Conclusion

Recent machine learning and deep learning methods for clinical risk prediction using electronic health records data have been deployed for many applications. However, the concerns about handling missingness, interpretability, and fairness issues have impeded its adoption in real healthcare settings. This dissertation focuses on developing machine learning and deep learning frameworks (Chapters 3, 4, 5, 6) for electronic health records data in addressing the three major limitations.

Chapter 3 presented a novel ensemble learning model (ELMV) to predict patient outcomes using EHR data with substantial missing values. ELMV makes ensemble predictions using multiple subsets with much lower missing rates and uses a support set from the training data to estimate the distribution of testing data to avoid biases. ELMV outperforms conventional missing value imputation methods and traditional ensemble learning models in both simulation and real-world experiments. One limitation of this model is that the effect of different missingness patterns or the reasons for missingness, such as MCAR (Missing Completely at Random), MAR (Missing At Random), and MNAR (Missing Not at Random) was not taken into account. Therefore, EHR data with different missingness patterns may cause the presented models to perform differently, which needs further caution and improvements. In the future, we plan to assess the effect of model performances using data with different missingness patterns and to modify the model architectures to be adapted to different types of missingness.

Chapter 4 presented a novel deep learning model (KGDAL) for rolling mortality prediction for temporal EHR data using prior knowledge and attention mechanism. With guidance from the knowledge graph, the two-dimensional attention mechanism

improves the model’s performance and interpretability. The experiment with two large healthcare datasets showed that KGDAL outperformed all the compared models. Also, KGDAL-derived patient risk trajectories may assist healthcare providers in making timely decisions and actions.

Chapter 5 presented a novel approach called Knowledge-guided Time-aware LSTM (KIT-LSTM). It is a new LSTM variant that uses two time-aware gates to address irregular and asynchronous multi-variable temporal EHR data issues and one knowledge-aware gate that infuses medical knowledge from ontology for better prediction power and interpretations. Experiments on real-world data demonstrate that KIT-LSTM performs better than the state-of-the-art baseline methods for predicting patients’ risk trajectories and model interpretation. As a result, KIT-LSTM can better support timely decision-making for clinicians.

In chapters 4 and 5, we introduced two deep learning models which used one sole ontology as the prior medical knowledge to guide the prediction and the interpretation of the model. However, including one ontology might impede performance improvement due to the concern about the completeness of medical ontology. In the future, we plan to integrate multiple ontologies for further improvement of the model and incorporate a personalized/customized knowledge graph to better capture the relationships between the temporal features for patients with different backgrounds.

Chapter 6 presented a masked triple attention transformer encoder (MTATE) to learn an unbiased and fair representation based on different subpopulations for Electronic Health Records (EHR) to address the unfairness issue for healthcare AI applications. Specifically, MTATE includes multiple domain classifiers to assist in learning diverse representations for different subpopulations by masking different randomly selected latent features. Furthermore, MTATE uses three attention mechanisms to learn attention regarding temporal relevance, feature relevance, and subpopulation relevance. Finally, the downstream prediction task is trained together with the domain

classifiers and the attention mechanisms. The experiments on real-world healthcare data show that MTATE performs better than the baseline models.

7.2 Future Direction

7.2.1 Transfer and Federated Learning for Transportable Healthcare AI

The transportability of AI models in healthcare refers to the ability of the model to be effectively and reliably used across different healthcare settings (i.e., produce accurate predictions on new sets of patients from different clinical settings [160]). Healthcare data can vary a lot across institutions due to different patient backgrounds, geographical locations, comorbidity status, etc., making it difficult to build models that maintain good prediction power and performance across different sites. The transportability of AI models is particularly critical to improving usability and scalability in healthcare settings.

Models presented in Chapter 4,5,6 are trained and validated on one dataset (either the proprietary EHR data from the University of Kentucky or the public MIMIC3 dataset). However, the transportability of all presented models across different institutions still needs to be discovered. Therefore, it would be worthwhile to explore the following two frameworks to improve the model transportability in healthcare settings:

- Use transfer learning to improve model transportability without requiring large amounts of labeled data from a different institution. The model is first trained on a large EHR dataset then the pre-trained model is adapted to make predictions on other EHR datasets from different hospitals. It can improve transportability by enabling the model to leverage knowledge learned from large, diverse datasets to adapt/transfer to new datasets from different hospitals.
- Use federated learning to improve model transportability without compromising

patient privacy. Data from multiple hospitals will be used to collaboratively train a shared model without sharing their raw data. First, each site trains and generates a local model on their local data and shares the model parameters with a central server. Then, the center server generates an aggregated model from different sites. Finally, the aggregated model is sent back to each local site for further training or testing.

7.2.2 Topic Modeling for EHR Data Harmonization

Training and validating DL models across different healthcare settings are challenging. For example, different hospitals may have heterogeneous data due to different data collection methods, terminology and codes, and various clinical practices such as diagnostic criteria, treatment protocols, and outcome measures. These variations result in non-interoperable data across hospitals, significantly hindering a model's transportability across different clinical settings. Techniques such as transferring previous EHR data into a common format and using a common data model are helpful. Or manual curation to harmonize different dataset are required to improve the transportability of AI models. However, these approaches can be time and labor-consuming. Therefore, in future work, we propose using topic modeling to support data harmonization from multiple healthcare systems. Specifically, topic modeling will be adopted to automatically identify and map different terminology and codes used by each hospital to the common topics. This would ensure faster and more accurate data harmonization for downstream prediction models.

