To the Graduate Council:
I am submitting herewith a dissertation written by Zekai Wang entitled "Advanced Machine Learning for Data-driven Disease Prediction." I have examined the final paper copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Industrial and Systems Engineering.

Bing Yao, Major Professor

We have read this dissertation
and recommend its acceptance:

Xueping Li

Anahita Khojandi

Lan Gao

Accepted for the Council:

Dixie L. Thompson

Vice Provost and Dean of the Graduate School

To the Graduate Council:
I am submitting herewith a dissertation written by Zekai Wang entitled "Advanced Machine Learning for Data-driven Disease Prediction." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Industrial and Systems Engineering.

Bing Yao, Major Professor

We have read this dissertation
and recommend its acceptance:

Xueping Li

Anahita Khojandi

Lan Gao

Accepted for the Council:

Dixie L. Thompson

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

Advanced Machine Learning for Data-driven Disease Prediction

A Dissertation Presented for the

Doctor of Philosophy

Degree

The University of Tennessee, Knoxville

Zekai Wang

August 2024

© by Zekai Wang, 2024
All Rights Reserved.

To My Dear Parents

Guixiang Wang and Yulan Liang

To My Beloved Wife

Yicheng Du

Acknowledgements

I am deeply grateful to the many individuals who have supported me throughout the journey of completing my dissertation.

First and foremost, I would like to express my profound gratitude to my advisor, Dr. Bing Yao. Under your guidance, I have grown from a novice researcher to a competent and confident scholar. Your mentorship has been invaluable, providing me with the skills, knowledge, and inspiration to pursue rigorous and impactful research. Your patience, insightful feedback, and unwavering support have been crucial to my academic and personal development.

I am grateful to Dr. Tieming Liu at Oklahoma State University. Your support during my job search, as well as your academic guidance and advice, have been immensely beneficial to me. Your encouragement and practical insights have greatly contributed to my professional growth and development.

I also want to express my deep appreciation to Dr. Suhao Chen. Your support has extended beyond academic advice to offering guidance on many aspects of my personal life. Your help and counsel have been invaluable, and I am deeply grateful for your kindness and mentorship.

Additionally, I am thankful to my dissertation committee members at the University of Tennessee Knoxville: Dr. Xueping Li, Dr. Anahita Khojandi, and Dr. Lan Gao, and at Oklahoma State University: Dr. Tieming Liu, Dr. Farzad Yousefian and Dr. Ye Liang. Your support, guidance, and insightful comments have been crucial in refining my research and completing this dissertation.

I am forever indebted to my parents, Guixiang Wang and Yulan Liang, for their unwavering support of my decisions. Whether it was coming to the United States for study, changing my major, or pursuing a Ph.D., you have always stood by me without hesitation. Your belief in my potential and your constant encouragement have been my greatest source of strength.

Lastly, I would like to express my deepest gratitude to my beloved wife, Yicheng Du. You have witnessed every step of my Ph.D. journey. From our meeting in Dallas to moving together for my studies in Oklahoma, Tennessee, and Georgia, your unwavering belief in me and your endless support have been my anchor. No matter the challenges we faced, your faith in me never wavered, and for that, I am eternally grateful.

Thank you all for your contributions and support. This dissertation is as much yours as it is mine.

Abstract

The rapid advancement in sensing and information technology has ushered us into an era of data explosion, where a large amount of data is now easily available and accessible in the clinical environment. This wealth of healthcare data offers new avenues for developing automated data-driven methods for disease diagnosis. Electronic Health Records (EHRs), serving as digital repositories of a patient's medical information, present unique opportunities to analyze and decipher clinical events and patterns within large populations. Given the rich information about a patient's health trajectory, leveraging EHRs through data-driven methodologies can significantly enhance clinical decision support systems.

However, utilizing real-world EHRs for reliable data-driven disease detection presents several challenges due to the observational nature of EHRs. Unlike well-defined, randomized longitudinal experiments, EHRs are recorded only when patients receive care, resulting in complex and highly heterogeneous data. This dissertation focuses on developing robust and reliable advanced machine-learning algorithms to investigate complex EHR data for disease prediction, addressing the challenges of diverse and complex data types, incomplete datasets, and imbalanced class distributions in EHRs.

Table of Contents

1 Introduction **1**

1.1 Motivation 1

1.2 Research Background 4

1.2.1 Data-driven methods for longitudinal EHR data 4

1.2.2 Data-driven methods for waveform data 8

1.2.3 Data-driven methods for imbalance and incompleteness issues 9

1.2.4 Research objectives 10

1.2.5 Organization of the dissertation 12

2 Multi-branching temporal convolutional network for sepsis prediction **14**

2.1 Introduction 15

2.2 Research Background 17

2.3 Research Methodology 20

2.3.1 Multivariate Time Series Classification 20

2.3.2 Temporal Convolutional Network (TCN) 21

2.3.3 Multi-Branching Outputs 23

2.3.4 Gaussian Process-based Active Learning for Optimal Network Hyperparameters 26

2.4 Experimental Design and Results 29

2.4.1 Data Description and Experimental Design 29

2.4.2 Experimental Results . . . 30
2.4.3 Comparison Study with Existing Literature . . . 34
2.5 Discussion . . . 35
2.5.1 Interpretation on Performance Improvement of MB-TCN . . . 35
2.5.2 Limitation of the MB-TCN Framework . . . 36
2.5.3 The Impact of Sample Length on Performance of MB-TCN . . . 37
2.5.4 The Impact of Branch Number on Performance of MB-TCN . . . 38
2.6 Conclusions . . . 38

3 Hierarchical deep learning with generative adversarial network for automatic cardiac diagnosis from ECG signals 40
3.1 Introduction . . . 41
3.2 Research Background . . . 43
3.3 Research methodology . . . 45
3.3.1 MadeGAN for First-level Anomaly Detection . . . 46
3.3.2 Transfer learning-enhanced second-level classification . . . 51
3.4 Experimental Design and Results . . . 54
3.4.1 Data Description and Experimental Design . . . 54
3.4.2 Training and Testing Procedures . . . 56
3.4.3 Experimental Results . . . 58
3.4.4 Long-term ST Database . . . 63
3.4.5 Experimental Results from the First-level MadeGAN . . . 64
3.4.6 Experimental Results from the Second-level Classification . . . 64
3.4.7 Discussion on the Limitation . . . 65
3.5 Conclusions . . . 66

4 Multi-branching temporal convolutional network with tensor data completion for diabetic retinopathy prediction 67
4.1 Introduction . . . 68
4.2 Research Background . . . 71

4.2.1 DR screening methods in current clinical practice 71
4.2.2 Machine Learning for Healthcare Data 71
4.2.3 Existing Work on Data-Driven DR Prediction 72
4.3 Research Methodology 75
4.3.1 Tensor Completion for Missing Data Imputation 75
4.3.2 Missingness-informed TCN 78
4.3.3 Multi-Branching Outputs 80
4.4 Experimental Design and Results 81
4.4.1 Data Source and Data Extraction 81
4.4.2 Experimental Design 82
4.4.3 Experimental Results 83
4.5 Discussion 85
4.5.1 Overfitting or underfitting issues in model training 85
4.5.2 Performance comparison with existing literature 86
4.5.3 Discussion on the performance improvement 87
4.5.4 Broad applications 88
4.5.5 Challenges in real-world clinical implementation 89
4.6 Conclusions 90
4.7 Acknowledgement 91
4.8 Data Availability Statement 91

5 MUSE-Net: Missingness-aware mUlti-branching Self-attention Encoder for Irregular Longitudinal Electronic Health Records 92
5.1 Introduction 93
5.2 Research Background 95
5.3 Research Methodology 98
5.3.1 Multi-task Gaussian Process (MGP) for Missing Value Imputation 99

5.3.2 Missingness-aware Multi-branching Self-Attention Encoder (MUSE-Net) . . . 102
5.4 Experimental Design and Results . . . 107
5.4.1 Experimental Results in Synthetic Case Study . . . 108
5.4.2 Experimental Results in Real-world Case Study . . . 111
5.5 Conclusions . . . 113
5.6 Acknowledgement . . . 113
5.7 Data Availability Statement . . . 114

6 Conclusions **115**

Bibliography **120**

Appendices **147**

A Supplement Materials of Chapter 2 **148**
A.1 Dynamic time wrapping (DTW)-based Hierarchical Clustering (DTW-HC) . . . 148

B Supplement Materials of Chapter 5 **150**
B.1 Parameter Specifications in Synthetic Data Generation . . . 150
B.2 Robust Classifier from Multi-branching Network Models . . . 150
B.2.1 Algorithm for GPU Acceleration in MGP Imputation . . . 155

C Figures **159**

D Tables **174**

Vita **181**

Chapter 1

Introduction

1.1 Motivation

The rapid advancement in sensing and information technology has ushered us into an era of data explosion where a large amount of data is now easily available and accessible in the clinical environment. The wealth of healthcare data offers new avenues for developing automated data-driven methods for disease diagnosis. For instance, there have been growing research interests in harnessing electronic health records (EHRs) to create data-driven solutions for clinical decision support. EHRs serve as digital repositories of a patient's medical information including demographics, progress notes, medications, vital signs, medical history, and lab results. Given the rich information about a patient's health trajectory, EHRs present unique opportunities to analyze and decipher clinical events and patterns within large populations through data-driven methodologies.

Data-driven methods for disease prediction are essential due to their ability to analyze vast amounts of complex healthcare data, uncover hidden patterns, and provide accurate, timely insights that traditional diagnostic methods might miss. As healthcare systems increasingly generate massive volumes of data from EHRs, medical imaging, and genomics, there is a growing need to leverage this data effectively.

Traditional diagnostic approaches often rely on manual analysis, which can be time-consuming, subject to human error, and limited by the clinician's expertise. In contrast, data-driven methods can process and analyze large datasets efficiently, identifying subtle correlations and trends that may not be apparent to human observers. These methods enhance the precision and speed of diagnosis, support personalized treatment plans by considering the unique health profiles of patients, and enable predictive analytics that can forecast disease progression and outcomes. By integrating data-driven approaches into healthcare, we can improve diagnostic accuracy, optimize resource allocation, and ultimately enhance patient outcomes, making these methods indispensable in modern medical practice.

However, data mining of EHR data poses distinctive research challenges due to the observational nature of EHRs. Unlike well-defined, randomized longitudinal experiments in clinical trials that collect data on a fixed schedule and ensure high data quality, EHRs are recorded only when patients receive care or doctors provide services. The information collected and the timing of its collection are not determined by researchers, resulting in EHR data that are often complex and highly heterogeneous. Effectively utilizing real-world EHRs for reliable data-driven disease detection presents several challenges. This dissertation will focus on addressing the following challenges:

(1) **Diversity and complexity of data types**. EHR datasets often contain diverse data types. For example, EHR datasets are commonly associated with multivariate longitudinal records (or sequences). The medical variables usually not only have intra-sequence correlation (i.e., autocorrelation within one individual sequence) but also have inter-sequence correlation (i.e., the correlation between different sequences). In addition, for many chronic patients, such as Diabetic Retinopathy (DR) patients, the EHR data are often documented during irregular patient visits, leading to non-uniform time intervals between successive measurements and a lack of synchronization across various medical variables or among different patients. Traditional time series models face challenges when applied to irregular

longitudinal data because they typically assume a parametric form of the temporal variables, making it difficult to effectively account for highly heterogeneous and irregular time intervals across different variables. Additionally, the widely used deep learning models such as convolution neural networks (CNNs) and recurrent neural networks (RNNs) for mining sequential or time series data are designed by assuming consecutive data points are collected at a uniform time interval. Those deep learning models do not consider the elapsed time between records and are less effective in modeling irregular longitudinal EHRs.

The other common data type in EHRs is waveform data. Waveform data are generated at a higher frequency than most other types of health data, resulting in large volumes of data over short periods. A common example of waveform data is electrocardiograms (ECGs). ECG signals are recordings of the electrical activity of the heart over a period of time. Real-world ECG signals contain complex patterns because cardiovascular systems are featured with nonlinear and nonstationary dynamics from the complicated interactions of many interconnected parts such as ion channels and gap junctions to perform cardiac functions. Thus, new machine learning algorithms that can effectively and automatically extract useful features are needed for reliable ECG signal analysis.

(2) **Incomplete datasets and imbalanced class distributions**. Real-world EHRs suffer from the issues of missing values and imbalanced data. Due to the nature of clinical practice, not all information is recorded for every patient visit, leading to incomplete datasets. Additionally, EHR data often exhibit a significantly imbalanced distribution, with certain health outcomes or characteristics being underrepresented compared to more common ones. For instance, rare diseases or adverse drug reactions may have very few instances compared to more prevalent conditions. In the existing literature, a wide array of statistical and machine learning techniques have been designed to tackle the missing value and imbalanced data issues, which, however, are less applicable in the context of modeling irregular longitudinal EHRs. The presence of missing values and imbalanced class distributions will introduce further difficulties

in effective model training using longitudinal EHRs, leading to biased or inaccurate predictions if not properly addressed.

Thus, this dissertation focuses on developing robust and reliable advanced machine-learning algorithms to investigate complex EHR data for disease prediction, addressing the challenges of diverse and complex data types, incomplete datasets, and imbalanced class distributions in EHRs.

1.2 Research Background

1.2.1 Data-driven methods for longitudinal EHR data

The integration of data-driven modeling methodologies with EHR data has transformed the healthcare field, offering unprecedented opportunities for clinical decision support (Yadav et al., 2018; Landi et al., 2020; Rajpurkar et al., 2022). Traditional machine learning methods, such as decision trees, naive Bayes classifiers, support vector machines (SVMs), and k-nearest neighbors (k-NN), have been widely used due to their simplicity, interpretability, and effectiveness in structured data analysis (Hastie et al., 2009). Extensive research has been conducted to employ machine learning models for disease identification using EHRs that consist of cross-sectional patient health information (Banda et al., 2018; Wang et al., 2022). For example, Huang et al. (Huang et al., 2007) employed naive Bayes, decision trees, and nearest neighbor algorithms with feature selection methods to determine key factors affecting type II diabetes control and identify individuals with suboptimal diabetes control status. Hong et al. (Hong et al., 2019a) developed a multi-class classification method to analyze clinical data, identifying patients with obesity and various comorbidities using logistic regression, support vector machines, and decision tree algorithms. A comprehensive review of machine learning applications to non-longitudinal EHRs can be found in (Xiao et al., 2018; Rajkomar et al., 2019).

However, these approaches face significant challenges when dealing with the complexities of longitudinal data. They often require extensive manual feature engineering to capture temporal dependencies, which is both time-consuming and prone to errors. Furthermore, traditional methods typically assume that data points are independent and identically distributed, which is not the case in longitudinal data where observations are correlated over time. This assumption limits their effectiveness in capturing the dynamic nature of patient health trajectories (Banda et al., 2018; Das et al., 2022). Despite these limitations, traditional machine learning methods have laid the foundation for more advanced techniques, providing initial insights and establishing a baseline for disease prediction models

With the growing availability of longitudinal EHRs, there is increasing interest in developing advanced data-driven models to capture critical temporal information for disease prediction. Due to their outstanding performance and strong capability in pattern recognition, deep learning techniques have been widely explored to mine complexly structured data by leveraging the nonlinear interactions among neurons within and across various hidden layers (Xie and Yao, 2022a; Xie et al., 2023; Xie and Yao, 2022c; Wang et al., 2023c). Unlike conventional statistical and machine learning methods, deep learning does not require sophisticated feature engineering or strict model assumptions. Deep Neural Networks (DNNs) instead operate as a form of representation learning, where models are provided with raw data and automatically develop the necessary representative features for pattern recognition. A DNN generally consists of L parametric functions, also known as layers where each layer is considered as a transformed representation of the input domain (Ismail Fawaz et al., 2019). Specifically, a layer l_i takes the output of the previous layer l_{i-1} as its input and applies a non-linear activation function, such as ReLu and sigmoid, to compute the output of the current layer.

In healthcare domain, many advanced network architectures have been crafted for modeling time series or sequential data (Miotto et al., 2018). For instance, Recurrent Neural Networks (RNNs) are the famous families of DNNs for modeling longitudinal

EHRs. Lipton et al (Lipton et al., 2015) were the pioneers in proposing the use of Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) for recognizing patterns within multivariate clinical time series data. Building on this research, many researchers employed models from the RNNs family to analyze regularly-spaced EHRs for various clinical tasks (Futoma et al., 2017; Choi et al., 2017; Che et al., 2018; Chen et al., 2022). In addition, Temporal Convolutional Networks (TCNs) have gained increasing popularity as a robust alternative to the RNNs family for modeling longitudinal EHRs (Wang and Yao, 2021; Wang et al., 2024; Chen et al., 2022). With the unique architecture of dilated and causal 1-D convolution and residual blocks (Bai et al., 2018), TCNs can capture long-term dependencies in data, which allows them to access a broader range of input information at each layer. This capability enables TCNs to outperform RNNs in various tasks involving complex temporal relationships within EHR data Wang and Yao (2021); Wang et al. (2024); Chen et al. (2022); Rosnati and Fortuin (2021); Moor et al. (2019).

To address the issue of irregular time intervals in some specific EHR data, many modified RNN architectures have been developed. For example, the time-aware LSTM (TLSTM) was designed to account for non-uniform sampling intervals through a time decay mechanism (Baytas et al., 2017). Che et al. (Che et al., 2018) proposed a GRU-D model for irregular sampled time series. It also integrates a decay mechanism to model the impact of missing sampled values over time, allowing the network to better capture temporal dependencies even when data points are irregular or incomplete. A comprehensive review of modified RNN architectures for irregular sampled time series can be found in (Sun et al., 2020; Weerakody et al., 2021).

However, those modified RNNs assume that the impact of historical risk factors on disease prediction proportionally decays over time, which may not accurately describe complex disease trajectories. Additionally, RNNs have been widely recognized as computationally inefficient for modeling large-scale, long sequential data due to their sequential processing nature. The self-attention encoder, a cornerstone of the transformer architecture (Vaswani et al., 2017), has revolutionized the field of

natural language processing (NLP) with its unique advantages. Unlike RNNs and TCNs that process data sequentially and require uniform sampling intervals, the self-attention mechanism allows the model to weigh the importance of different parts of the input sequences relative to each other. This feature enables parallel processing and enhances the model's ability to capture long-range dependencies in irregularly spaced longitudinal signals.

Various approaches leverage the transformer/self-attention architecture to improve prediction accuracy and robustness in time series analysis. For instance, the Vision Transformer for Irregularly Sampled Time Series (ViTST) transforms time series data into images and applies vision transformers to capture temporal dependencies effectively Li et al. (2024). The Self-Supervised Transformer for Sparse and Irregularly Sampled Multivariate Clinical Time-Series (STraTS) employs a continuous value embedding technique to handle continuous time and variable values without the need for imputation, thus preserving fine-grained information Tipirneni and Reddy (2022). Another approach, the Deformable Neighborhood Attention Transformer (DNA-T), introduces deformable attention mechanisms to capture local and global dependencies in medical time series data Huang et al. (2024). These models demonstrate significant improvements over traditional methods, particularly in handling the irregularities and sparsity inherent in many real-world datasets.

However, some drawbacks persist across these novel transformer-based architectures. The transformation of time series into images or triplets can lead to information loss and may not capture all nuances of the data. Moreover, these architectures often require substantial computational resources due to the complexity of the models, making them less practical for large-scale or real-time applications. In addition, despite the strengths of self-attention encoders/transformer architecture, it is worth noting that most standard deep learning algorithms are designed under the assumption that data is ideal, which cannot be directly applied to mine longitudinal EHR databases with missing values and imbalanced class distributions. Little has been done to design an effective framework to model real-world longitudinal EHRs

by accounting for the issues of missing values, imbalanced data distribution, and irregular time intervals for robust decision-making.

1.2.2 Data-driven methods for waveform data

Similar to the longitudinal EHR data, a wide range of statistical and machine learning algorithms have been developed for waveform data analysis and pattern recognition. Techniques ranging from dynamic spatiotemporal warping to XGBoost and wavelet-based approaches have been employed for classifications such as myocardial infarction and heartbeat detection (Yang et al., 2013; Hasnul et al., 2021; Bertsimas et al., 2021). However, most traditional data-driven methods depend heavily on manual feature engineering, which is labor-intensive, prone to trial-and-error, and generally limited by human expert knowledge (Guglin and Thatai, 2006; Schläpfer and Wellens, 2017).

Deep neural networks (DNNs) automate feature extraction, demonstrating superior performance in arrhythmia detection without the need for explicit feature engineering (LeCun et al., 2015). Nonetheless, most existing deep learning algorithms for waveform analysis are based on supervised learning, which requires a large volume of annotated waveform signals and suffers from the problem of extremely imbalanced data. Consequently, the application of unsupervised and semi-supervised learning in waveform analysis has been increasingly investigated. For example, autoencoders (AEs), a semi-supervised deep learning technique, have been widely used to study waveform data by extracting critical low-dimensional representations of the raw signals for disease prediction (Xia et al., 2018). Furthermore, GAN-based frameworks, another semi-supervised learning technique to capture inherent data distributions (Goodfellow et al., 2014), have been applied in waveform analysis. For example, Zhou *et al.* (Zhou et al., 2019) developed a BeatGAN structure to model ECG signals for anomaly detection.

However, most existing semi-supervised deep learning methods primarily focus on differentiating abnormal waveforms from normal ones (i.e., binary classification)

and are not capable of performing multi-class classification to identify different types of cardiac arrhythmias. Novel analytical models are needed to efficiently handle the imbalanced data issue and the data-lacking problem for both robust anomaly detection and accurate disease identification from ECG signals.

1.2.3 Data-driven methods for imbalance and incompleteness issues

Various strategies have been developed to address the issue of imbalanced data (Krawczyk, 2016). These strategies generally fall into two categories: data-level and algorithm-level methods. Data-level methods, including undersampling, oversampling, and synthetic data generation techniques like SMOTE (Chawla et al., 2002), aim to adjust the distribution of datasets to make them more balanced. Algorithm-level methods, such as cost-sensitive learning, modify the algorithms themselves to reduce bias towards the majority class (He and Garcia, 2009). However, these approaches have their drawbacks. Sampling methods like undersampling and oversampling can lead to the loss of crucial information or the addition of redundant data, respectively. Synthetic methods may face challenges such as lack of realism, limited variability, and difficulties in validating models. They are also not well-suited for longitudinal data, compromising their effectiveness in generalizing to real-world scenarios (He and Garcia, 2009). Additionally, setting precise cost parameters in cost-sensitive methods poses a significant challenge in practical applications, making it difficult to effectively implement these approaches in real-world scenarios (Krawczyk, 2016).

Similarly, to tackle missing values, various imputation strategies for clinical data have been studied, each with unique approaches and drawbacks (Cismondi et al., 2013). Simple imputation methods, such as mean/carry forward imputation and interpolation/extrapolation imputation, may result in the loss of critical information and potentially introduce bias into datasets. Advanced model-based methods, like

tree-based (Doove et al., 2014) imputation and multiple imputations by chained equations (MICE) (Azur et al., 2011), consider relationships between variables to predict missing values using machine learning models. However, many of these methods are primarily designed for cross-sectional imputation (i.e., imputing missing measurements at the same time point) and may not be suitable for longitudinal clinical data.

1.2.4 Research objectives

My research aims to develop robust and reliable advanced machine learning algorithms to investigate the real-world complex EHR data for disease prediction. Specifically, the objectives of this dissertation include:

- **Topic 1:** *Multi-branching temporal convolutional network for sepsis prediction. (Published in (Wang and Yao, 2021))*
  - Proposing a missingness-informed multi-branching temporal convolutional network (MB-TCN) to model the multivariate longitudinal medical data for robust sepsis prediction.
  - Proposing the multi-branching architecture and missing value Masks module to effectively handle the missing values and imbalanced data issues.
  - Providing an empirical method to quantify the uncertainty in data-driven predictive modeling.
- **Topic 2:** *Hierarchical deep learning with generative adversarial network for automatic cardiac diagnosis from ECG signals (Published in (Wang et al., 2023c))*
  - Designing a two-level hierarchical deep learning framework leveraging Generative Adversarial Network (GAN) for enhanced cardiac diagnosis from ECG signals.

  - Integrating memory-augmented deep autoencoder (AE) with adversarial training framework (MadeGAN) to recognize the diversity of patterns of normal signals and mitigate the over-fitting issues in traditional AE for anomaly detection in ECG signals.
  - Applying transfer learning from MadeGAN and coupled with a multi-branching architecture for robust multi-class arrhythmias classification, which is effective to address the data scarcity and imbalanced issues.

- **Topic 3:** *Multi-branching temporal convolutional network with tensor data completion for diabetic retinopathy prediction (Published in (Wang et al., 2024))*
  - Employing the CANDECOMP/PARAFAC (CP) decomposition to complete the tensor data for missing value imputation by simultaneously accounting for the multidimensional correlations between lab test variables, patients, and time sequences.
  - Adapting the missingness-informed multi-branching TCN framework to further handle the imbalanced data and model both the intra- and inter-sequence correlations in the longitudinal EHR data for DR prediction.

- **Topic 4:** *MUSE-Net: Missingness-aware mUlti-branching Self-attention Encoder for Irregular Longitudinal Electronic Health Records*
  - Introducing a novel framework of **M**issingness-aware m**U**lti-branching **S**elf-Attention **E**ncoder (MUSE-Net) for data-driven disease prediction.
  - Proposing to integrate a time-aware self-attention encoder with a multi-branching classifier to address the imbalanced data issue and further classify the irregular longitudinal EHRs for disease prediction.
  - To improve the computational efficiency, the training of both MGPs and MUSE-Net is accelerated by using GPU, ensuring efficient processing of large and high-dimensional EHR data.

1.2.5 Organization of the dissertation

This dissertation is structured around several manuscripts, with each of Chapter 2 through Chapter 5 composed as a research paper, either published or currently under review. The rest of the dissertation is arranged in the following manner:

In Chapter 2, we propose a novel predictive framework with Multi-Branching Temporal Convolutional Network (MB-TCN) to model the complexly structured medical data for robust prediction of sepsis. The MB-TCN framework not only efficiently handles the missing value and imbalanced data issues but also effectively captures the temporal pattern and heterogeneous variable interactions. We evaluate the performance of the proposed MB-TCN in predicting sepsis using real-world medical data from PhysioNet/Computing in Cardiology Challenge 2019. Experimental results show that MB-TCN outperforms existing methods that are commonly used in current practice.

In Chapter 3, we propose a two-level hierarchical deep learning framework with Generative Adversarial Network (GAN) for automatic diagnosis of ECG signals. The first-level model is composed of a Memory-Augmented Deep auto-Encoder with GAN (MadeGAN), which aims to differentiate abnormal signals from normal ECGs for anomaly detection. The second-level learning aims at robust multi-class classification for different arrhythmias identification, which is achieved by integrating the transfer learning technique to transfer knowledge from the first-level learning with the multi-branching architecture to handle the data-lacking and imbalanced data issue. We evaluate the performance of the proposed framework using real-world medical data from the MIT-BIH arrhythmia database. Experimental results show that our proposed model outperforms existing methods that are commonly used in current practice.

In Chapter 4, we propose a Multi-branching Temporal Convolutional Network with Tensor Data Completion (MB-TCN-TC) model to analyze the longitudinal EHRs collected from diabetic patients for Diabetic retinopathy (DR) prediction.

Experimental results demonstrate that the proposed MB-TCN-TC model not only effectively copes with the imbalanced data and missing value issues commonly seen in EHR datasets but also captures the temporal correlation and complicated interactions among medical variables in the longitudinal clinical records, yielding superior prediction performance compared to existing methods.

In Chapter 5, we propose a novel **M**issingness-aware m**U**lti-branching **S**elf-Attention **E**ncoder (MUSE-Net) to cope with the challenges in modeling longitudinal EHRs for data-driven disease prediction. The MUSE-Net leverages a multi-task Gaussian process (MGP) with GPU acceleration for data imputation, a multi-branching architecture to address the data imbalance problem, and a time-aware self-attention encoder to account for the irregularly spaced time interval in longitudinal EHRs. We evaluate the proposed MUSE-Net using both synthetic and real-world datasets. Experimental results show that our MUSE-Net outperforms existing methods that are widely used to investigate longitudinal signals.

Chapter 6 includes my research contributions, general conclusions, and future research directions.

Chapter 2

Multi-branching temporal convolutional network for sepsis prediction

Sepsis is among the leading causes of morbidity and mortality in modern intensive care units. Accurate sepsis prediction is of critical importance to save lives and reduce medical costs. The rapid advancements in sensing and information technology facilitate the effective monitoring of patients' health conditions, generating a wealth of medical data, and provide an unprecedented opportunity for data-driven diagnosis of sepsis. However, real-world medical data are often complexly structured with a high level of uncertainty (e.g., missing values, imbalanced data). Realizing the full data potential depends on developing effective analytical models. In this paper, we propose a novel predictive framework with Multi-Branching Temporal Convolutional Network (MB-TCN) to model the complexly structured medical data for robust prediction of sepsis. The MB-TCN framework not only efficiently handles the missing value and imbalanced data issues but also effectively captures the temporal pattern and heterogeneous variable interactions. We evaluate the performance of the proposed MB-TCN in predicting sepsis using real-world medical data from

PhysioNet/Computing in Cardiology Challenge 2019. Experimental results show that MB-TCN outperforms existing methods that are commonly used in current practice. *Index Terms*: Multi-Branching Temporal Convolutional Networks, Imbalanced Data, Missing Value, Gaussian Process

2.1 Introduction

Sepsis is among the leading causes of morbidity and mortality in modern intensive care units (ICU). More than 1.7 million people in the U.S. develop sepsis annually, among which about 270,000 people died, contributing to over one-third of all in-hospital deaths. Infection has been known to be the root cause of sepsis onset. For example, lower respiratory tract infections cause septic shock in 35-50% of ICU patients (Sprung et al., 2008). Sepsis is a life-threatening condition leading to tissue damage, organ failure, or death. It has been estimated that mortality risk from sepsis will increase by around 7.6% with every hour of delay in effective medical treatment after the onset of hypotension due to the irreversible organ damage (Kumar et al., 2006). Although professional intensive care associations have proposed clinical standards and guidelines for sepsis diagnosis and treatment, the reliable and timely identification of sepsis is still unsatisfactory due to its unclear source of infection and nature of complex symptoms (Reyna et al., 2020).

Rapid advancements in sensing and information technology facilitate the effective monitoring of patients' health conditions, which brings about a data-rich environment (Yao and Yang, 2016; Yao et al., 2017). The comprehensive and systematically collected electronic-health-record (EHR) data provide an unprecedented opportunity for data-driven diagnosis of sepsis. However, EHR data are often complexly structured with a high level of uncertainty. For example, the medical data is commonly associated with a multivariate time series, the length of which may be different for different patients. Fully utilizing medical data for reliable sepsis prediction calls for

new data-driven methods that are capable to cope with the complex data structure, which poses several challenges including

(1) **Missing data**: EHR data are commonly subject to the issue of considerable missing values, which occurs when no record is stored for some variables in an observation. For example, there are up to 80% missing values in the clinical data from the PhysioNet/Computing in Cardiology Challenge 2019 (Reyna et al., 2020). Missing data generally happens due to lack of collection or documentation and variability of adherence to practice guidelines during routine healthcare management. For example, vital signs such as respiratory rate, heart rate, and body temperature, are generally recorded every three to four hours, while laboratory variables, such as the fraction of inspired oxygen and aspartate transaminase, are recorded on a daily basis for ICU patients (Reyna et al., 2020). This recording procedure results in a large number of data missingness. The substantial missing information may potentially lead to significantly biased estimates or invalid conclusions if left unaddressed.

(2) **Imbalanced data**: Disease detection is generally formulated as a binary classification problem (i.e., with or without disease). It is common for clinical datasets to exhibit imbalanced distribution between different classes because the number of patients with diseases is much smaller compared with the whole population. Data-driven predictive modeling based on imbalanced data tends to ignore the minority class (i.e., the diseased class), leading to unsatisfactory prediction performance. Traditional data augmentation techniques such as random over-sampling and synthetic sampling based on K-nearest neighbors (KNN) are not generally applicable due to the complex structure of the medical data (i.e., multivariate time series with variable lengths). Thus, new machine learning algorithms that are able to effectively model the multivariate longitudinal data and account for both imbalanced and missing data are urgently needed for reliable sepsis identification.

This paper proposes a Multi-Branching Temporal Convolutional Network (MB-TCN) to model the multivariate longitudinal medical data for robust sepsis prediction. We first define a missing-value mask as one of the TCN inputs, which enables

the network to learn and capture the missingness pattern in the medical records. Second, we propose a Multi-Branching method composed of a core TCN and multi-branching outputs to handle the imbalanced data, without making any assumptions and feature engineering. Each branching output is trained with a balanced subset of the medical records, while the core TCN is trained with the whole dataset. Third, we employ Gaussian Process Upper-Confidence Bound (GP-UCB) to search for the best network hyperparameters. We evaluate the MB-TCN model on the data from PhysioNet/Computing in Cardiology Challenge 2019 for sepsis prediction. Experimental results show that the MB-TCN significantly outperforms existing methods that are commonly used in current practice.

2.2 Research Background

Various traditional statistical and machine learning methods have been proposed for data-driven prediction of sepsis. For example, Nemati *et al* (Nemati et al., 2018) developed a modified Weibull-Cox proportional hazards model, which assumes a Cox proportional hazards with a Weibull-base hazard rate, to predict the sepsis onset. Delahanty *et al* (Delahanty et al., 2019) applied the Gradient Boosting method to model the Rhee clinical surveillance sepsis criteria (Rhee et al., 2017) for automatic sepsis identification. Other modern machine learning algorithms, such as Random Forest and XGBoost, have also been implemented for sepsis disease detection. Mani *et al* (Mani et al., 2014) and Horng *et al* (Horng et al., 2017) developed data-driven clinical decision-support systems for sepsis prediction by integrating different machine learning algorithms including Support Vector Machine, KNN, and Naive Bayes. A comprehensive review of statistical and machine learning methods in sepsis detection can be found in (Fleuren et al., 2020; Islam et al., 2019). Most existing statistical methods solve the prediction problem by first assuming a parametric function for the effect of predictors to model the underlying stochastic process, and then estimating unknown parameters using available data. For example, Cox

proportional hazards models assume a linear relationship between the predictors and proportional hazard (Emmert-Streib and Dehmer, 2019), which may not be generally valid in real-world practice. Furthermore, traditional methods depend heavily on manual feature engineering, which is a time-consuming trial-and-error process and may require domain expertise (Calvert et al., 2016).

Deep neural network (DNN) is another powerful tool that has widespread applications in the area of data-driven disease prediction (LeCun et al., 2015; Betancur et al., 2018). Unlike conventional statistical and machine learning methods, DNN does not require sophisticated feature engineering or strict model assumptions. A DNN generally consists of L parametric functions, also known as layers where each layer is considered as a transformed representation of the input domain. Specifically, a layer l_i takes the output of the previous layer l_{i-1} as its input and applies a non-linear activation function, such as rectified linear unit (ReLU) and sigmoid, to compute the output of current layer. Traditional feedforward DNNs have been implemented to predict the sepsis onset and have outperformed conventional statistical methods such as logistic regression or Cox models (Jaimes et al., 2005; López-Martínez et al., 2019). However, traditional DNN methods lack the ability to fully account for the temporal pattern in the disease trajectory, leading to unsatisfactory prediction performance.

The family of Recurrent Neural Networks (RNNs) including Long Short-Term Memory networks (LSTMs) (Hochreiter and Schmidhuber, 1997) and Gated Recurrent Units (GRUs) (Chung et al., 2014; Cho et al., 2014) has been proved to be powerful methods for classification tasks involving sequential inputs such as language and longitudinal clinical data (Lipton et al., 2015; Choi et al., 2017). Different RNN architectures have been developed to study longitudinal medical records for sepsis prediction. For example, Kam and Kim (Kam and Kim, 2017) implemented the LSTM to learn effective representations from medical data and predict the onset of sepsis. Futoma *et al* (Futoma et al., 2017) combined Multitask Gaussian Process (GP) modeling with an LSTM classifier for sepsis prediction. Due to the strong prediction performance and the ability to capture variable-length observations and

long-term temporal dependency, the family of RNNs has been treated as default methods to model time series and sequence data. However, recent studies show that certain convolutional neural networks can outperform the performance of RNNs when studying longitudinal or sequence data (Yin et al., 2017; Bai et al., 1803). Specifically, Bai *et al* (Bai et al., 1803) compared Temporal Convolutional Networks (TCNs) and canonical recurrent architectures (LSTMs and GRUs) in a comprehensive set of sequence modeling tasks and have proved TCNs outperformed RNNs in all tasks.

TCNs have been recently widely used to investigate multivariate time series data for disease diagnosis (Jarrett et al., 2019; Catling and Wolff, 2020). Moor *et al* (Moor et al., 2019) and Kok *et al* (Kok et al., 2020) developed GP-based TCNs to predict sepsis onset. They implemented the Multi-task GP to estimate the missing values and quantify the estimation uncertainty. However, different encounters may follow different GP distributions, and the GP hyperparameters need to be trained and estimated for each patient individually. Additionally, GP modeling involves mathematical operations of the large kernel matrix with a dimensionality of $(N_v * N_t) \times (N_v * N_t)$ (where N_v and N_t denotes the cardinality of medical variables and temporal points, respectively). As such, the imputation procedure with multi-task GP is highly computationally intensive. Additionally, traditional imputation methods including carry-forward and mean imputation are not capable to effectively exploit the data structure and may introduce large imputation error into the data. Thus, novel imputation procedures are needed for robust and accurate prediction of sepsis.

Data-driven sepsis prediction also suffers from the problem with extremely imbalanced data. A variety of sampling methods have been proposed to create a balanced dataset out from the original imbalanced data for robust classification and prediction (He and Garcia, 2009). Commonly used methods include random over-sampling, random under-sampling, informed under-sampling (e.g., KNN-based under-sampling) (Mani and Zhang, 2003), and synthetic minority over-sampling (SMOTE) (Chawla et al., 2002). Random under- and over-sampling are currently

the most common methods for imbalanced problems. However, under-sampling methods remove a large number of majority cases, resulting in a huge information loss; over-sampling may cause over-fitting due to replicated data from the minority class. Although the KNN-based under-sampling and SMOTE achieve better performances in non-time series imbalanced learning (Chawla et al., 2002; Mani and Zhang, 2003), they are not generally applicable for time series analysis. Specifically, KNN-based under-sampling employs a KNN classifier to under-sample the majority class according to the Euclidean distance between data points. Similarly, SMOTE creates artificial data generally based on Euclidean distance. However, Euclidean distance cannot effectively evaluate the dissimilarity between two time series with different lengths. Dynamic Time Warping (DTW) has been proved to be effective in quantifying the dissimilarity between time series data (Berndt and Clifford, 1994; Yao, 2021), but synthesizing minority samples with DTW is still highly challenging due to the complex data structure and the non-differentiability of DTW distance (Cuturi and Blondel, 2017). Thus, robust data-driven prediction of sepsis calls for novel analytical models that are capable to not only efficiently handle the missing value and imbalanced problems but also learn effective representations from the complexly structured medical data.

2.3 Research Methodology

2.3.1 Multivariate Time Series Classification

Before defining our network architecture, we first introduce the nature of multivariate time series classification problem. Suppose that we have N_p patients indexed by $i \in \{1, \ldots, N_p\}$. Each case is associated with historical medical data, which is described by the tuple $(\boldsymbol{x}^p, y_p)$, where $\boldsymbol{x}^p$ is a multi-variate time series denoting the observations of medical variables over time for case p and y_p is the binary label with $y_p = 1$ indicating case p is diagnosed with sepsis and otherwise $y_p = 0$. As such, our complete

dataset is given by $\mathcal{D} = \{(\boldsymbol{x}^p, y_p)\}_{p=1}^{N_p}$. The objective of time series classification in our study is to predict the probability of sepsis for a new patient given a historical sequence of the clinical observations.

2.3.2 Temporal Convolutional Network (TCN)

TCN is a powerful neural network for time series modeling, demonstrating superior performance over traditional RNNs in various fields (Bai et al., 1803). Instead of extracting spatial correlations as in the traditional convolutional networks, TCNs apply filters in the temporal dimension to capture dynamic patterns in the disease progression trajectory. The key feature of TCN that makes it more effective in modeling longitudinal data compared with the traditional DNNs and RNNs is the dilated causal convolution, which is achieved by applying a dilated temporal convolution with a filter $f : \{0, 1, \ldots, k-1\}$ over the input longitudinal data as

$$C(t) = (\boldsymbol{x}_v^p *_d f)(t) = \sum_{a=0}^{k-1} f(a) \cdot \boldsymbol{x}_v^p(t - a \cdot d) \tag{2.1}$$

where $C(\cdot)$ denotes the convolutional output, $\boldsymbol{x}_v^p$ represents the observations of medical variable v over time for patient p, $*$ denotes the convolution operation, d is the dilation factor, and k is the kernel size. More importantly, $(t - a \cdot d)$ indicates the direction toward the past, which realizes the causal convolution and ensures that network output at time t will not depend on any future information, i.e., medical observations at $t+1, t+2, \ldots, N_t$. Additionally, by varying d, the TCN is able to achieve multi-resolution convolution to capture both the long-range and local temporal correlations, thereby facilitating the effective modeling of high-resolution longitudinal data.