7.2.3 Multi-modality Models for Personalized Medicine

Single modality models can achieve good prediction power, but the model is limited to the information contained within one modality. As a result, a single modality model may not be able to provide a complete picture of a patient's condition. For example,

models presented in Chapter 4,5,6 can identify and extract critical patterns and information for patients' risk prediction from laboratory measurements in EHR. Still, those models may miss critical information from other modalities, such as patient behavior observations and family disease history from clinical notes. Thus, developing multi-modality deep learning models that can handle variable data formats from different modalities such as laboratory measurement, unstructured text clinical notes, medical images, and genetic data is worthwhile. Multi-modality models can provide a more comprehensive view of a patient's condition and tailor treatment plans to each individual patient's needs.

Bibliography

- [1] Benjamin Shickel, Patrick James Tighe, Azra Bihorac, and Parisa Rashidi. Deep ehr: a survey of recent advances in deep learning techniques for electronic health record (ehr) analysis. *IEEE journal of biomedical and health informatics*, 22(5):1589–1604, 2017.
- [2] Pranjul Yadav, Michael Steinbach, Vipin Kumar, and Gyorgy Simon. Mining electronic health records (ehrs) a survey. *ACM Computing Surveys (CSUR)*, 50(6):1–40, 2018.
- [3] D Goldberg Goetz, Anton J Kuzel, Lisa Bo Feng, Jonathan P DeShazo, and Linda E Love. EhRs in primary care practices: benefits, challenges, and successful strategies. *The American journal of managed care*, 18(2):e48–54, 2012.
- [4] Chunhua Weng, Paul Appelbaum, George Hripcsak, Ian Kronish, Linda Busacca, Karina W Davidson, and J Thomas Bigger. Using ehRs to integrate research with patient care: promises and challenges. *Journal of the American Medical Informatics Association*, 19(5):684–687, 2012.
- [5] Rakesh Malhotra, Kianoush B Kashani, Etienne Macedo, Jihoon Kim, Josee Bouchard, Susan Wynn, Guangxi Li, Lucila Ohno-Machado, and Ravindra Mehta. A risk prediction score for acute kidney injury in the intensive care unit. *Nephrology Dialysis Transplantation*, 32(5):814–822, 2017.
- [6] Khaled Shawwa, Erina Ghosh, Stephanie Lanius, Emma Schwager, Larry Es-helman, and Kianoush B Kashani. Predicting acute kidney injury in critically ill patients using comorbid conditions utilizing machine learning. *Clinical Kidney Journal*, 2020.
- [7] Adnan Qayyum, Junaid Qadir, Muhammad Bilal, and Ala Al-Fuqaha. Secure and robust machine learning for healthcare: A survey. *IEEE Reviews in Biomedical Engineering*, 14:156–180, 2020.
- [8] Marzyeh Ghassemi, Tristan Naumann, Peter Schulam, Andrew L Beam, Irene Y Chen, and Rajesh Ranganath. A review of challenges and opportunities in machine learning for health. *AMIA Summits on Translational Science Proceedings*, 2020:191, 2020.
- [9] Ke Lin, Yonghua Hu, and Guilan Kong. Predicting in-hospital mortality of patients with acute kidney injury in the icu using random forest model. *International journal of medical informatics*, 125:55–61, 2019.
- [10] Thierry Verplancke, Stijn Van Looy, Dominique Benoit, Stijn Vansteelandt, Pieter Depuydt, Filip De Turck, and Johan Decruyenaere. Support vector machine versus logistic regression modeling for prediction of hospital mortality in critically ill patients with haematological malignancies. *BMC medical informatics and decision making*, 8(1):1–8, 2008.

- [11] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.
- [12] Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD’16, pages 785–794, New York, NY, USA, 2016. ACM.
- [13] Guilan Kong, Ke Lin, and Yonghua Hu. Using machine learning methods to predict in-hospital mortality of sepsis patients in the icu. *BMC medical informatics and decision making*, 20(1):1–10, 2020.
- [14] Tahani A Daghistani, Radwa Elshawi, Sherif Sakr, Amjad M Ahmed, Abdullah Al-Thwayee, and Mouaz H Al-Mallah. Predictors of in-hospital length of stay among cardiac patients: A machine learning approach. *International journal of cardiology*, 288:140–147, 2019.
- [15] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.
- [16] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- [17] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.
- [18] Edward Choi, Mohammad Taha Bahadori, Andy Schuetz, Walter F Stewart, and Jimeng Sun. Doctor ai: Predicting clinical events via recurrent neural networks. In *Machine learning for healthcare conference*, pages 301–318. PMLR, 2016.
- [19] G Maragatham and Shobana Devi. Lstm model for prediction of heart failure in big data. *Journal of medical systems*, 43(5):1–13, 2019.
- [20] Jiang Li, Xiaowei S Yan, Durgesh Chaudhary, Venkatesh Avula, Satish Mudiganti, Hannah Husby, Shima Shahjouei, Ardavan Afshar, Walter F Stewart, Mohammed Yeasin, et al. Imputation of missing values for electronic health record laboratory data. *npj Digital Medicine*, 4(1):1–14, 2021.
- [21] Zhen Hu, Genevieve B Melton, Elliot G Arsoniadis, Yan Wang, Mary R Kwaan, and Gyorgy J Simon. Strategies for handling missing clinical data for automated surgical site infection detection from the electronic health record. *Journal of biomedical informatics*, 68:112–120, 2017.
- [22] Muhammad Aurangzeb Ahmad, Carly Eckert, and Ankur Teredesai. Interpretable machine learning in healthcare. In *Proceedings of the 2018 ACM*