Another key feature of TCN is associated with the residual connections (He et al., 2016). Increasing number of layers in neural networks will probably introduce large training errors due to gradient vanishing or explosion, which is popularly known as

the degradation problem. The degradation problem hinders the performance of neural network modeling when the network goes deeper. The residual connection is generally added into TCNs to cope with the degradation problem and increase the depth of neural networks for better prediction performance. As shown in Fig. C.1(a), the temporal residual block consists of two branches. One branch is formed by a series of network transformations, $\mathcal{F}$, including the casual convolution, batch normalization, ReLU, and a dropout layer. The other branch is a shortcut from the input to the output, i.e., a direct projection of the input $\boldsymbol{x}_v$. As such, the output of the residual block is

$$o(\boldsymbol{x}_v^p) = \text{activation}(\mathcal{F}(\boldsymbol{x}_v^p) + \boldsymbol{x}_v^p) \tag{2.2}$$

Such residual connections allow the network to fit the residual mapping, i.e., focusing on the modifications to the identity mapping instead of the entire transformation, thereby mitigating the gradient vanishing or explosion problem in training DNNs. Note that an additional 1×1 convolutional layer is added to the shortcut branch to deal with the problem when the length of $\boldsymbol{x}_v^p$ is different from $\mathcal{F}(\boldsymbol{x}_v^p)$.

In our TCN model, there are two layers of causal convolution and ReLU within a residual block. We also employ the batch-normalization to each causal convolutional layer (i.e., normalizing the input layer by re-centering and re-scaling) to make the network training faster and more stable. Additionally, we add a dropout layer for each causal convolution to prevent over-fitting. The TCN is capable of capturing both the heterogeneous patterns among different medical variables and the temporal correlations in the longitudinal data when the information is complete (i.e., without missing values). However, missing data is a common challenge in health investigations, which will deteriorate the prediction performance of TCN and pose significant challenges for effective information retrieval. Here, we treat the missingness in clinical datasets as informative missingness and let the network learn the correlation

between patterns of missing and non-missing values. Typically, we create missing-value masks as $\boldsymbol{Z} = \{\boldsymbol{z}^p\}_{p=1}^{N_p}$ with $z_{v,t}^p = 1$ if $x_{v,t}^p$ is missing; otherwise $z_{v,t}^p = 0$.

Fig. C.1(b) shows our TCN architecture. Specifically, the network has two inputs: one is the multivariate time series, $\boldsymbol{x}^p$; the other is the missing-value mask, $\boldsymbol{z}^p$. The input data $\boldsymbol{x}^p$ is processed by the TCN and the mask $\boldsymbol{z}^p$ is processed by multiple causal convolutional layers with ReLU activation in a parallel stream. The outputs of both streams are concatenated, which is followed by a dropout layer to avoid overfitting. The final fully connected layer then combines the missingness structure, temporal correlations, and heterogeneous medical patterns in the clinical data captured by the previous temporal-convolutional layers. At the end of the network, we propose a multi-branching output structure to address the imbalanced data problem, which will be detailed in the next subsection. Note that our problem is the binary classification of sepsis, so each branching output is followed by a sigmoid activation function to generate the predicted probability of sepsis.

2.3.3 Multi-Branching Outputs

In this subsection, we develop a multi-branching TCN (MB-TCN) to address the problem with imbalanced data and improve the classification performance, as shown in Fig. C.2. Specifically, the MB-TCN consists of a core TCN and N_b branching outputs. We create N_b balanced datasets from the original data $\mathcal{D} = \{\mathcal{D}_-, \mathcal{D}_+\}$ by under-sampling the majority class $\mathcal{D}_-$ (without disease) as $\mathcal{D}_i = \{\mathcal{D}_-^i, \mathcal{D}_+\}$, where $\mathcal{D}_- = \cup_{i \in 1,\ldots,N_b} \mathcal{D}_-^i$, and further create multiple output layers corresponding to each of the balanced datasets. The core TCN architecture will be trained by all the N_b datasets, while each branching output will be trained by the corresponding balanced sub-dataset. The objective of neural network training is to search a set of optimal parameters such that the network outputs high probabilities for positive cases (i.e., sepsis patients) and low probabilities for negative cases (i.e., non-sepsis cases). As such, we select the binary cross-entropy as the loss function of MB-TCN, which is

given as:

$$\mathcal{L}(\boldsymbol{\theta}; \boldsymbol{X}, \boldsymbol{Z}) = -\sum_{p=1}^{N_p}\sum_{i=1}^{N_b}\mathcal{I}(p \in \mathcal{D}_i)\Big(y_p \log(\hat{P}_i(\boldsymbol{\theta}; \boldsymbol{x}^p, \boldsymbol{z}^p)) + (1-y_p)\log(1-\hat{P}_i(\boldsymbol{\theta}; \boldsymbol{x}^p, \boldsymbol{z}^p))\Big) \tag{2.3}$$

where $\boldsymbol{\theta}$ denotes the network parameter to be trained, $\mathcal{I}(\cdot)$ is an indicator function, and $\hat{P}_i(\boldsymbol{\theta}; \boldsymbol{x}^p, \boldsymbol{z}^p)$ is the predicted probability of sepsis from branching output i given the input $\boldsymbol{x}^p$ and $\boldsymbol{z}^p$. If the network predicts a probability close to 0 for a positive case (i.e., $y_p = 1$ and $\hat{P}_i(\boldsymbol{\theta}; \boldsymbol{x}^p, \boldsymbol{z}^p) \approx 0$), the term $y_p \log(\hat{P}_i(\boldsymbol{\theta}; \boldsymbol{x}^p, \boldsymbol{z}^p))$ will incur a large loss in $\mathcal{L}(\boldsymbol{\theta}; \boldsymbol{X}, \boldsymbol{Z})$. Similarly, the term $(1-y_p)\log(1-\hat{P}_i(\boldsymbol{\theta}; \boldsymbol{x}^p, \boldsymbol{z}^p))$ will become very large if the network predicts a probability close to 1 for a negative case (i.e., $y_p = 0$ and $\hat{P}_i(\boldsymbol{\theta}; \boldsymbol{x}^p, \boldsymbol{z}^p) \approx 1$), leading to a large $\mathcal{L}(\boldsymbol{\theta}; \boldsymbol{X}, \boldsymbol{Z})$. Thus, minimizing Eq. (2.3) will generate the optimal network parameters for accurate sepsis identification.

Algorithm 1 summarizes the training procedure for the proposed MB-TCN framework. Typically, in each iteration, we use all minority class (with disease) data $\mathcal{D}_+$ and sample a sub-dataset $\mathcal{D}_-^i$ with the same size as $\mathcal{D}_+$ from the majority class $\mathcal{D}_-$. Then, we select one branching output at random to train the network, update network parameters $\boldsymbol{\theta}$, and further make predictions based on the selected branching output and dataset. During the inference step, the multi-branching TCN generates N_b predicted probabilities of sepsis for one patient, which will further be utilized for robust sepsis prediction. Specifically, we will evaluate the following three different approaches to compute the final probability of sepsis:

(1) **Simple average (MB-TCN-AVE)**: We take the average of the N_b outputs as the final sepsis probability for patient p:

$$\hat{P}^p = \bar{\hat{P}}^p = \frac{1}{N_b}\sum_{i=1}^{N_b}\hat{P}_i^p \tag{2.4}$$

where $\hat{P}_i^p$ is the prediction probability of branching output i for patient p.

(2) **Confidence margin (MB-TCN-CM)**: The confidence margin of branching output i for patient p is defined as

$$cm_i^p = \begin{cases} \hat{P}_i^p - 0.5 & if \quad \hat{P}_i^p > 0.5 \\ 0.5 - \hat{P}_i^p & if \quad \hat{P}_i^p \leq 0.5 \end{cases} \tag{2.5}$$

We further calculate the weight of output i as $w_i^p = \frac{cm_i^p}{\sum_{j=1}^{N_b} cm_j^p}$. Then, the final prediction probability for case p is given by

$$\hat{P}^p = \sum_{j=1}^{N_b} w_j^p \hat{P}_j^p \tag{2.6}$$

(3) **Voting (MB-TCN-VO)**: In this method, each branching output makes prediction independently. If the number of predicted positive label is greater than or equal to the number of predicted negative label, then the final predicted label for patient p is sepsis, i.e., $\hat{y}^p = 1$, and $\hat{y}^p = 0$ otherwise. Mathematically, the predicted label is given by

$$\hat{y}^p = \mathcal{I}(\sum_{j=1}^{N_b} \mathcal{I}(\hat{y}_j^p = 1) \geq \sum_{j=1}^{N_b} \mathcal{I}(\hat{y}_j^p = 0)) \tag{2.7}$$

where $\hat{y}_j^p$ denotes the predicted label by branching output j.

The proposed multi-branching output is a simple architecture and is expected to enable the TCN to learn more efficiently and effectively from imbalanced dataset and simultaneously prevent the common problem of over-fitting and information loss. Addtionally, the proposed MB-TCN framework is capable to quantify the prediction uncertainty. Specifically, we define the prediction uncertainty for patient p as the empirical standard deviation (ESTD) of the predicted probabilities by the N_b branching outputs:

$$\hat{\sigma}^p = \sqrt{\frac{\sum_{i=1}^{N_b} (\hat{P}_i^p - \bar{\hat{P}}^p)^2}{N_b - 1}} \tag{2.8}$$

A small $\hat{\sigma}^p$ indicates that the branching outputs provide homogeneous predictive results, and the MB-TCN is confident with the prediction for patient p. On the other hand, if $\hat{\sigma}^p$ is large, there is a high level of variation in the branching outputs and the MB-TCN is less certain about the prediction. In such cases, the clinical practitioner needs to carefully monitor the patient's health condition and collect more medical data to increase the prediction confidence.

Algorithm 1: Training Procedure of the proposed MB-TCN

Input: $\{\boldsymbol{x}^p\}_{p=1}^{N_p}$, $\{\boldsymbol{z}^p\}_{p=1}^{N_p}$, N_b;
Output: Trained network parameters $\boldsymbol{\theta}^*$;
for $i = 1$ *to maximum epochs* I **do**
    for $j = 1$ *to the number of branching outputs* N_b **do**
        Create balanced dataset $\mathcal{D}_j$ by under-sampling;
    end
    for $k = 1$ *to maximum iterations* K **do**
        Sample minibatch M_k from $\mathcal{D}_j$ (which is randomly selected from $\{\mathcal{D}_1, \ldots, \mathcal{D}_{N_b}\}$);
        Compute binary cross-entropy loss based on M_k and branching output j;
        Update $\boldsymbol{\theta}$ excluding branching outputs not selected;
    end
    if *converge* **then**
        break
    end
end
Return $\boldsymbol{\theta}$

2.3.4 Gaussian Process-based Active Learning for Optimal Network Hyperparameters

The performance of the MB-TCN also depends heavily on the network hyperparameters $\boldsymbol{\omega}$ including the number of residual blocks n_r, number of filters n_f, kernel size

k, dilation factor d, dropout rate γ, and learning rate α, i.e., $\boldsymbol{\omega} = [n_r, n_f, k, d, \gamma, \alpha]^T$. Searching for the optimal hyperparameters, denoted as $\boldsymbol{\omega}^*$, requires repeatedly training the network with different parameter values, which can be time-consuming for large-scale deep learning. Here, we propose to actively tune $\boldsymbol{\omega}$ based on Gaussian Process Upper-Confidence-Bound (GP-UCB) (Srinivas et al., 2009). The GP-UCB consists of two components: GP regression as a surrogate model to approximate the optimization function and an effective acquisition function to determine the next query point in the search space.

A GP provides a distribution over functions, i.e., $g(\cdot) \sim \mathcal{GP}(m(\cdot), \mathcal{K}(\cdot))$, which is fully specified by the mean function $m(\cdot)$ and covariance function $\mathcal{K}(\cdot)$. The input of the GP in our setting is the network hyperparameters, i.e., $\boldsymbol{\omega}$'s, and the output is a metric to characterize the prediction performance of the MB-TCN, which is defined as

$$\delta(\boldsymbol{\omega}) = \text{AUROC}(\boldsymbol{\omega}; \mathcal{D}_{valid}) + \text{AUPRC}(\boldsymbol{\omega}; \mathcal{D}_{valid}) \tag{2.9}$$

where $\text{AUROC}(\boldsymbol{\omega}; \mathcal{D}_{valid})$ and $\text{AUPRC}(\boldsymbol{\omega}; \mathcal{D}_{valid})$ denote the Area Under Receiver-Operating-Characteristic Curve (AUROC) and Area Under Precision-Recall Curve (AUPRC) of the validation set $\mathcal{D}_{valid}$ given the hyperparameters $\boldsymbol{\omega}$. The ROC curve evaluates the tradeoff between the False Positive Rate (FPR) and True Positive Rate (TPR), and AUROC is a performance score across all thresholds, quantifying the general prediction performance. On the other hand, the AUPRC characterizes the relationship between Recall and Precision, which has been widely recognized to be more effective than AUROC when evaluating the classification performance with an imbalanced dataset (Flach and Kull, 2015; Soleymani et al., 2020). Combination of the two scores, i.e., $\delta(\boldsymbol{\omega})$, is expected to effectively quantify the performance of data-driven predictive modeling.

The GP regression in our study estimates the unobserved latent function $g(\boldsymbol{\omega})$ to generate $\delta(\boldsymbol{\omega})$ from the input $\boldsymbol{\omega}$: $\delta(\boldsymbol{\omega}) = g(\boldsymbol{\omega}) + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2)$ is a noise

factor added for numerical stability. In this study, we use the zero mean function and select the square exponential kernel as the covariance function, i.e., $\mathcal{K}(\boldsymbol{\omega}_1, \boldsymbol{\omega}_2) = a^2 \exp(-\frac{\|\boldsymbol{\omega}_1 - \boldsymbol{\omega}_2\|^2}{\lambda^2})$. Given the collected data $\{\boldsymbol{\Omega}_n, \boldsymbol{\delta}_n\}$, where $\boldsymbol{\Omega}_n = [\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_n]$ and $\boldsymbol{\delta}_n = [\delta(\boldsymbol{\omega}_1), \ldots, \delta(\boldsymbol{\omega}_n)]^T$, the GP parameters (i.e., a and λ) can be estimated with the maximum likelihood method. Additionally, the predictive distribution of $\delta(\boldsymbol{\omega})$ with an arbitrary unevaluated setting of $\boldsymbol{\omega}$ is given by $\delta(\boldsymbol{\omega})|\boldsymbol{\Omega}_n, \boldsymbol{\delta}_n, \boldsymbol{\omega} \sim \mathcal{N}(m_n(\boldsymbol{\omega}), \sigma_n^2(\boldsymbol{\omega}))$, with the predictive mean and variance as

$$
\begin{aligned}
m_n(\boldsymbol{\omega}) &= \mathcal{K}(\boldsymbol{\omega}, \boldsymbol{\Omega}_n)(\mathcal{K}(\boldsymbol{\Omega}_n, \boldsymbol{\Omega}_n) + \sigma_\epsilon^2 \boldsymbol{I})^{-1} \boldsymbol{\delta}_n && (2.10) \\
\sigma_n^2(\boldsymbol{\omega}) &= \mathcal{K}(\boldsymbol{\omega}, \boldsymbol{\omega}) - \mathcal{K}(\boldsymbol{\omega}, \boldsymbol{\Omega}_n)(\mathcal{K}(\boldsymbol{\Omega}_n, \boldsymbol{\Omega}_n \\
&\quad + \sigma_\epsilon^2 \boldsymbol{I})^{-1} \mathcal{K}(\boldsymbol{\Omega}_n, \boldsymbol{\omega}) && (2.11)
\end{aligned}
$$

Given the GP predictive model, we define the acquisition function for the next sampling point according to GP-UCB:

$$
\boldsymbol{\omega}_{n+1} = \text{argmax}_{\boldsymbol{\omega}} \{ m_n(\boldsymbol{\omega}) + \beta_n^{1/2} \sigma_n(\boldsymbol{\omega}) \} \tag{2.12}
$$

where β_n is a constant to balance the trade-off between the exploitation of current knowledge (i.e., $m_n(\boldsymbol{\omega})$) and the exploration of uncertain space (i.e., $\sigma_n(\boldsymbol{\omega})$). In other words, Eq. (2.12) enables us to search for the optimal network hyperparameters in a computation-efficient way. After the new point, $\boldsymbol{\omega}_{n+1}$, is selected, the MB-TCN will be trained to generate the corresponding performance metric $\delta(\boldsymbol{\omega}_{n+1})$. When the new input-output pair $\{\boldsymbol{\omega}_{n+1}, \delta(\boldsymbol{\omega}_{n+1})\}$ becomes available, the GP predictive model will be updated according to

$$
\begin{bmatrix} \boldsymbol{\delta}_n \\ \delta(\boldsymbol{\omega}_{n+1}) \end{bmatrix} \sim \mathcal{N} \left(0, \begin{bmatrix} \mathcal{K}(\boldsymbol{\Omega}_n, \boldsymbol{\Omega}_n) + \sigma_\epsilon^2 \mathbf{I} & \mathcal{K}(\boldsymbol{\Omega}_n, \boldsymbol{\omega}_{n+1}) \\ \mathcal{K}(\boldsymbol{\omega}_{n+1}, \boldsymbol{\Omega}_n) & \mathcal{K}(\boldsymbol{\omega}_{n+1}, \boldsymbol{\omega}_{n+1}) \end{bmatrix} \right) \tag{2.13}
$$

This active-learning procedure is repeatedly evaluated until $\boldsymbol{\omega}$ converges around a certain point where $\delta(\boldsymbol{\omega})$ is expected to have a maximum value, and the converged vector $\boldsymbol{\omega}^*$ will be selected as the optimal network hyperparameters.

2.4 Experimental Design and Results

2.4.1 Data Description and Experimental Design

The dataset we use is obtained from PhysioNet/Computing in Cardiology Challenge 2019 with medical records from two hospital systems: Beth Israel Deaconess Medical Center which contains 20,336 patients where sepsis patients account for 8.8% and Emory University Hospital, which contains 20,000 patients with 5.7% sepsis cases. The data were labeled according to the Sepsis-3 clinical criteria Reyna et al. (2020). For sepsis cases, the label is '0' up to 6 hours before sepsis onset and '1' otherwise. For non-sepsis cases, the label is always '0' over time. Both datasets consist of 40 clinical variables including around 2.5 million hourly time windows and 15 million data points. Specifically, there are 8 vital sign variables, 26 laboratory variables, and 6 demographic variables. Each clinical variable is recorded over time, generating a multivariate time series for each patient. Additionally, the medical records are with variable lengths and are subject to missing values as shown in Fig. C.3 and Fig. C.4. In our experiment, we use the padding method to process time series data so that we can encode variable-length data into a single 3-D tensor (samples$\times$ time $\times$ variables). The dimensionality of the "time" is the length of the longest time series. Specifically, samples that have shorter time series will be padded with -1 to achieve the same length as the longest time series. In addition, the missing-value masks will treat those padded values as missingness and their corresponding positions in the mask will be set to 1.

We select the data from Beth Israel Deaconess Medical Center for training and the data from Emory University Hospital for testing. We train the

MB-TCN with 80% of the training data and tune network hyperparameters by GP-UCB with the rest 20% training data. We further evaluate the performance of MB-TCN using the test dataset. Fig. C.5 illustrates our experimental design. The MB-TCN will be benchmarked with a pure TCN, TCN+over-sampling, TCN+under-sampling, and TCN+under-sampling+DTW-HC (see Appendix A), pure LSTM, Multi-Branching LSTM (MB-LSTM), LSTM+over-sampling, LSTM+under-sampling, and LSTM+under-sampling+DTW-HC. Note that the hyperparameters in the pure TCN, TCN+over-sampling, TCN+under-sampling, and TCN+under-sampling+DTW-HC are set the same as in the MB-TCN; those in MB-LSTM are actively selected with GP-UCB, and architectures of other LSTM models are designed with the same hyperparameters as the MB-LSTM. The proposed method will also be benchmarked with different imputation methods (i.e., MB-TCN+mean and MB-TCN+carry-forward). The model performance will be evaluated according to two metrics: AUROC and AUPRC.

2.4.2 Experimental Results

Table D.1 shows the performance comparison between MB-TCN-based methods and other benchmarks. Note that MB-TCN methods achieve better performance with AUROC and AUPRC scores of 0.892 and 0.527 for MB-TCN-AVE, 0.891 and 0.525 for MB-TCN-CM, and 0.885 and 0.499 for MB-TCN-VO compared with traditional TCN-based methods or LSTM-based methods. Specifically, MB-TCN-AVE improves on AUROC by 14% compared with TCN+under-sampling (0.782); improves by 7% compared with LSTM+under-sampling (0.834), 7% compared with pure LSTM (0.831), and 2.3% compared with LSTM+over-sampling (0.872). Moreover, the improvement achieved by MB-TCN-AVE is more significant in terms of the AUPRC: improving by 117% compared with TCN+under-sampling (0.242), 32% compared with LSTM+under-sampling (0.400), and 27% and 22% compared with LSTM+over-sampling (0.414) and pure LSTM (0.431). MB-TCN-AVE also improves on AUROC

by 7.3% compared with TCN+under-sampling+DTW-HC (0.831), by 8.3% compared with LSTM+under-sampling+DTW-HC (0.824). Moreover, MB-TCN-AVE achieves improvement on AUPRC by 17.9% compared with TCN+under-sampling+DTW-HC (0.447), and 20.9% compared with LSTM+under-sampling+DTW-HC (0.436). Note that the AUROC provided by MB-TCN-AVE achieves a marginal improvement compared with TCN+Over-sampling (i.e., 0.892 over 0.884). However, the improvement on AUPRC is significant with a 28% enhancement (i.e., 0.527 over 0.413). This improvement on AUPRC is less significant compared with the pure TCN (i.e., 0.527 over 0.449), but still achieves an 18% enhancement. Additionally, MB-TCN-AVE achieves a 10% improvement on AUROC compared with the pure TCN (i.e., 0.892 over 0.812). As such, MB-TCN framework performs significantly better compared with all the benchmarking methods.

Furthermore, there is no significant difference in AUC scores for different MB-TCN methods. Specifically, MB-TCN-AVE, MB-TCN-CM, and MB-TCN-VO achieve 0.892, 0.891, and 0.885 AUROC scores and 0.527, 0.525, and 0.499 AUPRC scores respectively. Note that the voting-based methods in both MB-TCN and MB-LSTM (i.e., MB-TCN-VO and MB-LSTM-VO) perform slightly worse compared with AVE- and CM-based methods. This is due to the fact that voting-based methods ignore the estimation uncertainty in the predicted probability for each branching output. For example, assume the estimated probability from each of the 10 branching outputs for patient p is 0.55, 0.51, 0.52, 0.53, 0.55, 0.54, 0.1, 0.05, 0.15, and 0.02, respectively. The final output of the estimated label by voting-based methods should be $\hat{y}^p = 1$ if the threshold is 0.5, because more than half of the branches output a label of 1. However, the estimation given by the first 6 outputs is quite uncertain with estimated probabilities very close to 0.5, while the rest four branches are with higher confidence about their estimations with a probability close to 0. As such, after taking the estimation uncertainty into account, the final estimated label should be $\hat{y}^p = 0$, which

matches the estimation given by AVE- and CM-based methods. Nevertheless, voting-based methods with multi-branching still outperform the pure, over- and under-sampling based models.

It is also worth noting that the MB-TCN based methods outperform the MB-LSTM based methods. Specifically, as shown in Table D.1, MB-TCN-AVE achieves 1.6% and 14.6% performance improvement on AUROC and AUPRC scores compared with MB-LSTM-AVE (from 0.878 and 0.460 to 0.892 and 0.527). MB-TCN-CM improves on the AUROC and AUPRC by 1.4% and 14.1%, respectively, compared with MB-LSTM-CM (from 0.879 and 0.460 to 0.891 and 0.525). MB-TCN-VO increases the AUROC and AUPRC from 0.878 and 0.458 to 0.885 and 0.499, respectively, compared with MB-LSTM-VO, achieving an improvement of 0.8% and 9.0% in AUROC and AUPRC scores. This is due to the effectiveness of TCN to handle the long-range multivariate longitudinal data.

Fig. C.6 (a) and (b) display the ROC and PR curves of all models. A good model is toward the top-left corner in ROC space (i.e. the lower FPR and higher TPR). According to Fig. C.6(a), the curve of MB-TCN-AVE method dominates in the ROC space, yielding the best AUROC score of 0.892. Correspondingly, a good classification model should achieve both higher recall and precision in the PR curve. According to Fig. C.6(b), the MB-TCN-AVE model dominates in PR space, which yields the best AUPRC score of 0.527.

Table D.2 further shows AUROC and AUPRC scores of MB-TCN-AVE with missing-value mask and other imputation methods that are commonly used in current practice. Specifically, the performance of the proposed MB-TCN-AVE is benchmarked with MB-TCN+mean imputation and MB-TCN+carry-forward imputation. Note that the MB-TCN-AVE method yields the best AUROC and AUPRC scores of 0.892 and 0.527 respectively compared with MB-TCN+mean (with AUROC and AUPRC scores of 0.821 and 0.389) and MB-TCN+carry-forward (with AUROC and AUPRC scores of 0.821 and 0.397). The performance improvement is due to the fact that the proposed missing-value mask effectively captures the

missingness structure in the data with no additional statistical assumptions or imputation uncertainty.

Fig. C.7 (a) and (b) show the ROC and PR curves of 10 branching outputs in MB-TCN methods. All curves in the ROC and PR space have similar trends, but the variation in PR curves is bigger compared with the ROC curves. Specifically, the 3 standard deviations for AUROC and AUPRC is 0.013 and 0.041, respectively. This is due to the effect of imbalanced dataset (only 5.7% positive cases in the testing dataset) and PR curve is more sensitive than ROC curve when the dataset is extremely imbalanced (Saito and Rehmsmeier, 2015). Note that the AUROC from each branching output with a minimal value of 0.878 outperforms the traditional methods, i.e., pure TCN (0.812), TCN+under-sampling (0.782), and TCN+under-sampling+DTW-HC (0.831), except for TCN+ over-sampling (0.884). However, the mean AUROC from the 10 outputs yields a higher score of 0.885 compared with TCN+over-sampling. Additionally, all AUPRC scores from the MB-TCN with a minimal value of 0.471 significantly outperform the pure TCN (0.449), TCN+over-sampling (0.413), TCN+under-sampling (0.242), and TCN+under-sampling+DTW-HC (0.447). Fig. C.8 shows the empirical standard deviation (ESTD) of predicted probabilities for the test dataset by the MB-TCN. Note that the distribution of ESTD is right-skewed, which indicates most of samples in the test dataset have a small ESTD (within 0.1). Specifically, 266 samples out of the 20,000 patients in the test dataset are predicted with an ESTD ≥ 0.3, which should be done further investigation by clinical practitioners for reliable prediction. Table D.3 shows the comparison of average missing-value rates of vital signs between samples with ESTD ≥ 0.3 and samples with ESTD < 0.3. Note that samples with ESTD ≥ 0.3 have higher rates for all vital signs (including Heart Rate (HR), Pulse Oximetry (O2Sat), Temperature (Temp), Systolic BP (SBP), Mean Arterial Pressure (MAP), and Respiration Rate (Resp)). As such, we can conclude that the high missing-value rate in vital signs will increase the prediction uncertainty. Clinicians need to collect more medical data by carefully monitoring the patient's health condition to increase the prediction confidence.

2.4.3 Comparison Study with Existing Literature

We first compare the MB-TCN model with 5 studies from the PhysioNet/Computing in Cardiology Challenge 2019 whose performance scores rank among the top 5. Similar to our investigation, these studies use the data from Beth Israel Deaconess Medical Center for training and the data from Emory University Hospital for testing. Table D.4 summarizes the comparison results (only one study reported AUPRC score). Note that all the evaluation metrics are calculated on the test dataset. Our proposed MB-TCN-AVE yields the best AUROC and AUPRC scores of 0.892 and 0.527, respectively.

In addition, we further compare our MB-TCN with the GP-based TCN (GP-TCN)(Kok et al., 2020). In the GP-TCN, the missing values are estimated with a Multi-task GP and the TCN is trained by minimizing the expected value of the loss function. Note that the GP-TCN paper merged datasets from both hospital systems, and then employed 10-fold stratified cross-validation to train and evaluate the model. In other words, 90% of the merged data is used for training and the remaining 10% is for validation at each fold. We add another experiment here to evaluate our MB-TCN with 10-fold stratified cross-validation. Fig. C.9 displays the ROC and PR curves obtained from MB-TCN-AVE. Our MB-TCN method yields mean AUROC and AUPRC scores of 0.954 and 0.770 respectively, which outperforms the GP-TCN with the mean AUROC and AUPRC scores of 0.93 and 0.70. In addition, our MB-TCN-AVE achieves a more robust prediction compared to the GP-TCN. Specifically, the standard deviation of the AUROC (AUPRC) score from MB-TCN-AVE is 0.008 (0.027) which is significantly smaller compared to the GP-TCN method whose standard deviation is 0.04 (0.1). Hence, our MB-TCN framework significantly outperforms the GP-TCN, achieving a more accurate and robust sepsis prediction.

2.5 Discussion

2.5.1 Interpretation on Performance Improvement of MB-TCN

The performance improvement of the proposed MB-TCN framework is mainly due to the following three modules:

(1) **Missing-value Mask**: We design a missing-value mask to characterize the missingness structure in the data without the need for additional statistical assumptions as most existing imputation approaches do. The missing-value mask is considered as one extra input of MB-TCN as shown in the network structure (Fig. C.1). As such, the network is trained to directly learn the missingness patterns in the data for accurate sepsis prediction, and no extra uncertainty or imputation error will be introduced into the predictive model as compared to most existing imputation methods.

(2) **Multi-branching Architecture**: The multi-branching architecture is another contributor to the performance enhancement. The strengths of the multi-branching architecture are due to: (a) *Balanced Training Datasets*: Balanced training datasets are created by under-sampling the majority class and combining them with the minority class. Each branching output is trained with a balanced dataset. As such, the imbalanced data issue is significantly relieved compared with the pure TCN or other machine learning models that are trained with a highly imbalanced dataset, leading to a significant performance improvement. (b) *Robust Prediction*: The multi-branching architecture is similar to the ensemble scheme, which pursues diversity to improve the generalization ability of MB-TCN by combining the decisions from multiple individual classifiers. Specifically, each branching output is trained by different balanced subset, which ensures that different branches learn different mapping functions for the same problem. The final prediction is further obtained by combining all branching outputs. As such the multi-branching scheme can lower

the variation of each individual branching output and increase the robustness of the MB-TCN prediction.

(3) **Core TCN**: The core TCN module is trained by the entire original training dataset. In other words, there is no information loss associated with the MB-TCN model as compared with the under-sampling methods. Additionally, it requires no artificially synthesized data which may introduce extra error or uncertainty into the predictive model if the synthesized data is of poor quality. As such, the core TCN module further boosts the prediction performance of MB-TCN.

2.5.2 Limitation of the MB-TCN Framework

One possible limitation is associated with the way to generate the balanced dataset for each branching output. We originally propose to generate the balanced dataset by randomly under-sampling the majority class to the size of the minority class and combining them with the minority data. However, this strategy may not fully maximize the potential in performance improvement of the proposed multi-branching framework. This is due to the fact that the sub-dataset of the majority class may contain redundant information if similar samples (i.e., samples with small discrepancy measures) are selected and included in the same subset. In order to further improve the prediction performance of MB-TCN, we propose a new sampling method of DTW-based Hierarchical Clustering (DTW-HC) to generate the balanced dataset, which is expected to increase the diversity of the majority-class samples in each sub-dataset. We denote this new method as MB-TCN+DTW-HC. The sampling procedure is illustrated in the Appendix A.

Table D.5 shows the performance comparison between original MB-TCN methods and MB-TCN+DTW-HC methods. Note that MB-TCN+DTW-HC methods achieve better performance with AUROC and AUPRC scores of 0.900 and 0.560 for AVE, 0.900 and 0.559 for CM, and 0.895 and 0.522 for VO compared with the original MB-TCN methods. The enhancement achieved by MB-TCN+DTW-HC methods is

more significant in terms of AUPRC: with an improvement of 6.3% compared with MB-TCN-AVE (0.527), 6.5% compared with MB-TCN-CM, and 4.6% compared with MB-TCN-CM. As such, increasing the data diversity in each subset further boosts the predictive performance of MB-TCN.

2.5.3 The Impact of Sample Length on Performance of MB-TCN

In the present investigation, we cut the initial points for the long-sequence samples so that each sample is within the length of 40- and 60-hours, respectively. We compare the performance of MB-TCN-AVE trained by shorter sequences with that trained by the original data in Table D.6. Note that here we train our MB-TCN framework with newly generated balanced data by DTW-HC. We omit DTW-HC here for notation convenience. Specifically, the AUROC scores provided by MB-TCN-AVE trained by sequences with 60-hour length decrease from 0.900 to 0.887 and the AUPRC scores decrease from 0.560 to 0.545 compared with the MB-TCN-AVE trained by full-length data. The AUROC and AUPRC scores trained by 40-hours length sequences decrease from 0.900 to 0.872 and from 0.560 to 0.487 respectively, compared with that trained by full-length data. Thus, training MB-TCN with shorter-sequence samples will have a negative impact on the final prediction performance. This is due to the fact that the core TCN applies dilated casual convolution over input sequences to capture both long-range and local temporal correlation. Important long-range information will be lost if we cut the original samples into shorter sequences and the resulted prediction performance will be diminished.

2.5.4 The Impact of Branch Number on Performance of MB-TCN

Fig. C.10 compares the AUROC and AUPRC of MB-TCN-AVE with 3, 5, 8, 10, and 15 branching outputs, respectively. Here we train the MB-TCN with newly generated balanced data by DTW-HC. We omit DTW-HC for notation convenience. Note that the MB-TCN-AVE with 10 branches has the best performance (i.e. AUROC=0.90, AUPRC=0.56). Too many (e.g., 15) or too few (e.g., 3) branches will introduce a negative impact on the model performance. This is due to the fact that the optimal number of branches depends on the number of balanced datasets that can be created from the training set. Specifically, there are 8.8% positive cases and 91.2% negative cases in the training set whose negative cases can be divided into around 10 subsets to form 10 balanced datasets combined with all the positive cases. Too few or too many branching outputs will result relatively imbalanced sub-dataset for each branch, leading to a deteriorated prediction performance. Nevertheless, the AUPRCs given by the MB-TCN-AVE with all the different number of branching outputs significantly outperform the existing deep learning-based methods. This is due to the fact that MB-TCN even designed with many (e.g., 15) or few (e.g., 3) branching outputs is trained with a relatively balanced training dataset. The ratio of the majority-class data over the minority is approximately 3.45:1 and 0.7:1 for the 3 and 15 branching outputs, respectively. The imbalanced data issue is significantly relieved compared with the pure TCN with an imbalanced ratio of 10.36:1, leading to a significant performance improvement.

2.6 Conclusions

In this paper, we develop a novel framework of Multi-Branching Temporal Convolutional Network (MB-TCN) to investigate multivariate longitudinal medical data for robust and reliable prediction of sepsis. We first define a missing-value mask as one of

the TCN inputs to capture the missingness pattern in the medical records. Second, we design a multi-branching architecture composed of a core TCN and multi-branching outputs to handle the imbalanced data. Additionally, we propose to actively search for MB-TCN hyperparameters based on the GP-UCB criterion. Finally, we implement the MB-TCN to predict sepsis using the medical data from PhysioNet/Computing in Cardiology Challenge 2019. Experimental results show that the MB-TCN method not only efficiently handles the missing value and imbalanced data problems but also effectively captures the temporal pattern and heterogeneous variable interactions in complex-structured medical data, yielding a better performance in predicting sepsis with a higher AUROC and AUPRC scores compared with existing methods. Moreover, the MB-TCN framework can be broadly implemented to study general multivariate time series classification problems with missing values and imbalanced data for chronic disease prediction and anomaly detection.

Chapter 3

Hierarchical deep learning with generative adversarial network for automatic cardiac diagnosis from ECG signals

Cardiac disease is the leading cause of death in the US. Accurate heart disease detection is of critical importance for timely medical treatment to save patients' lives. Routine use of electrocardiogram (ECG) is the most common method for physicians to assess the electrical activities of the heart and detect possible abnormal cardiac conditions. Fully utilizing the ECG data for reliable heart disease detection depends on developing effective analytical models. In this paper, we propose a two-level hierarchical deep learning framework with Generative Adversarial Network (GAN) for automatic diagnosis of ECG signals. The first-level model is composed of a Memory-Augmented Deep AutoEncoder with GAN (MadeGAN), which aims to differentiate abnormal signals from normal ECGs for anomaly detection. The second-level learning aims at robust multi-class classification for different arrhythmias identification, which is achieved by integrating the transfer learning technique to transfer knowledge from

the first-level learning with the multi-branching architecture to handle the data-lacking and imbalanced data issue. We evaluate the performance of the proposed framework using real-world medical data from the MIT-BIH arrhythmia database. Experimental results show that our proposed model outperforms existing methods that are commonly used in current practice.
Index Terms: Deep learning, Hierarchical Model, Generative Adversarial Network, Multi-branching Output

3.1 Introduction

Heart disease is the leading cause of death in the US (Virani et al., 2021). It is of critical importance to develop accurate and reliable heart disease diagnoses for timely medical treatments to save patients' lives (Yao, 2021; Yao and Yang, 2016). The heart rhythm is generated by the excitation, propagation, and coordination of electrical signals from the cardiac cells across different heart chambers. The resulting electrical signals on the body surface are described by the electrocardiogram (ECG), which consists of a P-wave, QRS-complex, and T-wave (Hurst, 1998). Changes in electrophysiological properties will vary the propagation pattern of electrodynamics and lead to different types of abnormalities and/or cardiac arrhythmias manifested in the variation of ECG waveform patterns (Garabelli et al., 2016; Borleffs et al., 2009). In recent years, rapid advancements in wearable sensing and information technology facilitate the effective monitoring of patients' heart health conditions (Zhu et al., 2018; Liu et al., 2020; Yao et al., 2021; Xie and Yao, 2022a; Iranzad et al., 2021; Yao and Yang, 2020). Routine use of ECG is the most common method for physicians in everyday clinical practice to assess cardiac electrical activities and detect possible abnormal conditions. Physicians generally identify cardiac arrhythmia by checking the ECG waveforms with naked eyes. This can be time-consuming and may require extensive human resources. Additionally, ECG misinterpretation may happen especially when there exists a large amount of data to inspect (Schull et al.,

2006). Auto arrhythmia detection based on machine learning algorithms can provide important assistance to physicians (Rizwan et al., 2020). However, the research on fully utilizing ECGs for data-driven disease detection poses several challenges including:

(1) **Nonlinear and nonstationary dynamics**: Real-world cardiovascular systems are featured with nonlinear and nonstationary dynamics from the complicated interactions of many interconnected parts such as ion channels and gap junctions to perform cardiac functions, generating ECG signals with nonlinear waveforms. Traditional statistical and machine learning methods depend heavily on manual feature engineering of such waveform data, which generally consists of two stages (Minchole et al., 2019): human experts extract useful features from raw ECGs at the first stage and then employ machine learning algorithms on the handcrafted features to generate predictive results at the second stage. However, this procedure is restricted by the data quality and human expert knowledge (Guglin and Thatai, 2006), and may result in information loss, which lacks the potential for real clinical implementation. Thus, new algorithms that are able to effectively and automatically extract useful features are urgently needed for reliable heart disease identification.

(2) **Lack of training labels and imbalanced data issue**: Most existing data-driven models for ECG analysis are achieved through supervised learning, which requires a large volume of annotated ECG cycles (with diagnostic labels such as normal, abnormal, or specific types of arrhythmia). However, the annotation process requires cardiologists to manually inspect the ECG signals and assign a label to each different pattern, which is time-consuming and labor-intensive. Additionally, it is impractical to collect enough data for each type of disease-altered signals in order to meet the requirement for sufficient supervised training. This is due to the fact that data associated with abnormal heart conditions is significantly less than data from healthy people. Moreover, the occurrence rate of different arrhythmias is highly diverse. Data-driven predictive modeling based on such imbalanced data tends to ignore the minority classes, leading to unsatisfactory detection performance. As such,

new methods that can effectively model the ECGs and account for the data-lacking and imbalanced data issues are needed for reliable disease identification.

This paper proposes a hierarchical deep learning framework with Generative Adversarial Network (GAN) to investigate ECG signals for automatic identification of different types of arrhythmias. We first propose a Memory-Augmented Deep AutoEncoder with Generative Adversarial Network (MadeGAN) to achieve the first-level anomaly detection (i.e., binary classification for normal and abnormal signals). Second, we employ the transfer learning technique to transfer knowledge learned from the first-level training for second-level multi-class classification to identify different types of arrhythmias. In addition, in the second-level network, we adapt the multi-branching architecture developed in our prior work (Wang and Yao, 2021) to solve the imbalanced data issue among different types of heart diseases. We evaluate our proposed hierarchical deep learning framework using the data from the MIT-BIH arrhythmia database (Moody and Mark, 2001). Experimental results show that our proposed method significantly outperforms existing approaches that are commonly used in current practice.

3.2 Research Background

A variety of statistical and machine learning algorithms have been developed for ECG data analysis and pattern recognition (Wang et al., 2013). For example, Yang *et al* (Yang et al., 2013) developed a dynamic spatiotemporal warping algorithm to measure dissimilarities between ECG signals and further employed the spatial embedding to transform the warping dissimilarity matrix into feature vectors for myocardial infarction classification. Aziz *et al* (Aziz et al., 2021) proposed two-event related moving-averages and fractional-Fourier-transform algorithms to analyze ECG signals. Bertsimas *et al* (Bertsimas et al., 2021) utilized the XGBoost algorithm to capture disease-altered patterns in ECG cycles for heart disease prediction. Wavelet-based and recurrence analysis approaches have also been widely implemented to learn

waveform features for ECG classification (Addison, 2005; Yao et al., 2017; Chen et al., 2019). Lyon *et al* (Lyon et al., 2018) investigated the linear and quadratic discriminants, support vector machine, random forest, and Bayesian network for heartbeat classification from ECG signals. A comprehensive review of statistical and machine learning methods in ECG detection can be found in (Minchole et al., 2019; Serhal et al., 2022). However, most existing traditional data-driven methods depend heavily on manual feature engineering, which is a labor-intensive trial-and-error process and is generally limited by human expert knowledge (Guglin and Thatai, 2006; Schläpfer and Wellens, 2017).