- international conference on bioinformatics, computational biology, and health informatics*, pages 559–560, 2018.
- [23] Alfredo Vellido. The importance of interpretability and visualization in machine learning for applications in medicine and health care. *Neural computing and applications*, 32(24):18069–18083, 2020.
- [24] Garrett Wilson and Diane J Cook. A survey of unsupervised deep domain adaptation. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(5):1–46, 2020.
- [25] Sherri Rose. Machine learning for prediction in electronic health data. *JAMA network open*, 1(4):e181404–e181404, 2018.
- [26] Baoyao Yang, Mang Ye, Qingxiong Tan, and Pong C Yuen. Cross-domain missingness-aware time-series adaptation with similarity distillation in medical applications. *IEEE Transactions on Cybernetics*, 2020.
- [27] Lucas Jing Liu, Hongwei Zhang, Jianzhong Di, and Jin Chen. Elmv: An ensemble-learning approach for analyzing electrical health records with significant missing values. In *Proceedings of the 11th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics, BCB '20*, New York, NY, USA, 2020. Association for Computing Machinery.
- [28] Lucas Jing Liu, Victor Ortiz-Soriano, Javier A. Neyra, and Jin Chen. Kgdal: Knowledge graph guided double attention lstm for rolling mortality prediction for aki-d patients. In *Proceedings of the 12th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics, BCB '21*, New York, NY, USA, 2021. Association for Computing Machinery.
- [29] L. Liu, V. Ortiz-Soriano, J. A. Neyra, and J. Chen. Kit-lstm: Knowledge-guided time-aware lstm for continuous clinical risk prediction. In *2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 1086–1091, Los Alamitos, CA, USA, dec 2022. IEEE Computer Society.
- [30] Sayan Putatunda and Kiran Rama. A comparative analysis of hyperopt as against other approaches for hyper-parameter optimization of xgboost. In *Proceedings of the 2018 International Conference on Signal Processing and Machine Learning*, pages 6–10, 2018.
- [31] Brian J Wells, Kevin M Chagin, Amy S Nowacki, and Michael W Kattan. Strategies for handling missing data in electronic health record derived data. *Egems*, 1(3), 2013.
- [32] Therese D Pigott. A review of methods for missing data. *Educational research and evaluation*, 7(4):353–383, 2001.

- [33] A Rogier T Donders, Geert JMG Van Der Heijden, Theo Stijnen, and Karel GM Moons. A gentle introduction to imputation of missing values. *Journal of clinical epidemiology*, 59(10):1087–1091, 2006.
- [34] Matthijs Blankers, Maarten WJ Koeter, and Gerard M Schippers. Missing data approaches in ehealth research: simulation study and a tutorial for nonmathematically inclined researchers. *Journal of medical Internet research*, 12(5):e54, 2010.
- [35] Melissa J Azur, Elizabeth A Stuart, Constantine Frangakis, and Philip J Leaf. Multiple imputation by chained equations: what is it and how does it work? *International journal of methods in psychiatric research*, 20(1):40–49, 2011.
- [36] Daniel McNeish. Missing data methods for arbitrary missingness with small samples. *Journal of Applied Statistics*, 44(1):24–39, 2017.
- [37] Julien Clavel, Gildas Merceron, and Gilles Escarguel. Missing data estimation in morphometrics: how much is too much? *Systematic biology*, 63(2):203–218, 2014.
- [38] Sukhdev Mishra and Diwakar Khare. On comparative performance of multiple imputation methods for moderate to large proportions of missing data in clinical trials: a simulation study. *J Med Stat Inform*, 2(1):9, 2014.
- [39] Yiran Dong and Chao-Ying Joanne Peng. Principled missing data methods for researchers. *SpringerPlus*, 2(1):222, 2013.
- [40] Janus Christian Jakobsen, Christian Gluud, Jørn Wetterslev, and Per Winkel. When and how should multiple imputation be used for handling missing data in randomised clinical trials—a practical guide with flowcharts. *BMC medical research methodology*, 17(1):162, 2017.
- [41] Joseph G Ibrahim, Haitao Chu, and Ming-Hui Chen. Missing data in clinical studies: issues and methods. *Journal of clinical oncology*, 30(26):3297, 2012.
- [42] Jia Li, Mengdie Wang, Michael S Steinbach, Vipin Kumar, and Gyorgy J Simon. Don’t do imputation: Dealing with informative missing values in ehr data analysis. In *2018 IEEE International Conference on Big Knowledge (ICBK)*, pages 415–422. IEEE, 2018.
- [43] Paul Madley-Dowd, Rachael Hughes, Kate Tilling, and Jon Heron. The proportion of missing data should not be used to guide decisions on multiple imputation. *Journal of clinical epidemiology*, 110:63–73, 2019.
- [44] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bannetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, et al. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information Fusion*, 58:82–115, 2020.