Deep Neural Network (DNN) is another powerful tool that has achieved promising results in the area of data-driven disease detection (LeCun et al., 2015). Unlike conventional statistical and machine learning methods, the main advantage of DNNs is that they do not require explicit feature engineering. Instead, feature extraction is automatically achieved by intermediate layers of the network. Due to the powerful ability of representation learning and feature extraction, DNNs has achieved success in many fields (Chen et al., 2022; Shi et al., 2022). It has been demonstrated that DNN-based features are more informative than handcrafted features for arrhythmia detection (Hong et al., 2019b; Galloway et al., 2019). As such, a variety of DNN models including convolutional neural networks (CNNs) (Albawi et al., 2018) have been designed for arrhythmia detection and have outperformed conventional statistical methods (Serhal et al., 2022; Ebrahimi et al., 2020; Murat et al., 2020; Eltrass et al., 2021). For example, Hannun *et al* (Hannun et al., 2019) employed 1D CNN to classify 12 rhythm classes and achieved high performance that is comparable to the diagnostic results provided by cardiology experts. Li *et al* (Li et al., 2018) combined a 2D CNN and a distance matrix to classify congestive heart failure. Shashikumar *et al* (Shashikumar et al., 2018) developed an attention-based model with a 2d CNN as the feature extractor and a bidirectional recurrent neural network to capture the temporal pattern in ECG signals.

However, most existing deep learning algorithms for ECG analysis are based on supervised learning, which requires a large volume of annotated ECG signals and also suffers from the problem of extremely imbalanced data. Thus, the application of unsupervised and semi-supervised learning in ECG analysis has been increasingly investigated. For example, AutoEncoder (AE), a semi-supervised deep learning technique, has been widely used to study ECG data by extracting critical low-dimension representation of the raw signals for disease prediction (Xia et al., 2018; Xiong et al., 2016). Furthermore, GAN-based framework, another semi-supervised learning technique to capture inherent data distributions (Goodfellow et al., 2014; Li et al., 2021b), has been applied in ECG analysis. For example, Zhou *et al* (Zhou et al., 2019) developed a BeatGAN structure to model ECG signals for anomaly detection. Shin *et al* (Shin et al., 2020) integrated the AnoGAN framework (Schlegl et al., 2017) with a decision boundary-based model for ECG anomaly detection. Additionally, because GANs framework has ability to generate synthetic samples that are similar to the real data, many researchers employed GANs framework for data augumentation to handle the imbalanced issue in ECG dataset (Wang et al., 2019b; Xia et al., 2023; Rath et al., 2021). However, most existing semi-supervised deep learning methods mainly focus on differentiating the abnormal ECGs from normal ones (i.e., binary classification) and they are not able to perform multi-class classification to identify different types of cardiac arrhythmia. Thus, novel analytical models are urgently needed to efficiently handle the imbalanced data issue and the data lacking problem for both robust anomaly detection and accurate disease identification from ECG signals.

3.3 Research methodology

As shown in Fig. C.11, this section presents the proposed hierarchical deep learning framework for automatic ECG diagnosis. We denote a single ECG cycle as $\boldsymbol{x} \in \mathbb{R}^{d_x \times 1}$, where d_x denotes the dimensionality of $\boldsymbol{x}$. Each ECG cycle is associated with a multiclass label y. As such, each training data point can be described by the tuple

$(\boldsymbol{x}, y)$ with $y = 0$ indicating normal signal and $y = 1, 2, \ldots, M$ corresponding to other different types of arrhythmias. Our objective is to first differentiate abnormal ECG signals from normal ones (i.e., first-level anomaly detection) and then classify the abnormal signals into different types of arrhythmias (i.e., second-level multi-class classification). Specifically, we propose a Mmemory-augmented deep autoencoder with generative adversarial network (MadeGAN) to achieve the first-level anomaly detection. The second-level classification network is constructed by integrating a shallow classifier with the part of trained discriminator from the first-level learning (i.e., transfer learning to handle the data-lacking problem) and a multi-branching layer (to handle the imbalanced data issue).

3.3.1 MadeGAN for First-level Anomaly Detection

The goal of first-level MadeGAN is to differentiate abnormal ECG cycles from normal ones. Specifically, MadeGAN is trained with normal samples and aims to generate similar reconstructed signals for normal ECG inputs with small anomaly scores and dissimilar reconstructions for abnormal ECGs with bigger anomaly scores. As shown in Fig. C.11(a), the MadeGAN is designed by integrating semi-supervised learning based on memory-augmented deep autoencoder (MemAE) (Gong et al., 2019) with adversarial training, which will be detailed in the following subsections.

Semi-supervised learning based on memory-augmented deep autoencoder (MemAE)

The MemAE consists of three key modules: an encoder to encode the input signal $\boldsymbol{x} \in \mathcal{X}$ and generate a query latent vector $\boldsymbol{z} \in \mathcal{Z}$, a decoder to reconstruct signal $\hat{\boldsymbol{x}}$ from the latent space, and a memory module with a prototypical database to store the encoded patterns of normal ECGs. In other words, MemAE is composed by adding a memory module into an autoencoder (AE).

An AE leverages feedforward neural networks for representation learning where the input is the same as the output. A traditional AE only contains an encoder and a

decoder. The encoder is characterized by a function $f_e(\cdot)$ from original data domain $\mathcal{X} \subset \mathbb{R}^{d_x \times 1}$ to a latent manifold domain $\mathcal{Z} \subset R^{d_z \times 1}$, i.e., $f_e(\cdot) : \mathcal{X} \to \mathcal{Z}$, which captures the critical information of the original signal $\boldsymbol{x}$ by designing a latent vector $\boldsymbol{z}$ with reduced dimensionality (i.e., $d_z < d_x$) through

$$\boldsymbol{z} = f_e(\boldsymbol{x}, \boldsymbol{\theta}_e) \tag{3.1}$$

where $\boldsymbol{\theta}_e$ denotes the parameter set of the encoder. The decoder is defined by the function $f_d(\cdot)$ to convert $\boldsymbol{z}$ back to the original space and generate the reconstructed signal as

$$\hat{\boldsymbol{x}} = f_d(\boldsymbol{z}, \boldsymbol{\theta}_d) = f_d(f_e(\boldsymbol{x}, \boldsymbol{\theta}_e), \boldsymbol{\theta}_d) \tag{3.2}$$

where $\boldsymbol{\theta}_d$ denotes the parameter set of the decoder. The objective of AE is to effectively compress the input data $\boldsymbol{x}$ into a latent-space representation $\boldsymbol{z}$ and reconstruct a signal $\hat{\boldsymbol{x}}$ that is close to $\boldsymbol{x}$ from the encoded representation. This optimization objective is generally achieved by minimizing the following reconstruction error:

$$\mathcal{L}_{ed}(\boldsymbol{\theta}_e, \boldsymbol{\theta}_d) \quad = \quad \frac{1}{N} \sum_{i=1}^{N} \|\hat{\boldsymbol{x}} - \boldsymbol{x}\|^2 \tag{3.3}$$

where N denotes the size of the training set. If the AE is trained purely based on normal data, it is expected that bigger reconstruction errors will be produced for abnormal input signals than the normal inputs. As such, AE has been widely applied for anomaly detection in various areas (Gong et al., 2019).

However, traditional AE-based approaches for anomaly detection often fail to explicitly consider variations in normal signals. In particular, nonlinear and non-stationary dynamics are inherent in the real-world cardiovascular system, generating ECG signals with nonlinear waveforms. There are variations in the shape, amplitude,

and phase among different ECG cycles even under normal conditions. Due to the powerful representation capability of DNN, the AE trained by normal signals with diverse patterns tends to generalize well and may reconstruct abnormal signals with small errors (Gong et al., 2019). To mitigate the drawback of traditional AEs, we leverage a memory module to recognize the diversity of normal patterns and reduce the generalizability of AE to abnormal signals as inspired by the work from (Gong et al., 2019).

The memory module is added in between the encoder and decoder as shown in Fig. C.11(a) and is formed by a memory database with a matrix of $\mathbf{\Omega} \subset \mathbb{R}^{N_\Omega \times d_z}$ to store the representative latent features of normal ECG signals, where N_Ω denotes the number of prototypical vectors stored in the memory module. Given a latent variable $\mathbf{z}$, the memory module will generate a retrieved latent representation according to

$$\hat{\mathbf{z}} = \mathbf{\Omega}^T \boldsymbol{w} \ = \sum_{i=1}^{N_\Omega} w_i \boldsymbol{\omega}_i \tag{3.4}$$

where $\boldsymbol{\omega}_i$ denotes the ith memory element in the memory database $\mathbf{\Omega}$, and $\boldsymbol{w}$ is a weight vector with $w_i \geq 0$ and $\sum_{i=1}^{N_\Omega} w_i = 1$ to characterize the contribution of each prototypical vector in $\mathbf{\Omega}$ when constructing $\hat{\mathbf{z}}$. Note that the weight vector $\boldsymbol{w}$ is designed based on the similarity between the memory elements and the latent query vector $\boldsymbol{z}$ as

$$w_i = \frac{\exp(d(\mathbf{z}, \boldsymbol{\omega}_i))}{\sum_{i=1}^{N_\Omega} \exp(d(\mathbf{z}, \boldsymbol{\omega}_i))} \tag{3.5}$$

where $d(\cdot, \cdot)$ is a similarity measure, which is selected as the cosine similarity $d(\mathbf{z}, \boldsymbol{\omega}_i) = \frac{\mathbf{z}^T \boldsymbol{\omega}_i}{\|\mathbf{z}\| \|\boldsymbol{\omega}_i\|}$ in this study. Hence, the memory module is designed to extract prototypical vectors in $\mathbf{\Omega}$ with high similarity with $\boldsymbol{z}$ to generate the representation vector $\hat{\boldsymbol{z}}$. $\hat{\boldsymbol{z}}$ will then serve as the input of the decoder to reversely map the latent domain back to the original space and generate the reconstructed signal $\hat{\boldsymbol{x}}$, i.e., $\hat{\boldsymbol{x}} = f_d(\hat{\boldsymbol{z}}, \boldsymbol{\theta}_d)$.

Adversarial-training enhanced anomaly detection

To further optimize the AE, minimize potential overfitting possibility, and enlarge the gap in reconstruction error between anomalous and normal signals, we propose to leverage the GAN framework (Goodfellow et al., 2014) to incorporate adversarial training into the MemAE. The GAN consists of two modules: a generator G and a discriminator D. G is designed to generate synthetic samples that are similar to the real data, while D aims to classify the data as real or artificial. The competition between G and D enables GAN to learn the underlying data distribution and further generate high-quality synthetic samples by optimizing the objective function:

$$\min_{\boldsymbol{\theta}_G} \max_{\boldsymbol{\theta}_D} \mathcal{V}(\boldsymbol{\theta}_G; \boldsymbol{\theta}_D) = \mathbb{E}_{\boldsymbol{x} \sim P_{\boldsymbol{x}}}[\log D(\boldsymbol{x}; \boldsymbol{\theta}_D)] + \mathbb{E}_{\hat{\boldsymbol{z}} \sim P_{\boldsymbol{z}}}[\log(1 - D(G(\hat{\boldsymbol{z}}; \boldsymbol{\theta}_G); \boldsymbol{\theta}_D))] \quad (3.6)$$

where $\boldsymbol{\theta}_D$ and $\boldsymbol{\theta}_G$ denote the parameter sets of the discriminator and generator respectively, and $P_{\boldsymbol{x}}$ and $P_{\boldsymbol{z}}$ represent probability distributions of the real sample and latent variable spaces, respectively. The discriminator D tries to maximize the objective function in Eq. (3.6) with respect to $\boldsymbol{\theta}_D$ given a fixed generator $G(\cdot; \boldsymbol{\theta}_G)$ by assigning probability of 1 to real data points $\boldsymbol{x}$ and 0 to generated samples $\hat{\boldsymbol{x}}$. The objective of generator, G, is to reconstruct a signal $\hat{\boldsymbol{x}}$ as similar as the original input and fool discriminator D into classifying the synthetic signal as real by minimizing Eq. (3.6) with respect to $\boldsymbol{\theta}_G$ given a fixed discriminator $D(\cdot; \boldsymbol{\theta}_D)$.

In the present investigation, we adopt the MemAE as the generator of GAN, i.e., $G(\cdot) = f_d(f_e(\cdot; \boldsymbol{\theta}_e, \boldsymbol{\Omega}), \boldsymbol{\theta}_d)$, to design a Memory-Augmented Deep AutoEncoder with Generative Adversarial Network (MadeGAN) for the first-level anomaly detection. Hence, the objective function in Eq. (3.6) becomes

$$\min_{\boldsymbol{\theta}_G} \max_{\boldsymbol{\theta}_D} \mathcal{V}'(\boldsymbol{\theta}_G; \boldsymbol{\theta}_D) = \mathbb{E}_{\boldsymbol{x} \sim P_{\boldsymbol{x}}}[\log D(\boldsymbol{x}; \boldsymbol{\theta}_D) + \log(1 - D(f_d(f_e(\boldsymbol{x}; \boldsymbol{\theta}_e, \boldsymbol{\Omega}), \boldsymbol{\theta}_d)); \boldsymbol{\theta}_D)] \quad (3.7)$$

where $\boldsymbol{\theta}_G = \{\boldsymbol{\theta}_e, \boldsymbol{\Omega}, \boldsymbol{\theta}_d\}$, which denotes the parameter set in MemAE. In other words, we add adversarial training into the MemAE to improve the performance in anomaly

detection. Additionally, in order to guarantee the success of the training procedure, we add feature matching loss (Salimans et al., 2016) into the objective function of the generator to minimize the difference of meaningful features that are learned by hidden layers of the discriminator between the real and synthetic signals. Specifically, the feature matching loss is defined as:

$$\mathcal{L}_{fm}(\boldsymbol{\theta}_G) = \mathbb{E}_{\boldsymbol{x}\sim P_{\boldsymbol{x}}}\|h_D(\boldsymbol{x}) - h_D(\hat{\boldsymbol{x}})\| \tag{3.8}$$

where $h_D(\cdot)$ denotes feature vector learned from intermediate layers of the discriminator D.

Moreover, in order to increase the computation efficiency, we propose to promote the sparsity of the weight vector such that the representation latent vector $\hat{\boldsymbol{z}}$ can be reconstructed with a restricted number of phenotypical patterns in the memory database. This is achieved by minimizing the L1 norm of the weight vector $\boldsymbol{w}$, i.e.,

$$\mathcal{L}_w(\boldsymbol{\Omega}) = \|\boldsymbol{w}\|_1 \tag{3.9}$$

As such, the loss function of the MemAE is adapted by the adversarial training given a fixed discriminator $D(\cdot;\boldsymbol{\theta}_D)$ as

$$\mathcal{L}_{edA}(\boldsymbol{\theta}_G;\boldsymbol{\theta}_D) \quad = \quad \mathbb{E}_{\boldsymbol{x}\sim P_{\boldsymbol{x}}}\|\boldsymbol{x} - \hat{\boldsymbol{x}}\|^2 + \mathcal{L}_w(\boldsymbol{\Omega}) + \mathbb{E}_{\boldsymbol{x}\sim P_{\boldsymbol{x}}}\|h_D(\boldsymbol{x}) - h_D(\hat{\boldsymbol{x}})\| \tag{3.10}$$

where $\hat{\boldsymbol{x}} = f_d(f_e(\boldsymbol{x};\boldsymbol{\theta}_e,\boldsymbol{\Omega}),\boldsymbol{\theta}_d)$. $\mathcal{L}_{edA}(\boldsymbol{\theta}_G;\boldsymbol{\theta}_D)$ is also considered as the loss function of the generator in the proposed MadeGAN to improve the robustness and stability of network training. Thus, the overall objective of MadeGAN is given by

$$\min_{\boldsymbol{\theta}_G}\max_{\boldsymbol{\theta}_D}\mathcal{L}(\boldsymbol{\theta}_G,\boldsymbol{\theta}_D) = \mathcal{V}'(\boldsymbol{\theta}_G;\boldsymbol{\theta}_D) + \mathcal{L}_{edA}(\boldsymbol{\theta}_G;\boldsymbol{\theta}_D) \tag{3.11}$$

By optimizing $\mathcal{L}(\boldsymbol{\theta}_G,\boldsymbol{\theta}_D)$, the adversarial training pushes the generator (i.e., the MemAE) to improve in generating realistic reconstructions from normal inputs such

that the discriminator cannot distinguish between the real and artificial signals. As such, the proposed MadeGAN is expected to enhance the anomaly detection performance of traditional AE and MemAE to better preserve the phenotypical patterns of normal ECG data, which will output low reconstruction errors for normal signals and high errors for abnormal ones.

Anomaly detection

During the inference stage, the anomaly degree of a new query signal $\boldsymbol{x}$ is evaluated by the following anomaly score:

$$s(\boldsymbol{x}) = \|\boldsymbol{x} - G(\boldsymbol{x})\| \tag{3.12}$$

which quantifies the discrepancy between the true signal and the signal reconstructed by patterns extracted from normal signals by G. The network is expected to output big anomaly scores for abnormal signals. On the other hand, a small $s(\boldsymbol{x})$ indicates that the test signal $\boldsymbol{x}$ shares similar patterns as seen in normal ECGs, which will then be classified as normal by the proposed MadeGAN.

3.3.2 Transfer learning-enhanced second-level classification

The objective of second-level learning is to classify the abnormal ECGs identified from first-level MadeGAN into different types of arrhythmias for accurate disease detection, as shown in Fig. C.11(b). Note that the performance of deep learning-based classification depends, to a great extent, on the quality and quantity of training data. However, due to the expensive data collection and labeling process, data associated with abnormal heart conditions is significantly less than that from the healthy condition. Moreover, the occurrence rates of different diseases are highly diverse, depending on the underlying characteristics of the patient's health condition, which leads to imbalanced data sizes for different disease types. Traditional predictive modeling based on such small and imbalanced data tends to generate biased estimates, leading to unsatisfactory detection performance. As such, innovative design of

network architecture and novel training technique are urgently needed to study ECG signals for accurate disease identification.

Here, we propose to take advantage of the large volume of normal ECG cycles to improve the second-level classification through transfer learning (Pan and Yang, 2009). Note that even though the exact distribution and morphological shapes between normal and abnormal ECGs are different, they share common patterns on the macroscale. For example, each ECG consists of a P-wave, QRS-complex, and T-wave. Hence, such macroscale pattern information in normal ECGs can be transferred to the second-level learning to improve the classification performance.

Transfer learning generally involves two different domains: source domain $\mathcal{X}_S$, and target domain $\mathcal{X}_T$, and the corresponding two different learning tasks $\mathcal{T}_S$ and $\mathcal{T}_T$, respectively. The objective of transfer learning is to leverage the knowledge learned from $\mathcal{T}_S$ from the source domain $\mathcal{X}_S$ to improve the learning task $\mathcal{T}_T$ in the target domain $\mathcal{X}_T$. Specifically, in the present investigation, the source domain $\mathcal{X}_S$ consists of the normal ECG data and the associated task $\mathcal{T}_S$ is anomaly detection. Our target task $\mathcal{T}_T$ for second-level learning is multi-class classification to identify different types of arrhythmias from the data in the target domain $\mathcal{X}_T$ (i.e., abnormal ECGs). To achieve the knowledge transfer, we adopt the discriminator D in the MadeGAN with learned network parameters $\hat{\boldsymbol{\theta}}_D$ from the first-level training to improve the learning of the multi-class classification $\mathcal{T}_T$ in the second level.

In addition, because the common problem of imbalanced data may still exist in different types of abnormal signals, we add the multi-branching (MB) architecture (Wang and Yao, 2021) as shown in Fig. C.11(c) to the 1-D CNN to address the imbalanced data issue and further improve the classification performance. We denote the abnormal dataset as $\mathcal{D}_A = \{\boldsymbol{X}_A, \boldsymbol{y}_A\}$, where $\boldsymbol{X}_A$ is the set of abnormal ECGs and $\boldsymbol{y}_A$ denotes the corresponding label set. Assume there are M different types of arrhythmia or $\mathcal{D}_A$ consists of M different sub-groups, i.e., $\mathcal{D}_A = \{\mathcal{D}_1, \ldots, \mathcal{D}_M\}$, and assume class l contains the smallest data samples, i.e., $|\mathcal{D}_l| \leq |\mathcal{D}_m|$ for $m \neq l$ and $m \in \{1, \ldots, M\}$. We generate N_b balanced datasets,

$\mathcal{D}_{Ai}$'s ($i \in \{1, \ldots, N_b\}$), from $\mathcal{D}_A$ by under-sampling the class with larger sample size, i.e., $\mathcal{D}_{Ai} = \{\mathcal{D}_{1i}, \ldots, \mathcal{D}_{(l-1)i}, \mathcal{D}_l, \mathcal{D}_{(l+1)i}, \ldots, \mathcal{D}_{Mi}\}$, where $\mathcal{D}_{mi}$ is a subset by under-sampling the data $\mathcal{D}_m$ from type m arrhythmia ($m \in \{1, \ldots, l-1, l+1, \ldots, M\}$) to the same size with $\mathcal{D}_l$. The corresponding N_b branching outputs for the N_b balanced datasets are further created and attached to the 1D CNN for robust and accurate disease identification.

The shallow 1D CNN will be trained by all the N_b datasets, and each branching output will be trained by the corresponding balanced sub-dataset. This MB architecture and training procedure will enable the network to learn more effectively from the imbalanced dataset for reliable classification (Wang and Yao, 2021). The objective of second-level learning is to search a set of optimal parameters such that the network outputs a high probability for the true label and low probabilities for others. As such, we select the multi-class cross-entropy as the loss function of the second-level network:

$$\mathcal{L}(\boldsymbol{\theta}_{2nd}; \boldsymbol{X}_A) = -\sum_{p=1}^{N_A}\sum_{i=1}^{N_b} \mathcal{I}(p \in \mathcal{D}_{Ai}) \Big(\sum_{m=1}^{M} y_p^{(m)} \log(\hat{P}_i^{(m)}(\boldsymbol{\theta}_{2nd}; \bar{D}(\boldsymbol{x}_A^p; \hat{\boldsymbol{\theta}}_D))) \Big) \quad (3.13)$$

where $\boldsymbol{\theta}_{2nd}$ denotes the parameter set of the second-level network, N_A is the sample size of the abnormal dataset, $\mathcal{I}(\cdot)$ is an indicator function, and $\hat{P}_i^{(m)}(\boldsymbol{\theta}_{2nd}; \bar{D}(\boldsymbol{x}_A^p; \hat{\boldsymbol{\theta}}_D))$ is the predicted probability of type m arrhythmia from branching output i given the features $\bar{D}(\boldsymbol{x}_A^p; \hat{\boldsymbol{\theta}}_D))$ extracted by the discriminator from the input $\boldsymbol{x}_A^p$. Please see more details about the MB network in our prior work (Wang and Yao, 2021). The final predicted probability of class m for input $\boldsymbol{x}_A^p$ is given by

$$\hat{P}^{(m)} = \frac{1}{N_b} \sum_{i=1}^{N_b} \hat{P}_i^{(m)}(\boldsymbol{\theta}_{2nd}; \bar{D}(\boldsymbol{x}_A^p; \hat{\boldsymbol{\theta}}_D)) \quad (3.14)$$

Integrating the MB structure and the discriminator-based transfer learning with the 1D CNN enables our second-level model (MB-Conv1d-Discrim) to overcome

the data-lacking and imbalanced data issues, and further improve the classification performance for accurate disease identification.

3.4 Experimental Design and Results

3.4.1 Data Description and Experimental Design

The dataset used in this study is obtained from MIT-BIH database (Moody and Mark, 2001). This dataset contains 48 half-hour ECG recordings, which is obtained from 47 subjects from the Beth Israel Hospital Arrhythmia Laboratory between 1975 and 1979. Each recording is digitized at 360 samples per second with 11-bit resolution. Two or more independent cardiologists annotate the beats in each recording. We remove four recordings (102, 104, 107, 217) because of their poor signal quality. Each heartbeat is segmented through the following procedure: (1) The Pan-Tompkins QRS detection algorithm (Pan and Tompkins, 1985) is first employed to detect all R-peak locations in original ECG recordings; (2) 320 samples are then selected for each heartbeat with 140 samples before the corresponding R-peak and 180 samples after it. This procedure guarantees that segmented signals have the same dimension and approximately cover one heart cycle. In total, our dataset contains 97,553 annotated beats including: 86,717 normal signals (N), 7,008 premature ventricular contractions (V), 3,026 supraventricular premature beats (S), and 802 ventricular fusion beats (F). Additionally, we employ the high-pass finite impulse response (FIR) filter to reduce multiple types of noise commonly seen in ECG signals such as power-line interference and baseline wandering (An and K Stylios, 2020).

The performance of our first-level MadeGAN in anomaly detection will be compared with a pure AutoEncoder (AE), AE with memory module (MemAE), and BeatGAN (Zhou et al., 2019). The pure AE only contains an encoder and a decoder trained by only normal signals. However, due to the powerful representation capability of DNN, the AE trained by normal signals with diverse patterns tends to

generalize well and may reconstruct abnormal signals with small errors, leading to unsatisfactory performance for anomaly detection. BeatGAN is a semi-supervised learning framework consisting of traditional AE with adversarial training. The adversarial training can improve the model robustness, but BeatGAN still suffers from the limitations of traditional AE. MemAE is a variation of traditional AE, where a memory module is added between encoder and decoder. The memory module can recognize the diversity of normal patterns and reduce the generalizability of AE to abnormal signals. However, it is less robust compared to the models trained by adversarial training framework. This is due to the fact that the adversarial training pushes the generator to improve in generating realistic reconstructions from normal inputs, further enlarging the gap in reconstruction error between anomalous and normal signals.

The second-level model (i.e., MB-Conv1d-Discrim) will be benchmarked with a pure 1D CNN (Conv1d), 1D CNN with the generator feature extractor (Conv1d-Gene), 1D CNN with the encoder feature extractor (Conv1d-Encoder), 1D CNN with the discriminator feature extractor (Conv1d-Discrim), 1D CNN with an MB layer (MB-Conv1d), Conv1d-Gene with an MB layer (MB-Conv1d-Gene), and Conv1d-Encoder with an MB layer (MB-Conv1d-Encoder). The pure 1D CNN can be used as a classifier for time series data, but it lacks the ability to handle extremely imbalanced datasets and requires a large number of training data for good prediction performance. The 1D CNN combined with transfer learning method (i.e., Conv1d-Encoder, Conv1d-Discrim, Conv1d-Gene) can mitigate the problem of lacking of training data, but it cannot handle the imbalanced data issue. Three different feature extractors (i.e., Encoder, generator, discriminator) can be chosen to do transfer learning. However, only the discriminator (i.e., trained by fake and real data) performs a classification task. As such, the discriminator-enabled transfer learning (i.e., Conv1d-Discrim) can capture more accurate features for the second-level task (which is also a classification task) compared with other feature extractors. The multi-branching (MB) framework (Wang and Yao, 2021) is effective to handle the imbalanced data issue, which is

combined with the transfer learning, i.e., MB-Conv1d-Encoder, MB-Conv1d-Discrim, MB-Conv1d-Gene, to further improve the classification performance.

The performance of both the first and second-level learning will be evaluated according to: two overall performance metrics, i.e., Area Under Receiver-Operating-Characteristic Curve (AUROC), and Area Under Precision-Recall Curve (AUPRC), and five-point performance metrics, i.e., *Recall*, *Precision*, $f-score$, *Accuracy*, and $Balanced-Accuracy$. The ROC characterizes the relationship between the False Positive Rate (FPR) and True Positive Rate (TPR), and AUROC is a score quantifying the general prediction performance across all thresholds. Similarly, the AUPRC evaluates the overall relationship between *Recall* and *Precision*. Note that the $Balanced-Accuracy$ for the performance of second-level models is equivalent to the average of *Recall* scores for all the abnormal classes.

To evaluate the performance metrics for the first-level anomaly detection, we calculate the anomaly score, s_i, for each sample in the test set, generating the set $S_A = \{s_1, s_2, \ldots, s_{N_T}\}$, where N_T denotes the number of samples in the test set. Then, we apply feature scaling to convert the anomaly scores into the probabilistic range of [0,1]:

$$s_i^{'} = \frac{s_i - \min(S_A)}{\max(S_A) - \min(S_A)} \tag{3.15}$$

As such, different thresholds on the normalized score $s_i^{'}$ will be applied to calculate FPR, TPR, *Recall*, and *Precision*, and further compute the AUROC and AUPRC scores for the first-level anomaly detection.

3.4.2 Training and Testing Procedures

Fig. C.12 shows the detailed architecture of the proposed two-level framework. Our first-level MadeGAN contains four core modules: encoder, decoder, memory module, and discriminator. We use five 1D convolutional (Conv1d) layers in the encoder. Note that the notation of $(n_{filter}, n_{size}, n_{stride})$ represents that there are n_{filter} filters

with a filter size of n_{size} and the stride of n_{stride}. For example, (32-4-2) means that this Conv1d layer is constructed by 32 filters with the filter size of 4 and stride of 2. The latent space dimension of the encoder is 50, i.e., $d_{\boldsymbol{z}} = 50$. Additionally, we use batch-normalization and leaky ReLU activation to each Conv1d layer to make the model training more stable. The decoder is a mirrored version of the encoder with five transposed Conv1d layers. The memory model contains 2000 prototypical vectors of normal ECGs, i.e., $N_\Omega = 2000$. The discriminator has the same core architecture as the encoder with an additional Conv1d and fully connected layers for classification. Our second-level model consists of a Conv1d layer, an MB layer with 4 branching outputs, and a feature extractor. Note that the feature extractor is composed of the discriminator without the last Conv1d and fully connected layers (i.e., the part of discriminator inside the dashed box in Fig. C.12). Specifically, the abnormal signal will first go through Conv1d (32-4-2), Conv1d (64-4-2), Conv1d (128-4-2), Conv1d (256-4-2), and Conv1d (512-4-2) layers in the discriminator including all Leaky Relu and Batch normalized layers. Then, the output from Conv1d (512-4-2) layer will go to the shallow 1D CNN layer (i.e., Conv1d (32-1-2)) and an MB layer to make the final second-level prediction. Adam optimizer with an initial rate of $lr = 0.0002$ and momentums of $\beta_1 = 0.5$, $\beta_2 = 0.999$ is used for network training. The network hyper-parameters, i.e., $\{n_{filter}, n_{size}, n_{stride}, d_z, N_\Omega, lr, \beta_1, \beta_2\}$ are selected by empirical fine-tuning.

Fig. C.13 shows the training and testing procedures for the proposed model. Specifically, during the first-level MadeGAN training, we use 90% normal signals as the training set and the rest of 10% normal signals as the test set. In order to evaluate the anomaly detection performance of MadeGAN in the testing phase, all abnormal ECGs are combined with the rest 10% normal ones to form the test set. During the second-level training, we use 90% abnormal ECGs as the training set, which contains 3 different types of arrhythmias (i.e., the V, S, and F samples). The rest of 10% abnormal signals is used as the test set to evaluate the performance of the second-level classification. Note that during the second-level training, the parameter $\hat{\boldsymbol{\theta}}_D$ of

discriminator D adopted from MadeGAN is held frozen to achieve the knowledge transfer from the first-level learning.

3.4.3 Experimental Results

First-level Anomaly Detection

Table D.7 shows the performance comparison between MadeGAN and other benchmarks in the first-level anomaly detection. Note that MadeGAN achieves the best performance with AUROC and AUPRC scores of 0.954 and 0.936. Specifically, MadeGAN improves on AUROC by 5.3% compared with the pure AE (0.906), by 2.9% compared with MemAE (0.927), and by 1.4% compared with BeatGAN (0.941). Furthermore, the improvement is more significant in terms of AUPRC: with the improvement of 5.6%, 3.8%, and 1.5% compared with the pure AE (0.886), MemAE (0.902), and BeatGAN (0.922), respectively.

Fig. C.14 (a) and (b) display the AUROC and AUPRC over 350 epochs in the test dataset for different methods. Note that the pure AE and MemAE converge in the first 100 epochs but achieve relatively low performance scores, i.e., AUROC of 0.906 and 0.927, and AUPRC of 0.886 and 0.902, respectively. On the other hand, the performance scores of MadeGAN and BeatGAN improve over epochs and achieve better AUROC of 0.954 and 0.941, and AUPRC of 0.936 and 0.922, respectively. This is due to the fact that both MadeGAN and BeatGAN incorporate the adversarial training into the reconstruction, which alleviates the potential overfitting problem in AE training. In addition, the memory module enables MadeGAN to effectively account for the possible variations in normal signals and reduce the possibility of reconstructing abnormal ECGs with small errors. This further enlarges the gap in reconstruction errors between normal and abnormal signals, which makes MadeGAN achieve better performance compared with BeatGAN. Table D.8 further shows the performance comparison in terms of point metrics. Note that our MadeGAN yields

the best *Recall*, *Precision*, f-score, *Accuracy*, and $Balanced - Accuracy$ of 0.954, 0.856, 0.902, 0.885, and 0.913 respectively among the five methods.

Fig. C.15 displays the examples of reconstructed ECG signals generated by our first-level MadeGAN and other benchmarks for normal ECGs and different types of abnormal ECGs. Here, we adopt the reconstruction error (i.e. $\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|^2$) to evaluate the performance for different approaches. Because all models are trained purely by normal ECGs, we expect that a small reconstruction error will be produced from the normal signal and a large error will be generated from the abnormal signal. According to Fig.C.15 (a), although the reconstructed signals from all the methods match with the input normal signal closely, our MadeGAN generates the smallest reconstruction error (0.042) compared to other approaches (0.109 for BeatGAN, 0.074 for AE, and 0.053 for MemAE). On the other hand, according to Fig. C.15 (b)-(d), MadeGAN is able to generate the most dissimilar reconstructions and produces the largest reconstruction errors for abnormal signals. For example, the reconstruction error generated by MadeGAN for the S class is 6.488, which is significantly bigger than 2.402 by BeatGAN, 0.305 by AE, and 0.612 by MemAE.

Second-level Arrhythmia Identification

Table. D.9 shows the AUROC and AUPRC of our MB-Conv1d-Discrim and other MB-based and Conv1D-based benchmarks in the second-level arrhythmia classification. Note that a good model will have a large AUROC and AUPRC. According to Table. D.9, our MB-Conv1d-Discrim has the best AUROC and AUPRC among other methods. Specifically, compared with MB-Conv1d-Gene and MB-Conv1d-Encoder, our MB-Conv1d-Discrim increases AUROC from 0.983 and 0.963 to 0.988, and increases AUPRC from 0.981 and 0.955 to 0.989 for the S class. Similarly, for the V class, MB-Conv1d-Discrim improves AUROC from 0.985 and 0.960 to 0.988, and increases AUPRC from 0.981 and 0.938 to 0.987 compared with MB-Conv1d-Gene and MB-Conv1d-Encoder. The improvement is more significant in the F class,

which has the smallest sample size among the three classes. Specifically, MB-Conv1d-Discrim achieves 1.5% and 3.8% performance improvement on AUROC, and 3.8% and 10.0% improvement on AUPRC compared with MB-Conv1d-Gene and MB-Conv1d-Encoder for the F class.

Similarly, MB-Conv1d-Discrim increases AUROC from 0.979 and 0.974 to 0.988, and improves AUPRC from 0.961 and 0.956 to 0.989 compared with Conv1d-Gene and Conv1d-Encoder for the S class. For the V class, MB-Conv1d-Discrim enhances AUROC from 0.980 and 0.979 to 0.984, and increases AUPRC from 0.981 and 0.986 to 0.987 compared with Conv1d-Gene and Conv1d-Encoder. For the F class, MB-Conv1d-Discrim achieves 3.9% (from 0.959 to 0.997) and 4.7% (from 0.952 to 0.997) performance improvement on AUROC compared with Conv1d-Gene and Conv1d-Encoder, and achieves 19.3% (from 0.828 to 0.988) and 37.4% (from 0.719 to 0.988) improvement on AUPRC. More importantly, our MB-Conv1D-Discrim achieves a significant improvement of 7.3% on AUROC (from 0.924 to 0.997) and 85% on AUPRC (from 0.534 to 0.988) for F class compared with the pure Conv1D without any transfer learning or MB layer.

According to Table D.9, the discriminator-based feature extractor outperforms the generator-based and encoder-based feature extractors. This is due to the fact that the discriminator is trained to classify the signal as fake or real. In other words, classification is the learning task of the discriminator in the first-level training, which is similar to the second-level task (i.e., multi-class classification). However, the learning tasks for the generator or the encoder are signal reconstruction and latent-space generation respectively, which are different from the second-level learning task. As such, the discriminator-based feature extractor yields the best performance to improve the second-level classification.

It is also worth noting that MB-based methods outperform non-MB-based methods. This improvement is more significant for the class with a small sample size (i.e., F class). Specifically, as shown in Table. D.9, MB-Conv1d increases AUROC and AUPRC from 0.924 to 0.981 and from 0.534 to 0.939 respectively, compared with the

pure Conv1d model. MB-Conv1d-Discrim increases AUROC and AUPRC from 0.963 to 0.997 and from 0.911 to 0.988 respectively, compared with the Conv1d-Discrim. This is due to the fact that the MB architecture guarantees each branching output is trained by a balanced sub-dataset to effectively relieve the imbalanced data issue. As such, the integration of transfer learning and MB architecture has significantly improved classification performance in the second-level learning.

Table D.10 further shows the confusion matrix on the test dataset provided by the proposed second-level model (i.e., MB-Conv1d-Discrim). Note that even for the F class, our model provides pretty good classification performance. Specifically, 80 F samples are correctly predicted out of the 81 total F samples. Table D.11 shows the performance comparison of point metrics between MB-Conv1d-Discrim and other benchmarks. Note that MB-Conv1d-Discrim yields the best *Recall*, *Precision*, f-score, and *Accuracy* of 0.976, 0.976, 0.975, and 0.976, respectively. Note that MB-based methods also outperform non-MB methods in terms of the point metrics. For example, MB-Conv1d-Discrim, MB-Conv1d-Encoder, MB-Conv1d-Generator, and MB-Conv1d achieve 2.5%, 7.3%, 4.7%, and 22.3% improvement on the f-score compared with their non-MB counterparts. It is also worth noting that MB-Conv1d-Discrim achieves a more significant improvement on *Recall*, *Precision*, and f-score than *Accuracy* compared to the pure Conv1d without any transfer learning or MB layer. Specifically, MB-Conv1d-Discrim achieves 29.8% performance improvement on *Recall*, 21.8% on *Precision*, and 20.5% on f-score. This is due to the fact that *Recall*, *Precision*, and f-score are more robust evaluation metrics when there exists an imbalanced issue, which can be effectively addressed by the MB architecture and transfer learning.

Comparison Study with Existing Literature

In order to benchmark with existing literature, we use the 5-fold cross validation to further evaluate the proposed first- and second-level models. The 5-fold cross validation has been proved to be able to generate a less biased estimate of model

performance and is widely utilized to evaluate machine learning models. Table D.12 summarizes the comparison of AUROC and AUPRC between MadeGAN and existing methods that used the MIT-BIH dataset for ECG anomaly detection. Our MadeGAN yields the average of the 5 folds ± 1 standard deviation (std) for AUROC and AUPRC of 0.950±0.002 and 0.922±0.003, respectively, which outperforms existing methods (Zhou et al., 2019; Shin et al., 2020).

Table D.13 summarizes the comparison of the performance scores between our second-level model (i.e., MB-Conv1d-Discrim) and existing approaches. Our method yields the average (±1 std) of the 5 folds for AUROC of 0.989(±0.001), AUPRC of 0.984(±0.002), *Recall* of 0.964(±0.006), *Precision* of 0.967(±0.005), and f-score of 0.965(±0.005), which significantly outperforms existing methods. Note that the *Accuracy* of our model (i.e., 0.967 ± 0.005) is slightly smaller than the accuracy (i.e., 0.987) in (Mousavi et al., 2021). However, the *Accuracy* alone may not serve as a reliable performance measure when evaluating machine learning models with imbalanced data issue (Flach and Kull, 2015). For example, if we have a dataset with 10,000 negative samples and 10 positive samples, we can obtain an *Accuracy* as high as 99.9% if we train a model to predict all samples as negative. However, this model is meaningless because we are more interested in the ability of the model to predict positive samples, which are commonly evaluated by *Recall* and *Precision*. Note that our model has significantly better *Recall* and *Precision* (i.e., 0.964 and 0.967) compared with (Mousavi et al., 2021) (i.e., 0.831 and 0.923).

The performance improvement of the proposed hierarchical deep learning framework compared with existing literature is mainly due to the following four modules:

(1) **Memory module**: The traditional AutoEncoder trained by normal signals may potentially generalize and reconstruct abnormal signals well due to the powerful representation capability of deep neural networks. The memory module added between the encoder and decoder will mitigate the drawback of traditional AutoEncoder by recognizing the diversity of normal patterns and reducing the generalization ability to the abnormal signals.

(2) **Adversarial training**: We propose to leverage the Generative Adversarial Network (GAN) framework to incorporate adversarial training into the Memory AutoEncoder. The adversarial training pushes the generator to improve in generating realistic reconstructions from normal inputs such that the discriminator cannot distinguish between the real and artificial signals. This technique will improve the performance of AutoEncoder, minimize potential overfitting possibility, and enlarge the gap in reconstruction error between anomalous and normal signals.