- [45] Ionuț ȚĂRANU. Data mining in healthcare: decision making and precision. *Database Systems Journal BOARD*, 33, 2016.
- [46] W James Murdoch, Chandan Singh, Karl Kumbier, Reza Abbasi-Asl, and Bin Yu. Interpretable machine learning: definitions, methods, and applications. *arXiv preprint arXiv:1901.04592*, 2019.
- [47] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. ” why should i trust you?” explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.
- [48] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4765–4774. Curran Associates, Inc., 2017.
- [49] Stan Lipovetsky and Michael Conklin. Analysis of regression in game theory approach. *Applied Stochastic Models in Business and Industry*, 17(4):319–330, 2001.
- [50] Erik Štrumbelj and Igor Kononenko. Explaining prediction models and individual predictions with feature contributions. *Knowledge and information systems*, 41(3):647–665, 2014.
- [51] Anupam Datta, Shayak Sen, and Yair Zick. Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems. In *2016 IEEE symposium on security and privacy (SP)*, pages 598–617. IEEE, 2016.
- [52] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.
- [53] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*, 2015.
- [54] Sneha Chaudhari, Gungor Polatkan, Rohan Ramanath, and Varun Mithal. An attentive survey of attention models. *arXiv preprint arXiv:1904.02874*, 2019.
- [55] Edward Choi, Mohammad Taha Bahadori, Joshua A Kulas, Andy Schuetz, Walter F Stewart, and Jimeng Sun. Retain: An interpretable predictive model for healthcare using reverse time attention mechanism. *arXiv preprint arXiv:1608.05745*, 2016.
- [56] Ying Sha and May D Wang. Interpretable predictions of clinical outcomes with an attention-based recurrent neural network. In *Proceedings of the 8th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, pages 233–240, 2017.

- [57] Fenglong Ma, Radha Chitta, Jing Zhou, Quanzeng You, Tong Sun, and Jing Gao. Dipole: Diagnosis prediction in healthcare via attention-based bidirectional recurrent neural networks. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1903–1911, 2017.
- [58] Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 2: Short papers)*, pages 207–212, 2016.
- [59] Maxat Kulmanov, Fatima Zohra Smaili, Xin Gao, and Robert Hoehndorf. Machine learning with biomedical ontologies. *biorxiv*, 2020.
- [60] Barry Smith, Werner Ceusters, Bert Klagges, Jacob Köhler, Anand Kumar, Jane Lomax, Chris Mungall, Fabian Neuhaus, Alan L Rector, and Cornelius Rosse. Relations in biomedical ontologies. *Genome biology*, 6(5):1–15, 2005.
- [61] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Neural Information Processing Systems (NIPS)*, pages 1–9, 2013.
- [62] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864, 2016.
- [63] Fan Gong, Meng Wang, Haofen Wang, Sen Wang, and Mengyue Liu. Smr: Medical knowledge graph embedding for safe medicine recommendation. *Big Data Research*, 23:100174, 2021.
- [64] Zhihuang Lin, Dan Yang, and Xiaochun Yin. Patient similarity via joint embeddings of medical knowledge graph and medical entity descriptions. *IEEE Access*, 8:156663–156676, 2020.
- [65] Vinay Srinivas Bharadhwaj, Mehdi Ali, Colin Birkenbihl, Sarah Mubeen, Jens Lehmann, Martin Hofmann-Apitius, Charles Tapley Hoyt, and Daniel Domingo-Fernández. Clep: A hybrid data-and knowledge-driven framework for generating patient representations. *Bioinformatics*, 2021.
- [66] Edward Choi, Mohammad Taha Bahadori, Le Song, Walter F Stewart, and Jimeng Sun. Gram: graph-based attention model for healthcare representation learning. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 787–795, 2017.
- [67] Changchang Yin, Rongjian Zhao, Buyue Qian, Xin Lv, and Ping Zhang. Domain knowledge guided deep learning with electronic health records. In *2019 IEEE International Conference on Data Mining (ICDM)*, pages 738–747. IEEE, 2019.

- [68] Lihong Song, Chin Wang Cheong, Kejing Yin, William K Cheung, Benjamin CM Fung, and Jonathan Poon. Medical concept embedding with multiple ontological representations. In *IJCAI*, volume 19, pages 4613–4619, 2019.
- [69] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *ACM Comput. Surv.*, 54(6), jul 2021.
- [70] Ravi B Parikh, Stephanie Teeple, and Amol S Navathe. Addressing bias in artificial intelligence in health care. *Jama*, 322(24):2377–2378, 2019.
- [71] Trishan Panch, Heather Mattie, and Rifat Atun. Artificial intelligence and algorithmic bias: implications for health systems. *Journal of global health*, 9(2), 2019.
- [72] Sara Gerke, Timo Minssen, and Glenn Cohen. Ethical and legal challenges of artificial intelligence-driven healthcare. In *Artificial intelligence in healthcare*, pages 295–336. Elsevier, 2020.
- [73] Robert Adragna, Elliot Creager, David Madras, and Richard Zemel. Fairness and robustness in invariant learning: A case study in toxicity classification. *arXiv preprint arXiv:2011.06485*, 2020.
- [74] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016.
- [75] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014.
- [76] Jian Shen, Yanru Qu, Weinan Zhang, and Yong Yu. Wasserstein distance guided representation learning for domain adaptation. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.
- [77] Sanjay Purushotham, Wilka Carvalho, Tanachat Nilanon, and Yan Liu. Variational recurrent adversarial deep domain adaptation. 2016.
- [78] Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C Courville, and Yoshua Bengio. A recurrent latent variable model for sequential data. *Advances in neural information processing systems*, 28, 2015.
- [79] Yuan Zhang, Xi Yang, Julie Ivy, and Min Chi. Time-aware adversarial networks for adapting disease progression modeling. In *2019 IEEE International Conference on Healthcare Informatics (ICHI)*, pages 1–11. IEEE, 2019.
- [80] Farzaneh Khoshnevisan and Min Chi. An adversarial domain separation framework for septic shock early prediction across ehr systems. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 64–73. IEEE, 2020.