(3) **Transfer learning-enhanced second-level classification**: Due to the lack of labeled abnormal signals, we also take advantage of the large volume of normal ECG cycles to improve the second-level classification through transfer learning. In order to achieve the knowledge transfer, we leverage the discriminator D from the MadeGAN with learned network parameters $\hat{\boldsymbol{\theta}}_D$ from the 1st-level training to improve the learning of the 2nd-level multi-class classification.

(4) **Multi-branching architecture to address the imbalanced-data issue**: The multi-branching architecture is another contributor to the performance enhancement. Balanced sub-datasets are created during the training procedure such that each branching output is trained with a balanced sub-dataset. This multi-branching architecture and training procedure will enable the network to learn more efficiently and effectively from the imbalanced dataset for reliable classification.

3.4.4 Long-term ST Database

In order to further evaluate the performance of our method, we implement the hierarchical deep learning framework to study the ECG signals for anomaly detection and classification from the long-term ST database (Jager et al., 2003; Goldberger et al., 2000). In this dataset, the ECG recordings are digitalized at 250 samples per second with 12-bit resolution. Heartbeats are annotated by expert annotators independently using SEMIA, a program written by the group in Ljubljana (Jager et al., 2003). We randomly select 86,321 normal heartbeats (N), 7,026 premature

ventricular contractions (V), 3,029 supraventricular premature beats (S), and 405 ventricular fusion heartbeats (F) from the database to evaluate our method through 5-fold cross validation. Similar to the evaluation procedure for the MIT-BIH database, the performance of our 1st-level model (i.e., MadeGAN) is benchmarked with AE, MemAE, and BeatGAN. The 2nd-level model (i.e., MB-Conv1d- Discrim) is benchmarked with Conv1d, Conv1d-Gene, Conv1d-Encoder, Conv1d-Discrim, MB-Conv1d, MB-Conv1d-Gene, and MB-Conv1d-Encoder.

3.4.5 Experimental Results from the First-level MadeGAN

Table D.14 shows the mean evaluation metrics (± 1 std) from the 5-fold cross validation obtained from MadeGAN and other benchmarks. Note that our MadeGAN yields the best average AUROC and AUPRC of 0.984 and 0.986, respectively. In addition, our MadeGAN generates a more robust anomaly detection with the smallest std for AUROC (AUPRC) of 0.001 (0.002) compared with other benchmarks. Similarly, our MadeGAN achieves the best mean *Recall*, *Precision*, f-score, and *Accuracy* of 0.956, 0.954, 0.947, and 0.955 respectively, and generates a more robust anomaly detection performance with the smallest std of 0.004, 0.005, 0.004, and 0.003, respectively.

3.4.6 Experimental Results from the Second-level Classification

Table D.15 shows the evaluation metrics obtained from the second-level multi-classification model. Note that our MB-Conv1d-Discrim model yields the best average performance scores with AUROC of 0.991±0.002, AUPRC of 0.953± 0.003, *Recall* of 0.917±0.005, *Precision* of 0.955± 0.003, f-score of 0.920±0.003, and *Accuracy* 0.972±0.003 compared with other benchmarks.

3.4.7 Discussion on the Limitation

One potential limitation of this work is that we only focus on single-lead ECG analysis. Although single-lead ECG analysis has great application potential for wearable everyday health monitoring, cardiologists usually make the medical diagnosis based on multi-lead ECG signals (e.g., 12-lead ECGs) from patients with severe heart disease. It is worth noting that 12-lead ECGs are multi-channel time-series signals, which can be considered as 3D tensors characterized by (n_b, n_c, n_t), where n_b is the batch size, n_c is the number of channels ($n_t = 12$ for 12-lead ECGs), and n_t is the number of time ticks of one heartbeat. By re-designing the input layer to incorporate the channel information, the 3D tensor data can be fed to and train the network for anomaly detection and arrhythmia identification. However, the predictive performance depends to a great extent on the network structure details and hyperparameter selection. One of our future research directions will focus on adapting the proposed hierarchical deep learning framework and designing effective network structures to capture both the temporal dynamics within each channel and the correlations across different channels to analyze multi-channel ECG signals for heart disease detection.

Another potential limitation is that the finite impulse response (FIR) filter for ECG denoising may not be effective to remove motion artifacts (Satija et al., 2018; Li et al., 2014; Satija et al., 2017). In order to further increase the applicability of the proposed method in real clinical diagnosis, more advanced signal quality assessment methods are needed in preprocessing step (Satija et al., 2018; Li et al., 2014; Satija et al., 2017), which will be investigated and combined with our model for ECG signal analysis in our future work.

3.5 Conclusions

In this paper, we develop a two-level hierarchical deep learning framework with Generative Adversarial Network to investigate ECG data for robust and reliable identification of heart diseases. In the first-level learning, we propose a Memory-Augmented Deep Auto-Encoder with Generative Adversarial Network (MadeGAN) to achieve anomaly detection (i.e., binary classification of normal and abnormal signals). Specifically, the MadeGAN is designed by integrating semi-supervised learning based on memory-augmented deep auto-encoder (MemAE) with adversarial training. The MemAE recognizes the diversity of normal patterns and reduces the generalization ability to abnormal signals. Additionally, the adversarial training minimizes potential overfitting possibility and enlarges the gap in reconstruction error between anomalous and normal signals. In the second-level learning, we employ transfer learning by adopting the discriminator with learned network parameters from the first-level training to handle the data lacking problem and achieve the multi-class classification among different types of arrhythmias. Additionally, the multi-branching technique is used in the second-level model to handle the imbalanced data issue and enhance the classification performance. Experimental results show that our framework effectively captures the disease-altered feature patterns from ECG signals, yielding better performance in predicting heart disease with higher performance scores compared with existing methods. Moreover, this hierarchical deep learning framework can be broadly implemented to study other waveform data such as electroencephalography (EEG) and photoplethysmography (PPG) for smart anomaly detection and multi-class classification.

Chapter 4

Multi-branching temporal convolutional network with tensor data completion for diabetic retinopathy prediction

Diabetic retinopathy (DR), caused by damage to the blood vessels in the tissue of the retina, is a microvascular complication of diabetes. DR is the leading cause of vision loss among working-aged adults. However, due to the low compliance rate of DR screening and expensive medical devices for ophthalmic exams, many DR patients did not seek proper medical attention until DR develops to irreversible stages (i.e., vision loss). Fortunately, the widely available electronic health record (EHR) databases provide an unprecedented opportunity to develop cost-effective machine-learning tools for DR detection. This paper proposes a Multi-branching Temporal Convolutional Network with Tensor Data Completion (MB-TCN-TC) model to analyze the longitudinal EHRs collected from diabetic patients for DR prediction. Experimental results demonstrate that the proposed MB-TCN-TC model not only effectively copes with the imbalanced data and missing value issues commonly

seen in EHR datasets but also captures the temporal correlation and complicated interactions among medical variables in the longitudinal clinical records, yielding superior prediction performance compared to existing methods. Specifically, our MB-TCN-TC model provides AUROC and AUPRC scores of 0.949 and 0.793 respectively, achieving an improvement of 6.27% on AUROC, 11.85% on AUPRC, and 19.3% on F1 score compared with the traditional TCN model.
Index Terms: Diabetic retinopathy, Electronic health record, Tensor data completion, Temporal convolutional network, Multi-branching outputs, Missing value, Imbalanced data

4.1 Introduction

Diabetic retinopathy (DR), caused by damage to the blood vessels in the tissue of the retina, is a microvascular complication of diabetes. It is a leading cause of vision loss among working-aged people around the world (Yau et al., 2012). According to the American Diabetes Association (ADA), almost all type 1 patients and more than 60% of type 2 patients have varying degrees of retinopathy after 20 years of diabetes (Association, 2002a). Vision loss can be prevented if DR is diagnosed at an early stage. Unfortunately, many diabetic patients are unaware of their potential risk of DR because DR is asymptomatic at the early stage. Thus, they miss the most effective time window to halt the progression of DR, because vision loss or blindness is inevitable (Chen et al., 2022). Additionally, although ADA suggests ophthalmic examinations in the annual DR screening guidelines, the compliance rates are below expectation, which results in 19% of DR patients being undiagnosed (Yau et al., 2012; Chen et al., 2022). Furthermore, the shortage of expensive medical devices and experts for ophthalmic exams also limits the availability of DR screening, especially in rural and remote areas. As such, reliable and cost-effective DR detection tools are urgently needed to facilitate the screening of DR.

With the rapid development of medical information technology, we now live in an era of data explosion where a large amount of data are readily available and accessible in the clinical environment (T. Denton, 2023; Alagoz et al., 2021; Yao et al., 2017; Yao and Yang, 2016). This data-rich environment provides an unprecedented opportunity for developing automated tools for disease diagnosis. For example, many data-driven methods have been developed to support clinical decision-making by leveraging electronic health record (EHR) data (Wang and Yao, 2021; Liu et al., 2022; Shickel et al., 2017) , which contain rich health information including patient demographics, vital signs, and lab tests. However, EHR data are generally complexly structured with a high level of heterogeneity. Fully utilizing EHR data for reliable data-driven DR diagnosis poses the following challenges.

(1) **Poor data quality**. Real-world EHRs suffer from the issues of imbalanced data and missing values. An imbalanced dataset with skewed distribution between two classes results from the fact that the number of negative patients (without a specific disease) is generally much larger than that of positive patients. Disease detection tasks are often formulated as binary classification problems, where each training sample is from two classes (e.g., positive vs. negative). With imbalanced training data, the majority class will dominate the classifier, yielding unsatisfactory performance in predicting the minority class. Missing value problems occur due to the heterogeneity in clinical needs and lack of documentation (Janssen et al., 2010; Li et al., 2021a; Groenwold, 2020). For example, patients often do not take all types of lab tests at every clinic visit. A patient may do Complete Blood Count test to measure the count of red and white blood cells at one visit, while he/she may take Comprehensive Metabolic Panel test to measure the level of Alanine Aminotransferase (ALT) and Blood Urea Nitrogen at the next visit. Such data quality issues will result in significant biases in decision-making if left unaddressed, constraining the clinical value of EHRs.

(2) **Multivariate longitudinal records**. EHR datasets are commonly associated with multivariate longitudinal records (or sequences), especially for patients

with chronic diseases such as DR. The medical variables usually not only have intra-sequence correlation (i.e., autocorrelation within one individual sequence) but also have inter-sequence correlation (i.e., the correlation between different sequences). For example, the observation of ALT of one DR patient at a given time stamp is correlated with the observation at the nearby time step. On the other hand, the ALT could also be correlated with the observations of Aspartate Aminotransferase (AST) at the same or different time steps. Note that EHRs present a wealth of information on the dynamic evolution of medical variables for large populations. This leads to a new three-way tensor form of data with patients $\times$ variables $\times$ sequences, as opposed to the table form of predictor and response variables in traditional predictive modeling. Novel machine learning approaches are urgently needed to effectively handle such complex data structure for disease prediction.

In this paper, we propose a Multi-Branching Temporal Convolutional Network with Tensor Data Completion (MB-TCN-TC) to model multivariate longitudinal EHRs for reliable DR prediction by addressing both the imbalanced data and missing value issues. We first utilize the CANDECOMP/PARAFAC (CP) decomposition to complete the tensor data for missing value imputation by simultaneously accounting for the multidimensional correlations between lab test variables, patients, and time sequences. Second, to further account for the imputation uncertainty, we propose to add the missing value masks as another input stream to our predictive model to address the possible discrepancy between the imputed values and true observations. Third, we adapt the multi-branching TCN framework (Wang and Yao, 2021) to handle the imbalanced data and model both the intra- and inter-sequence correlations in the longitudinal EHR data for DR prediction. We evaluate the proposed MB-TCN-TC on a large real-world clinical dataset. Experimental results demonstrate that the proposed MB-TCN-TC model achieves better performances for DR detection compared with existing methods.

4.2 Research Background

4.2.1 DR screening methods in current clinical practice

The most common screening method for DR is a dilated comprehensive eye examination (DCEE) performed by an ophthalmologist or optometrist (Association, 2002b). According to the guidelines of diabetes care by ADA (ElSayed et al., 2023), adults with diabetes should have an initial DCEE within 5 years after the onset of type 1 diabetes or at the diagnosis time of type 2 diabetes, and have subsequent exams every 1-2 years if there is no evidence of DR, or at least annually if any level of DR is present at the initial exam. During the exam, doctors apply eye drops to enlarge patients' pupils so that they can examine patients' retinas with devices such as a slit lamp. Within 4-6 hours after dilation, the patient's vision may be blurry and the patient is advised not to drive. This causes a significant inconvenience, especially for patients living in rural areas that are a few hours away from the ophthalmologist or optometrist's clinic.

An alternative screening method for DR is fundus photography. Trained clinicians take fundus pictures of diabetic patients. The digital color fundus photographs of the retina are then evaluated by doctors or certified technicians either at point-of-care or remotely. This diagnosis process requires experts to manually inspect the fundus pictures and evaluate the presence of lesions associated with the vascular abnormalities, which is time-consuming and depends heavily on the clinicians' domain knowledge.

4.2.2 Machine Learning for Healthcare Data

The fast-growing healthcare data presents remarkable prospects for data-driven scientific knowledge discovery and clinical decision support. The utilization of machine learning in healthcare data analysis has witnessed exponential growth in recent years. Numerous studies have demonstrated the potential transformative

impact of machine learning algorithms in deciphering complicated healthcare data including medical images and EHRs to improve the precision of diagnostic and treatment strategies (T. Denton, 2023; Holzinger et al., 2023).

Medical imaging such as Magnetic Resonance Imaging, Computed Tomography, and ophthalmoscopy enables non-intrusive assessment of the inner structures of the body for disease diagnosis. A substantial body of research has been conducted to develop effective machine learning methods to analyze the imaging data (Calisto et al., 2022a; Sivaraman et al., 2023; Gleeson et al., 2023; Calisto et al., 2023; Xie and Yao, 2023; Setzer et al., 2020). In particular, Convolutional Neural Networks (CNNs) have gained great interest in the field of medical image analysis, largely attributable to their unique ability to discern intricate structural relationships among neighboring pixels in images (Tajbakhsh et al., 2016). For example, Sanghvi. et al. (Sanghvi et al., 2023) developed a CNN model for the detection of COVID and Pneumonia using Chest X-ray images. Calisto et al. (Calisto et al., 2022b) and Dontchos (Dontchos et al., 2021) demonstrated the promising potential of CNN models for breast cancer diagnosis using breast images.

The EHR is an electronic version of a patient's medical information over time including demographics, medications, vital signs, laboratory data, etc. A variety of statistical and machine learning models including logistic regression, support vector machine (SVM), random forest, and deep neural networks (DNNs) have been developed to investigate the EHRs with applications in data-driven prediction of numerous diseases such as sepsis (Wang and Yao, 2021), diabetes (Zhu et al., 2020), and myocardial infarction (Wang et al., 2022). A comprehensive review on machine learning of healthcare data can be referred to (Shickel et al., 2017; Xiao et al., 2018)

4.2.3 Existing Work on Data-Driven DR Prediction

In the past decade, a variety of machine-learning models have been developed to analyze fundus images for automated DR detection. For example, Sopharak et al.

(Sopharak et al., 2010) proposed a machine learning model for exudate detection in retinal images of diabetic patients. Specifically, they first leveraged ophthalmologists' expertise to perform feature extraction and selection. Then, an SVM classifier was built based on the selected features for DR detection. Roychowdhury et al. (Roychowdhury et al., 2013) evaluated AdaBoost, k-nearest neighbor (KNN), SVM, and Gaussian Mixture Model for classifying retinopathy lesions from fundus images. Reza et al. (Reza et al., 2011) proposed a machine learning algorithm based on marker-controlled watershed segmentation to detect optic disc, exudates, and cotton wool spots in fundus images and further identified DR. In general, most traditional machine learning methods still depend heavily on manual feature extraction and selection, which is not only a trial-and-error labor-intensive process but also depends on human expertise (Sopharak et al., 2010; Hasan et al., 2021; Das et al., 2022).

Deep learning models have been proposed to analyze retinal images. The primary advantage of DNNs over traditional machine learning methods is that DNNs have the ability to learn complex representations of the input data without the need for explicit feature engineering (LeCun et al., 2015). DNN-based features have been proved to be more informative than handcrafted features for data-driven disease detection (Xie and Yao, 2022a; Wang et al., 2023c; Xie and Yao, 2022b; Zheng et al., 2020). As a result, DNN models, especially CNNs, have been developed to analyze retinal images for DR prediction with better performances than conventional machine learning methods. For example, Ting et al. (Ting et al., 2017) proposed a deep learning model for screening DR and other eye diseases from retinal images. Wang et al. (Wang et al., 2020) developed a multi-channel generative adversarial network to investigate retinal images for DR diagnosis. Comprehensive reviews of data-driven DR diagnosis based on retinal images can be found in (Hasan et al., 2021; Das et al., 2022).

Despite the impressive performances of DNNs in fundus photography recognition, expensive imaging devices (e.g., digital fundus cameras) and ophthalmic imaging skills are still needed, limiting the use of this approach to well-funded healthcare providers with trained technicians. More cost-effective screening tools based on EHR

data, which is widely available and accessible to all healthcare settings, have been developed such as Cox's proportional hazard model (Semeraro et al., 2011), decision trees-based methods (Ogunyemi and Kermah, 2015), ensemble models (Piri et al., 2017), and extreme gradient boosting (XGBoosting) (Wang et al., 2021). However, many of the methods, e.g., decision trees-based models and gradient boosting-based models, lack the ability to model the dynamic trajectory of the DR disease. Our prior work (Chen et al., 2022) has shown that the longitudinal information of the disease dynamics is conducive to data-driven disease prediction. Note that Cox proportional hazard models generally assume there is a linear relationship between the medical variables and proportional hazard (Emmert-Streib and Dehmer, 2019). This linearity assumption often is not valid for real-world EHR data, which generally describes the nonlinear and non-stationary disease dynamics.

Table D.16 summarizes the limitations in existing machine learning methods for DR prediction using EHRs. Specifically, data-driven DR prediction based on EHR data suffers from the problem of missing values and extremely imbalanced data as we discussed in Section I. In the literature, there are various studies on the imputation of missing medical data, which generally estimate the missing values by investigating the data structure and exploiting the correlation patterns. Common imputation techniques include case deletion, mean imputation, interpolation and extrapolation, and advanced machine-learning-based imputation (Cismondi et al., 2013; Imani et al., 2019). However, most of those methods focus on addressing the missing value issue in matrix-form data but are not directly applicable to capture the multi-dimensional correlation among patients, variables, and sequences in longitudinal EHR, and thus are not well-suited for tensor-form data imputation.

Similarly, various methods have been proposed to handle the imbalanced data issue, such as random under/over-sampling, informed under-sampling, and synthetic minority over-sampling (SMOTE) (He and Garcia, 2009; Krawczyk, 2016; Mani and Zhang, 2003; Chawla et al., 2002). However, under/over-sampling methods suffer from huge information loss or over-fitting problems. Informed under-sampling (i.e.,

KNN-based imputation), and SMOTE are not generally applicable for longitudinal data analysis because, in those methods, the similarity calculation is based on the Euclidean distance, which may not be able to capture the true closeness between longitudinal sequences (He and Garcia, 2009; Yao, 2021). Therefore, novel analytical models are urgently needed to effectively learn from complexly structured longitudinal EHRs for reliable DR prediction.

4.3 Research Methodology

This section presents the proposed MB-TCN-TC framework for DR prediction from longitudinal EHRs. Suppose there are N patients indexed by $p \in \{p_1, \ldots, p_N\}$ and each patient is described by the medical records denoted as $(\boldsymbol{v}^p, \boldsymbol{t}^p, y^p)$: $\boldsymbol{v}^p \in \mathbb{R}^M$ denotes the medical variables; $\boldsymbol{t}^p \in \mathbb{R}^T$ is the time point when medical variables are observed; $y^p \in \{0, 1\}$ is the label with $y^p = 1$ indicating that patient p is diagnosed with DR and $y^p = 0$ otherwise. As shown in Fig. C.16(a), the data for all patients can be summarized as a tuple $\mathcal{D} = (\boldsymbol{X}, \boldsymbol{y})$, where $\boldsymbol{X} \in \mathbb{R}^{N \times M \times T}$ is 3-way tensor data of size $N \times M \times T$ with N, M, T denoting the number of patients, variables, and time stamps respectively, and $\boldsymbol{y} \in \mathbb{R}^N$ is the label vector. As illustrated in Fig. C.16(b), the missing values will first be imputed with tensor completion. Second, missing value masks will be generated to capture the missingness patterns, which will be combined with the imputed tensor to generate two streams of balanced subsets by under-sampling the majority class, as shown Fig. C.16(c). Third, the balanced datasets are fed into the MB-TCN to train the network and further predict the DR probability for new patients as shown in Fig. C.16(d).

4.3.1 Tensor Completion for Missing Data Imputation

Tensor factorization has been proven to be an efficient method to study the latent structure in tensor-form data for missing value imputation (Cichocki et al., 2007).

The critical idea of the tensor-factorization-based imputation is to jointly consider the multi-dimensional correlations among medical variables, time, and patients in a compute-efficient way. The CANDECOMP/PARAFAC (CP) decomposition (Cichocki et al., 2007) is a widely used approach to study tensor data by projecting it into a linear combination of rank-1 tensors as shown in Fig. C.17. Specifically, suppose the rank of 3D tensor $\boldsymbol{X}$ (i.e., medical records) is R, where $R \leq \min\{N, M, T\}$ and the element of $\boldsymbol{X}$ is denoted as x_{pvt}, where $p \in \{1, \ldots, N\}$, $v \in \{1, \ldots, M\}$, and $t \in \{1, \ldots, T\}$. Then, the CP decomposition of a complete tensor $\boldsymbol{X}$ is represented as: $x_{pvt} \triangleq [\![\boldsymbol{U}, \boldsymbol{H}, \boldsymbol{S}]\!]_{pvt} = \sum_{r=1}^{R} u_{pr} h_{vr} s_{tr}$, where u_{pr}, h_{vr}, and s_{tr} are elements of the factor matrices $\boldsymbol{U} \in \mathbb{U}^{N \times R}$, $\boldsymbol{H} \in \mathbb{R}^{M \times R}$, and $\boldsymbol{S} \in \mathbb{R}^{T \times R}$. As such, the high-dimensional tensor data $\boldsymbol{X}$ is characterized by the low-dimensional latent space defined by the factor matrices $\boldsymbol{U}$, $\boldsymbol{H}$, and $\boldsymbol{S}$. However, such a decomposition procedure is not viable if tensor $\boldsymbol{X}$ is incomplete. With missing data, CP decomposition is achieved by minimizing the mean squared error between the observed data and reconstructed tensor as:

$$J_{\boldsymbol{W}}(\boldsymbol{U}, \boldsymbol{H}, \boldsymbol{S}) \triangleq \frac{1}{2} \|\boldsymbol{X} - [\![\boldsymbol{U}, \boldsymbol{H}, \boldsymbol{S}]\!]\|_{\boldsymbol{W}}^2 = \frac{1}{2} \sum_{p=1}^{N} \sum_{v=1}^{M} \sum_{t=1}^{T} \left\{ w_{pvt} \left(x_{pvt} - \sum_{r=1}^{R} u_{pr} h_{vr} s_{tr} \right)^2 \right\} \tag{4.1}$$

where $\boldsymbol{W}$ is a 3D indicator tensor with $w_{pvt} = 0$ if x_{pvt} is missing and $w_{pvt} = 1$ otherwise.

We employ the CP-weighted optimization algorithm (Acar et al., 2011) to solve this minimization problem in Eq. (4.1) and estimate the factor matrices $\boldsymbol{U}$, $\boldsymbol{H}$, and $\boldsymbol{S}$. Specifically, at each iteration, we compute $\boldsymbol{X}_1 = \boldsymbol{W} * \boldsymbol{X}$ and $\boldsymbol{X}_2 = \boldsymbol{W} * [\![\boldsymbol{U}, \boldsymbol{H}, \boldsymbol{S}]\!]$ (where $*$ denotes element-wise multiplication) and further derive the gradient of the objective function in Eq. (4.1) by computing the partial derivatives of $J_{\boldsymbol{W}}(\boldsymbol{U}, \boldsymbol{H}, \boldsymbol{S})$

with respect to each element, i.e., u_{pr}, h_{vr}, s_{tr}, of the factor matrices:

$$\begin{aligned}
\frac{\partial J_{\boldsymbol{W}}(\boldsymbol{U},\boldsymbol{H},\boldsymbol{S})}{\partial \boldsymbol{U}} &= (\boldsymbol{X}_{2(U)} - \boldsymbol{X}_{1(U)})(\boldsymbol{H} \odot \boldsymbol{S}) \\
\frac{\partial J_{\boldsymbol{W}}(\boldsymbol{U},\boldsymbol{H},\boldsymbol{S})}{\partial \boldsymbol{H}} &= (\boldsymbol{X}_{2(H)} - \boldsymbol{X}_{1(H)})(\boldsymbol{U} \odot \boldsymbol{S}) \\
\frac{\partial J_{\boldsymbol{W}}(\boldsymbol{U},\boldsymbol{H},\boldsymbol{S})}{\partial \boldsymbol{S}} &= (\boldsymbol{X}_{2(S)} - \boldsymbol{X}_{1(S)})(\boldsymbol{U} \odot \boldsymbol{H})
\end{aligned} \tag{4.2}$$

where $\boldsymbol{X}_{1(I)}$ and $\boldsymbol{X}_{2(I)}$ are the corresponding matrices resulted from the matricization (i.e., flattening or unfolding) of tensors $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ (Kolda and Bader, 2009; Sidiropoulos et al., 2017): $\boldsymbol{X}_{1(U)}$, $\boldsymbol{X}_{2(U)} \in \mathbb{R}^{N \times MT}$, $\boldsymbol{X}_{1(H)}$, $\boldsymbol{X}_{2(H)} \in \mathbb{R}^{M \times NT}$, and $\boldsymbol{X}_{1(S)}$, $\boldsymbol{X}_{2(S)} \in \mathbb{R}^{T \times NM}$. The symbol $\odot$ denotes the Khatri-Rao product, which generates a matrix of sizes $NM \times R$ for two matrices $\boldsymbol{U} \in \mathbb{R}^{N \times R}$ and $\boldsymbol{H} \in \mathbb{R}^{M \times R}$ according to:

$$\boldsymbol{U} \odot \boldsymbol{H} = [\boldsymbol{u}_1 \otimes \boldsymbol{h}_1 \quad \boldsymbol{u}_2 \otimes \boldsymbol{h}_2, \cdots, \boldsymbol{u}_R \otimes \boldsymbol{h}_R] \tag{4.3}$$

where $\otimes$ denotes Kronecker product of two vectors:

$$(\boldsymbol{u}_i \otimes \boldsymbol{h}_i)^T = [u_{i,1}\boldsymbol{h}_i^T \quad u_{i,2}\boldsymbol{h}_i^T \quad \cdots \quad u_{i,N}\boldsymbol{h}_i^T] \tag{4.4}$$

Given the computed gradients, the nonlinear conjugate gradient algorithm based on Hestenes-Stiefel (Nocedal and Wright, 1999) updates and the More-Thuente line search method for step size (Moré and Thuente, 1994) is engaged to solve the optimization problem and estimate the reconstructed tensor as $\hat{\boldsymbol{X}} = [\![\hat{\boldsymbol{U}}, \hat{\boldsymbol{H}}, \hat{\boldsymbol{S}}]\!]$. As such, the original incomplete tensor $\boldsymbol{X}$ is imputed as $\bar{\boldsymbol{X}} = \boldsymbol{W} * \boldsymbol{X} + (\boldsymbol{1} - \boldsymbol{W}) * \hat{\boldsymbol{X}}$. Algorithm 2 summarizes the detailed procedure for tensor-based missing value imputation.

Algorithm 2: Tensor-based missing value imputation

Data: $\boldsymbol{X}$, $\boldsymbol{W}$, N_m=max iterations, ϵ=error tolerance
Result: Imputed tensor $\bar{\boldsymbol{X}}$

Initialization: $\hat{\boldsymbol{X}}_0 = [\![\boldsymbol{U}_0, \boldsymbol{H}_0, \boldsymbol{S}_0]\!]$, $k \leftarrow 0$, $\boldsymbol{x}_0 = \text{Vect}(\boldsymbol{U}_0, \boldsymbol{H}_0, \boldsymbol{S}_0) \in \mathbb{R}^{R(N+M+T)\times 1}$

Evaluate : $\boldsymbol{g}_0 = \nabla J_{\boldsymbol{W}}(\boldsymbol{U_0}, \boldsymbol{H_0}, \boldsymbol{S_0}) \in \mathbb{R}^{R(N+M+T)\times 1}$, $\boldsymbol{p}_0 \leftarrow -\boldsymbol{g}_0$, $J_0 = \frac{1}{2}\|\boldsymbol{X} - \hat{\boldsymbol{X}}_0\|^2_{\boldsymbol{W}}$

while $k < N_m$ *or* $\frac{|J_{k+1}-J_k|}{J_k} > \epsilon$ **do**

  Compute α_k based on More-Thuente line search

  Set $\boldsymbol{x}_{k+1} \leftarrow \boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k$, then update $\hat{\boldsymbol{X}}_{k+1} = [\![\hat{\boldsymbol{U}}_{k+1}, \hat{\boldsymbol{H}}_{k+1}, \hat{\boldsymbol{S}}_{k+1}]\!] = \text{Tensorize}(\boldsymbol{x}_{k+1})$

  Evaluate $\boldsymbol{g}_{k+1} = \nabla J_{\boldsymbol{W}}(\hat{\boldsymbol{U}}_{k+1}, \hat{\boldsymbol{H}}_{k+1}, \hat{\boldsymbol{S}}_{k+1})$

  Compute β^{HS}_{k+1} based on Hestenes-Stiefel updates

  $\boldsymbol{p}_{k+1} \leftarrow -\boldsymbol{g}_{k+1} + \beta^{HS}_{k+1}\boldsymbol{p}_k$

  Compute $J_{k+1} = \frac{1}{2}\|\boldsymbol{X} - \hat{\boldsymbol{X}}_{k+1}\|^2_{\boldsymbol{W}}$

  $k \leftarrow k + 1$

end

Tensorize $\boldsymbol{x}_k$ to $\hat{\boldsymbol{X}} \in \mathbb{R}^{N\times M\times T}$

Return: $\bar{\boldsymbol{X}} = \boldsymbol{W} * \boldsymbol{X} + (\boldsymbol{1} - \boldsymbol{W}) * \hat{\boldsymbol{X}}$

4.3.2 Missingness-informed TCN

TCN is an effective tool to capture both temporal correlations in longitudinal data (i.e., intra-sequence correlation) and heterogeneous patterns between different features (i.e., inter-sequence correlation) (Bai et al., 2018). Here, we propose a missingness-informed TCN to process the imputed tensor data $\bar{\boldsymbol{X}}$ to capture critical temporal and variable correlation while being aware of missingness patterns for better prediction. Specifically, as shown in Figure C.18, there are four key features in this model:

(1) **Multi-channel input**: TCNs treat the multi-variable longitudinal data as multi-channel sequences and apply trainable filters to process such multi-channel input at the same time. Fig. C.19 shows the fundamental procedure of how TCNs handle multi-variable longitudinal data. Specifically, each variable in the time series is considered as an individual input channel (e.g., 2 channels in the figure means there

are 2 variables in the input). TCNs then use convolutional filters to process temporal data where the filter size determines how many time points the network examines in one operation, and the channel number of the filter is the same as the number of input channels. As the filters are applied across the input channels, the network can learn from multiple interrelated sequences simultaneously, effectively capturing the intersequence correlations.

(2)**Dilated causal convolution:** Figure C.18(c) shows the procedure of dilated causal convolution. Specifically, TCNs achieve the dilated causal convolution with filters $f_e : \{0, 1, \ldots, k_e - 1\}$ over longitudinal input as

$$Conv(t) = (\boldsymbol{x}_{p,v} *_d f_e)(t) = \sum_{\tau=0}^{k-1} f_e(\tau) \cdot \boldsymbol{x}_{p,v}(t - \tau \cdot d) \tag{4.5}$$

where $\boldsymbol{x}_{p,v}$ denotes the observations of variable v for patient p over time, $*$ represents the convolutional operation, d, f_e, k_e are the dilated factor, filters, and kernel size, respectively. The causal convolution is realized by $t-\tau \cdot d$, which ensures the direction of convolutional operation is toward the past and the network output at the current time t is not a function of any future information after time t. By tuning dilated factor d and kernel size k_e, TCNs have the ability to capture both local- and long-range correlations in the temporal domain.

(3) **Residual block:** Residual block (Fig. C.18(b))(He et al., 2016) is another critical attribute for effective longitudinal data modeling, which is leveraged to solve the problem of gradient degradation when the network goes deeper for modeling complexly structured data. The output of residual block is $o(\boldsymbol{x}_v^p) = \sigma(\mathcal{F}(\boldsymbol{x}_v^p) + \boldsymbol{x}_v^p)$, where $\sigma(\cdot)$ is the activation function, and $\mathcal{F}$ is a series of operations including the dilated causal convolution, batch normalization, nonlinear activation, and dropout. Such residual operation enables the network to focus on the residual of the identity mapping to cope with the gradient degradation problem in DNN training (He et al., 2016). Our proposed model consists of two layers of causal convolution and ReLU activation within a residual block. In addition, we implement batch normalization

for each causal convolutional layer to re-center and re-scale the input (or the output of previous layer), which helps to improve the speed and stability of network training (Ioffe and Szegedy, 2015). Moreover, a dropout layer is added for each causal convolution to avoid the possible over-fitting problem (Srivastava et al., 2014).

(3) **Missing value masks:** To account for the possible discrepancy between the imputed values and true observations, the missing value masks, i.e., the 3D indicator tensor $\boldsymbol{W}$, will serve as another input of the TCN model as shown in Fig. C.18(a). This will enable the network to learn the missingness pattern, i.e., correlation between non-missing and missing values, for mitigating the impact of possible imputation error incurred from the imputation procedure on the prediction performance.

4.3.3 Multi-Branching Outputs

In this subsection, we adapt the multi-branching (MB) architecture (Wang and Yao, 2021) to cope with the imbalanced issue in the DR dataset. The MB network structure consists of a core TCN and an MB output layer. The core TCN can be treated as a feature extractor or backbone of the network, and the MB layer is the output classifier. Existing literature has shown that the classifier component of the network suffers more from the imbalanced data issue, while the backbone portion is less susceptible during training (Zhou et al., 2020; Huo et al., 2022). As such, we propose to train the core TCN with the whole dataset and use a balanced sub-dataset to train each branching output in the MB layer to alleviate the impact of imbalanced data on the classifier.

We denote the original dataset as $\mathcal{D}$, which consists of the minority class, $\mathcal{D}_P$ (i.e., DR patients), and majority class $\mathcal{D}_N$ (i.e., non-DR patients). As shown in Fig. C.16(c), N_b balanced datasets are created from $\mathcal{D}$ by under-sampling the non-DR class $\mathcal{D}_N$ as $\mathcal{D}_i = \{\mathcal{D}_N^i, \mathcal{D}_P\}$, where $\mathcal{D}_N = \cup_{i\in 1,\ldots,N_b}\mathcal{D}_N^i$. The same operation will also be conducted for the missing value masks. Furthermore, an MB layer with N_b outputs is added at the end of the core TCN to process the N_b balanced sub-datasets. Fig. C.18 shows the details of our MB network architecture. Note that each balanced

dataset consists of the imputed EHRs and missing value masks, which are processed by two parallel sequential residual blocks. Then, the results of two parallel streams are concatenated and further flow to the fully connected layer. In the end, the MB output layer will generate N_b predicted DR probabilities based on the outputs of the fully connected layer. As such, each balanced dataset serves as the training data for the corresponding branch of the output layer, and the core TCN structure will be optimized by all the N_b training subsets. Specifically, the optimization will be achieved by minimizing the binary cross-entropy loss:

$$\mathcal{L}(\boldsymbol{\theta}; \boldsymbol{X}, \boldsymbol{W}) = -\sum_{p=1}^{N} \sum_{i=1}^{N_b} \mathcal{I}(p \in \mathcal{D}_i) \Big(y_p \log(\hat{P}_i(\boldsymbol{\theta}; \boldsymbol{x}^p, \boldsymbol{w}^p)) + (1 - y_p) \log(1 - \hat{P}_i(\boldsymbol{\theta}; \boldsymbol{x}^p, \boldsymbol{w}^p)) \Big)$$

where $\boldsymbol{\theta}$ represents the model parameters, $\mathcal{I}(p \in \mathcal{D}_i)$ denotes the indicator function indicating whether patient p belongs to subset $\mathcal{D}_i$, and $\hat{P}_i(\boldsymbol{\theta}; \boldsymbol{x}^p, \boldsymbol{w}^p)$ is the prediction of DR probability provided by i-th branching output for the input signal $\boldsymbol{x}^p$ and the corresponding mask $\boldsymbol{w}^p$ of patient p. The final prediction of DR probability for patient p is the mean of the N_b predicted probabilities:

$$\hat{P}(\boldsymbol{x}^p) = \sum_{i=1}^{N_b} \hat{P}_i(\boldsymbol{\theta}; \boldsymbol{x}^p, \boldsymbol{w}^p)/N_b \tag{4.6}$$

4.4 Experimental Design and Results

4.4.1 Data Source and Data Extraction

The dataset we use in this study is obtained from the 2018 Cerner Health Facts® data warehouse, one of the largest Health Insurance Portability and Accountability Act (HIPAA)–compliant databases in the U.S. storing de-identified clinical data of more than 63 million patients (Chen et al., 2022). This database contains comprehensive clinical information including patient demographics, hospital visits,

diagnoses, procedures, medication prescriptions, vital signs, lab tests, etc., providing an unprecedented opportunity for data-driven diagnosis of DR. The labels are derived by examining whether one or more DR diagnosis codes exist and "1" represents a DR patient while "0" represents a non-DR diabetic patient.

We select independent variables based on literature (Yau et al., 2012; Wang et al., 2020) and include 21 routine blood tests for diabetic patients, 5 comorbidity variables (i.e., neuropathy, nephropathy, hypertension, obesity, and coronary artery disease), 3 demographic variables (i.e., gender, age, and race), and the duration of diabetes, which is measured in years from the first diabetic diagnosis to the beginning of the prediction window. Note that all 21 blood tests are subject to missing values, the missing rates of which are shown in Fig. C.20. The final dataset consists of 414,199 diabetic patients with a 3% (12,590) DR positive rate. We randomly partition the dataset into a training set (70%) for model training, a validation set (10%) for hyperparameters tuning, and a testing set (20%) for performance evaluation.

4.4.2 Experimental Design

As shown in Fig. C.21, the performance of our MB-TCN-TC will be compared with TCN + different imputation techniques and TCN + imbalanced data handling methods. Specifically, our MB-TCN-TC method will be benchmarked with TCN + carry forward imputation (TCN+CF), TCN + mean imputation (TCN+mean), TCN + under-sampling, TCN + over-sampling, and pure TCN. Additionally, we will evaluate the number of branching outputs (i.e., 5, 10, and 20 branches) on the performance of MB-TCN-TC. The performance of both our model and benchmarking methods will be evaluated according to 6 metrics: Confusion Matrix, Area Under Receiver-Operating-Characteristic Curve (AUROC), Area Under Precision-Recall Curve (AUPRC), Recall, Precision, and F1 score. The ROC curve characterizes the tradeoff between the True Positive Rate (TPR) and the False Positive Rate (FPR), and the PRC curve captures the relationship between Precision and Recall. Hence,

AUROC and AUPRC serve as the overall metrics across all thresholds, quantifying the overall prediction performance.

Our model is built using Pytorch leveraging the computational power of an NVIDIA RTX A4500 GPU. In the CP decomposition-based imputation, we select the rank of the tensor $R = 25$, maximum number of iterations $N_m = 10,000$, and the error tolerance $\epsilon = 10^{-16}$. The hyperparameters of our neural network are selected by empirical fine-tuning. Specifically, our final network model consists of 3 residual blocks and 10 branching outputs (i.e., $N_b = 10$); the number of filters is 64 and kernel size is $k_e = 3$ in the causal convolution; the dilation factor d is selected as 1 for the first and 2 for the second dilated convolutional layer in each residual block; the dropout rate is 0.5; we select Adam optimizer with an initial rate of 0.0002 and momentum values of $\beta_1 = 0.9$, $\beta_2 = 0.999$ during the network training.

4.4.3 Experimental Results

Fig. C.22 (a) and (b) illustrate the ROC and PR curves of our MB-TCN-TC with 10 branching outputs (MB-TCN-TC-10) and other methods. Note that a good model in the ROC space is located towards the top-left corner, indicating a low FPR and high TPR. According to Fig. C.22(a), our method, i.e., MB-TCN-TC-10 with mask, dominates the ROC space with the highest AUROC of 0.949. On the other hand, in the PR space, a good classification model should achieve both higher Recall and Precision, towards the top-right corner. As shown in Fig. C.22(b), the MB-TCN-TC-10 with mask outperforms other methods, yielding the highest AUPRC of 0.793. Notably, our MB-TCN-TC-10 method achieves an improvement of 6.27% on AUROC and 11.85% on AUPRC compared with the pure TCN without the tensor-based imputation and MB architecture (with AUROC of 0.893 and AUPRC of 0.709).

Table D.17 further shows the performance comparison in terms of Confusion Matrix along with Recall, Precision, and F1 metrics. Specifically, our MB-TCN-TC-10 yields the highest true positive rate with 1,880 DR patients correctly identified,

indicating a strong Recall in detecting DR. Our model also generates a lower false negative with 680 patients, which is critical in medical diagnostics as it reduces the risk of missing a potential DR diagnosis. Moreover, our model maintains the highest overall Precision (0.723) with the smallest false positives (i.e., 722 patients). Hence, our model makes correct DR prediction more often than other benchmark methods. Additionally, the highest Recall of 0.734 suggests that our MB-TCN-TC-10 has the best capability to identify all DR patients. Our model also yields the highest F1 score of 0.728, achieving the best balance between Recall and Precision.