- [81] Geert Molenberghs and Michael Kenward. *Missing data in clinical studies*, volume 61. John Wiley & Sons, 2007.
- [82] Roderick JA Little and Donald B Rubin. *Statistical analysis with missing data*, volume 793. John Wiley & Sons, 2019.
- [83] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.
- [84] Bradley Efron. Missing data, imputation, and the bootstrap. *Journal of the American Statistical Association*, 89(426):463–475, 1994.
- [85] S van Buuren and Karin Groothuis-Oudshoorn. mice: Multivariate imputation by chained equations in r. *Journal of statistical software*, pages 1–68, 2010.
- [86] Zahra Ghazanfari, Ali Akbar Haghdoost, Sakineh Mohammad Alizadeh, Jamileh Atapour, and Farzaneh Zolala. A comparison of hba1c and fasting blood sugar tests in general population. *International journal of preventive medicine*, 1(3):187, 2010.
- [87] William S Noble. What is a support vector machine? *Nature biotechnology*, 24(12):1565–1567, 2006.
- [88] Richard Bellman. Dynamic programming. *Science*, 153(3731):34–37, 1966.
- [89] Mohamed Aly. Survey on multiclass classification methods. *Neural Netw*, 19:1–9, 2005.
- [90] Claude Sammut and Geoffrey I. Webb, editors. *Leave-One-Out Cross-Validation*, pages 744–744. Springer US, Boston, MA, 2017.
- [91] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.
- [92] Theodore K Alexandrides, George Skroubis, and Fotis Kalfarentzos. Resolution of diabetes mellitus and metabolic syndrome following roux-en-y gastric bypass and a variant of biliopancreatic diversion in patients with morbid obesity. *Obesity surgery*, 17(2):176–184, 2007.
- [93] Dimitris Papadias and Yufei Tao. *Reverse Nearest Neighbor Query*, pages 2434–2438. Springer US, Boston, MA, 2009.
- [94] Joe H Ward Jr. Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301):236–244, 1963.
- [95] Michael J Berridge, Peter Lipp, and Martin D Bootman. The versatility and universality of calcium signalling. *Nature reviews Molecular cell biology*, 1(1):11–21, 2000.

- [96] Carina Ämmälä, Frances M Ashcroft, and Patrik Rorsman. Calcium-independent potentiation of insulin release by cyclic amp in single β -cells. *Nature*, 363(6427):356–358, 1993.
- [97] Paweena Susantitaphong, Dinna N Cruz, Jorge Cerda, Maher Abulfaraj, Fahad Alqahtani, Ioannis Koulouridis, and Bertrand L Jaber. World incidence of aki: a meta-analysis. *Clinical Journal of the American Society of Nephrology*, 8(9):1482–1493, 2013.
- [98] Lynda A Szczech, William Harmon, Thomas H Hostetter, Paul E Klotman, Neil R Powe, John R Sedor, Paul Smedberg, and Jonathan Himmelfarb. World kidney day 2009: problems and challenges in the emerging epidemic of kidney disease, 2009.
- [99] LaTonya J Hickson, Sanjay Chaudhary, Amy W Williams, John J Dillon, Suzanne M Norby, James R Gregoire, Robert C Albright Jr, James T McCarthy, Bjorg Thorsteinsdottir, and Andrew D Rule. Predictors of outpatient kidney function recovery among patients who initiate hemodialysis in the hospital. *American Journal of Kidney Diseases*, 65(4):592–602, 2015.
- [100] Lakhmir S Chawla, Paul W Eggers, Robert A Star, and Paul L Kimmel. Acute kidney injury and chronic kidney disease as interconnected syndromes. *New England Journal of Medicine*, 371(1):58–66, 2014.
- [101] Lakhmir S Chawla, Richard L Amdur, Susan Amodeo, Paul L Kimmel, and Carlos E Palant. The severity of acute kidney injury predicts progression to chronic kidney disease. *Kidney international*, 79(12):1361–1369, 2011.
- [102] Michael Heung, Diane E Steffick, Kara Zivin, Brenda W Gillespie, Tanushree Banerjee, Chi-yuan Hsu, Neil R Powe, Meda E Pavkov, Desmond E Williams, Rajiv Saran, et al. Acute kidney injury recovery pattern and subsequent risk of ckd: an analysis of veterans health administration data. *American Journal of Kidney Diseases*, 67(5):742–752, 2016.
- [103] Charuhas V Thakar, Annette Christianson, Jonathan Himmelfarb, and Anthony C Leonard. Acute kidney injury episodes and chronic kidney disease risk in diabetes mellitus. *Clinical journal of the American Society of Nephrology*, 6(11):2567–2572, 2011.
- [104] Vin-Cent Wu, Che-Hsiung Wu, Tao-Min Huang, Cheng-Yi Wang, Chun-Fu Lai, Chih-Chung Shiao, Chia-Hsui Chang, Shuei-Liong Lin, Yen-Yuan Chen, Yung-Ming Chen, et al. Long-term risk of coronary events after aki. *Journal of the American Society of Nephrology*, 25(3):595–605, 2014.
- [105] Ayodele Odutayo, Christopher X Wong, Michael Farkouh, Douglas G Altman, Sally Hopewell, Connor A Emdin, and Benjamin H Hunn. Aki and long-term risk for cardiovascular events and mortality. *Journal of the American Society of Nephrology*, 28(1):377–387, 2017.