Table D.18 shows AUROC and AUPRC scores of our MB-TCN-TC and other imputation methods commonly used in current practice with and without (w/o) missing-value mask. Note that all the network models compared in Table D.18 contain a classifier layer with 10 branching outputs. Our MB-TCN-TC with missing-value mask method provides the highest AUROC and AUPRC performance scores of 0.949 and 0.793 respectively, as compared with MB-TCN+CF with missing-value mask (with AUROC of 0.925 and AUPRC of 0.734), MB-TCN+CF w/o missing-value mask (with AUROC of 0.915 and AUPRC of 0.724), MB-TCN+mean with masks (with AUROC of 0.916 and AUPRC of 0.726), MB-TCN+mean w/o missing-value masks (with AUROC of 0.920 and AUPRC of 0.727), and MB-TCN-TC w/o missing-value mask method (with AUROC of 0.937 and AUPRC of 0.775).

Table D.19 further shows the comparison between our MB-TCN-TC-10 model and other methods in addressing the imbalanced data issue. Note that all methods compared in this table engage the tensor decomposition-based imputation and missing value masks to handle the missing value issue. Our MB-TCN-TC-10 method achieves better performance with AUROC and AUPRC scores of 0.949 and 0.793, respectively. Specifically, our MB-TCN-TC-10 achieves 1.71% and 4.89% improvements on AUROC and AUPRC respectively, compared with the under-sampling method; increases AUROC and AUPRC performance scores by 2.82% and 2.72%, respectively, compared with the pure TCN that is trained by the original

imbalanced dataset; increases the AUROC and AUPRC metrics by 1.61% and 2.85% respectively, compared to the over-sampling method.

4.5 Discussion

4.5.1 Overfitting or underfitting issues in model training

Fig. C.23 shows the values of training and validation losses of our MB-TCN-TC-10 model over 50 epochs. The training loss keeps decreasing with some random fluctuations when the epoch increases, while the validation loss first decreases and then stabilizes around a low value after about 20 epochs. A similar trend is also observed in the evolution of AUROC over epochs as shown in Fig. C.23 (b). In our experiments, we select the total epoch number as 26 to increase the computational efficiency and guarantee the network model is well trained according to Fig. C.23. Note that the validation loss (AUROC) does not further increase (decrease) as we increase the training epochs, indicating that there is no overfitting issue. This is attributable to the dropout layer in our network. Specifically, in our training procedure, we select the dropout rate as $\gamma = 0.5$. The critical idea of dropout is to randomly freeze a subset of the DNN neurons with a pre-defined probability (i.e., $\gamma = 0.5$) at each training iteration (Srivastava et al., 2014). This technique introduces stochasticity into network training to avoid the possible overfitting issue, which is equivalent to the $L2$ regularization. In fact, the dropout loss function $\mathcal{L}_{drop}$ can be written as the summation of the original loss function $\mathcal{L}(\boldsymbol{\theta}; \boldsymbol{X}, \boldsymbol{W})$ and an $L2$ regularization on the network parameters:

$$\mathcal{L}_{drop} = \mathcal{L}(\boldsymbol{\theta}; \boldsymbol{X}, \boldsymbol{W}) + \lambda \sum_{i=1}^{l} \theta_i^2 \tag{4.7}$$

where regularization parameter λ is a function of the dropout rate γ, l denotes the number of hidden layers, and θ_i stands for the weights and bias for the ith hidden layer. This regularization technique effectively avoids the overfitting issue.

Furthermore, it is worth noting that the training loss (AUROC) is larger (lower) than the validation loss (AUROC). This is also due to the use of dropout layer during model training. The dropout technique is only used in the training phase to avoid overfitting, while all neurons are active and contribute to the prediction during the validation or testing phase. In other words, $L2$ regularization (i.e., the second term on the right hand) in Eq. (4.7) is removed during the validation or testing phase, leading to a lower validation loss compared to the training loss. Additionally, because all extracted features by the hidden layers remain active for prediction during the validation phase, a higher AUROC is generated for the validation set compared with the training phase.

4.5.2 Performance comparison with existing literature

Table D.20 presents a comparison study of the performance scores between our MB-TCN-TC-10 model and traditional machine learning methods in existing literature. Note that traditional machine learning methods (i.e., random forest, XGBoost, simple neural network, logistic regression) are not directly applicable in handling longitudinal data. To facilitate a meaningful comparison, the longitudinal dataset is transformed into a non-temporal dataset by only keeping the last encounter of routine blood tests for each patient to train traditional machine learning models. According to Table D.20, our model outperforms the benchmarks across various metrics, notably better than the highest-performance benchmark, XGBoost. Specifically, the MB-TCN-TC-10 yields an improvement of 3.7% on AUROC and 8.1% in AUPRC compared to XGBoost.

Additionally, our model achieves the best Recall of 0.734 and the highest Precision of 0.723. The high Recall and Precision demonstrate that our model is capable of not

only identifying DR patients more accurately but also achieving a superior reduction in false-positive rates. Moreover, the highest F1 score of 0.728 further demonstrates our model's enhanced capability to maintain an optimal balance between Recall and Precision. Meanwhile, the MB-TCN-TC-10 provides the highest true positive rate with 1,880 DR patients correctly identified. Our model also yields the smallest false negative rate with 680 patients, which is critical in medical diagnostics as it reduces the risk of missing a potential DR diagnosis. Additionally, our model generates the smallest false positives (i.e., 722), maintaining the highest overall precision (0.723), which indicates that our model is able to make correct DR prediction more often than other benchmark methods. The superior performance of our model is due to its advanced architecture design (i.e., tensor decomposition-based imputation, missingness-informed TCN, and multi-branching output) to model multi-variates longitudinal data with missing value and imbalanced issues. In contrast, traditional machine learning methods cannot model longitudinal datasets to capture the dynamic disease trajectory, especially when dealing with missing information and imbalanced data issues, resulting in suboptimal prediction performance.

4.5.3 Discussion on the performance improvement

According to Tables D.18 and D.19, our MB-TCN-TC model effectively solved the missing value and imbalanced data issues, which are very common problems in data-driven disease detection. Specifically, given the same classification layer with 10 branching outputs, our MB-TCN-TC with missing-value mask model yields the best performance in DR identification (see Table D.18) compared with other imputation methods. This improvement is due to two unique features of our framework: (1) the tensor decomposition-based imputation is an effective method to jointly consider the multi-dimensional correlations among medical variables, time, and patients, which is conducive to reliably imputing the missing data; (2) the missing-value mask effectively

captures the missingness structure for mitigating the impact of possible imputation error on the prediction performance.

Similarly, given the same imputation method (i.e., tensor decomposition-based imputation with missing value mask), our MB-TCN-TC-10 provides the most accurate prediction performance compared to other techniques in handling imbalanced data (see Table D.19). This improvement is due to the MB-enhanced classifier layer, where each branching output is trained with a balanced dataset to mitigate the potential bias induced by the imbalanced data. Additionally, the MB architecture also combines the predictions from multiple independent classifiers according to Eq. (4.6): the branching outputs learn different mapping functions, combining which significantly lowers the variation and increases the robustness of the prediction performance.

Table D.21 further evaluates the prediction performance of the MB-TCN-TC with 5, 10, and 20 branching outputs. When increasing the MB outputs from 5 to 10, both the AUROC and AUPRC slightly improve. This is due to the fact that the imbalanced ratio in the training sub-dataset for each branching output improves from 6:1 to 3:1 when using 10 instead of 5 MB outputs. Note that there is a nonnegligible decrease in the performance scores when further increasing the branching outputs to 20. One possible reason is that the sample size of the balanced sub-dataset to train each branching output will decrease as the number of branches increases, which may lead to ineffective or insufficient training of each branching output. Additionally, increasing the branching outputs too much will also increase the model size, resulting in a more complex network that is more difficult to converge during the training.

4.5.4 Broad applications

Our predictive model is built based on EHR data, which is readily available and accessible to all healthcare settings. It does not require fundus images taken by retinal cameras, which are expensive for underserved populations. Our approach provides a cost-effective non-image-based tool for DR screening, which has a promising potential

to increase the compliance rate of recommended ophthalmic exams among asymptotic patients and overcome barriers to providing ubiquitous diabetic eye care in rural or underserved areas.

Furthermore, our approach provides effective solutions to the long-lasting challenges in modeling EHRs including incomplete datasets, imbalanced class distribution, and multi-variate longitudinal records, thereby ensuring reliable data-driven disease detection. The versatility of MB-TCN-TC extends beyond DR prediction, making it applicable for automatic prediction of a wide range of diseases based on EHRs. Notably, chronic conditions such as Alzheimer's disease, Myasthenia Gravis, and Hidradenitis Suppurativa, which typically exhibit gradual, long-term progression (Liu et al., 2014; Garg et al., 2017; Chang et al., 2022), can significantly benefit from our MB-TCN-TC approach. Our model can capture critical information about the heterogenous variable interaction and disease trajectory, which are difficult to discern through manual inspection of high-dimensional longitudinal EHR data. This capability facilitates accurate disease detection and has the potential to revolutionize chronic disease monitoring. These examples underscore the broad applicability of our research, indicating its potential for further exploration and utilization in diverse healthcare contexts.

4.5.5 Challenges in real-world clinical implementation

The model implementation in real-world clinical scenarios faces some challenges. One challenge is associated with the computational burden, which is a common issue in large-scale machine learning. The unique architecture of our MB-TCN-TC model allows it to capture intricate patterns in complexly structured EHR data. This, on the other hand, requires substantial computational resources to execute, especially when handling a large volume of EHR data. Specifically, our dataset contains 414,199 diabetic patients with multiple medical variables changing over time, resulting in approximately 217 million data points. In our study, we utilize NVIDIA RTX A4500

Graphics Processing Units (GPUs) to improve computational efficiency by parallel processing. As a result, the total training time for 50 epochs is approximately 10 hours.

Furthermore, DNN models are widely recognized to be difficult to optimize in addition to the computational burden. The objective function is generally non-convex, meaning there are multiple local optimal solutions. Traditional convex optimization algorithms and simple gradient descent methods (Bottou, 2010) often struggle with this issue, possibly ending up with a suboptimal solution. The situation worsens with many flat regions in non-convex functions, where the function value is nearly constant. The gradient in such flat regions is tiny, leading the algorithms stuck at sub-optimal solutions. In this project, we implement the ADAM algorithm (Kingma and Ba, 2014), a first-order gradient-based method that employs adaptive estimates of lower-order moments, to optimize our objective function. ADAM is an appealing choice due to its computational efficiency, minimal memory requirements, and suitability for handling large datasets or parameters. Its applicability to large-scale non-convex optimization enhances the optimization process of large DNN models.

4.6 Conclusions

This paper presents a novel method of Multi-branching Temporal Convolutional Network with tensor completion (MB-TCN-TC) to investigate multi-variate longitudinal clinical data for reliable DR identification. We first leverage the method of tensor data completion to estimate the missing values in the incomplete EHR data. Second, we adapt the multi-branching temporal neural network framework to further cope with the imbalanced issue and model the longitudinal sequences. Experimental results show that the proposed MB-TCN-TC model is effective to capture critical temporal information and complicated variable interactions in the longitudinal EHR data by accounting for both the imbalanced data and missing value issues. Our MB-TCN-TC method yields superior performance when identifying DR from longitudinal

EHR data with better AUROC and AUPRC metrics compared to existing methods. More importantly, this MB-TCN-TC framework can be generally applicable to model complex longitudinal sequences with imbalanced data and missing value issues for reliable health monitoring and diagnostics.

4.7 Acknowledgement

This research work was supported by the National Eye Institute of the National Institutes of Health under Award Number R01EY033861. We also acknowledge the Cerner Corporation and Center for Health Systems Innovation (CHSI) at Oklahoma State University for sharing the Health Facts® EHR database to support this research. Additionally, the authors acknowledge the University of Tennessee High-Performance Scientific Computing group for providing computational resources for this work.

4.8 Data Availability Statement

The data use agreement with Cerner Health Corporation prohibits authors from redistributing the data. Other researchers, who are interested in obtaining the data used in the study, may directly contact Cerner Heath Corporation for data access.

Chapter 5

MUSE-Net: Missingness-aware mUlti-branching Self-attention Encoder for Irregular Longitudinal Electronic Health Records

The era of big data has made vast amounts of clinical data readily available, particularly in the form of electronic health records (EHRs), which provides unprecedented opportunities for developing data-driven diagnostic tools to enhance clinical decision making. However, the application of EHRs in data-driven modeling faces challenges such as irregularly spaced multi-variate time series, issues of incompleteness, and data imbalance. Realizing the full data potential of EHRs hinges on the development of advanced analytical models. In this paper, we propose a novel **M**issingness-aware m**U**lti-branching **S**elf-Attention **E**ncoder (MUSE-Net) to cope with the challenges in modeling longitudinal EHRs for data-driven disease prediction. The MUSE-Net leverages a multi-task Gaussian process (MGP) with GPU acceleration for data imputation, a multi-branching architecture to address the data imbalance problem, and a time-aware self-attention encoder to account for the

irregularly spaced time interval in longitudinal EHRs. We evaluate the proposed MUSE-Net using both synthetic and real-world datasets. Experimental results show that our MUSE-Net outperforms existing methods that are widely used to investigate longitudinal signals.

Index Terms: Irregularly spaced time series, Multivariate longitudinal records, Data imputation, Imbalanced dataset, Multi-task Gaussian process, Self-attention encoder

5.1 Introduction

Rapid advancements in sensing and information technology have ushered us into an era of data explosion where a large amount of data is now easily available and accessible in the clinical environment (Yang and Yao, 2023; Yao et al., 2021; Yao and Yang, 2020). The wealth of healthcare data offers new avenues for developing automated data-driven methods for disease diagnosis. For instance, there have been growing research interests in harnessing electronic health records (EHRs) to create data-driven solutions for clinical decision support (Rajkomar et al., 2018; Wang et al., 2024). EHRs serve as digital repositories of a patient's medical information including demographics, medications, vital signs, and lab results (Yadav et al., 2018; Shickel et al., 2017; Yao et al., 2017), curated over time by healthcare providers, leading to a longitudinal database. With rich information about a patient's health trajectory, longitudinal EHRs present unique opportunities to analyze and decipher clinical events and patterns within large populations through data-driven machine learning.

However, data mining of longitudinal EHRs poses distinctive research challenges due to the observational nature of EHRs. Unlike well-defined, randomized longitudinal experiments in clinical trials that collect data on a fixed schedule and ensure high data quality, EHRs are recorded only when patients receive care or doctors provide services. The information collected and the timing of its collection are not determined by researchers, resulting in EHR data that are often complex and highly

heterogeneous (Xiao et al., 2018). Specifically, fully utilizing real-world EHRs for reliable data-driven decision-making calls upon addressing several challenges:

(1) **Irregularly spaced time series**. EHR data are often documented during irregular patient visits, leading to non-uniform time intervals between successive measurements and a lack of synchronization across various medical variables or among different patients. Traditional time series models face challenges when applied to irregular longitudinal data because they typically assume a parametric form of the temporal variables, making them difficult to effectively account for highly heterogeneous and irregular time intervals across different variables. Additionally, the widely used deep learning models such as convolution neural networks (CNNs) and recurrent neural networks (RNNs) for mining sequential or time series data are designed by assuming consecutive data points are collected at a uniform time interval. Those deep learning models do not consider the elapsed time between records and are less effective in modeling irregular longitudinal EHRs.

(2) **Incomplete datasets and imbalanced class distributions**. Real-world EHRs suffer from the issues of missing values and imbalanced data. Due to the nature of clinical practice, not all information is recorded for every patient visit, leading to incomplete datasets. Additionally, EHR data often exhibit a significantly imbalanced distribution, with certain health outcomes or characteristics being underrepresented compared to more common ones. For instance, rare diseases or adverse drug reactions (Schieppati et al., 2008; Wang et al., 2023b) may have very few instances compared to more prevalent conditions. In the existing literature, a wide array of statistical and machine learning techniques have been designed to tackle the missing value and imbalanced data issues (Emmanuel et al., 2021; He and Garcia, 2009), which, however, are less applicable in the context of modeling irregular longitudinal EHRs. The presence of missing values and imbalanced class distributions will introduce further difficulties in effective model training using longitudinal EHRs, leading to biased or inaccurate predictions if not properly addressed.

To address the multifaceted challenges presented by real-world longitudinal EHRs, this paper introduces a novel framework of **M**issingness-aware m**U**lti-branching **S**elf-Attention **E**ncoder (MUSE-Net) for data-driven disease prediction. First, multi-task Gaussian processes (MGPs) are employed for missing value imputation in irregularly sampled time series. Second, we propose to add missing value masks that record the locations of missing observations as another input stream to our predictive model, which enables the learning of correlations between non-missing and missing values for mitigating the impact of possible imputation errors incurred from the imputation procedure on the prediction performance. Third, we propose to integrate a time-aware self-attention encoder with a multi-branching classifier to address the imbalanced data issue and further classify the irregular longitudinal EHRs for disease prediction. Finally, to improve the computational efficiency, the training of both MGPs and MUSE-Net is accelerated by using GPU, ensuring efficient processing of large and high-dimensional EHR data. We evaluate our proposed framework using both simulation data and real-world EHRs. Experimental results show that our proposed method significantly outperforms existing approaches that are widely used in current practice.

5.2 Research Background

The integration of data-driven modeling methodologies and EHR data has transformed the healthcare field, which provides unprecedented opportunities for clinical decision support (Yadav et al., 2018; Landi et al., 2020; Rajpurkar et al., 2022). Extensive research has been conducted to develop data-driven models for disease identification using non-longitudinal EHRs that consist of static or cross-sectional patient health information (Banda et al., 2018; Wang et al., 2022). For example, Huang et al. (Huang et al., 2007) employed naive Bayes, decision tree, and nearest neighbor algorithms incorporating feature selection methods to determine key factors affecting type II diabetes control and further identify individuals who exhibit

suboptimal diabetes control status. Hong et al. (Hong et al., 2019a) developed a multi-class classification method to analyze clinical data in identifying patients with obesity and various comorbidities using logistic regression, support vector machine, and decision tree algorithms. A comprehensive review on machine learning of non-longitudinal EHRs can be referred to (Xiao et al., 2018; Rajkomar et al., 2019). However, those traditional methods are limited in capturing temporal patterns in patient health status over time. This limitation can result in less accurate predictions for conditions that are heavily dependent on longitudinal health trajectories (Banda et al., 2018; Das et al., 2022). Moreover, traditional machine learning approaches often require manual feature engineering, which is not only time-consuming and prone to error but also demands significant domain expertise.

With the growing availability of longitudinal EHRs, increasing interests have been devoted to developing advanced data-driven models to capture critical temporal information for disease prediction. Owing to the outstanding performance and strong capability in pattern recognition, deep learning has been widely explored to mine complexly structured data leveraging the nonlinear interactions among neurons within and across various hidden layers (Xie and Yao, 2022a; Xie et al., 2023; Xie and Yao, 2022c; Wang et al., 2023c). Advanced network architectures have been crafted for modeling time series or sequential data. For example, recurrent neural networks (RNNs) including long short-term memory (LSTM) are among the most commonly used deep learning models to analyze medical time series for various clinical tasks (Choi et al., 2017; Che et al., 2018; Chen et al., 2022). Additionally, temporal convolutional networks (TCNs) have been recognized as a robust alternative to RNNs for modeling longitudinal clinical signals (Wang et al., 2024; Wang and Yao, 2021). The unique architecture of dilated and causal 1-D convolution and residual blocks (Bai et al., 2018) allows TCNs to access a broader range of sequential information for capturing long-term temporal dependencies, thereby outperforming RNNs in various tasks involving mining temporal data (Wang et al., 2024; Wang and Yao, 2021; Rosnati and Fortuin, 2021). However, most existing RNNs and TCNs are designed

with the assumption that the longitudinal records are collected at a constant rate and require the neighboring samples to appear at fixed distances to facilitate the convolution or recurrent operations. This assumption is not valid in many real-world databases, making traditional RNNs or TCNs less effective in modeling irregular-spaced longitudinal EHRs.

To cope with the issue of irregular time intervals, many modified RNN architectures have been developed. For example, the time-aware LSTM (TLSTM) was designed to account for non-uniform sampling intervals through a time decay mechanism (Baytas et al., 2017). Che et al. (Che et al., 2018) proposed a GRU-D model for irregular sampled time series. It also integrates a decay mechanism to model the impact of missing sampled values over time, allowing the network to better capture temporal dependencies even when data points are irregular or incomplete. A comprehensive review of modified RNN architectures for irregular sampled time series can be found in (Sun et al., 2020; Weerakody et al., 2021). However, those modified RNNs assume that the impact of historical risk factors on disease prediction proportionally decays over time, which may not accurately describe complex disease trajectories. Additionally, RNNs have been widely recognized as computationally inefficient for modeling large-scale, long sequential data due to their sequential processing nature.

The self-attention encoder, a cornerstone of the transformer architecture (Vaswani et al., 2017), has revolutionized the field of natural language processing (NLP) with its unique advantages. Unlike RNNs and TCNs that process data sequentially and require uniform sampling intervals, the self-attention mechanism allows the model to weigh the importance of different parts of the input sequences relative to each other. This feature enables parallel processing and enhances the model's ability to capture long-range dependencies in irregularly spaced longitudinal signals.

Various approaches leverage the transformer/self-attention architecture to improve prediction accuracy and robustness in time series analysis. For instance, the Vision Transformer for Irregularly Sampled Time Series (ViTST) transforms

time series data into images and applies vision transformers to capture temporal dependencies effectively (Li et al., 2024). The Self-Supervised Transformer for Sparse and Irregularly Sampled Multivariate Clinical Time-Series (STraTS) employs a continuous value embedding technique to handle continuous time and variable values without the need for imputation, thus preserving fine-grained information (Tipirneni and Reddy, 2022). Another approach, the Deformable Neighborhood Attention Transformer (DNA-T), introduces deformable attention mechanisms to capture local and global dependencies in medical time series data (Huang et al., 2024). These models demonstrate significant improvements over traditional methods, particularly in handling the irregularities and sparsity inherent in many real-world datasets.

However, some drawbacks persist across these novel transformer-based architectures. The transformation of time series into images or triplets can lead to information loss and may not capture all nuances of the data. Moreover, these architectures often require substantial computational resources due to the complexity of the models, making them less practical for large-scale or real-time applications. In addition, despite the strengths of self-attention encoders/transformer architecture, it is worth noting that most standard deep learning algorithms are designed under the assumption that data is ideal, which cannot be directly applied to mine longitudinal EHR databases with missing values and imbalanced class distributions. Little has been done to design an effective framework to model real-world longitudinal EHRs by accounting for the issues of missing values, imbalanced data distribution, and irregular time intervals for robust decision-making.

5.3 Research Methodology

Suppose that there are N patients indexed by $p \in \{1, \ldots, N\}$ and we denote the dataset as $\mathcal{D} = \{(\boldsymbol{t}_p, Y_p, l_p)\}_{p=1}^{N}$. Each patient is associated with longitudinal data described by the tuple $(\boldsymbol{t}_p, Y_p, l_p)$: $\boldsymbol{t}_p = [t_{p,1}, t_{p,2}, \cdots, t_{p,T_p}]$ is the time index set for patient p; $Y_p = [\boldsymbol{y}_{p,1}, \boldsymbol{y}_{p,2}, \cdots, \boldsymbol{y}_{p,M}] \in \mathbb{R}^{T_p \times M}$ represents the values of M medical

variables at $\boldsymbol{t}_p$; l_p is the binary label with $l_p = 1$ indicating patient p is positive (e.g., with disease) and otherwise $l_p = 0$. The length of the time series for each patient is highly variable (i.e., $T_p \neq T_{p'}$, if $p \neq p'$), and the times series are often irregularly spaced (i.e., $t_{p,i+1} - t_{p,i} \neq t_{p,j+1} - t_{p,j}$, if $j \neq i$) in real-world longitudinal EHR databases. Additionally, the EHR dataset is often incomplete, and we denote the complete set of values of the M variables as $\boldsymbol{y}_p = \text{Vect}(Y_p) = [\boldsymbol{y}_{p,1}^T, \boldsymbol{y}_{p,2}^T, \cdots, \boldsymbol{y}_{p,M}^T]^T \in \mathbb{R}^{T_p M}$. Then, we define an index set that contains observed values as $I_o = \{(t,m)|$ if $[Y_p]_{t,m}$ is observed$\}$ and the corresponding values are denoted by $\boldsymbol{y}_p^o \in \mathbb{R}^{O_p}$. Similarly, we define $I_u = \{(t,m)|$ if $[Y_p]_{t,m}$ is missing$\}$ as the index set of missing values, and the missing value vector is denoted by $\boldsymbol{y}_p^u \in \mathbb{R}^{U_p}$, where $U_p + O_p = T_p M$. Fig. C.25 shows the flowchart of the proposed MUSE-Net to classify irregularly spaced and incomplete longitudinal data. Each component in the flowchart is described in detail in the following subsections.

5.3.1 Multi-task Gaussian Process (MGP) for Missing Value Imputation

A Gaussian process (GP) is a flexible non-parametric Bayesian model where any collection of random variables follows a joint Gaussian distribution Williams and Rasmussen (2006); Xie and Yao (2023); Yao (2021), which have been widely used to model complex time series data (Moor et al., 2019; Zhang et al., 2022). GP-based temporal models provide a way to determine the distribution of a variable at any arbitrary point in time, making them intrinsically capable of dealing with missing value imputations that involve irregularly spaced longitudinal data. Note that single-task GPs are limited in their ability to model correlations across multiple related tasks (i.e., different medical variables), making them less effective for modeling multi-variate EHR data. To account for the multi-variate nature, we adopt a multi-task GP (MGP) (Bonilla et al., 2007), which is an extension to single-task GPs, to capture

both variable interactions and temporal correlations for imputing missing values in irregular multi-variate longitudinal EHRs.

We denote $f_{pm}(t)$ as a latent function representing the true values of variable m for patient p at time t, and a patient-independent MGP prior is placed over the latent function $f_{pm}(t)$ with a zero mean and the covariance function as:

$$\begin{aligned} \text{cov}(f_{pm}(t), f_{pm'}(t^{'})) &= \mathcal{K}_M(m, m^{'})\mathcal{K}_t(t, t^{'}) \\ y_{pm}(t) &\sim \mathcal{N}(f_{pm}(t), \sigma_m^2) \end{aligned} \tag{5.1}$$

where $y_{pm}(t)$ is the observed value of variable m for patient p at time t, σ_m^2 is the noise term for the mth task (variable), $\mathcal{K}_M(m, m^{'})$ captures the similarities between tasks, and $\mathcal{K}_t(t, t^{'})$ is a temporal correlation function. Note that a fundamental property of time series data is that observations at neighboring points tend to be similar. As such, we use the squared exponential kernel function $\mathcal{K}_t(t, t^{'}) = \exp(-\frac{(t-t^{'})^2}{2\theta^2})$ with a lengthscale parameter θ. Hence, the prior distribution for the fully observed multivariate longitudinal records, $\boldsymbol{y}_p^o$, can be represented by:

$$\boldsymbol{y}_p^o \sim \mathcal{N}(\mathbf{0}, \Sigma_p^o), \qquad \Sigma_p^o = \mathcal{K}_M \odot \mathcal{K}_{O_p} + E \tag{5.2}$$

We formulate the kernel function Σ_p^o for the observed records as the combination of the Hadamard product (i.e., element-wise multiplication) of $\mathcal{K}_M$ and $\mathcal{K}_{O_p}$, and a noise term E: $\mathcal{K}_M$ is a $O_p \times O_p$ positive semi-definite covariance matrix over medical variables, which is parameterized by a low-rank matrix B as LBB^TL^T where $B \in \mathbb{R}^{M \times q}$, and $L \in \mathbb{R}^{O_p \times M}$ is an indicator matrix with $L_{im} = 1$ if the i^{th} observation belongs to task m and $\sum_{m=1}^{M} L_{im} = 1$; $\mathcal{K}_{O_p}$ is a $O_p \times O_p$ squared exponential correlation matrix over time with elements defined as $\mathcal{K}_{O_p}((m,t),(m',t')) = \mathcal{K}_t(t, t^{'})$ if $(m,t) \in I_o$ and $(m',t') \in I_o$; E is a noise matrix with $L\boldsymbol{\sigma}^2$ on its main diagonal ($\boldsymbol{\sigma}^2 \in \mathbb{R}^M$ and $[\boldsymbol{\sigma}^2]_m = \sigma_m^2$).

In the inference step, we impute the missing value for patient p using the posterior mean, $\mu_{\boldsymbol{y}_p^u}$:

$$\mu_{\boldsymbol{y}_p^u} = (\mathcal{K}_{M_*M} \odot \mathcal{K}_{U_pO_p})(\Sigma_p^o)^{-1}\boldsymbol{y}_p^o \tag{5.3}$$

where $\mathcal{K}_{M_*M} = L_*BB^TL^T \in \mathbb{R}^{U_p \times O_p}$ is the task covariance matrix for missing values, and $L_* \in \mathbb{R}^{U_p \times M}$, $L_{*,im} = 1$ if the i^{th} missing value belongs to task m; $\mathcal{K}_{U_pO_p}$ represents the correlation matrix between the time points of the observed and missing values. The hyperparameters in the MGP model include the lower triangular matrix B, error term $\text{Diag}(E)$, and the lengthscale θ in $\mathcal{K}_{O_p}$: $\boldsymbol{\Theta} = \{\text{Vect}(B), \text{Diag}(E), \theta\}$, which are learned by minimizing the negative log marginal likelihood given by (with derivative)

$$\begin{aligned} \mathcal{L}(\boldsymbol{\Theta}) &= \log|\Sigma_p^o| + (\boldsymbol{y}_p^o)^T(\Sigma_p^o)^{-1}\boldsymbol{y}_p^o + \frac{O_p}{2}\log(2\pi) \\ \frac{d\mathcal{L}}{d\boldsymbol{\Theta}} &= (\boldsymbol{y}_p^o)^T(\Sigma_p^o)^{-1}\frac{d\Sigma_p^o}{d\boldsymbol{\Theta}}(\Sigma_p^o)^{-1}\boldsymbol{y}_p^o + \text{Tr}\left((\Sigma_p^o)^{-1}\frac{d\Sigma_p^o}{d\boldsymbol{\Theta}}\right) \end{aligned} \tag{5.4}$$

Finally, the missing values in Y_p are estimated as $\hat{\boldsymbol{y}}_p^u = \mu_{\boldsymbol{y}_p^u}$, which are combined with the observed values $\boldsymbol{y}_p^o$ to form an imputed dataset $\hat{\mathcal{D}} = \{(\boldsymbol{t}_p, \hat{Y}_p, l_p)\}_{p=1}^N = \{((\boldsymbol{t}_p^o, \text{Vect}^{-1}(\boldsymbol{y}_p^o)), (\boldsymbol{t}_p^u, \text{Vect}^{-1}(\hat{\boldsymbol{y}}_p^u)), l_p)\}_{p=1}^N$, where $\text{Vect}^{-1}(\cdot)$ is the inverse operation of $\text{Vect}(\cdot)$.

In addition, to accelerate MGP Imputation, we adopt the blackbox matrix-matrix multiplication framework (Gardner et al., 2018) to integrate the modified preconditioned conjugate gradient (mPCG) approach with GPU acceleration into the data imputation workflow. Please refer to Appendix B for the detailed procedure for GPU acceleration of MGP imputation.

5.3.2 Missingness-aware Multi-branching Self-Attention Encoder (MUSE-Net)

We further propose a novel missingness-aware multi-branching self-attention encoder network (MUSE-Net) to process the imputed longitudinal EHRs. This model is designed to not only effectively capture crucial temporal correlations in multi-variate irregular longitudinal data but also recognize missingness patterns and account for the imbalanced data issue for enhanced classification performance. As shown in Fig. C.25(b) and (c), this model consists of 3 key modules: time-aware self-attention encoder, missing value masks, and multi-branching outputs.

Time-aware Self-attention Encoder

We propose to adapt the traditional self-attention encoder (Vaswani et al., 2017) to incorporate the information of elapsed time between consecutive records in irregular longitudinal EHRs. Specifically, there are three building blocks in our time-aware self-attention encoder:

Elapsed time-based positional encoding: The time order of the input longitudinal sequence plays a crucial role in time series analysis. However, this information is often ignored in traditional attention-based encoders because it does not incorporate any recurrent or convolution operations. To address this issue, an elapsed time-based positional encoding is added into the input sequence at the beginning of the self-attention encoder as shown in Fig. C.25(b), such that the temporal information regarding the irregularly sampled EHRs is incorporated into the model. In our study, we employ the following sinusoidal encoding:

$$PE_{t_{p,i},2m} = \sin(t_{p,i}/10000^{2m/M}) \qquad PE_{t_{p,i},2m+1} = \cos(t_{p,i}/10000^{2m/M}) \tag{5.5}$$

where $t_{p,i}$ is the time position of each observation to account for the irregularly spaced time interval, and $2m$ and $2m+1$ represent the $(2m)$-th and $(2m+1)$-th

variable dimensions, respectively. The sinusoid prevents positional encodings from becoming too large, introducing extra difficulties in network optimization. The embedded elapsed time-based positional features will then be combined with the imputed longitudinal sequence to generate the input of the self-attention module: $X_p = PE_p + \hat{Y}_p$, where $PE_p \in \mathbb{R}^{T_p \times M}$ is a matrix with elements defined in Eq. (5.5).

Self-attention module: The self-attention module is the fundamental mechanism that allows the network to learn dependencies in the longitudinal data. It maps a query set and a set of key-value pairs to an output. In a typical single-head self-attention, the key, query, and value matrices, denoted as $K \in \mathbb{R}^{T_p \times M}$, $Q \in \mathbb{R}^{T_p \times M}$, and $V \in \mathbb{R}^{T_p \times M}$, are computed by taking the input X_p:

$$K = \mathcal{F}_K(X_p), \quad Q = \mathcal{F}_Q(X_p), \quad V = \mathcal{F}_V(X_p) \tag{5.6}$$

where $\mathcal{F}_K(\cdot)$, $\mathcal{F}_Q(\cdot)$, and $\mathcal{F}_V(\cdot)$ represent the network operations to calculate the key, query, and value matrices respectively, which are often selected as linear transformations. The corresponding output, $\text{Output}_S \in \mathbb{R}^{T_p \times M}$, is computed by applying the scaled-dot production attention:

$$\text{Output}_S = \text{softmax}(\frac{QK^T}{\sqrt{M}}) \times V \tag{5.7}$$

where $\text{softmax}(\boldsymbol{x})_i = \frac{e^{x_i}}{\sum_j e^{x_j}}$ outputs the weight assigned to each element in V. Note that the dot product in the softmax function is scaled down by $\sqrt{M}$. This is essential to prevent the dot product values from growing too large, especially when the dimensionality of $\hat{Y}_p$ or X_p is large, introducing instabilities in the training process. The self-attention mechanism enables the network to access all the information of the input with the flexibility of focusing on certain important elements over the entire sequence, unlike the RNNs or CNNs which focus more on the neighboring elements.

The multi-head attention module is an extension of single-head self-attention to capture different relations among multivariate time series, which is adopted in our

study to model complex longitudinal EHRs. Specifically, we define multiple matrices for the keys, queries, and values by applying multiple network operations, $\mathcal{F}_K^l(\cdot)$'s, $\mathcal{F}_Q^l(\cdot)$'s, and $\mathcal{F}_V^l(\cdot)$'s, to the input:

$$K_l = \mathcal{F}_K^l(X_p) \in \mathbb{R}^{T_p \times \frac{M}{h}}, \quad Q_l = \mathcal{F}_Q^l(X_p) \in \mathbb{R}^{T_p \times \frac{M}{h}}, \quad V_l = \mathcal{F}_V^l(X_p) \in \mathbb{R}^{T_p \times \frac{M}{h}} \tag{5.8}$$

where h is the number of attention heads and $l \in \{1, \ldots, h\}$. The output of each attention head, $\text{Output}_l \in \mathbb{R}^{T_p \times \frac{M}{h}}$, is calculated by:

$$\text{Output}_l = \text{softmax}(\frac{Q_l K_l^T}{\sqrt{M/h}}) \times V_l \tag{5.9}$$

The final output of the multi-head attention, $\text{Output}_M \in \mathbb{R}^{T_p \times M}$, is the multiplication of the concatenation of the outputs of all heads and a weight matrix, $W^O \in \mathbb{R}^{M \times M}$:

$$\text{Output}_M = [\text{Output}_1, \text{Output}_2, \cdots, \text{Output}_h] W^O \tag{5.10}$$

This multi-head attention module enables the network to jointly attend to information from different subspaces of multivariate time series at different time points. Simple visualization of the computing procedure of multi-head attention can be found in Fig. C.24.

Feed-forward network: To stabilize the training process, Output_M will pass through a layer normalization operation Ba et al. (2016), and the normalized output will be combined with the original input using skip-connection (He et al., 2016) to generate the output of the "Add & Norm" block (see Fig. C.25(b)):

$$\text{Output}_{AN} = \text{LayerNorm}(\text{Output}_M) + X_p \tag{5.11}$$

Output_{AN} will serve as the input of a feed-forward network with GELU Hendrycks and Gimpel (2016) activation and additional "Add & Norm" blocks to further induce nonlinearity degree into the self-attention encoder. The multi-head attention and

feed-forward modules will be repeated multiple times to improve the generalizability of the model for capturing more informative features of the input sequence.

Missing Value Masks

To further account for potential discrepancies between imputed values and actual observations, the missing value masks (see Fig. C.25(a)) are incorporated as an additional input to the proposed model. We define the missing value masks for patient p as $\tilde{X}_p$ with $[\tilde{X}_p]_{t,m} = 1$ if $[Y_p]_{t,m}$ is missing; otherwise $[\tilde{X}_p]_{t,m} = 0$. The masks will be processed by an independent time-aware self-attention encoder in parallel with the imputed sequences as shown in Fig. C.25(b). This missingness-aware design enables the network to capture the relationship between non-missing and missing values. As a result, the network is capable of mitigating the effect of potential imputation errors during the MGP imputation process, thereby enhancing the overall predictive performance of the model.

Multi-branching Outputs

The self-attention encoder serves as a powerful feature extractor from irregular longitudinal EHRs for downstream tasks, such as binary classification. The classifier network, typically with a fully connected layer, is added after the feature extractor to interpret the abstracted features and make a final classification of the original input. The imbalanced data issue, a common problem in healthcare databases, incurs a more pronounced and direct negative impact on the classifier than on the feature extractor network. This is due to the fact that the classifier network directly interacts with the labels and is heavily influenced by the majority class during the training process Zhou et al. (2020). This leads to a bias towards the majority class, visibly affecting the model's performance. As such, careful design of the classifier network is needed to cope with the imbalanced data issue.

We propose to incorporate the Multi-branching (MB) architecture (Wang and Yao, 2021) (see Fig.C.25 (c)) into the classifier network to tackle the imbalanced issue. Specifically, in the training phase, the self-attention encoder will be trained with the whole dataset, and each of the MB outputs in the classifier network will be trained with a balanced sub-dataset to mitigate the negative influence of imbalanced data. The original imputed dataset $\hat{\mathcal{D}}$ consists of the majority class $\hat{\mathcal{D}}_M$, and the minority class $\hat{\mathcal{D}}_N$. We create N_b balanced sub-datasets by under-sampling $\hat{\mathcal{D}}_M$ to form $\mathcal{D}_i = \{\hat{\mathcal{D}}_N, \hat{\mathcal{D}}_M^i\}$, where the $\hat{\mathcal{D}}_M = \hat{\mathcal{D}}_M^1 \cup \hat{\mathcal{D}}_M^2 \cup \cdots \cup \hat{\mathcal{D}}_M^{N_b}$. This under-sampling operation is also applied to the missing value masks. Correspondingly, N_b output branches are created, each aligned with one of the balanced sub-datasets. Thus, each balanced sub-dataset serves as the training data for the respective branch in the output layer, and the self-attention encoder will be optimized by using all the N_b balanced subsets. In the end, the MB output layer will produce N_b predicted probabilities. The optimization process is guided by minimizing the cross-entropy loss function:

$$\mathcal{L}(\boldsymbol{\omega}; \hat{\mathcal{D}}) = -\sum_{p=1}^{N}\sum_{i=1}^{N_b} I(p \in \mathcal{D}_i)\left(l_p \log(\hat{P}^{(i)}(\hat{Y}_p, \tilde{X}_p; \boldsymbol{\omega})) + (1-l_p)\log(1-\hat{P}^{(i)}(\hat{Y}_p, \tilde{X}_p; \boldsymbol{\omega}))\right) \quad (5.12)$$

where $\boldsymbol{\omega}$ denotes the network parameters from both the self-attention encoder and the MB classifier; $I(\cdot)$ is an indicator function; $\hat{P}^{(i)}(\hat{Y}_p, \tilde{X}_p; \boldsymbol{\omega})$ is the predicted probability by output branch i given the input data $\hat{Y}_p$ and corresponding missing value mask $\tilde{X}_p$. The final predicted probability for patient p is computed as the average of the N_b predicted probabilities: $\hat{P}_{MB}(\hat{Y}_p, \tilde{X}_p; \boldsymbol{\omega}) = \sum_{i=1}^{N_b} \hat{P}^{(i)}(\hat{Y}_p, \tilde{X}_p; \boldsymbol{\omega})/N_b$, which is a more robust estimator compared to the single-branch (SB) counterpart as stated in Theorem I.

Theorem I: If both MB and SB classifiers are sufficiently trained with the following assumptions:

$$\begin{aligned} \bar{\hat{P}}^{(i)}(Y_p) &= \bar{\hat{P}}_{SB}(Y_p), \qquad (i \in \{1, \ldots, N_b\}) \\ \mathbb{E}_{\boldsymbol{\omega}}[D_{KL}(\bar{\hat{P}}^{(i)}(Y_p)||\hat{P}^{(i)}(Y_p;\boldsymbol{\omega}))] &= \mathbb{E}_{\boldsymbol{\omega}}[D_{KL}(\bar{\hat{P}}_{SB}(Y_p)||\hat{P}_{SB}(Y_p;\boldsymbol{\omega}))], \end{aligned} \tag{5.13}$$

where $\hat{P}_{SB}(Y_p;\boldsymbol{\omega})$ is the predicted distribution by the SB classifier for input Y_p, $D_{KL}(\cdot)$ is the Kullback–Leibler divergence, $\bar{\hat{P}}_{SB}(Y_p) = \arg\min_{P_*} \mathbb{E}_{\boldsymbol{\omega}}[D_{KL}(P_*(Y_p)||\hat{P}_{SB}(Y_p;\boldsymbol{\omega}))]$, and $\bar{\hat{P}}^{(i)}(Y_p) = \arg\min_{P_*} \mathbb{E}_{\boldsymbol{\omega}}[D_{KL}(P_*(Y_p)||\hat{P}^{(i)}(Y_p;\boldsymbol{\omega}))]$, then the variance of the MB classifier is no larger than the variance of the SB classifier, i.e., $V_{MB} \leq V_{SB}$, where

$$V_{MB} = \mathbb{E}_{Y_p \sim P_{data}, \boldsymbol{w}}[D_{KL}(\bar{\hat{P}}_{MB}(Y_p)||\hat{P}_{MB}(Y_p;\boldsymbol{w}))] \tag{5.14}$$

$$V_{SB} = \mathbb{E}_{Y_p \sim P_{data}, \boldsymbol{w}}[D_{KL}(\bar{\hat{P}}_{SB}(Y_p)||\hat{P}_{SB}(Y_p;\boldsymbol{w}))] \tag{5.15}$$

Here, P_{data} denotes the true data distribution and

$$\bar{\hat{P}}_{MB}(Y_p) = \arg\min_{P_*} \mathbb{E}_{\boldsymbol{\omega}}[D_{KL}(P_*(Y_p)||\hat{P}_{MB}(Y_p;\boldsymbol{\omega}))] \tag{5.16}$$

Theorem I shows that the MB classifier is more robust than the SB classifier (see Appendix B for the theoretical proof).

5.4 Experimental Design and Results

Fig. C.26 shows our experimental design to evaluate the performance of the proposed MUSE-Net. We implement our MUSE-Net to investigate both a synthetic and real-world dataset for performance evaluation. The MUSE-Net is benchmarked with three popular model structures widely used in sequential modeling: LSTM (Hochreiter and Schmidhuber, 1997), gated recurrent unit (GRU) (Chung et al., 2014), and TCN (Bai et al., 2018). Additionally, we evaluate the prediction performance provided by different imputation methods, i.e., the proposed MGP+mask, MGP, single Gaussian

process (GP), GP+mask, the baseline mean imputation, and Mean+mask. We also evaluate the impact of number of branches on the final prediction performance. The dataset in both synthetic and real-world case studies is split into training (80%), validation (10%), and testing (10%) sets to facilitate a comprehensive training and evaluation process. The model performance is evaluated based on two key metrics: area under the receiver operating characteristic curve (AUROC) and area under the precision-recall curve (AUPRC). AUROC serves as a measure of the model's ability to discriminate between classes, with a value of 1.0 indicating perfect classification and 0.5 denoting no discriminative power. AUPRC is especially valuable for assessing performance in classifying imbalanced datasets, reflecting the balance between precision and recall across various thresholds.

5.4.1 Experimental Results in Synthetic Case Study

Algorithm 1 shows the process to generate synthetic data from the autoregressive moving average (ARMA) model (Adhikari and Agrawal, 2013). Specifically, we first generate three baseline time series variables by ARMA(p_m,q_m) models with moving average of order q_m and autoregressive of order p_m: $v_{m,t} = \phi_{m,1}v_{m,t-1} + \cdots + \phi_{m,p_m}v_{m,t-p_m} + \psi_{m,1}\epsilon_{m,t-1} + \cdots + \psi_{m,q_m}\epsilon_{m,t-q_m} + \epsilon_{m,t}$, where $\{\epsilon_{m,t}\}$ are uncorrelated random variables for variable m; $\Omega_m = \{\phi_{m,1}, \cdots, \phi_{m,p_m}, \psi_{m,1}, \cdots, \psi_{m,q_m}\}$ is the set of coefficients for variable m. We initialize Ω_m randomly from a standard normal distribution and randomly select p_m,q_m from set $\{1, 2, 3\}$. We generate the remaining seven variables through feature engineering applied to the three baseline time series, ensuring correlations exist among the variables and differences exist between different classes. Detailed information on the feature engineering process and the simulation parameters is available in the appendix. The generated synthetic dataset is a 3-D tensor containing 5,000 samples with 90% from the majority class. Each sample has 10 variables measured across 50 time steps with irregular intervals. Notably,

Algorithm 3: Synthetic Data Generation Process

Require: Number of time steps: $n\_obs = 200$; Number of observations sampled from each time series: $sub\_time = 50$; Number of variables: $n\_variables = 10$; Number of samples to generate: $n\_samples = 5000$; Percentage of the majority samples: $percent\_negative = 90\%$

Initialize Ω_m for ARMA(p_m,q_m), $m \in \{1, 2, 3\}$, and apply feature engineering to generate remaining ARMA(p_m,q_m) $m \in \{4, 5, \cdots, 2 \times n\_variables\}$;

Generate a label list, *labels_list*, including *n_samples* binary labels (0 or 1) based on *percent_negative* ;

Generate empty 3-D tensor, *all_samples*, to store the generated data ;

for $j \leftarrow 1$ **to** *n_samples* **do**

  Initialize an empty temporal list: $variables \leftarrow [\quad]$;

  for $m \leftarrow 1$ **to** *n_variables* **do**

    if *label_list*[j] *is* 0 *(negative instance)* **then**

      Generate *n_variables* time series data using ARMA(p_m, q_m) ;

    else

      Generate *n_variables* time series data using ARMA($p_{m+n\_variables}, q_{m+n\_variables}$) ;

    Add generated time series to *variables* ;

  Combine *variables* into a matrix with *n_variables* columns and add it to *all_samples* ;

Initialize an empty tensor: $final\_dataset$;

for $i \leftarrow 1$ **to** *n_samples* **do**

  subtime_indices $\leftarrow$ Maintain the temporal order and randomly select *sub_time* unique time steps ;

  $sub\_ts\_i \leftarrow all\_samples[i, subtime\_indices, :]$;

  Add generated sub_ts_i to $final\_dataset$;

return $final\_dataset$

each variable is characterized by a distinct rate of missing values, varying randomly between 30% and 60%, mimicking real-world data complexities.

Fig. C.27 shows the AUROC and AUPRC scores over training epochs for the MUSE-Net-9 (i.e., with 9 branching outputs) utilizing different imputation methods with missing value masks on the validation set of our simulated data. The 'complete' represents the scenario where MUSE-Net is trained on synthetic data without any missing values. The MUSE-Net-9+MGP method exhibits superior performance in

both AUROC/AUPRC metrics and convergence speed. In contrast, the MUSE-Net-9+GP and MUSE-Net-9+mean methods display delayed convergence and lower scores, especially in the AUPRC metric.

Fig. C.28 displays the AUROC and AUPRC scores across training epochs for MUSE-Net+MGP with different numbers of MB outputs on the simulated validation set. It is evident that within 8 epochs, all the MUSE-Nets achieve optimal performance, reaching near-perfect AUROC and AUPRC scores. This exceptional performance is partly due to the simplistic nature of the synthetic dataset and the high efficacy of MGP imputation and effective feature extraction achieved by the MUSE-Net. However, we observe variations in the convergence speed among different MB numbers. Specifically, as we increase the MB outputs from 1 to 9, the speed of convergence improves. This enhancement can be attributed to the improved balance in the training sub-dataset for each branching output. With only one output, the imbalanced ratio is as skewed as 9:1, presenting a significant challenge for the model to learn from the minority class. As the number of MB outputs increases to 9, the ratio between the positive and negative classes is approximately 1:1, providing a more balanced learning environment for each branch. As such, the network model can learn more effectively from both classes, leading to faster convergence.

The impact of imbalanced data is more significant when modeling datasets with poor quality. Fig. C.29 shows the AUROC and AUPRC scores across epochs for MUSE-Net with mean imputation and different numbers of MB outputs on the synthetic validation set. The mean imputation is known for introducing a high bias due to its simplistic assumption, resulting in a less reliable imputed dataset. According to Fig. C.29, the MUSE-Net-9 converges within 6 epochs and achieves a good AUROC/AUPRC score, which indicates that our MB architecture together with the missing value masks effectively mitigates the negative impact of imbalanced datasets, even when the imputed training data is with low quality. In contrast, the MUSE-Net does not converge in 12 epochs if the MB architecture is not incorporated

(i.e., MB 1) and the final AUROC/AUPRC scores decrease to 0.921/0.773 from 0.988/0.895.

5.4.2 Experimental Results in Real-world Case Study

We further conduct a real-world case study using the Diabetic Retinopathy (DR) dataset, obtained from the 2018 Cerner Health Facts data warehouse Wang et al. (2024); Chen et al. (2022), to evaluate our MUSE-Net. The medical variables selected for this study include 21 routine blood tests, 5 comorbidity indicators, 3 demographic variables, and the duration of diabetes. More details on variable selection can be referred to Chen et al. (2022); Wang et al. (2021). Notably, the 21 blood tests are all subject to missing values, and the final dataset consists of records from 23,245 diabetic patients, with a minority of 8.9% diagnosed with DR. We use label "1" and "0" to indicate that a patient diagnosed with and without DR, respectively.

Fig. C.30 shows the performance of our MUSE-Net compared to other benchmark models on the DR validation set. Note that to ensure a fair comparison, all models are designed to have similar sizes, with the number of model parameters approximately 4,300, as shown in Table D.22. Furthermore, all models utilize the MGP imputation with missing value masks, 10 MB outputs, and Adam with decoupled weight decay optimizer Loshchilov and Hutter (2017). According to Fig. C.30, our MUSE-Net consistently outperforms other benchmarks, achieving the highest AUROC and AUPRC scores on the validation set. This trend is also evident in the results for the test set summarized in Table D.23, where MUSE-Net maintains the AUROC score of 0.949 and the AUPRC score of 0.883, which dominates those yielded by TCN, LSTM, and GRU models. Specifically, the MUSE-Net has an improvement over TCN by 2.3% in AUROC and 5.3% in AUPRC, and an improvement over LSTM by 3.3% in AUROC and 7.1% in AUPRC, and an improvement over GRU 1.9% in AUROC and 4.6% in AUPRC. This performance demonstrates that MUSE-Net is more effective at capturing the critical patterns inherent in irregular longitudinal DR

data. The higher AUROC and AUPRC scores reflect the superior performance of our MUSE-Net in terms of both discrimination and precision-recall balance, which are critical for clinical decision making.

Table D.24 presents the performance comparison in AUROC and AUPRC between our MUSE-Net-10 with MGP imputation and other imputation methods on the DR test set. It is worth noting that the MUSE-Net-10 with MGP+masks outperforms other methods, yielding an AUROC of 0.949 and an AUPRC of 0.883, which represents an improvement over GP+masks by 1.2% in AUROC and 6.1% in AUPRC, and an improvement over mean+masks by 3.5% in AUROC and 7.8% in AUPRC. Moreover, consistent performance improvement is achieved across all imputation methods when missing value masks are applied. Specifically, MGP+masks achieves an improvement over the non-mask MGP by 4.3% in AUROC and 8.5% in AUPRC. Similarly, employing masks with the GP imputation yields a 3.2% and 4.3% boost in AUROC and AUPRC respectively; the application of masks in mean imputation contributes to a 1.8% and 6.4% improvement in AUROC and AUPRC, respectively. The results show that the missing value masks enable the model to effectively account for the missingness patterns and mitigate potential discrepancies between imputed values and actual observations that may arise during the imputation process.

Table D.25 shows the AUROC and AUPRC scores for MUSE-Net with MGP imputation across various numbers of MB outputs in the DR test set. MB 1 corresponds to a model with a single output handling the original dataset with an imbalance ratio of approximately 1:10, and MB 10 represents a model with 10 outputs, each of which is trained with one of the 10 balanced sub-datasets with a ratio of approximately 1:1. This table suggests an overall increasing trend in both AUROC and AUPRC when the number of MB outputs increases. The highest AUROC/AUPRC scores of 0.949/0.883 are achieved with 10 MB outputs, demonstrating the enhanced predictive capability of our MUSE-Net in a balanced training environment. The performance gradually declines with a decrease in the number of MB outputs, reaching its lowest AUROC and AUPRC of 0.935 and 0.860

at MB 1. Specifically, the MUSE-Net with MB 10 achieves an improvement over the MUSE-Net with MB 1 by 1.5% in AUROC and 2.7% in AUPRC.

5.5 Conclusions

In this paper, we introduce a novel framework: **M**issingness-aware m**U**lti-branching **S**elf-Attention **E**ncoder (MUSE-Net) to model irregular longitudinal EHRs with missingness and imbalanced data issues. First, multi-task Gaussian processes (MGPs) are leveraged for missing value imputation in irregularly sampled longitudinal signals. Second, we propose a time-aware self-attention encoder augmented with a missing value mask and multi-branching architectures to classify irregular longitudinal EHRs. Finally, the training of both MGPs and MUSE-Net is accelerated by using GPU to improve the computation efficiency in processing large and high-dimensional longitudinal EHRs. We evaluate our proposed framework using both simulation data and real-world EHRs. Experimental results show that our MUSE-Net significantly outperforms existing approaches that are widely used to investigate longitudinal signals in current practice. More importantly, this framework can be broadly applicable to model complex longitudinal data with issues of multivariate irregularly spaced time series, incompleteness, and imbalanced class distributions.

5.6 Acknowledgement

This research work was supported partially by the National Eye Institute of the National Institutes of Health under Award Number R01EY033861 and partially by the National Heart, Lung, And Blood Institute of the National Institutes of Health under Award Number R01HL172292. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health. We also acknowledge the Cerner Corporation and Center for Health Systems Innovation (CHSI) at Oklahoma State University for sharing the Health Facts®

EHR database to support this research. Additionally, the authors acknowledge the University of Tennessee High-Performance Scientific Computing group for providing computational resources for this work.

5.7 Data Availability Statement

The data use agreement with Cerner Health Corporation prohibits authors from redistributing the data. Other researchers, who are interested in obtaining the data used in the study, may directly contact Cerner Heath Corporation for data access.

Chapter 6

Conclusions

This dissertation has explored advanced machine learning techniques to address the challenges posed by the complex and heterogeneous nature of Electronic Health Records (EHRs) for disease prediction. The research has demonstrated significant advancements in developing robust and reliable algorithms to improve the accuracy and efficiency of disease diagnosis and prediction.

The key contributions of this dissertation are as follows:

- **Multi-branching Temporal Convolutional Network for Sepsis Prediction**: In Chapter 2, we propose a novel predictive framework with Multi-Branching Temporal Convolutional Network (MB-TCN) to model the complexly structured medical data for robust prediction of sepsis. The MB-TCN framework not only efficiently handles the missing value and imbalanced data issues but also effectively captures the temporal pattern and heterogeneous variable interactions. We evaluate the performance of the proposed MB-TCN in predicting sepsis using real-world medical data from PhysioNet/Computing in Cardiology Challenge 2019. Experimental results show that MB-TCN outperforms existing methods that are commonly used in current practice.

- **Hierarchical Deep Learning with GAN for Cardiac Diagnosis**: In Chapter 3, we propose a two-level hierarchical deep learning framework with

Generative Adversarial Network (GAN) for automatic diagnosis of ECG signals. The first-level model is composed of a Memory-Augmented Deep AutoEncoder with GAN (MadeGAN), which aims to differentiate abnormal signals from normal ECGs for anomaly detection. The second-level learning aims at robust multi-class classification for different arrhythmias identification, which is achieved by integrating the transfer learning technique to transfer knowledge from the first-level learning with the multi-branching architecture to handle the data-lacking and imbalanced data issue. We evaluate the performance of the proposed framework using real-world medical data from the MIT-BIH arrhythmia database. Experimental results show that our proposed model outperforms existing methods that are commonly used in current practice.

- **Tensor Data Completion for Diabetic Retinopathy Prediction**: In Chapter 4, we propose a Multi-branching Temporal Convolutional Network with Tensor Data Completion (MB-TCN-TC) model to analyze the longitudinal EHRs collected from diabetic patients for DR prediction. Experimental results demonstrate that the proposed MB-TCN-TC model not only effectively copes with the imbalanced data and missing value issues commonly seen in EHR datasets but also captures the temporal correlation and complicated interactions among medical variables in the longitudinal clinical records, yielding superior prediction performance compared to existing methods. Specifically, our MB-TCN-TC model provides AUROC and AUPRC scores of 0.949 and 0.793 respectively, achieving an improvement of 6.27% on AUROC, 11.85% on AUPRC, and 19.3% on F1 score compared with the traditional TCN model.

- **MUSE-Net for Irregular Longitudinal EHRs**: In Chapter 5, we propose a novel **M**issingness-aware m**U**lti-branching **S**elf-Attention **E**ncoder (MUSE-Net) to cope with the challenges in modeling longitudinal EHRs for data-driven disease prediction. The MUSE-Net leverages a multi-task Gaussian process (MGP) with GPU acceleration for data imputation, a multi-branching

architecture to address the data imbalance problem, and a time-aware self-attention encoder to account for the irregularly spaced time interval in longitudinal EHRs. We evaluate the proposed MUSE-Net using both synthetic and real-world datasets. Experimental results show that our MUSE-Net outperforms existing methods that are widely used to investigate longitudinal signals.

Building on my past research endeavors outlined above, my upcoming research intentions encompass:

- **Multi-modal machine learning in complex healthcare systems**: Modern healthcare systems are deeply rooted in the integration of information from multi-modalities. For example, EHRs are representative examples of multi-modal data, encompassing not only structured laboratory findings or vital signs, but also imaging data (e.g., X-rays and CT scans), waveform data (e.g., ECG and EEG), unstructured narrative records (e.g., clinical notes and pathology reports). Each modality offers a unique perspective on a patient's health and fusing them to create an informative state will improve the predictive power of machine learning models. However, the growing volume and variety of data types present significant challenges (e.g., incomplete multi-modality and data heterogeneity) for data-driven decision-making. My research objective is to delve into multi-modal machine learning techniques to effectively integrate data from multiple sources, offering a unified view of patient health. Specifically, this research aims to 1) expand upon my prior work to design advanced DNNs for medical images and textual data to cope with the challenges in vision language understanding and generation; 2) develop cutting-edge models to integrate information from multi-modalities (e.g., images, textual, waveform signals) for robust multi-task learning. Upon successful completion, this research will yield 1) comprehensive understanding of patient's health condition; 2)

reduced clinical workload, freeing professionals for patient-centered tasks; and 3) improved diagnostic precision.

- **Advanced deep learning methods for modeling and monitoring complex spatiotemporal systems**: Distributed sensing offers a unique opportunity to observe the space-time dynamics of nonlinear complex systems, enhancing spatiotemporal data analysis and informed decision-making. Complex spatiotemporal systems typically feature intricate 3D geometries, and spatial- and temporal-varying properties. Traditional machine learning methods require considerable amounts of training data to model such complex systems, and they often fail to offer insights into the underlying mechanisms of system states. My research objective is to develop temporal graph neural network (GNN) leveraging the physics-based principles for modeling spatiotemporal dynamic systems. Specifically, this research aims to 1) extend my preceding contributions to develop a GNN-based models to analyze multi-channel ECGs, i.e., 3D spatiotemporal measurements, for automatic cardiac diagnosis; 2) incorporate physics-based principles of the cardiac electric wave propagation to improve the model robustness and interpretability; 3) investigate the GNN-based model with physics constraints in other 3D spatiotemporal systems such as additive manufacturing systems. This research aims to introduce advanced predictive models that will deepen our understanding of intricate 3D spatiotemporal systems across healthcare and advanced manufacturing sectors.

- **Fairness-aware artificial intelligence for healthcare systems**: As artificial intelligence (AI) research becomes increasingly prevalent within the healthcare field, ensuring fairness has become a paramount consideration to improve the usability of AI-tools for automated decision making. Disparities in health outcomes have long persisted across various patient demographics, with numerous underlying causes such as socio-economic conditions, genetic factors, and historical biases. When integrating AI systems into healthcare,

there is a risk of amplifying the disparities if the algorithms inherit or exacerbate biases from the training data. Moreover, due to the black-box property of most advanced AI systems, biases may also arise during interactions between the AI and clinicians, as well as the AI and patients. My research objective is to develop novel fairness-aware, interpretable machine learning algorithms for healthcare applications. Specifically, this research aims to 1) propose novel fairness metrics that are sensitive to the nuances of healthcare data; 2) extend my preceding contributions to develop interpretable-AI tools, providing transparency in decision-making processes and reducing biases in the interaction between AI & clinicians as well as AI & patients; 3) combine domain knowledge with machine learning and data mining techniques to develop fairness framework that not only detect and mitigate biases in existing datasets, but also monitor trained models in real-world scenarios to ensure they do not introduce biases when new data added over time. This research aims to ensure that the benefits of machine learning are equitably shared across all patient groups, contributing to more equitable health outcomes and improved trust in AI-powered medical decisions.

Bibliography

Acar, E., Dunlavy, D. M., Kolda, T. G., and Mørup, M. (2011). Scalable tensor factorizations for incomplete data. *Chemometrics and Intelligent Laboratory Systems*, 106(1):41–56. 76

Acharya, U. R., Oh, S. L., Hagiwara, Y., Tan, J. H., Adam, M., Gertych, A., and San Tan, R. (2017). A deep convolutional neural network model to classify heartbeats. *Computers in biology and medicine*, 89:389–396. 177

Addison, P. S. (2005). Wavelet transforms and the ecg: a review. *Physiological measurement*, 26(5):R155. 44

Adhikari, R. and Agrawal, R. K. (2013). An introductory study on time series modeling and forecasting. *arXiv preprint arXiv:1302.6613*. 108

Alagoz, O., Lowry, K. P., Kurian, A. W., Mandelblatt, J. S., Ergun, M. A., Huang, H., Lee, S. J., Schechter, C. B., Tosteson, A. N., Miglioretti, D. L., et al. (2021). Impact of the covid-19 pandemic on breast cancer mortality in the us: estimates from collaborative simulation modeling. *JNCI: Journal of the National Cancer Institute*, 113(11):1484–1494. 69

Albawi, S., Bayat, O., Al-Azawi, S., and Ucan, O. N. (2018). Social touch gesture recognition using convolutional neural network. *Computational Intelligence and Neuroscience*, 2018(1):6973103. 44

An, X. and K Stylios, G. (2020). Comparison of motion artefact reduction methods and the implementation of adaptive motion artefact reduction in wearable electrocardiogram monitoring. *Sensors*, 20(5):1468. 54

Association, A. D. (2002a). Diabetic retinopathy. *Diabetes Care*, 25:s90–s93. 68

Association, A. D. (2002b). Diabetic retinopathy. *Diabetes Care*, 25:s90–s93. 71

Aziz, S., Ahmed, S., and Alouini, M.-S. (2021). Ecg-based machine-learning algorithms for heartbeat classification. *Scientific reports*, 11(1):18738. 43

Azur, M. J., Stuart, E. A., Frangakis, C., and Leaf, P. J. (2011). Multiple imputation by chained equations: what is it and how does it work? *International journal of methods in psychiatric research*, 20(1):40–49. 10

Ba, J. L., Kiros, J. R., and Hinton, G. E. (2016). Layer normalization. *arXiv preprint arXiv:1607.06450*. 104

Bai, S., Kolter, J. Z., and Koltun, V. (1803). An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arxiv 2018. *arXiv preprint arXiv:1803.01271*. 19, 21

Bai, S., Kolter, J. Z., and Koltun, V. (2018). An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*. 6, 78, 96, 107

Banda, J. M., Seneviratne, M., Hernandez-Boussard, T., and Shah, N. H. (2018). Advances in electronic phenotyping: from rule-based definitions to machine learning models. *Annual review of biomedical data science*, 1:53–68. 4, 5, 95, 96

Baytas, I. M., Xiao, C., Zhang, X., Wang, F., Jain, A. K., and Zhou, J. (2017). 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. 6, 97

Berndt, D. J. and Clifford, J. (1994). Using dynamic time warping to find patterns in time series. In *KDD workshop*, volume 10, pages 359–370. Seattle, WA, USA:. 20

Bertsimas, D., Mingardi, L., and Stellato, B. (2021). Machine learning for real-time heart disease prediction. *IEEE Journal of Biomedical and Health Informatics*, 25(9):3627–3637. 8, 43

Betancur, J., Commandeur, F., Motlagh, M., Sharir, T., Einstein, A. J., Bokhari, S., Fish, M. B., Ruddy, T. D., Kaufmann, P., Sinusas, A. J., et al. (2018). Deep learning for prediction of obstructive disease from fast myocardial perfusion spect: a multicenter study. *JACC: Cardiovascular Imaging*, 11(11):1654–1663. 18

Bonilla, E. V., Chai, K., and Williams, C. (2007). Multi-task gaussian process prediction. *Advances in neural information processing systems*, 20. 99

Borleffs, C. J. W., Scherptong, R. W., Man, S.-C., van Welsenes, G. H., Bax, J. J., van Erven, L., Swenne, C. A., and Schalij, M. J. (2009). Predicting ventricular arrhythmias in patients with ischemic heart disease: clinical application of the ecg-derived qrs-t angle. *Circulation: Arrhythmia and Electrophysiology*, 2(5):548–554. 41

Bottou, L. (2010). Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010: 19th International Conference on Computational StatisticsParis France, August 22-27, 2010 Keynote, Invited and Contributed Papers*, pages 177–186. Springer. 90

Calisto, F. M., Fernandes, J., Morais, M., Santiago, C., Abrantes, J. M., Nunes, N., and Nascimento, J. C. (2023). Assertiveness-based agent communication for

a personalized medicine on medical imaging diagnosis. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–20. 72

Calisto, F. M., Nunes, N., and Nascimento, J. C. (2022a). Modeling adoption of intelligent agents in medical imaging. *International Journal of Human-Computer Studies*, 168:102922. 72

Calisto, F. M., Santiago, C., Nunes, N., and Nascimento, J. C. (2022b). Breastscreening-ai: Evaluating medical intelligent agents for human-ai interactions. *Artificial Intelligence in Medicine*, 127:102285. 72

Calvert, J. S., Price, D. A., Chettipally, U. K., Barton, C. W., Feldman, M. D., Hoffman, J. L., Jay, M., and Das, R. (2016). A computational approach to early sepsis detection. *Computers in biology and medicine*, 74:69–73. 18

Catling, F. J. and Wolff, A. H. (2020). Temporal convolutional networks allow early prediction of events in critical care. *Journal of the American Medical Informatics Association*, 27(3):355–365. 19

Chang, C.-C., Yeh, J.-H., Chiu, H.-C., Chen, Y.-M., Jhou, M.-J., Liu, T.-C., and Lu, C.-J. (2022). Utilization of decision tree algorithms for supporting the prediction of intensive care unit admission of myasthenia gravis: A machine learning-based approach. *Journal of Personalized Medicine*, 12(1):32. 89

Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357. 9, 19, 20, 74

Che, Z., Purushotham, S., Cho, K., Sontag, D., and Liu, Y. (2018). Recurrent neural networks for multivariate time series with missing values. *Scientific reports*, 8(1):6085. 6, 96, 97

Chen, R., Imani, F., and Yang, H. (2019). Heterogeneous recurrence analysis of disease-altered spatiotemporal patterns in multi-channel cardiac signals. *IEEE journal of biomedical and health informatics*, 24(6):1619–1631. 44

Chen, S., Wang, Z., Yao, B., and Liu, T. (2022). Prediction of diabetic retinopathy using longitudinal electronic health records. In *2022 IEEE 18th International Conference on Automation Science and Engineering (CASE)*, pages 949–954. IEEE. 6, 44, 68, 74, 81, 96, 111, 178

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*. 18

Choi, E., Schuetz, A., Stewart, W. F., and Sun, J. (2017). Using recurrent neural network models for early detection of heart failure onset. *Journal of the American Medical Informatics Association*, 24(2):361–370. 6, 18, 96

Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*. 18, 107

Cichocki, A., Zdunek, R., and Amari, S.-i. (2007). Nonnegative matrix and tensor factorization [lecture notes]. *IEEE signal processing magazine*, 25(1):142–145. 75, 76

Cismondi, F., Fialho, A. S., Vieira, S. M., Reti, S. R., Sousa, J. M., and Finkelstein, S. N. (2013). Missing data in medical databases: Impute, delete or classify? *Artificial intelligence in medicine*, 58(1):63–72. 9, 74

Cuturi, M. and Blondel, M. (2017). Soft-dtw: a differentiable loss function for time-series. In *International conference on machine learning*, pages 894–903. PMLR. 20

Das, D., Biswas, S. K., and Bandyopadhyay, S. (2022). A critical review on diagnosis of diabetic retinopathy using machine learning and deep learning. *Multimedia Tools and Applications*, 81(18):25613–25655. 5, 73, 96

Delahanty, R. J., Alvarez, J., Flynn, L. M., Sherwin, R. L., and Jones, S. S. (2019). Development and evaluation of a machine learning model for the early identification of patients at risk for sepsis. *Annals of emergency medicine*, 73(4):334–344. 17

Dontchos, B. N., Yala, A., Barzilay, R., Xiang, J., and Lehman, C. D. (2021). External validation of a deep learning model for predicting mammographic breast density in routine clinical practice. *Academic Radiology*, 28(4):475–480. 72

Doove, L. L., Van Buuren, S., and Dusseldorp, E. (2014). Recursive partitioning for missing data imputation in the presence of interaction effects. *Computational statistics & data analysis*, 72:92–104. 10

Du, J. A., Sadr, N., and de Chazal, P. (2019). Automated prediction of sepsis onset using gradient boosted decision trees. In *2019 Computing in Cardiology (CinC)*, pages Page–1. IEEE. 175

Ebrahimi, Z., Loni, M., Daneshtalab, M., and Gharehbaghi, A. (2020). A review on deep learning methods for ecg arrhythmia classification. *Expert Systems with Applications: X*, 7:100033. 44

ElSayed, N. A., Aleppo, G., Aroda, V. R., Bannuru, R. R., Brown, F. M., Bruemmer, D., Collins, B. S., Gibbons, C. H., Giurini, J. M., Hilliard, M. E., et al. (2023). 12. retinopathy, neuropathy, and foot care: Standards of care in diabetes—2023. *Diabetes Care*, 46(Supplement_1):S203–S215. 71

Eltrass, A. S., Tayel, M. B., and Ammar, A. I. (2021). A new automated cnn deep learning approach for identification of ecg congestive heart failure and arrhythmia using constant-q non-stationary gabor transform. *Biomedical signal processing and control*, 65:102326. 44

Emmanuel, T., Maupong, T., Mpoeleng, D., Semong, T., Mphago, B., and Tabona, O. (2021). A survey on missing data in machine learning. *Journal of Big Data*, 8(1):1–37. 94

Emmert-Streib, F. and Dehmer, M. (2019). Introduction to survival analysis in practice. *Machine Learning and Knowledge Extraction*, 1(3):1013–1038. 18, 74

Flach, P. A. and Kull, M. (2015). Precision-recall-gain curves: Pr analysis done right. In *NIPS*, volume 15. 27, 62

Fleuren, L. M., Klausch, T. L., Zwager, C. L., Schoonmade, L. J., Guo, T., Roggeveen, L. F., Swart, E. L., Girbes, A. R., Thoral, P., Ercole, A., et al. (2020). Machine learning for the prediction of sepsis: a systematic review and meta-analysis of diagnostic test accuracy. *Intensive care medicine*, 46(3):383–400. 17

Futoma, J., Hariharan, S., and Heller, K. (2017). Learning to detect sepsis with a multitask gaussian process rnn classifier. In *International conference on machine learning*, pages 1174–1182. PMLR. 6, 18, 155

Galloway, C., Treiman, D., Shreibati, J., Schram, M., Karbaschi, Z., Valys, A., Albert, D., and Stavrakis, S. (2019). 5105 a deep neural network predicts atrial fibrillation from normal ecgs recorded on a smartphone-enabled device. *European Heart Journal*, 40(Supplement_1):ehz746–0041. 44

Garabelli, P., Stavrakis, S., Albert, M., Koomson, E., Parwani, P., Chohan, J., Smith, L., Albert, D., Xie, R., Xie, Q., et al. (2016). Comparison of qt interval readings in normal sinus rhythm between a smartphone heart monitor and a 12-lead ecg for healthy volunteers and inpatients receiving sotalol or dofetilide. *Journal of cardiovascular electrophysiology*, 27(7):827–832. 41

Gardner, J., Pleiss, G., Weinberger, K. Q., Bindel, D., and Wilson, A. G. (2018). Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu

acceleration. *Advances in neural information processing systems*, 31. 101, 155, 156, 157

Garg, A., Kirby, J. S., Lavian, J., Lin, G., and Strunk, A. (2017). Sex-and age-adjusted population analysis of prevalence estimates for hidradenitis suppurativa in the united states. *JAMA dermatology*, 153(8):760–764. 89

Gleeson, F., Revel, M.-P., Biederer, J., Larici, A. R., Martini, K., Frauenfelder, T., Screaton, N., Prosch, H., Snoeckx, A., Sverzellati, N., et al. (2023). Implementation of artificial intelligence in thoracic imaging—a what, how, and why guide from the european society of thoracic imaging (esti). *European Radiology*, pages 1–10. 72

Goldberger, A. L., Amaral, L. A., Glass, L., Hausdorff, J. M., Ivanov, P. C., Mark, R. G., Mietus, J. E., Moody, G. B., Peng, C.-K., and Stanley, H. E. (2000). Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. *circulation*, 101(23):e215–e220. 63

Golub, G. H. and Van Loan, C. F. (2013). *Matrix computations*. JHU press. 156, 157

Gong, D., Liu, L., Le, V., Saha, B., Mansour, M. R., Venkatesh, S., and Hengel, A. v. d. (2019). Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1705–1714. 46, 47, 48

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. *Advances in neural information processing systems*, 27. 8, 45, 49

Groenwold, R. H. (2020). Informative missingness in electronic health record systems: the curse of knowing. *Diagnostic and prognostic research*, 4(1):1–6. 69

Guglin, M. E. and Thatai, D. (2006). Common errors in computer electrocardiogram interpretation. *International journal of cardiology*, 106(2):232–237. 8, 42, 44

Hannun, A. Y., Rajpurkar, P., Haghpanahi, M., Tison, G. H., Bourn, C., Turakhia, M. P., and Ng, A. Y. (2019). Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network. *Nature medicine*, 25(1):65–69. 44

Harbrecht, H., Peters, M., and Schneider, R. (2012). On the low-rank approximation by the pivoted cholesky decomposition. *Applied numerical mathematics*, 62(4):428–440. 156

Hasan, D. A., Zeebaree, S. R., Sadeeq, M. A., Shukur, H. M., Zebari, R. R., and Alkhayyat, A. H. (2021). Machine learning-based diabetic retinopathy early detection and classification systems-a survey. In *2021 1st Babylon International Conference on Information Technology and Science (BICITS)*, pages 16–21. IEEE. 73

Hasnul, M. A., Ab Aziz, N. A., et al. (2021). Ecg based emotion recognition system using fusion of feature extractions, feature selection, and classifiers. -. 8

Hastie, T., Tibshirani, R., Friedman, J. H., and Friedman, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer. 4

He, H. and Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, 21(9):1263–1284. 9, 19, 74, 75, 94

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778. 21, 79, 104

Hendrycks, D. and Gimpel, K. (2016). Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*. 104

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780. 6, 18, 107

Holzinger, A., Keiblinger, K., Holub, P., Zatloukal, K., and Müller, H. (2023). Ai for life: Trends in artificial intelligence for biotechnology. *New Biotechnology*, 74:16–24. 72

Hong, N., Wen, A., Stone, D. J., Tsuji, S., Kingsbury, P. R., Rasmussen, L. V., Pacheco, J. A., Adekkanattu, P., Wang, F., Luo, Y., et al. (2019a). Developing a fhir-based ehr phenotyping framework: A case study for identification of patients with obesity and multiple comorbidities from discharge summaries. *Journal of biomedical informatics*, 99:103310. 4, 96

Hong, S., Zhou, Y., Wu, M., Shang, J., Wang, Q., Li, H., and Xie, J. (2019b). Combining deep neural networks and engineered features for cardiac arrhythmia detection from ecg recordings. *Physiological measurement*, 40(5):054009. 44

Horng, S., Sontag, D. A., Halpern, Y., Jernite, Y., Shapiro, N. I., and Nathanson, L. A. (2017). Creating an automated trigger for sepsis clinical decision support at emergency department triage using machine learning. *PloS one*, 12(4):e0174708. 17

Huang, J., Yang, B., Yin, K., and Xu, J. (2024). Dna-t: Deformable neighborhood attention transformer for irregular medical time series. *IEEE Journal of Biomedical and Health Informatics*. 7, 98

Huang, Y., McCullagh, P., Black, N., and Harper, R. (2007). Feature selection and classification model construction on type 2 diabetic patients' data. *Artificial intelligence in medicine*, 41(3):251–262. 4, 95

Huo, Z., Qian, X., Huang, S., Wang, Z., and Mortazavi, B. J. (2022). Density-aware personalized training for risk prediction in imbalanced medical data. In *Machine Learning for Healthcare Conference*, pages 101–122. PMLR. 80

Hurst, J. W. (1998). Naming of the waves in the ecg, with a brief account of their genesis. *Circulation*, 98(18):1937–1942. 41

Imani, F., Cheng, C., Chen, R., and Yang, H. (2019). Nested gaussian process modeling and imputation of high-dimensional incomplete data under uncertainty. *IISE Transactions on Healthcare Systems Engineering*, 9(4):315–326. 74

Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr. 80

Iranzad, R., Liu, X., Chaovalitwongse, W. A., Hippe, D., Wang, S., Han, J., Thammasorn, P., Duan, C., Zeng, J., and Bowen, S. (2021). Gradient boosted trees for spatial data and its application to medical imaging data. *IISE Transactions on Healthcare Systems Engineering*, (just-accepted):1–33. 41

Islam, M. M., Nasrin, T., Walther, B. A., Wu, C.-C., Yang, H.-C., and Li, Y.-C. (2019). Prediction of sepsis patients using machine learning approach: a meta-analysis. *Computer methods and programs in biomedicine*, 170:1–9. 17

Ismail Fawaz, H., Forestier, G., Weber, J., Idoumghar, L., and Muller, P.-A. (2019). Deep learning for time series classification: a review. *Data mining and knowledge discovery*, 33(4):917–963. 5

Jager, F., Taddei, A., Moody, G. B., Emdin, M., Antolič, G., Dorn, R., Smrdel, A., Marchesi, C., and Mark, R. G. (2003). Long-term st database: a reference for the development and evaluation of automated ischaemia detectors and for the study of the dynamics of myocardial ischaemia. *Medical and Biological Engineering and Computing*, 41(2):172–182. 63

Jaimes, F., Farbiarz, J., Alvarez, D., and Martínez, C. (2005). Comparison between logistic regression and neural networks to predict death in patients with suspected sepsis in the emergency room. *Critical care*, 9(2):1–7. 18

Janssen, K. J., Donders, A. R. T., Harrell Jr, F. E., Vergouwe, Y., Chen, Q., Grobbee, D. E., and Moons, K. G. (2010). Missing covariate data in medical research: to

impute is better than to ignore. *Journal of clinical epidemiology*, 63(7):721–727. 69

Jarrett, D., Yoon, J., and van der Schaar, M. (2019). Dynamic prediction in clinical survival analysis using temporal convolutional networks. *IEEE journal of biomedical and health informatics*, 24(2):424–436. 19

Kachuee, M., Fazeli, S., and Sarrafzadeh, M. (2018). Ecg heartbeat classification: A deep transferable representation. In *2018 IEEE international conference on healthcare informatics (ICHI)*, pages 443–444. IEEE. 177

Kam, H. J. and Kim, H. Y. (2017). Learning representations for the early detection of sepsis with deep neural networks. *Computers in biology and medicine*, 89:248–255. 18

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*. 90, 158

Kok, C., Jahmunah, V., Oh, S. L., Zhou, X., Gururajan, R., Tao, X., Cheong, K. H., Gururajan, R., Molinari, F., and Acharya, U. R. (2020). Automated prediction of sepsis using temporal convolutional network. *Computers in Biology and Medicine*, 127:103957. 19, 34

Kolda, T. G. and Bader, B. W. (2009). Tensor decompositions and applications. *SIAM review*, 51(3):455–500. 77

Krawczyk, B. (2016). Learning from imbalanced data: open challenges and future directions. *Progress in Artificial Intelligence*, 5(4):221–232. 9, 74

Kumar, A., Roberts, D., Wood, K. E., Light, B., Parrillo, J. E., Sharma, S., Suppes, R., Feinstein, D., Zanotti, S., Taiberg, L., et al. (2006). Duration of hypotension before initiation of effective antimicrobial therapy is the critical determinant of survival in human septic shock. *Critical care medicine*, 34(6):1589–1596. 15

Landi, I., Glicksberg, B. S., Lee, H.-C., Cherng, S., Landi, G., Danieletto, M., Dudley, J. T., Furlanello, C., and Miotto, R. (2020). Deep representation learning of electronic health records to unlock patient stratification at scale. *NPJ digital medicine*, 3(1):96. 4, 95