- [106] Henrik Gammelager, Christian Fynbo Christiansen, Martin Berg Johansen, Else Tønnesen, Bente Jespersen, and Henrik Toft Sørensen. Three-year risk of cardiovascular disease among intensive care patients with acute kidney injury: a population-based cohort study. *Critical care*, 18(5):1–10, 2014.
- [107] Lowell J Lo, Alan S Go, Glenn M Chertow, Charles E McCulloch, Dongjie Fan, Juan D Ordoñez, and Chi-yuan Hsu. Dialysis-requiring acute renal failure increases the risk of progressive chronic kidney disease. *Kidney international*, 76(8):893–899, 2009.
- [108] Areef Ishani, Jay L Xue, Jonathan Himmelfarb, Paul W Eggers, Paul L Kimmel, Bruce A Molitoris, and Allan J Collins. Acute kidney injury increases risk of esrd among elderly. *Journal of the American Society of Nephrology*, 20(1):223–228, 2009.
- [109] Raymond K Hsu and Chi-yuan Hsu. The role of acute kidney injury in chronic kidney disease. In *Seminars in nephrology*, volume 36, pages 283–292. Elsevier, 2016.
- [110] Abraham Schoe, Ferishta Bakhshi-Raiey, Nicolette de Keizer, Jaap T van Dissel, and Evert de Jonge. Mortality prediction by sofa score in icu-patients after cardiac surgery; comparison with traditional prognostic-models. *BMC anesthesiology*, 20(1):1–8, 2020.
- [111] Jay L Koyner, Kyle A Carey, Dana P Edelson, and Matthew M Churpek. The development of a machine learning inpatient acute kidney injury prediction model. *Critical care medicine*, 46(7):1070–1077, 2018.
- [112] Alistair EW Johnson and Roger G Mark. Real-time mortality prediction in the intensive care unit. In *AMIA Annual Symposium Proceedings*, volume 2017, page 994. American Medical Informatics Association, 2017.
- [113] Soo Yeon Kim, Saehoon Kim, Joongbum Cho, Young Suh Kim, In Suk Sol, Youngchul Sung, Inhyeok Cho, Minseop Park, Haerin Jang, Yoon Hee Kim, et al. A deep learning model for real-time mortality prediction in critically ill children. *Critical care*, 23(1):1–10, 2019.
- [114] Anima Singh, Girish Nadkarni, Omri Gottesman, Stephen B Ellis, Erwin P Bottinger, and John V Guttag. Incorporating temporal ehr data in predictive models for risk stratification of renal function deterioration. *Journal of biomedical informatics*, 53:220–228, 2015.
- [115] Sebastian Köhler, Michael Gargano, Nicolas Matentzoglou, Leigh C Carmody, David Lewis-Smith, Nicole A Vasilevsky, Daniel Danis, Ganna Balagura, Gareth Baynam, Amy M Brower, et al. The human phenotype ontology in 2021. *Nucleic acids research*, 49(D1):D1207–D1217, 2021.

- [116] Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3:160035, 2016.
- [117] Sushrut S Waikar, Ron Wald, Glenn M Chertow, Gary C Curhan, Wolfgang C Winkelmayer, Orfeas Liangos, Marie-Anne Sosa, and Bertrand L Jaber. Validity of international classification of diseases, ninth revision, clinical modification codes for acute renal failure. *Journal of the American Society of Nephrology*, 17(6):1688–1694, 2006.
- [118] Peter N Robinson, Sebastian Köhler, Sebastian Bauer, Dominik Seelow, Denise Horn, and Stefan Mundlos. The human phenotype ontology: a tool for annotating and analyzing human hereditary disease. *The American Journal of Human Genetics*, 83(5):610–615, 2008.
- [119] Alexis Sardá-Espinosa. Time-series clustering in r using the dtwclust package. *The R Journal*, 2019.
- [120] Yu Cheng, Fei Wang, Ping Zhang, and Jianying Hu. Risk prediction with electronic health records: A deep learning approach. In *Proceedings of the 2016 SIAM International Conference on Data Mining*, pages 432–440. SIAM, 2016.
- [121] M.W. Kang, J. Kim, D.K. Kim, K. Oh, K.W Joo, Y.S Kim, and S. S. Han. Machine learning algorithm to predict mortality in patients undergoing continuous renal replacement therapy. *Critical Care*, 24(1):1–9, 2020.
- [122] J.R.A. Solares, F.E.D. Raimondi, Y. Zhu, F. Rahimian, D. Canoy, J. Tran, A.C.P. Gomes, A.H. Payberah, M. Zottoli, M. Nazarzadeh, et al. Deep learning for electronic health records: A comparative review of multiple deep neural architectures. *Journal of biomedical informatics*, 101:103337, 2020.
- [123] Kun-Hsing Yu, Andrew L Beam, and Isaac S Kohane. Artificial intelligence in healthcare. *Nature biomedical engineering*, 2(10):719–731, 2018.
- [124] Hun-Sung Kim, Dai-Jin Kim, and Kun-Ho Yoon. Medical big data is not yet available: why we need realism rather than exaggeration. *Endocrinology and Metabolism*, 34(4):349–354, 2019.
- [125] Nariman Noorbakhsh-Sabet, Ramin Zand, Yanfei Zhang, and Vida Abedi. Artificial intelligence transforms the future of health care. *The American journal of medicine*, 132(7):795–801, 2019.
- [126] S. Wu, S. Liu, S. Sohn, S. Moon, C. Wi, Y. Juhn, and H. Liu. Modeling asynchronous event sequences with rnns. *Journal of biomedical informatics*, 83:167–177, 2018.