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *nature*, 521(7553):436–444. 8, 18, 44, 73

Li, J., Yan, X. S., Chaudhary, D., Avula, V., Mudiganti, S., Husby, H., Shahjouei, S., Afshar, A., Stewart, W. F., Yeasin, M., et al. (2021a). Imputation of missing values for electronic health record laboratory data. *NPJ digital medicine*, 4(1):147. 69

Li, Q., Rajagopalan, C., and Clifford, G. D. (2014). A machine learning approach to multi-level ecg signal quality classification. *Computer methods and programs in biomedicine*, 117(3):435–447. 65

Li, T. and Zhou, M. (2016). Ecg classification using wavelet packet entropy and random forests. *Entropy*, 18(8):285. 177

Li, Y., Shi, Z., Liu, C., Tian, W., Kong, Z., and Williams, C. B. (2021b). Augmented time regularized generative adversarial network (atr-gan) for data augmentation in online process anomaly detection. *IEEE Transactions on Automation Science and Engineering*. 45

Li, Y., Zhang, Y., Zhao, L., Zhang, Y., Liu, C., Zhang, L., Zhang, L., Li, Z., Wang, B., Ng, E., et al. (2018). Combining convolutional neural network and distance distribution matrix for identification of congestive heart failure. *IEEE Access*, 6:39734–39744. 44

Li, Z., Li, S., and Yan, X. (2024). Time series as images: Vision transformer for irregularly sampled time series. *Advances in Neural Information Processing Systems*, 36. 7, 98

Lipton, Z. C., Kale, D. C., Elkan, C., and Wetzel, R. (2015). Learning to diagnose with lstm recurrent neural networks. *arXiv preprint arXiv:1511.03677*. 6, 18

Liu, N., Kumara, S., and Reich, E. (2020). Gaining insights into patient satisfaction through interpretable machine learning. *IEEE journal of biomedical and health informatics*, 25(6):2215–2226. 41

Liu, S., Liu, S., Cai, W., Pujol, S., Kikinis, R., and Feng, D. (2014). Early diagnosis of alzheimer's disease with deep learning. In *2014 IEEE 11th international symposium on biomedical imaging (ISBI)*, pages 1015–1018. IEEE. 89

Liu, Z., Khojandi, A., Li, X., Mohammed, A., Davis, R. L., and Kamaleswaran, R. (2022). A machine learning–enabled partially observable markov decision process framework for early sepsis prediction. *INFORMS Journal on Computing*, 34(4):2039–2057. 69

López-Martínez, F., Núñez-Valdez, E. R., Gomez, J. L., and García-Díaz, V. (2019). A neural network approach to predict early neonatal sepsis. *Computers & Electrical Engineering*, 76:379–388. 18

Loshchilov, I. and Hutter, F. (2017). Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*. 111, 179

Lyon, A., Mincholé, A., Martínez, J. P., Laguna, P., and Rodriguez, B. (2018). Computational techniques for ecg analysis and interpretation in light of their contribution to medical advances. *Journal of The Royal Society Interface*, 15(138):20170821. 44

Mani, I. and Zhang, I. (2003). knn approach to unbalanced data distributions: a case study involving information extraction. In *Proceedings of workshop on learning from imbalanced datasets*, volume 126. ICML United States. 19, 20, 74

Mani, S., Ozdas, A., Aliferis, C., Varol, H. A., Chen, Q., Carnevale, R., Chen, Y., Romano-Keeler, J., Nian, H., and Weitkamp, J.-H. (2014). Medical decision support

using machine learning for early detection of late-onset neonatal sepsis. *Journal of the American Medical Informatics Association*, 21(2):326–336. 17

Minchole, A., Camps, J., Lyon, A., and Rodríguez, B. (2019). Machine learning in the electrocardiogram. *Journal of electrocardiology*, 57:S61–S64. 42, 44

Miotto, R., Wang, F., Wang, S., Jiang, X., and Dudley, J. T. (2018). Deep learning for healthcare: review, opportunities and challenges. *Briefings in bioinformatics*, 19(6):1236–1246. 5

Moody, G. B. and Mark, R. G. (2001). The impact of the mit-bih arrhythmia database. *IEEE engineering in medicine and biology magazine*, 20(3):45–50. 43, 54

Moor, M., Horn, M., Rieck, B., Roqueiro, D., and Borgwardt, K. (2019). Early recognition of sepsis with gaussian process temporal convolutional networks and dynamic time warping. In *Machine Learning for Healthcare Conference*, pages 2–26. PMLR. 6, 19, 99, 155

Moré, J. J. and Thuente, D. J. (1994). Line search algorithms with guaranteed sufficient decrease. *ACM Transactions on Mathematical Software (TOMS)*, 20(3):286–307. 77

Morrill, J., Kormilitzin, A., Nevado-Holgado, A., Swaminathan, S., Howison, S., and Lyons, T. (2019). The signature-based model for early detection of sepsis from electronic health records in the intensive care unit. In *2019 Computing in Cardiology (CinC)*, pages Page–1. IEEE. 175

Mousavi, S., Afghah, F., Khadem, F., and Acharya, U. R. (2021). Ecg language processing (elp): A new technique to analyze ecg signals. *Computer Methods and Programs in Biomedicine*, 202:105959. 62, 177

Murat, F., Yildirim, O., Talo, M., Baloglu, U. B., Demir, Y., and Acharya, U. R. (2020). Application of deep learning techniques for heartbeats detection using ecg signals-analysis and review. *Computers in biology and medicine*, 120:103726. 44

Nemati, S., Holder, A., Razmi, F., Stanley, M. D., Clifford, G. D., and Buchman, T. G. (2018). An interpretable machine learning model for accurate prediction of sepsis in the icu. *Critical care medicine*, 46(4):547–553. 17

Nocedal, J. and Wright, S. J. (1999). *Numerical optimization*. Springer. 77

Ogunyemi, O. and Kermah, D. (2015). Machine learning approaches for detecting diabetic retinopathy from clinical and public health records. In *AMIA Annual Symposium Proceedings*, volume 2015, page 983. American Medical Informatics Association. 74, 178

Pan, J. and Tompkins, W. J. (1985). A real-time qrs detection algorithm. *IEEE transactions on biomedical engineering*, (3):230–236. 54

Pan, S. J. and Yang, Q. (2009). A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359. 52

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32. 155

Pfau, D. (2013). A generalized bias-variance decomposition for bregman divergences. -. 151

Piri, S., Delen, D., Liu, T., and Zolbanin, H. M. (2017). A data analytics approach to building a clinical decision support system for diabetic retinopathy: developing and deploying a model ensemble. *Decision Support Systems*, 101:12–27. 74, 178

Rajkomar, A., Dean, J., and Kohane, I. (2019). Machine learning in medicine. *New England Journal of Medicine*, 380(14):1347–1358. 4, 96

Rajkomar, A., Oren, E., Chen, K., Dai, A. M., Hajaj, N., Hardt, M., Liu, P. J., Liu, X., Marcus, J., Sun, M., et al. (2018). Scalable and accurate deep learning with electronic health records. *NPJ digital medicine*, 1(1):18. 93

Rajpurkar, P., Chen, E., Banerjee, O., and Topol, E. J. (2022). Ai in health and medicine. *Nature medicine*, 28(1):31–38. 4, 95

Rath, A., Mishra, D., Panda, G., and Satapathy, S. C. (2021). Heart disease detection using deep learning methods from imbalanced ecg samples. *Biomedical Signal Processing and Control*, 68:102820. 45

Reyna, M. A., Josef, C. S., Jeter, R., Shashikumar, S. P., Westover, M. B., Nemati, S., Clifford, G. D., and Sharma, A. (2020). Early prediction of sepsis from clinical data: the physionet/computing in cardiology challenge 2019. *Critical care medicine*, 48(2):210–217. 15, 16, 29

Reza, A. W., Eswaran, C., and Dimyati, K. (2011). Diagnosis of diabetic retinopathy: automatic extraction of optic disc and exudates from retinal images using marker-controlled watershed transformation. *Journal of medical systems*, 35:1491–1501. 73

Rhee, C., Dantes, R., Epstein, L., Murphy, D. J., Seymour, C. W., Iwashyna, T. J., Kadri, S. S., Angus, D. C., Danner, R. L., Fiore, A. E., et al. (2017). Incidence and trends of sepsis in us hospitals using clinical vs claims data, 2009-2014. *Jama*, 318(13):1241–1249. 17

Rizwan, A., Zoha, A., Mabrouk, I. B., Sabbour, H. M., Al-Sumaiti, A. S., Alomainy, A., Imran, M. A., and Abbasi, Q. H. (2020). A review on the state of the art in atrial fibrillation detection enabled by machine learning. *IEEE reviews in biomedical engineering*, 14:219–239. 42

Rosnati, M. and Fortuin, V. (2021). Mgp-atttcn: An interpretable machine learning model for the prediction of sepsis. *Plos one*, 16(5):e0251248. 6, 96, 155

Roychowdhury, S., Koozekanani, D. D., and Parhi, K. K. (2013). Dream: diabetic retinopathy analysis using machine learning. *IEEE journal of biomedical and health informatics*, 18(5):1717–1728. 73

Saad, Y. (2003). *Iterative methods for sparse linear systems*. SIAM. 157

Saito, T. and Rehmsmeier, M. (2015). The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PloS one*, 10(3):e0118432. 33

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. (2016). Improved techniques for training gans. *Advances in neural information processing systems*, 29:2234–2242. 50

Sanghvi, H. A., Patel, R. H., Agarwal, A., Gupta, S., Sawhney, V., and Pandya, A. S. (2023). A deep learning approach for classification of covid and pneumonia using densenet-201. *International Journal of Imaging Systems and Technology*, 33(1):18–38. 72

Satija, U., Ramkumar, B., and Manikandan, M. S. (2017). Automated ecg noise detection and classification system for unsupervised healthcare monitoring. *IEEE Journal of biomedical and health informatics*, 22(3):722–732. 65

Satija, U., Ramkumar, B., and Manikandan, M. S. (2018). A review of signal processing techniques for electrocardiogram signal quality assessment. *IEEE reviews in biomedical engineering*, 11:36–52. 65

Schieppati, A., Henter, J.-I., Daina, E., and Aperia, A. (2008). Why rare diseases are an important medical and social issue. *The Lancet*, 371(9629):2039–2041. 94

Schläpfer, J. and Wellens, H. J. (2017). Computer-interpreted electrocardiograms: benefits and limitations. *Journal of the American College of Cardiology*, 70(9):1183–1192. 8, 44

Schlegl, T., Seeböck, P., Waldstein, S. M., Schmidt-Erfurth, U., and Langs, G. (2017). Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In *International conference on information processing in medical imaging*, pages 146–157. Springer. 45

Schull, M. J., Vermeulen, M. J., and Stukel, T. A. (2006). The risk of missed diagnosis of acute myocardial infarction associated with emergency department volume. *Annals of emergency medicine*, 48(6):647–655. 41

Semeraro, F., Parrinello, G., Cancarini, A., Pasquini, L., Zarra, E., Cimino, A., Cancarini, G., Valentini, U., and Costagliola, C. (2011). Predicting the risk of diabetic retinopathy in type 2 diabetic patients. *Journal of Diabetes and its Complications*, 25(5):292–297. 74, 178

Serhal, H., Abdallah, N., Marion, J.-M., Chauvet, P., Oueidat, M., and Humeau-Heurtier, A. (2022). Overview on prediction, detection, and classification of atrial fibrillation using wavelets and ai on ecg. *Computers in Biology and Medicine*, 142:105168. 44

Setzer, F. C., Shi, K. J., Zhang, Z., Yan, H., Yoon, H., Mupparapu, M., and Li, J. (2020). Artificial intelligence for the computer-aided detection of periapical lesions in cone-beam computed tomographic images. *Journal of endodontics*, 46(7):987–993. 72

Shashikumar, S. P., Shah, A. J., Clifford, G. D., and Nemati, S. (2018). Detection of paroxysmal atrial fibrillation using attention-based bidirectional recurrent neural networks. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 715–723. 44

Shi, Z., Mamun, A. A., Kan, C., Tian, W., and Liu, C. (2022). An lstm-autoencoder based online side channel monitoring approach for cyber-physical attack detection in additive manufacturing. *Journal of Intelligent Manufacturing*, pages 1–17. 44

Shickel, B., Tighe, P. J., Bihorac, A., and Rashidi, P. (2017). 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. 69, 72, 93

Shin, D.-H., Park, R. C., and Chung, K. (2020). Decision boundary-based anomaly detection model using improved anogan from ecg data. *IEEE Access*, 8:108664–108674. 45, 62, 176

Sidiropoulos, N. D., De Lathauwer, L., Fu, X., Huang, K., Papalexakis, E. E., and Faloutsos, C. (2017). Tensor decomposition for signal processing and machine learning. *IEEE Transactions on Signal Processing*, 65(13):3551–3582. 77

Singh, J., Oshiro, K., Krishnan, R., Sato, M., Ohkuma, T., and Kato, N. (2019). Utilizing informative missingness for early prediction of sepsis. In *2019 Computing in Cardiology (CinC)*, pages 1–4. IEEE. 175

Sivaraman, V., Bukowski, L. A., Levin, J., Kahn, J. M., and Perer, A. (2023). Ignore, trust, or negotiate: understanding clinician acceptance of ai-based treatment recommendations in health care. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–18. 72

Soleymani, R., Granger, E., and Fumera, G. (2020). F-measure curves: A tool to visualize classifier performance under imbalance. *Pattern Recognition*, 100:107146. 27

Sopharak, A., Dailey, M. N., Uyyanonvara, B., Barman, S., Williamson, T., Nwe, K. T., and Moe, Y. A. (2010). Machine learning approach to automatic exudate detection in retinal images from diabetic patients. *Journal of Modern optics*, 57(2):124–135. 73

Sprung, C. L., Annane, D., Keh, D., Moreno, R., Singer, M., Freivogel, K., Weiss, Y. G., Benbenishty, J., Kalenka, A., Forst, H., et al. (2008). Hydrocortisone therapy

for patients with septic shock. *New England Journal of Medicine*, 358(2):111–124. 15

Srinivas, N., Krause, A., Kakade, S. M., and Seeger, M. (2009). Gaussian process optimization in the bandit setting: No regret and experimental design. *arXiv preprint arXiv:0912.3995*. 27

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958. 80, 85

Sun, C., Hong, S., Song, M., and Li, H. (2020). A review of deep learning methods for irregularly sampled medical time series data. *arXiv preprint arXiv:2010.12493*. 6, 97

T. Denton, B. (2023). Frontiers of medical decision-making in the modern age of data analytics. *IISE Transactions*, 55(1):94–105. 69, 72

Tajbakhsh, N., Shin, J. Y., Gurudu, S. R., Hurst, R. T., Kendall, C. B., Gotway, M. B., and Liang, J. (2016). Convolutional neural networks for medical image analysis: Full training or fine tuning? *IEEE transactions on medical imaging*, 35(5):1299–1312. 72

Ting, D. S. W., Cheung, C. Y.-L., Lim, G., Tan, G. S. W., Quang, N. D., Gan, A., Hamzah, H., Garcia-Franco, R., San Yeo, I. Y., Lee, S. Y., et al. (2017). Development and validation of a deep learning system for diabetic retinopathy and related eye diseases using retinal images from multiethnic populations with diabetes. *Jama*, 318(22):2211–2223. 73

Tipirneni, S. and Reddy, C. K. (2022). Self-supervised transformer for sparse and irregularly sampled multivariate clinical time-series. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 16(6):1–17. 7, 98

Ubaru, S., Chen, J., and Saad, Y. (2017). Fast estimation of tr(f(a)) via stochastic lanczos quadrature. *SIAM Journal on Matrix Analysis and Applications*, 38(4):1075–1099. 156

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30. 6, 97, 102, 155

Virani, S. S., Alonso, A., Aparicio, H. J., Benjamin, E. J., Bittencourt, M. S., Callaway, C. W., Carson, A. P., Chamberlain, A. M., Cheng, S., Delling, F. N., et al. (2021). Heart disease and stroke statistics—2021 update: a report from the american heart association. *Circulation*, 143(8):e254–e743. 41

Wang, K., Pleiss, G., Gardner, J., Tyree, S., Weinberger, K. Q., and Wilson, A. G. (2019a). Exact gaussian processes on a million data points. *Advances in neural information processing systems*, 32. 157

Wang, P., Hou, B., Shao, S., and Yan, R. (2019b). Ecg arrhythmias detection using auxiliary classifier generative adversarial network and residual network. *Ieee Access*, 7:100910–100922. 45

Wang, R., Liang, Y., Miao, Z., and Liu, T. (2023a). Bayesian analysis for imbalanced positive-unlabelled diagnosis codes in electronic health records. *The annals of applied statistics*, 17(2):1220. 178

Wang, R., Miao, Z., Liu, T., Liu, M., Grdinovac, K., Song, X., Liang, Y., Delen, D., and Paiva, W. (2021). Derivation and validation of essential predictors and risk index for early detection of diabetic retinopathy using electronic health records. *Journal of Clinical Medicine*, 10(7):1473. 74, 111, 178

Wang, S., Wang, X., Hu, Y., Shen, Y., Yang, Z., Gan, M., and Lei, B. (2020). Diabetic retinopathy diagnosis using multichannel generative adversarial network

with semisupervision. *IEEE Transactions on Automation Science and Engineering*, 18(2):574–585. 73, 82

Wang, X., Gui, Q., Liu, B., Jin, Z., and Chen, Y. (2013). Enabling smart personalized healthcare: a hybrid mobile-cloud approach for ecg telemonitoring. *IEEE journal of biomedical and health informatics*, 18(3):739–745. 43

Wang, Z., Chen, S., Liu, T., and Yao, B. (2024). Multi-branching temporal convolutional network with tensor data completion for diabetic retinopathy prediction. *IEEE Journal of Biomedical and Health Informatics*. 6, 11, 93, 96, 111

Wang, Z., Crawford, A., Lee, K. L., and Jaganathan, S. (2023b). reslife: Residual lifetime analysis tool in r. *arXiv preprint arXiv:2308.07410*. 94

Wang, Z., Liu, C., and Yao, B. (2022). Multi-branching neural network for myocardial infarction prediction. In *2022 IEEE 18th International Conference on Automation Science and Engineering (CASE)*, pages 2118–2123. IEEE. 4, 72, 95

Wang, Z., Stavrakis, S., and Yao, B. (2023c). Hierarchical deep learning with generative adversarial network for automatic cardiac diagnosis from ecg signals. *Computers in Biology and Medicine*, 155:106641. 5, 10, 73, 96

Wang, Z. and Yao, B. (2021). Multi-branching temporal convolutional network for sepsis prediction. *IEEE journal of biomedical and health informatics*, 26(2):876–887. 6, 10, 43, 52, 53, 55, 69, 70, 72, 80, 96, 106

Weerakody, P. B., Wong, K. W., Wang, G., and Ela, W. (2021). A review of irregular time series data handling with gated recurrent neural networks. *Neurocomputing*, 441:161–178. 6, 97

Williams, C. K. and Rasmussen, C. E. (2006). *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA. 99

Xia, Y., Xu, Y., Chen, P., Zhang, J., and Zhang, Y. (2023). Generative adversarial network with transformer generator for boosting ecg classification. *Biomedical Signal Processing and Control*, 80:104276. 45

Xia, Y., Zhang, H., Xu, L., Gao, Z., Zhang, H., Liu, H., and Li, S. (2018). An automatic cardiac arrhythmia classification system with wearable electrocardiogram. *IEEE Access*, 6:16529–16538. 8, 45

Xiao, C., Choi, E., and Sun, J. (2018). 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. 4, 72, 94, 96

Xie, J., Stavrakis, S., and Yao, B. (2023). Automated identication of atrial fibrillation from single-lead ecgs using multi-branching resnet. *arXiv preprint arXiv:2306.15096*. 5, 96

Xie, J. and Yao, B. (2022a). Physics-constrained deep active learning for spatiotemporal modeling of cardiac electrodynamics. *Computers in Biology and Medicine*, 146:105586. 5, 41, 73, 96

Xie, J. and Yao, B. (2022b). Physics-constrained deep active learning for spatiotemporal modeling of cardiac electrodynamics. *Computers in Biology and Medicine*, 146:105586. 73

Xie, J. and Yao, B. (2022c). Physics-constrained deep learning for robust inverse ecg modeling. *IEEE Transactions on Automation Science and Engineering*, 20(1):151–166. 5, 96

Xie, J. and Yao, B. (2023). Hierarchical active learning for defect localization in 3d systems. *IISE Transactions on Healthcare Systems Engineering*, pages 1–15. 72, 99

Xiong, P., Wang, H., Liu, M., Lin, F., Hou, Z., and Liu, X. (2016). A stacked contractive denoising auto-encoder for ecg signal denoising. *Physiological measurement*, 37(12):2214. 45

Yadav, P., Steinbach, M., Kumar, V., and Simon, G. (2018). Mining electronic health records (ehrs) a survey. *ACM Computing Surveys (CSUR)*, 50(6):1–40. 4, 93, 95

Yang, H., Kan, C., Liu, G., and Chen, Y. (2013). Spatiotemporal differentiation of myocardial infarctions. *IEEE Transactions on Automation Science and Engineering*, 10(4):938–947. 8, 43

Yang, H. and Yao, B. (2023). *Sensing, Modeling and Optimization of Cardiac Systems: A New Generation of Digital Twin for Heart Health Informatics*. Springer Nature. 93

Yang, M., Wang, X., Gao, H., Li, Y., Liu, X., Li, J., and Liu, C. (2019). Early prediction of sepsis using multi-feature fusion based xgboost learning and bayesian optimization. In *The IEEE Conference on Computing in Cardiology (CinC)*, volume 46, pages 1–4. 175

Yao, B. (2021). Spatiotemporal modeling and optimization for personalized cardiac simulation. *IISE Transactions on Healthcare Systems Engineering*, pages 1–35. 20, 41, 75, 99

Yao, B., Chen, Y., and Yang, H. (2021). Constrained markov decision process modeling for optimal sensing of cardiac events in mobile health. *IEEE Transactions on Automation Science and Engineering*. 41, 93

Yao, B. and Yang, H. (2016). Physics-driven spatiotemporal regularization for high-dimensional predictive modeling: A novel approach to solve the inverse ecg problem. *Scientific reports*, 6(1):39012. 15, 41, 69

Yao, B. and Yang, H. (2020). Spatiotemporal regularization for inverse ecg modeling. *IISE Transactions on Healthcare Systems Engineering*, 11(1):11–23. 41, 93

Yao, B., Zhu, R., and Yang, H. (2017). Characterizing the location and extent of myocardial infarctions with inverse ecg modeling and spatiotemporal regularization. *IEEE journal of biomedical and health informatics*, 22(5):1445–1455. 15, 44, 69, 93

Yau, J. W., Rogers, S. L., Kawasaki, R., Lamoureux, E. L., Kowalski, J. W., Bek, T., Chen, S.-J., Dekker, J. M., Fletcher, A., Grauslund, J., et al. (2012). Global prevalence and major risk factors of diabetic retinopathy. *Diabetes care*, 35(3):556–564. 68, 82

Yin, W., Kann, K., Yu, M., and Schütze, H. (2017). Comparative study of cnn and rnn for natural language processing. *arXiv preprint arXiv:1702.01923*. 19

Zabihi, M., Kiranyaz, S., and Gabbouj, M. (2019). Sepsis prediction in intensive care unit using ensemble of xgboost models. In *2019 Computing in Cardiology (CinC)*, pages Page–1. IEEE. 175

Zhang, K., Karanth, S., Patel, B., Murphy, R., and Jiang, X. (2022). A multi-task gaussian process self-attention neural network for real-time prediction of the need for mechanical ventilators in covid-19 patients. *Journal of Biomedical Informatics*, 130:104079. 99, 155

Zheng, Z., Yan, H., Setzer, F. C., Shi, K. J., Mupparapu, M., and Li, J. (2020). Anatomically constrained deep learning for automating dental cbct segmentation and lesion detection. *IEEE Transactions on Automation Science and Engineering*, 18(2):603–614. 73

Zhou, B., Cui, Q., Wei, X.-S., and Chen, Z.-M. (2020). Bbn: Bilateral-branch network with cumulative learning for long-tailed visual recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9719–9728. 80, 105

Zhou, B., Liu, S., Hooi, B., Cheng, X., and Ye, J. (2019). Beatgan: Anomalous rhythm detection using adversarially generated time series. In *IJCAI*, volume 2019, pages 4433–4439. 8, 45, 54, 62, 176

Zhu, R., Yao, B., Yang, H., and Leonelli, F. (2018). Optimal sensor placement for space–time potential mapping and data fusion. *IEEE sensors letters*, 3(1):1–4. 41

Zhu, T., Li, K., Herrero, P., and Georgiou, P. (2020). Deep learning for diabetes: a systematic review. *IEEE Journal of Biomedical and Health Informatics*, 25(7):2744–2757. 72

Appendices

Appendix A

Supplement Materials of Chapter 2

A.1 Dynamic time wrapping (DTW)-based Hierarchical Clustering (DTW-HC)

We propose a new sampling method of dynamic time wrapping (DTW)-based Hierarchical Clustering (DTW-HC) to generate the balanced dataset, which is expected to increase the diversity of the majority-class samples in each sub-dataset. The sampling procedure is illustrated as follows:

(1) We first calculate the pairwise DTW distance among the majority-class samples (i.e. non-sepsis class) to be used for clustering. Note that Euclidean distance cannot be used as the similarity measure here because data are time series with different lengths, while DTW has been proved to be effective to quantify the similarity of time series with variable length.

(2) Second, we employ the hierarchical clustering method based on the DTW distance matrix to divide the non-sepsis samples into K_- different clusters. The samples within the same cluster are more similar to each other, while the samples from different clusters are more different from each other.

(3) Third, in order to increase the sample diversity in each balanced dataset, we select N_+/K_- samples (where N_+ denotes the sample size of the sepsis class) from

each of the $K_{\text{-}}$ clusters to be combined with the sepsis samples and form a balanced dataset. This repeats for each branching output, generating N_b balanced datasets.

Appendix B

Supplement Materials of Chapter 5

B.1 Parameter Specifications in Synthetic Data Generation

Tables B.1 and B.2 provide the detail information of parameter specification in ARMA models and the feature engineering process.

Table B.1: Parameter specifications of the three baseline ARMA models

	p_m, q_m	Ω_m
ARMA_1, $m = 1$	(2, 2)	{-0.75, 0.25, 0.65, 0.35}
ARMA_2, $m = 2$	(1, 1)	{-0.8, 0.5}
ARMA_3, $m = 3$	(3, 3)	{-0.65, 0.45, -0.2, 0.70, 0.45, 0.25}

B.2 Robust Classifier from Multi-branching Network Models

This appendix shows the proof of Theorem I, which is based on Lemma I and Theorem II.

Table B.2: Feature engineering process to generate time series for each variable

	Neg. samples (label = 0)	Pos. samples (label = 1)
$m = 1$	ARMA_1	$\text{ARMA}_{11} = \text{ARMA}_1$
$m = 2$	ARMA_2	$\text{ARMA}_{12} = \text{ARMA}_2$
$m = 3$	ARMA_3	$\text{ARMA}_{13} = \text{ARMA}_3$
$m = 4$	$\text{ARMA}_4 = 0.8 \times \text{ARMA}_1$	$\text{ARMA}_{14} = 1.1 \times \text{ARMA}_{11}$
$m = 5$	$\text{ARMA}_5 = \text{ARMA}_1 + \text{ARMA}_2$	$\text{ARMA}_{15} = \text{ARMA}_{11} + \text{ARMA}_{12}$
$m = 6$	$\text{ARMA}_6 = \text{ARMA}_3 + \text{ARMA}_4$	$\text{ARMA}_{16} = \text{ARMA}_{13} + \text{ARMA}_{14}$
$m = 7$	$\text{ARMA}_7 = \text{ARMA}_1 + \text{ARMA}_2 + \text{ARMA}_3$	$\text{ARMA}_7 = \text{ARMA}_{11} + \text{ARMA}_{12} + \text{ARMA}_{13}$
$m = 8$	$\text{ARMA}_8 = \text{ARMA}_1 \times \text{ARMA}_2 + \text{ARMA}_3 \times \text{ARMA}_4$	$\text{ARMA}_{18} = \text{ARMA}_{11} \times \text{ARMA}_{12} + \text{ARMA}_{13} \times \text{ARMA}_{14}$
$m = 9$	$\text{ARMA}_9 = \text{ARMA}_3 + \text{ARMA}_4 + \text{ARMA}_1 \times \text{ARMA}_2$	$\text{ARMA}_{19} = 1.1 \times \text{ARMA}_{13} + 1.1 \times \text{ARMA}_{14} + \text{ARMA}_{11} \times \text{ARMA}_{12}$
$m = 10$	$\text{ARMA}_{10} = -\text{ARMA}_5 + \text{ARMA}_9$	$\text{ARMA}_{20} = -\text{ARMA}_{15} + \text{ARMA}_{19}$

Definition I (Bregman Divergence): If $F : \mathcal{X} \to \mathbb{R}$ is a convex differentiable function, the Bregman Divergence based on F is a function $D_F : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$, defined as

$$D_F[x_1||x_2] \equiv F(x_1) - F(x_2) - \langle \nabla F(x_2), x_1 - x_2 \rangle, \qquad x_1, x_2 \in \mathcal{X} \tag{B.1}$$

Lemma I (Generalized bias-variance decomposition (Pfau, 2013)): Let $F : \mathcal{X} \to \mathbb{R}$ be a convex differentiable function, $g(Y_p)$ is the true function, and $f(Y_p; \boldsymbol{\omega})$ is the prediction, where $\boldsymbol{\omega}$ is the model parameter, the generalized bias-variance decomposition based on the Bregman divergence D_F is

$$\mathbb{E}_{Y_p,\boldsymbol{\omega}}[D_F(g(Y_p)||f(Y_p;\boldsymbol{\omega}))] = \mathbb{E}_{Y_p,\boldsymbol{\omega}}[D_F(g(Y_p)||\bar{f}(Y_p))] + \mathbb{E}_{Y_p,\boldsymbol{\omega}}[D_F(\bar{f}(Y_p)||f(Y_p;\boldsymbol{\omega}))] \tag{B.2}$$

where $\bar{f}(Y_p) = \arg\min_z \mathbb{E}_{\boldsymbol{\omega}}[D_F(z||f(Y_p;\boldsymbol{\omega}))]$.

Proof. (Lemma I) According to the definition of $\bar{f}(Y_p)$, we first have the following results:

$$\begin{aligned}
0 &= \nabla_z \mathbb{E}_{\boldsymbol{\omega}}[D_F(z||f(Y_p;\boldsymbol{\omega}))]|_{z=\bar{f}(Y_p)} \\
&= \nabla_z \mathbb{E}_{\boldsymbol{\omega}}[F(z) - F(f(Y_p;\boldsymbol{\omega})) - \langle \nabla F(f(Y_p;\boldsymbol{\omega})), z - f(Y_p;\boldsymbol{\omega})\rangle]_{z=\bar{f}(Y_p)} \\
&= [\nabla F(z) - \nabla_z \mathbb{E}_{\boldsymbol{\omega}}[\langle \nabla F(f(Y_p;\boldsymbol{\omega})), z\rangle]]_{z=\bar{f}(Y_p)} \\
&= [\nabla F(z) - \mathbb{E}_{\boldsymbol{\omega}}[\nabla F(f(Y_p;\boldsymbol{\omega}))]]_{z=\bar{f}(Y_p)} \\
\Rightarrow \quad & \nabla F(\bar{f}(Y_p)) = \mathbb{E}_{\boldsymbol{\omega}}[\nabla F(f(Y_p;\boldsymbol{\omega}))] \qquad \text{(B.3)}
\end{aligned}$$

Then, the right-hand side of Eq. (B.2) can be rearranged as

$$\begin{aligned}
& \mathbb{E}_{Y_p,\boldsymbol{\omega}}[D_F(g(Y_p)||\bar{f}(Y_p))] + \mathbb{E}_{Y_p,\boldsymbol{\omega}}[D_F(\bar{f}(Y_p)||f(Y_p;\boldsymbol{\omega}))] \\
= \; & \mathbb{E}_{Y_p,\boldsymbol{\omega}}[F(g(Y_p)) - F(\bar{f}(Y_p)) - \langle \nabla F(\bar{f}(Y_p)), g(Y_p) - \bar{f}(Y_p)\rangle] + \\
& \mathbb{E}_{Y_p,\boldsymbol{\omega}}[F(\bar{f}(Y_p)) - F(f(Y_p;\boldsymbol{\omega})) - \langle \nabla F(f(Y_p;\boldsymbol{\omega})), \bar{f}(Y_p) - f(Y_p;\boldsymbol{\omega})\rangle] \\
= \; & \mathbb{E}_{Y_p,\boldsymbol{\omega}}[F(g(Y_p)) - F(f(Y_p;\boldsymbol{\omega})) \\
- \; & \langle \nabla F(\bar{f}(Y_p)), g(Y_p) - \bar{f}(Y_p)\rangle - \langle \nabla F(f(\boldsymbol{x};\boldsymbol{\omega})), \bar{f}(Y_p) - f(Y_p;\boldsymbol{\omega})\rangle] \\
= \; & \mathbb{E}_{Y_p,\boldsymbol{\omega}}[F(g(Y_p)) - F(f(Y_p;\boldsymbol{\omega})) - \langle \nabla F(f(Y_p;\boldsymbol{\omega})), g(Y_p) - f(Y_p;\boldsymbol{\omega})\rangle] \\
= \; & \mathbb{E}_{Y_p,\boldsymbol{\omega}}[D_F(g(Y_p)||f(Y_p;\boldsymbol{\omega}))] \qquad \text{(B.4)}
\end{aligned}$$

□

The bias-variance decomposition for classification analysis with cross-entropy loss is provided in Theorem II:

Theorem II (Bias-variance decomposition for classification tasks): We denote the true class distribution for input Y_p as $P(Y_p)$ and the predicted distribution given by the classification model as $\hat{P}(Y_p, \boldsymbol{w})$, where $\boldsymbol{w}$ is the model parameter set. We assume there are J classes in total. Then, according to the generalized decomposition for Bregman divergence, the bias-variance decomposition for classification tasks is given as

$$\begin{aligned}\mathbb{E}_{Y_p,\boldsymbol{\omega}}[D_{KL}(P(\boldsymbol{x})||\hat{P}(Y_p,\boldsymbol{w}))] &= \mathbb{E}_{Y_p,\boldsymbol{\omega}}[D_{KL}(P(Y_p)||\bar{\hat{P}}(Y_p))] \\ &+ \mathbb{E}_{Y_p,\boldsymbol{\omega}}[D_{KL}(\bar{\hat{P}}(Y_p)||\hat{P}(Y_p;\boldsymbol{\omega}))] \qquad \text{(B.5)}\end{aligned}$$

where $\bar{\hat{P}}(Y_p) = \arg\min_{P_*} \mathbb{E}_{\boldsymbol{\omega}}[D_{KL}(P_*(Y_p)||\hat{P}(Y_p;\boldsymbol{\omega}))]$, and $D_{KL}(P||\hat{P}) = \sum_{j=1}^{J} p_j \log \frac{p_j}{\hat{p}_j}$ is the Kullback-Leibler (KL) divergence. The first term in Eq. (B.5) characterizes the squared bias, and the second term captures the variance of the classifier model.

Proof. (Theorem II) The first important note is that minimizing KL divergence is equivalent to minimizing cross-entropy loss in classification tasks. The cross-entropy loss is defined as

$$H(P||\hat{P}) = -\sum_{j=1}^{J} p_j \log \hat{p}_j = D_{KL}(P||\hat{P}) - \sum_{j}(p_j \log p_j) \qquad \text{(B.6)}$$

where $\sum_j (p_j \log p_j)$ is often considered as a constant. Hence, minimizing $H(P||\hat{P})$ is equivalent to minimizing $D_{KL}(P||\hat{P})$. Additionally, KL divergence is a special case of the Bregman divergence. Specifically, if we define the function in Definition I as $F(P) = \sum_i p_i \log p_i$, then

$$\begin{aligned}D_F(P||\hat{P}) &= \sum_i (p_i \log p_i) - \sum_i (\hat{p}_i \log \hat{p}_i) - \sum_i \langle \log \hat{p}_i + 1, p_i - \hat{p}_i \rangle \\ &= \sum_i (p_i \log p_i) - \sum_i (p_i \log \hat{p}_i) = \sum_i p_i \log \frac{p_i}{\hat{p}_i} = D_{KL}(P||\hat{P}) \qquad \text{(B.7)}\end{aligned}$$

As such, we can leverage Lemma I for the Bregman divergence to prove Theorem II. Specifically, Theorem II follows by defining $F(P) = \sum_i p_i \log p_i$, and replacing $g(Y_p)$ by the true distribution $P(Y_P)$ and replacing the prediction function $f(Y_p;\boldsymbol{\omega})$ by $\hat{P}(Y_p;\boldsymbol{\omega})$.

□

Finally, the following shows the proof of Theorem I:

Proof. (Theorem I) According to Theorem II, the variances of the SB and MB classifiers are defined as (we omit subscripts Y_p for notation convenience)

$$V_{SB} = \mathbb{E}_{\boldsymbol{\omega}}[D_{KL}(\bar{\hat{P}}_{SB}||\hat{P}_{SB}(\boldsymbol{\omega}))], \quad V_{MB} = \mathbb{E}_{\boldsymbol{\omega}}[D_{KL}(\bar{\hat{P}}_{MB}||\hat{P}_{MB}(\boldsymbol{\omega}))] \tag{B.8}$$

where $\hat{P}_{SB}(\boldsymbol{\omega})$ and $\hat{P}_{MB}(\boldsymbol{\omega})$ are the predicted distribution provided by the SB and MB classifiers, respectively. In the MB model, the predicted probability for class j is $\hat{p}_{MB,j}(\boldsymbol{\omega}) = \frac{1}{N_b}\sum_{i=1}^{N_b}\hat{p}_j^{(i)}(\boldsymbol{\omega})$, where $\hat{p}_j^{(i)}(\boldsymbol{\omega})$ is the predicted probability for class j by branching output i. Given the assumptions in Theorem I, i.e., $\bar{\hat{P}}^{(i)} = \bar{\hat{P}}_{SB}$ and $\mathbb{E}_{\boldsymbol{\omega}}[D_{KL}(\bar{\hat{P}}^{(i)}||\hat{P}^{(i)}(\boldsymbol{\omega}))] = \mathbb{E}_{\boldsymbol{\omega}}[D_{KL}(\bar{\hat{P}}_{SB}||\hat{P}_{SB}(\boldsymbol{\omega}))]$ ($i \in \{1,\ldots,N_b\}$) if all the branches and the SB model are sufficiently trained, we have

$$\begin{aligned}
V_{SB} &= \frac{1}{N_b}\sum_{i=1}^{N_b}\mathbb{E}_{\boldsymbol{\omega}}[D_{KL}(\bar{\hat{P}}_{SB}||\hat{P}_{SB}(\boldsymbol{\omega}))] \\
&= \frac{1}{N_b}\sum_{i=1}^{N_b}\mathbb{E}_{\boldsymbol{\omega}}[D_{KL}(\bar{\hat{P}}^{(i)}||\hat{P}^{(i)}(\boldsymbol{\omega}))] \\
&= \frac{1}{N_b}\sum_{i=1}^{N_b}\mathbb{E}_{\boldsymbol{\omega}}\left[\sum_{j=1}^{J}\bar{\hat{p}}_j^{(i)}\log\frac{\bar{\hat{p}}_j^{(i)}}{\hat{p}_j^{(i)}(\boldsymbol{\omega})}\right] \\
&= \mathbb{E}_{\boldsymbol{\omega}}\left[\sum_{j=1}^{J}\left(\bar{\hat{p}}_{SB,j}\log\bar{\hat{p}}_{SB,j} - \sum_{i=1}^{N_b}\bar{\hat{p}}_{SB,j}\frac{\log\hat{p}_j^{(i)}(\boldsymbol{\omega})}{N_b}\right)\right] \\
&= \mathbb{E}_{\boldsymbol{\omega}}\left[\sum_{j=1}^{J}\left(\bar{\hat{p}}_{SB,j}\log\bar{\hat{p}}_{SB,j} - \bar{\hat{p}}_{SB,j}\log(\Pi_{i=1}^{N_b}\hat{p}_j^{(i)}(\boldsymbol{\omega}))^{\frac{1}{N_b}}\right)\right] \\
&\geq \mathbb{E}_{\boldsymbol{\omega}}\left[\sum_{j=1}^{J}\left(\bar{\hat{p}}_{SB,j}\log\bar{\hat{p}}_{SB,j} - \bar{\hat{p}}_{SB,j}\log(\frac{1}{N_b}\sum_{i=1}^{N_b}\hat{p}_j^{(i)}(\boldsymbol{\omega}))\right)\right] \\
&= \mathbb{E}_{\boldsymbol{\omega}}\left[\sum_{j=1}^{J}\left(\bar{\hat{p}}_{SB,j}\log\bar{\hat{p}}_{SB,j} - \bar{\hat{p}}_{SB,j}\log\hat{p}_{MB,j}(\boldsymbol{\omega})\right)\right] \\
&= \mathbb{E}_{\boldsymbol{\omega}}[D_{KL}(\bar{\hat{P}}_{SB}||\hat{P}_{MB}(\boldsymbol{\omega}))] \geq \mathbb{E}_{\boldsymbol{\omega}}[D_{KL}(\bar{\hat{P}}_{MB}||\hat{P}_{MB}(\boldsymbol{\omega}))] = V_{MB}
\end{aligned} \tag{B.9}$$

where the first inequality is true due to the fact that the geometric mean of nonnegative variables is always less than or equal to the arithmetic mean, and the

second inequality is true due to the definition of $\bar{\hat{P}}_{MB}$ as given in Eq. (5.16). As such, the MB classifier is more robust than the SB classifier, i.e., $V_{MB} \leq V_{SB}$, as stated in Theorem I. □

B.2.1 Algorithm for GPU Acceleration in MGP Imputation

Recent years have seen remarkable advancements in deep learning (Vaswani et al., 2017), largely attributed to new software frameworks that allow easy use of GPUs for efficient optimization (Paszke et al., 2019). In our study, we utilize GPU acceleration for training MUSE-Net, which has become a fundamental practice in modern deep-learning methodologies. However, using GPUs to speed up MGP imputation remains relatively unexplored and requires further investigation. MGP-based imputation can be computationally intensive because it involves operations of large kernel matrices.