- [127] Inci M Baytas, Cao Xiao, Xi Zhang, Fei Wang, Anil K Jain, and Jiayu Zhou. Patient subtyping via time-aware lstm networks. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 65–74, 2017.
- [128] Daniel Neil, Michael Pfeiffer, and Shih-Chii Liu. Phased lstm: Accelerating recurrent network training for long or event-based sequences. *arXiv preprint arXiv:1610.09513*, 2016.
- [129] Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. Recurrent neural networks for multivariate time series with missing values. *Scientific reports*, 8(1):1–12, 2018.
- [130] Wei Cao, Dong Wang, Jian Li, Hao Zhou, Lei Li, and Yitan Li. Brits: Bidirectional recurrent imputation for time series. *Advances in neural information processing systems*, 31, 2018.
- [131] Yuan Zhang. Attain: Attention-based time-aware lstm networks for disease progression modeling. In *In Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI-2019)*, pp. 4369–4375, Macao, China., 2019.
- [132] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu. Learning entity and relation embeddings for knowledge graph completion. In *Twenty-ninth AAAI conference on artificial intelligence*, 2015.
- [133] L.J. Liu, V. Ortiz-Soriano, J.A. Neyra, and J. Chen. Kgdal: knowledge graph guided double attention lstm for rolling mortality prediction for aki-d patients. In *Proceedings of the 12th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics*, pages 1–10, 2021.
- [134] I.T. Jolliffe and J. Cadima. Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065):20150202, 2016.
- [135] S. Santurkar, D. Tsipras, and A. Madry. Breeds: Benchmarks for subpopulation shift. *arXiv preprint arXiv:2008.04859*, 2020.
- [136] K. Chaudhary, A. Vaid, Á. Duffy, I. Paranjpe, S. Jaladanki, M. Paranjpe, K. Johnson, A. Gokhale, P. Pattharanitima, K. Chauhan, et al. Utilization of deep learning for subphenotype identification in sepsis-associated acute kidney injury. *Clinical Journal of the American Society of Nephrology*, 15(11):1557–1565, 2020.
- [137] Ben Shneiderman. Human-centered artificial intelligence: Reliable, safe & trustworthy. *International Journal of Human–Computer Interaction*, 36(6):495–504, 2020.

- [138] Milena A Gianfrancesco, Suzanne Tamang, Jinoos Yazdany, and Gabriela Schmajuk. Potential biases in machine learning algorithms using electronic health record data. *JAMA internal medicine*, 178(11):1544–1547, 2018.
- [139] Muhammad Aurangzeb Ahmad, Arpit Patel, Carly Eckert, Vikas Kumar, and Ankur Teredesai. Fairness in machine learning for healthcare. In *Proceedings of the 26th acm sigkdd international conference on knowledge discovery & data mining*, pages 3529–3530, 2020.
- [140] Cao Xiao, Edward Choi, and Jimeng Sun. Opportunities and challenges in developing deep learning models using electronic health records data: a systematic review. *Journal of the American Medical Informatics Association*, 25(10):1419–1428, 2018.
- [141] Farah Shamout, Tingting Zhu, and David A Clifton. Machine learning for clinical outcome prediction. *IEEE reviews in Biomedical Engineering*, 14:116–126, 2020.
- [142] Sanjay Purushotham, Wilka Carvalho, Tanachat Nilanon, and Yan Liu. Variational recurrent adversarial deep domain adaptation. In *International Conference on Learning Representations*, 2017.
- [143] Farzaneh Khoshnevisan and Min Chi. Unifying domain adaptation and domain generalization for robust prediction across minority racial groups. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 521–537. Springer, 2021.
- [144] Alexandra Chouldechova and Aaron Roth. The frontiers of fairness in machine learning. *arXiv preprint arXiv:1810.08810*, 2018.
- [145] Harini Suresh, Jen J Gong, and John V Guttag. Learning tasks for multitask learning: Heterogenous patient populations in the icu. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 802–810, 2018.
- [146] Sharmin Afrose, Wenjia Song, Charles B Nemeroff, Chang Lu, and Danfeng Daphne Yao. Subpopulation-specific machine learning prognosis for under-represented patients with double prioritized bias correction. *Communications medicine*, 2(1):1–14, 2022.
- [147] Changhee Lee and Mihaela Van Der Schaar. Temporal phenotyping using deep predictive clustering of disease progression. In *International Conference on Machine Learning*, pages 5767–5777. PMLR, 2020.
- [148] Henrique Aguiar, Mauro Santos, Peter Watkinson, and Tingting Zhu. Learning of cluster-based feature importance for electronic health record time-series. In *International Conference on Machine Learning*, pages 161–179. PMLR, 2022.

- [149] Hyeon-Ju Lee and Youn-Jung Son. Factors associated with in-hospital mortality after continuous renal replacement therapy for critically ill patients: a systematic review and meta-analysis. *International journal of environmental research and public health*, 17(23):8781, 2020.
- [150] Javier A Neyra and Girish N Nadkarni. Continuous kidney replacement therapy of the future: Innovations in information technology, data analytics, and quality assurance systems. *Advances in Chronic Kidney Disease*, 28(1):13–19, 2021.
- [151] Suvi T Vaara, Pavan K Bhatraju, Natalja L Stanski, Blaithin A McMahon, Kathleen Liu, Michael Joannidis, and Sean M Bagshaw. Subphenotypes in acute kidney injury: a narrative review. *Critical Care*, 26(1):1–10, 2022.
- [152] Hsin-Hsiung Chang, Jung-Hsien Chiang, Chi-Shiang Wang, Ping-Fang Chiu, Khaled Abdel-Kader, Huiwen Chen, Edward D Siew, Jonathan Yabes, Raghavan Murugan, Gilles Clermont, et al. Predicting mortality using machine learning algorithms in patients who require renal replacement therapy in the critical care unit. *Journal of Clinical Medicine*, 11(18):5289, 2022.
- [153] Sanjay Purushotham, Chuizheng Meng, Zhengping Che, and Yan Liu. Benchmarking deep learning models on large healthcare datasets. *Journal of biomedical informatics*, 83:112–134, 2018.
- [154] William Caicedo-Torres and Jairo Gutierrez. Iseeu: Visually interpretable deep learning for mortality prediction inside the icu. *Journal of biomedical informatics*, 98:103269, 2019.
- [155] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.
- [156] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschiot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33:18661–18673, 2020.
- [157] Isotta Landi, Benjamin S Glicksberg, Hao-Chih Lee, Sarah Cherng, Giulia Landi, Matteo Danieletto, Joel T Dudley, Cesare Furlanello, and Riccardo Miotto. Deep representation learning of electronic health records to unlock patient stratification at scale. *NPJ digital medicine*, 3(1):96, 2020.
- [158] Michael Feldman, Sorelle A Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. Certifying and removing disparate impact. In *proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 259–268, 2015.
- [159] Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. *Advances in neural information processing systems*, 29, 2016.

- [160] Xing Song, Alan SL Yu, John A Kellum, Lemuel R Waitman, Michael E Matheny, Steven Q Simpson, Yong Hu, and Mei Liu. Cross-site transportability of an explainable artificial intelligence model for acute kidney injury prediction. *Nature communications*, 11(1):5668, 2020.

Vita

Personal Information

- Name: Jing (Lucas) Liu

Education

- M.S. Electrical Engineering, Michigan Technological University, Houghton, MI, December 2016
- B.S. Electrical Engineering, Michigan Technological University, Houghton, MI, May 2014

Professional Experience

- Research Assistant, Department of Computer Science And Institute for Biomedical Informatics, University of Kentucky, Fall 2017-present
- Teaching Assistant, Department of Computer Science, University of Kentucky, Spring 2019
- Teaching Assistant, Department of Electrical Engineering, Michigan Technological University, 2014-2016
- Math Lab Consultant, Department of Mathematical Sciences, Michigan Technological University, 2010 - 2011

Publications

1. **Liu, L. J.**, Takeuchi, T., Chen, J., & Neyra, J. A. (2023). Artificial Intelligence in Continuous Kidney Replacement Therapy. *Clinical Journal of the American Society of Nephrology*, 10-2215.
2. **Liu L. J.**, Ortiz-Soriano, V., Neyra, J. A., & Chen, J. (2022, December). KIT-LSTM: Knowledge-guided Time-aware LSTM for Continuous Clinical Risk Prediction. In *2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)* (pp. 1086-1091). IEEE Computer Society.
3. **Liu L. J.**, Ortiz-Soriano, V., Neyra, J. A., Chen, J. (2021, August). KGDAL: knowledge graph guided double attention LSTM for rolling mortality prediction for AKI-D patients. In *Proceedings of the 12th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics* (pp. 1-10).
4. **Liu L. J.**, Zhang, H., Di, J., Chen, J. (2020, September). ELMV: an Ensemble-Learning Approach for Analyzing Electrical Health Records with Significant Missing Values. In *Proceedings of the 11th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics* (pp. 1-10).

5. **Liu J.**, Abeysinghe, R., Zheng, F., Cui, L. (2019, June). Pattern-based Extraction of Disease Drug Combination Knowledge from Biomedical Literature. In 2019 IEEE International Conference on Healthcare Informatics (ICHI) (pp. 1-7). IEEE.
6. Ortiz-Soriano, V., Cama-Olivares, A., **Liu L. J.**, Armentrout, B., Colohan, D., Paladiya, R., ... & Neyra, J. A. The optimization of outpatient hemodialysis management for AKI-D patients: A quality improvement study. *American Journal of Nephrology*. (2023).
7. Neyra, J. A., Ortiz-Soriano, V., **Liu L. J.**, Smith, T. D., Li, X., Xie, D., ... & Chen, J. (2023). Prediction of mortality and major adverse kidney events in critically ill patients with acute kidney injury. *American Journal of Kidney Diseases*, 81(1), 36-47.
8. Bastin, M. L. T., Stromberg, A. J., Nerusu, S. N., **Liu L. J.**, Mayer, K. P., Liu, K. D., ... & Neyra, J. A. (2022). Association of phosphate-containing versus phosphate-free solutions on ventilator days in patients requiring continuous kidney replacement therapy. *Clinical Journal of the American Society of Nephrology*, 17(5), 634-642.
9. Ly, H., Ortiz-Soriano, V., **Liu L. J.**, Liu, Y., Chen, J., Chang, A. R., ..., Neyra, J. A. (2021). Characteristics and Outcomes of Survivors of Critical Illness and Acute Kidney Injury Followed in a Pilot Acute Kidney Injury Clinic. *Kidney International Reports*.
10. Malluche, H. H., Chen, J., Lima, F., **Liu L. J.**, Monier-Faugere, M. C., Pienkowski, D. Bone Quality and Fractures in Women with Osteoporosis Treated with Bisphosphonates for One to Fourteen Years. *JBMR Plus*, e10549.
11. Jordan, M., Ortiz-Soriano, V., Pruitt, A., Chism, L., **Liu L. J.**, Chaaban, N., ..., Neyra, J. A. (2021). Kidney Recovery in Patients With Acute Kidney Injury Treated in Outpatient Hemodialysis or Rehabilitation Facilities. *Kidney Medicine*.
12. Bharath, L.P., Agrawal, M., McCambridge, G., Nicholas, D.A., Hasturk, H., **Liu L. J.**, Jiang, K., Liu, R., Guo, Z., Deeney, J. and Apovian, C.M., 2020. Metformin enhances autophagy and normalizes mitochondrial function to alleviate aging-associated inflammation. *Cell metabolism*, 32(1), pp.44-55. [**Cell Metabolism Best of 2020**]