MGP-based imputation involves two steps: training (i.e., learning $\boldsymbol{\Theta}$ by gradient-based algorithms) and inference (i.e., calculating the posterior mean for imputation). Three key components contribute to the computational intensity: multiplication of the inverse kernel matrix with observed values $(\Sigma_p^o)^{-1}\boldsymbol{y}_p^o$, computation of log determinant $\log|\Sigma_p^o|$ and trace $\text{Tr}\left((\Sigma_p^o)^{-1}\frac{d\Sigma_p^o}{d\boldsymbol{\Theta}}\right)$. In most existing MGP imputation techniques (Zhang et al., 2022; Futoma et al., 2017; Moor et al., 2019; Rosnati and Fortuin, 2021), calculation of the three quantities hinges on Cholesky decomposition of $(\Sigma_p^o)^{-1}$, which has cubic time complexity and thus poses significant computational demands for large datasets. Furthermore, the nonparallelizable nature of Cholesky decomposition makes it not suitable for GPU acceleration. To address this challenge, we adopt the blackbox matrix-matrix multiplication framework Gardner et al. (2018) to integrate the modified preconditioned conjugate gradient (mPCG) approach with GPU acceleration into the data imputation workflow.

Algorithm 4 summarizes the procedure for MGP imputation with mPCG. The inputs are initialized MGP hyperparameters $\boldsymbol{\Theta}_0$, observed value $\boldsymbol{y}_p^o$, covariance matrix Σ_p^o, a matrix $\Delta = [\boldsymbol{d}_0, \boldsymbol{d}_1, \cdots, \boldsymbol{d}_t] \in \mathbb{R}^{O_p \times (t+1)}$, where $\boldsymbol{d}_0 = \boldsymbol{y}_p^o$, and $\boldsymbol{d}_1, \boldsymbol{d}_2, \cdots, \boldsymbol{d}_t$ are

$i.i.d$ random vectors from a probability distribution with $\mathbb{E}(\boldsymbol{d}_i) = 0$ and $\mathbb{E}(\boldsymbol{d}_i \boldsymbol{d}_i^T) = \mathcal{P}$ ($i \in \{1, \ldots, t\}$), $\mathcal{P} = CC^T = (C_0 C_0^T + E) \approx \Sigma_p^o$, where $C_0 C_0^T \approx \mathcal{K}_M \odot \mathcal{K}_{O_p}$ is a low-rank matrix approximation of $\mathcal{K}_M \odot \mathcal{K}_{O_p}$ generated by pivotal Cholesky decomposition with $C_0 \in \mathbb{R}^{O_p \times r_0}$ ($r_0 \ll O_p$) (Harbrecht et al., 2012). $\mathcal{P}^{-1}$ will then be employed as a preconditioner for Σ_p^o to enhance the efficiency of the mPCG (Gardner et al., 2018). The output of Algorithm 4 is the posterior mean for missing value $\mu_{\boldsymbol{y}_p^u}$. Algorithm 4 consists of two main steps: the MGP training step with mPCG acceleration to generate optimal hyperparameters, $\boldsymbol{\Theta}^*$, followed by the imputation step to the end of the algorithm.

In MGP training, the key is to compute

$$(\Sigma_p^o)^{-1}\Delta = [(\Sigma_p^o)^{-1}\boldsymbol{d}_0, (\Sigma_p^o)^{-1}\boldsymbol{d}_1, (\Sigma_p^o)^{-1}\boldsymbol{d}_2, \cdots, (\Sigma_p^o)^{-1}\boldsymbol{d}_t] \tag{B.10}$$

in a parallel way in GPU using mPCG. After k iterations, the output of mPCG, i.e., $U_k \approx (\Sigma_p^o)^{-1}\Delta$ and $\mathcal{T}_1, \mathcal{T}_2, \cdots, \mathcal{T}_t \in \mathbb{R}^{k \times k}$, which are partial Lanczos tridiagonalizations (Golub and Van Loan, 2013) of Σ_p^o, can be used to compute $(\Sigma_p^o)^{-1}\boldsymbol{y}_p^o$, $\log|\Sigma_p^o|$, and $\mathrm{Tr}\left((\Sigma_p^o)^{-1}\frac{d\Sigma_p^o}{d\boldsymbol{\Theta}}\right)$ efficiently. Specifically, $(\Sigma_p^o)^{-1}\boldsymbol{y}_p^o$ can be obtained directly as $(\Sigma_p^o)^{-1}\boldsymbol{y}_p^o \approx [U_k]_{:,1}$, where $[U_k]_{:,1}$ is the first column of matrix U_k. The computation of $\mathrm{Tr}\left((\Sigma_p^o)^{-1}\frac{d\Sigma_p^o}{d\boldsymbol{\Theta}}\right)$ depends on stochastic trace estimation (Gardner et al., 2018; Ubaru et al., 2017), which is a method used to efficiently approximate the trace of a large matrix. The basic idea is to use random vectors, i.e., $\boldsymbol{d}_1, \boldsymbol{d}_2, \cdots, \boldsymbol{d}_t$, to probe the matrix as:

$$\begin{aligned}
\mathrm{Tr}\left((\Sigma_p^o)^{-1}\frac{d\Sigma_p^o}{d\boldsymbol{\Theta}}\right) &= \mathrm{Tr}\left((\Sigma_p^o)^{-1}\frac{d\Sigma_p^o}{d\boldsymbol{\Theta}} \mathop{\mathbb{E}}_{\boldsymbol{d}_i \sim \mathcal{N}(0,\mathcal{P})}[\mathcal{P}^{-1}\boldsymbol{d}_i\boldsymbol{d}_i^T]\right) \\
&= \mathop{\mathbb{E}}_{\boldsymbol{d}_i \sim \mathcal{N}(0,\mathcal{P})}\left[\mathrm{Tr}\left((\Sigma_p^o)^{-1}\frac{d\Sigma_p^o}{d\boldsymbol{\Theta}}\mathcal{P}^{-1}\boldsymbol{d}_i\boldsymbol{d}_i^T\right)\right] \\
&= \mathop{\mathbb{E}}_{\boldsymbol{d}_i \sim \mathcal{N}(0,\mathcal{P})}\left[\boldsymbol{d}_i^T(\Sigma_p^o)^{-1}\left(\frac{d\Sigma_p^o}{d\boldsymbol{\Theta}}\mathcal{P}^{-1}\boldsymbol{d}_i\right)\right] \\
&\approx \frac{1}{t}\sum_{i=1}^{t}\left(\boldsymbol{d}_i^T(\Sigma_p^o)^{-1}\right)\left(\frac{d\Sigma_p^o}{d\boldsymbol{\Theta}}\mathcal{P}^{-1}\boldsymbol{d}_i\right)
\end{aligned} \tag{B.11}$$

where $\boldsymbol{d}_i^T(\Sigma_p^o)^{-1} \approx ([U_k]_{:,i+1})^T$ are computed by mPCG, and $\mathcal{P}^{-1}$ can be calculated using the Matrix Inversion Lemma, i.e., $\mathcal{P}^{-1} = (C_0C_0^T + E)^{-1} = E^{-1} - E^{-1}(I + C_0^TE^{-1}C_0)^{-1}C_0^TE^{-1}$, which avoids the inversion computation of the $O_p \times O_p$ matrix using the inversion of a smaller $r_0 \times r_0$ matrix. To estimate $\log|\Sigma_p^o|$, we integrate stochastic trace estimation with partial Lanczos tridiagonalizations (Golub and Van Loan, 2013; Saad, 2003), i.e., $\mathcal{T}_1, \mathcal{T}_2, \cdots, \mathcal{T}_t$, from mPCG. The key idea of partial Lanczos tridiagonalizations is that instead of calculating the full tridiagonalization, i.e., $(C^{-1})^T\Sigma_p^oC^{-1} = H^T\mathcal{T}H$ where H is orthonormal, we run k iterations ($k \ll O_p$) of this algorithm t times so that we obtain t partial Lanczos tridiagonalizations decompositions $\mathcal{T}_i$. Specifically, the log determinant $\log|\Sigma_p^o|$ can be calculated by:

$$\begin{aligned}
\log|\Sigma_p^o| &= \log|\Sigma_p^oC^{-1}\mathcal{P}(C^{-1})^T| = \log|(C^{-1})^T\Sigma_p^oC^{-1}| + \log|\mathcal{P}| \\
\log|(C^{-1})^T\Sigma_p^oC^{-1}| &= \mathrm{Tr}(\log\mathcal{T}) = \mathop{\mathbb{E}}_{\boldsymbol{d}_i\sim\mathcal{N}(0,\mathcal{P})}[(\boldsymbol{d}_i^T(C^{-1})^T)(\log\mathcal{T})(C^{-1}\boldsymbol{d}_i)] \\
&= \mathop{\mathbb{E}}_{\boldsymbol{d}_i\sim\mathcal{N}(0,\mathcal{P})}[(\boldsymbol{d}_i^T(C^{-1})^T)H_i^T(\log\mathcal{T}_i)H_i(C^{-1}\boldsymbol{d}_i)] \\
&\approx \frac{1}{t}\sum_{i=1}^{t}\boldsymbol{e}_1^T(\log T_i)\boldsymbol{e}_1 \qquad \text{(B.12)}
\end{aligned}$$

where $\log|\mathcal{P}|$ can be computed using the Matrix Determinant Lemma as $\log|\mathcal{P}| = \log|C_0C_0^T + E| = \log|C_0^TE^{-1}C_0 + I| + \log|E|$, which converts the determinant calculation of $O_p \times O_p$ matrix into the determinant calculation of $r_0 \times r_0$ matrix; $\mathcal{T}_i \in \mathbb{R}^{k\times k}$ are partial Lanczos tridiagonal matrices, which can approximate the eigenvalue of $(C^{-1})^T\Sigma_p^oC^{-1}$ and can be obtained directly from mPCG (Golub and Van Loan, 2013; Saad, 2003); H_i is orthonormal matrix generated by $C^{-1}\boldsymbol{d}_i$ so that $H_i(C^{-1}\boldsymbol{d}_i) = \boldsymbol{e}_1$, where $\boldsymbol{e}_1$ is the first column of identity matrix. Many existing studies (Gardner et al., 2018; Wang et al., 2019a) have shown that by setting an iteration number $k \ll O_p$ and utilizing GPU to compute matrix operations in parallel, the mPCG algorithm can approximate exact solutions of MGP but with much faster speed compared to Cholesky decomposition.

The calculated $(\Sigma_p^o)^{-1}\boldsymbol{y}_p^o$, $\log|\Sigma_p^o|$, and $\mathrm{Tr}\left((\Sigma_p^o)^{-1}\frac{d\Sigma_p^o}{d\boldsymbol{\Theta}}\right)$ are used for computing $\mathcal{L}(\boldsymbol{\Theta})$ and its derivative $d\mathcal{L}/d\boldsymbol{\Theta}$. These values will be further utilized to estimate the optimal MGP hyperparameters through a gradient-based optimization algorithm (i.e., ADAM Optimizer (Kingma and Ba, 2014)), denoted as $\boldsymbol{\Theta}^*$, which is subsequently utilized to compute the posterior mean, completing data imputation.

Algorithm 4: Multi-task Gaussian Process (MGP) Imputation with Modified Preconditioned Conjugate Gradients (mPCG) Acceleration

Input : Initialization of $\boldsymbol{\Theta}_0$; $\boldsymbol{y}_p^o$; Σ_p^o; $\Delta = [\boldsymbol{d}_0, \boldsymbol{d}_1, \cdots, \boldsymbol{d}_t]$;
$\mathcal{P}^{-1} = (CC^T)^{-1} = (C_0C_0^T + E)^{-1}$, $C_0C_0^T$ generated by decomposing $\mathcal{K}_M \odot \mathcal{K}_{O_p}$ using pivot Cholesky decomposition; $l \leftarrow 0$

Output: The posterior mean for missing values $\mu_{\boldsymbol{y}_p^u}$

while *stopping criterion not met* **do**

  $U_0 \leftarrow 0$; $R_0 \leftarrow \Sigma_p^o U_0 - \Delta$; $Z_0 \leftarrow \mathcal{P}^{-1}(R_0)$; $S_0 \leftarrow -Z_0$; $\mathcal{T}_1, \ldots, \mathcal{T}_t \leftarrow 0$ **for** $j = 0$ **to** $k-1$ **do**

    $\boldsymbol{\alpha}_j \leftarrow (R_j \odot Z_j)^T\mathbf{1}/(S_j \odot (\Sigma_p^o S_j))\mathbf{1}$ $U_{j+1} \leftarrow U_j + S_j\mathrm{diag}(\boldsymbol{\alpha}_j)$
    $R_{j+1} \leftarrow R_j + \Sigma_p^o S_j\mathrm{diag}(\boldsymbol{\alpha}_j)$ **if** $\forall i \in 1, 2, \cdots, t \quad \|[R_{j+1}]_{:,i}\|_2 < tolerance$ **then**

      yield U_{j+1} **break**

    $Z_{j+1} \leftarrow \mathcal{P}^{-1}(R_{j+1})$ $\boldsymbol{\beta}_j \leftarrow (Z_{j+1} \odot Z_{j+1})^T\mathbf{1}/(Z_j \odot Z_j)^T\mathbf{1}$
    $S_{j+1} \leftarrow -Z_{j+1} + S_j\mathrm{diag}(\boldsymbol{\beta}_j)$ $\forall i \in 1, 2, \cdots, t \quad [\mathcal{T}_i]_{j+1,j+1} \leftarrow \frac{1}{[\boldsymbol{\alpha}_j]_i} + \frac{[\boldsymbol{\beta}_{j-1}]_i}{[\boldsymbol{\alpha}_{j-1}]_i}$ if $j > 0$, otherwise $[\mathcal{T}_i]_{j+1,j+1} \leftarrow 1/[\boldsymbol{\alpha}_j]_i$ $\forall i \in 1, 2, \cdots, t, j > 0$
    $[\mathcal{T}_i]_{j+1,j}, [\mathcal{T}_i]_{j,j+1} \leftarrow \frac{\sqrt{[\boldsymbol{\beta}_j]_i}}{[\boldsymbol{\alpha}_j]_i}$

  yield $U_k, \mathcal{T}_1, \ldots, \mathcal{T}_t$ ▷ Used for computation of $\mathcal{L}(\boldsymbol{\Theta}_l)$ and $\frac{d\mathcal{L}}{d\boldsymbol{\Theta}_l}$ Compute:
  $(\Sigma_p^o)^{-1}\boldsymbol{y}_p^o \leftarrow [U_k]_{:,1}$, $\log|\Sigma_p^o| \leftarrow \sum_{i=1}^t \boldsymbol{e}_1^T(\log T_i)\boldsymbol{e}_1 + \log|\mathcal{P}|$,
  $\mathrm{Tr}\left((\Sigma_p^o)^{-1}\frac{d\Sigma_p^o}{d\boldsymbol{\Theta}_l}\right) \leftarrow \frac{1}{t}\sum_{i=1}^t([U_k]_{:,i+1})^T\left(\frac{d\Sigma_p^o}{d\boldsymbol{\Theta}_l}\mathcal{P}^{-1}\boldsymbol{d}_i\right)$
  $\mathcal{L}(\boldsymbol{\Theta}_l) \leftarrow \log|\Sigma_p^o| + (\boldsymbol{y}_p^o)^T(\Sigma_p^o)^{-1}\boldsymbol{y}_p^o + \frac{O_p}{2}\log(2\pi)$
  $\frac{d\mathcal{L}}{d\boldsymbol{\Theta}_l} \leftarrow (\boldsymbol{y}_p^o)^T(\Sigma_p^o)^{-1}\frac{d\Sigma_p^o}{d\boldsymbol{\Theta}_l}(\Sigma_p^o)^{-1}\boldsymbol{y}_p^o + \mathrm{Tr}\left((\Sigma_p^o)^{-1}\frac{d\Sigma_p^o}{d\boldsymbol{\Theta}_l}\right)$
  $\boldsymbol{\Theta}_{l+1} \leftarrow ADAMOptimizer\left(\mathcal{L}(\boldsymbol{\Theta}_l), \frac{d\mathcal{L}}{d\boldsymbol{\Theta}_l}\right)$ ▷ Update $\boldsymbol{\Theta}_{l+1}$ by ADAM optimizer
  $l \leftarrow l + 1$

yield $\boldsymbol{\Theta}^* = \{\mathrm{Vect}(B)^*, \mathrm{Diag}(E)^*, \theta^*\}$ ▷ Complete training step

$\mu_{\boldsymbol{y}_p^u} \leftarrow (\mathcal{K}(\mathrm{Vect}(B)^*)_{M*M} \odot \mathcal{K}(\theta^*)_{U_pO_p})(\Sigma_p^o(\boldsymbol{\Theta}^*))^{-1}\boldsymbol{y}_p^o$ ▷ Complete imputation step

return $\mu_{\boldsymbol{y}_p^u}$

Appendix C

Figures

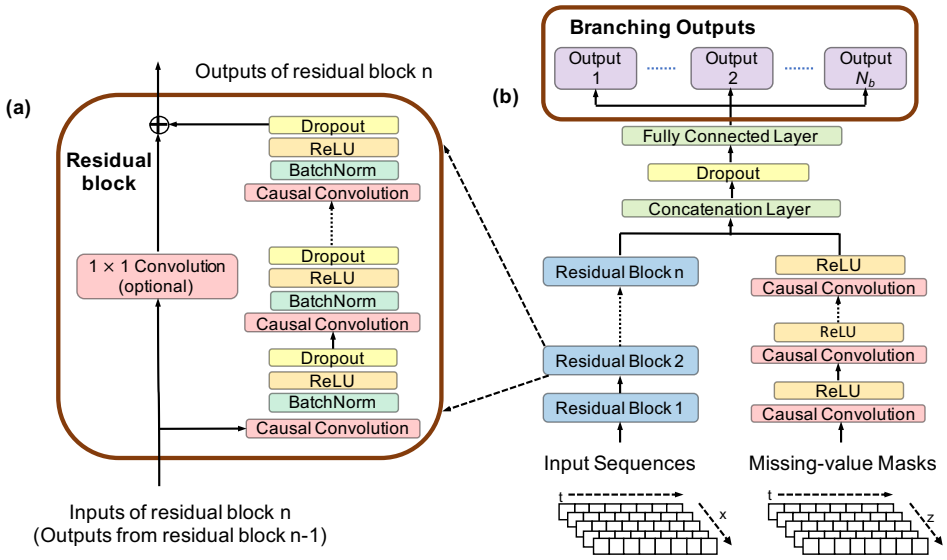


Figure C.1: (a) The architecture of a single residual block (b) The proposed framework of Multi-branching Temporal Convolutional Network (MB-TCN)

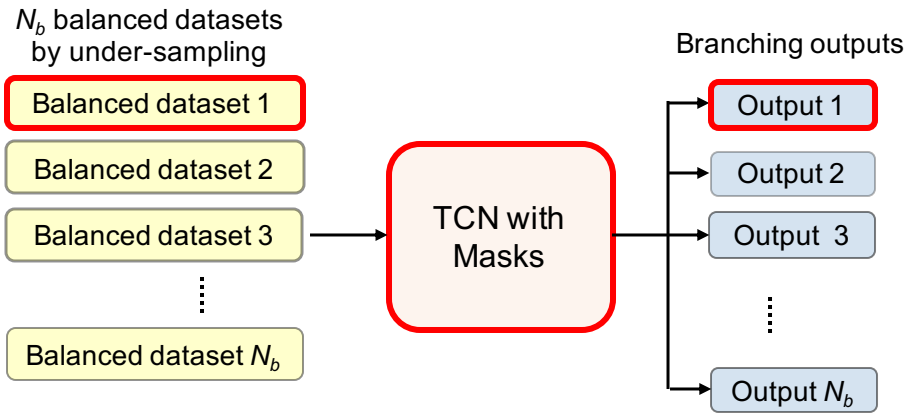


Figure C.2: Multi-branching outputs

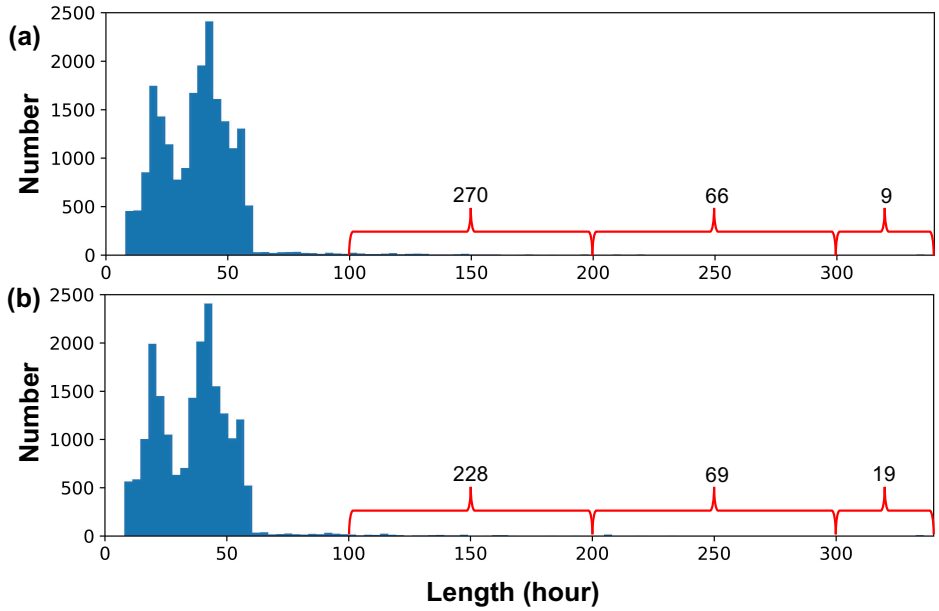


Figure C.3: Length distribution of the data from (a) Beth Israel Deaconess Medical Center (i.e., training set) (b) Emory University Hospital (i.e., testing set).

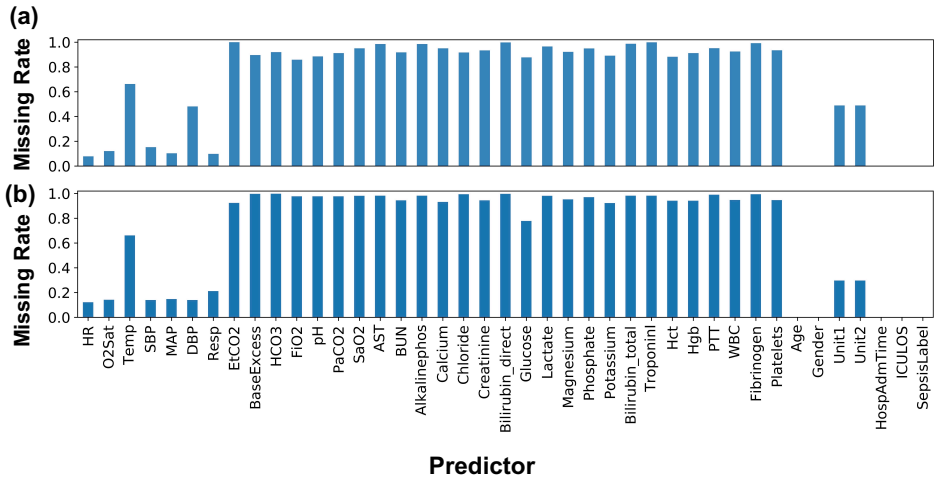


Figure C.4: Missing-value rate for each medical variable in the dataset from (a) the training and (b) testing set.

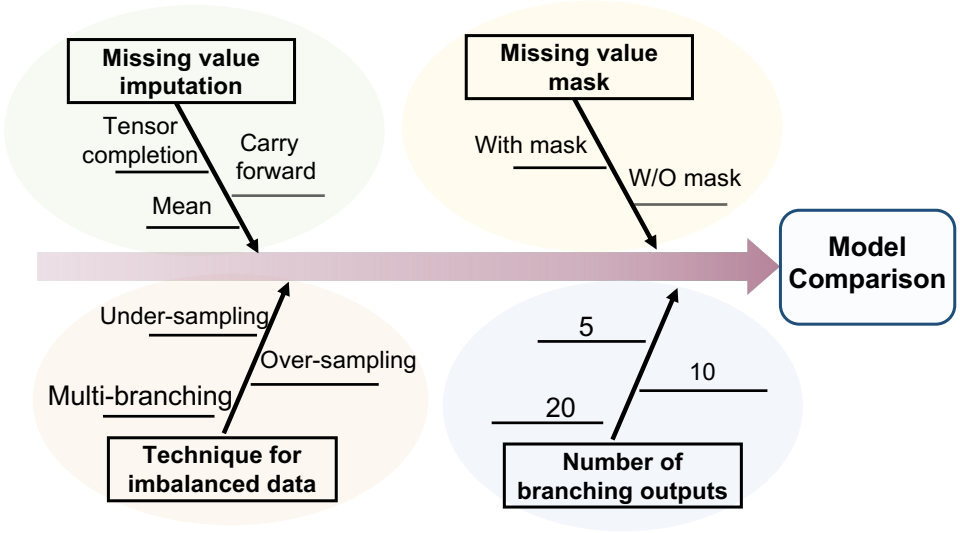


Figure C.5: Experimental Design for evaluating the performance of MB-TCN.

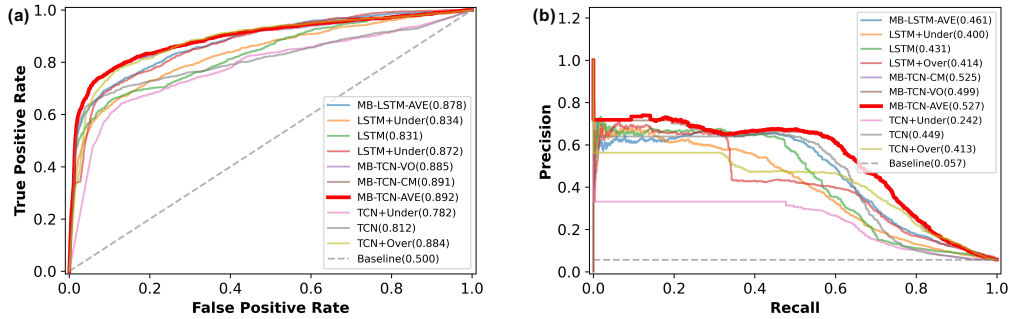


Figure C.6: The comparison of (a) ROC (b) PR curve between the proposed MB-TCN approach and other deep learning-based methods.

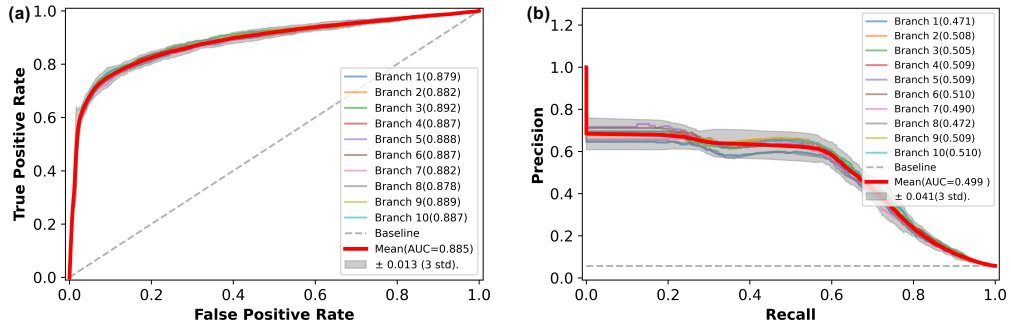


Figure C.7: (a) ROC (b) PR curves of all branching outputs in the MB-TCN model.

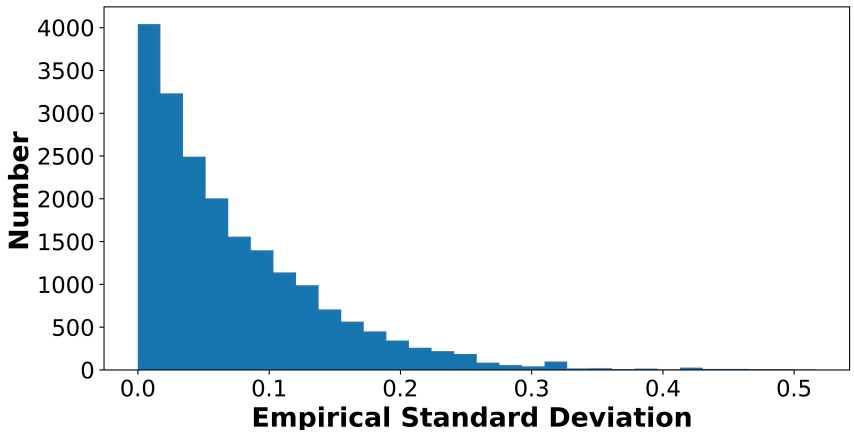


Figure C.8: Distribution of the ESTD of predicted probabilities in the test dataset.

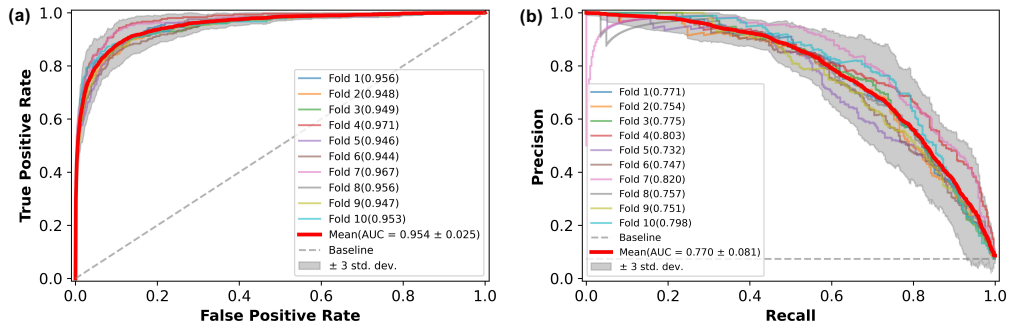


Figure C.9: (a) ROC (b) PR curves for the 10 folds obtained from our MB-TCN-AVE model.

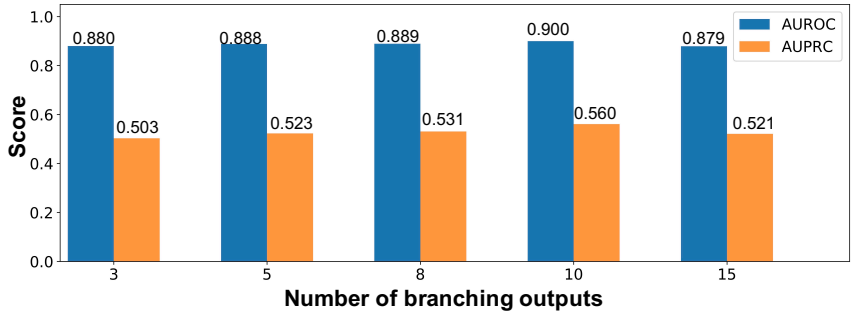


Figure C.10: The variation of predicted performance of MB-TCN-AVE with different number of branching outputs.

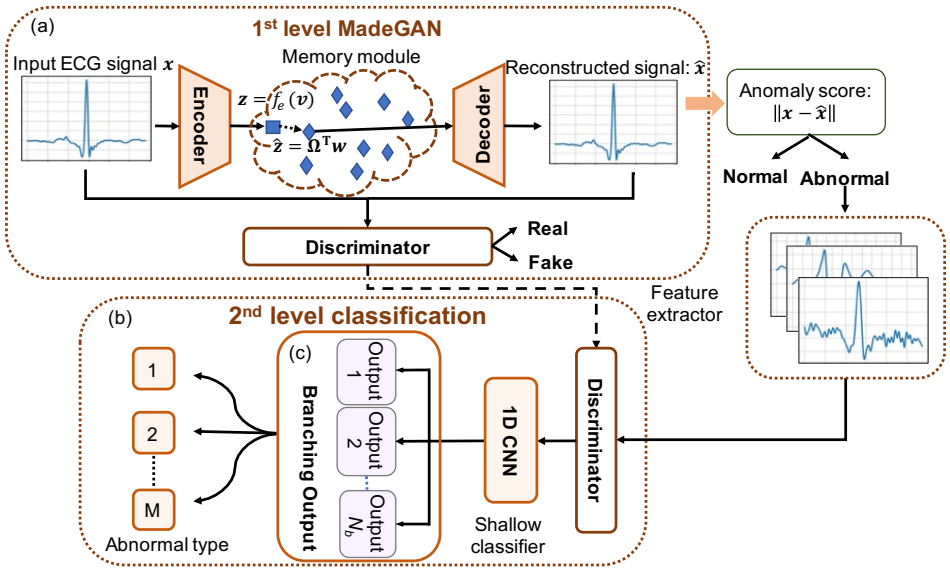


Figure C.11: The proposed two-level hierarchical deep learning framework: (a) first-level MadeGAN for anomaly detection; (b) second-level classification for arrhythmia type identification; (c) multi-branching output.

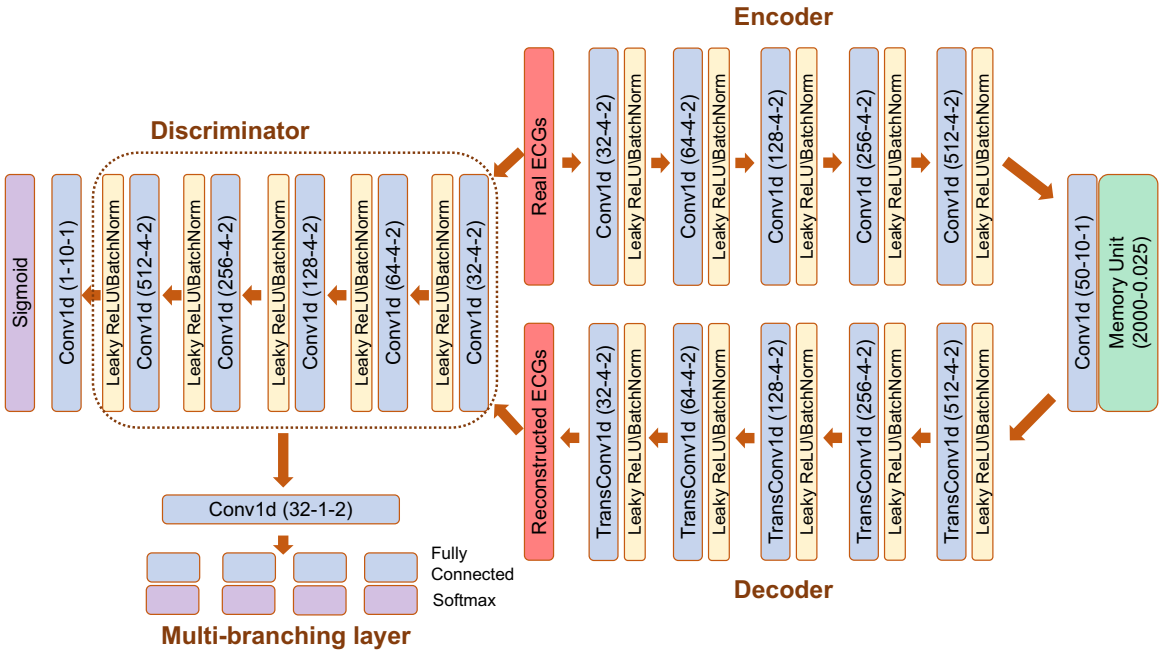


Figure C.12: The architecture details of our proposed hierarchical deep learning framework.

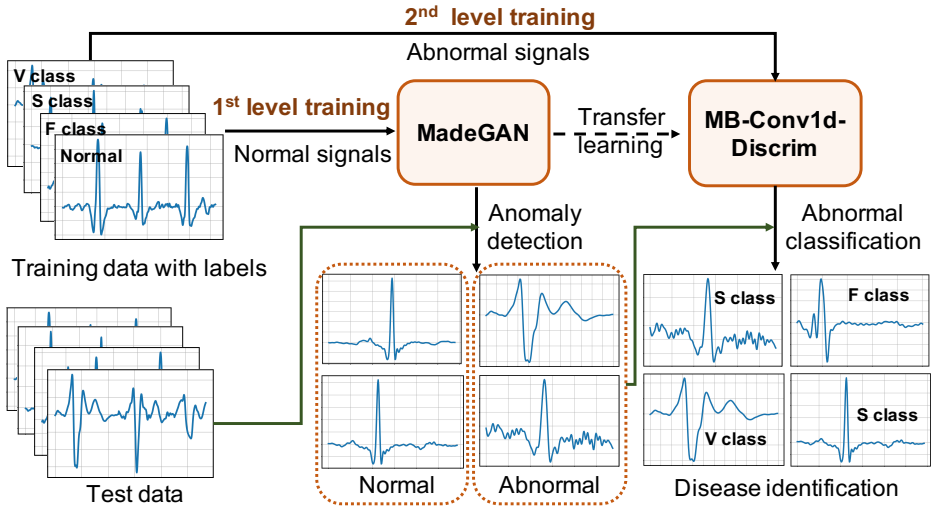


Figure C.13: Training and testing procedures for the proposed hierarchical deep learning framework.

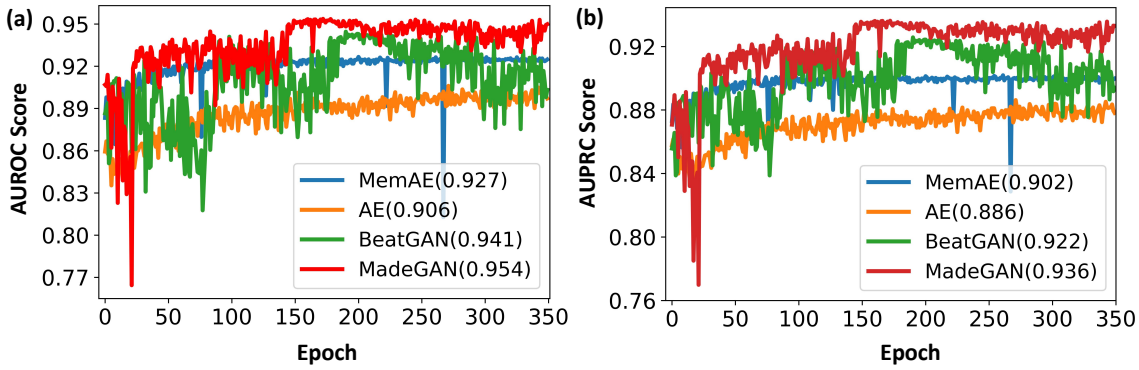


Figure C.14: The variations of (a) AUROC score and (b) AUPRC score over epochs in test set for different methods during the first-level anomaly detection.

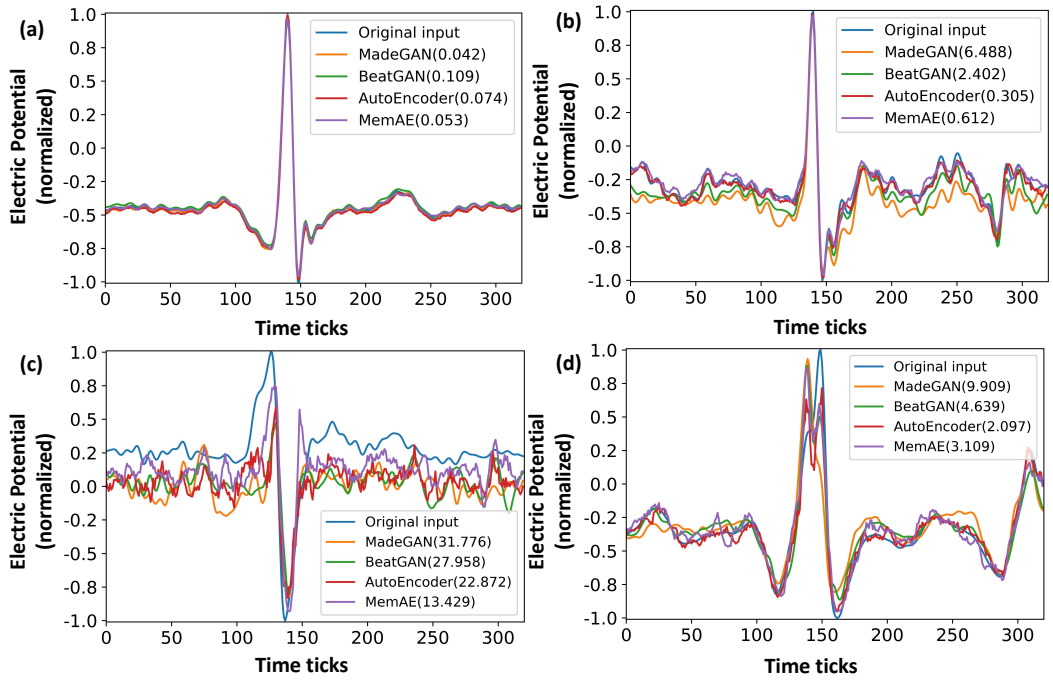


Figure C.15: Reconstructed ECG signals by the proposed MadeGAN and other benchmarks for (a) N class (b) S class (c) V class (d) F class.

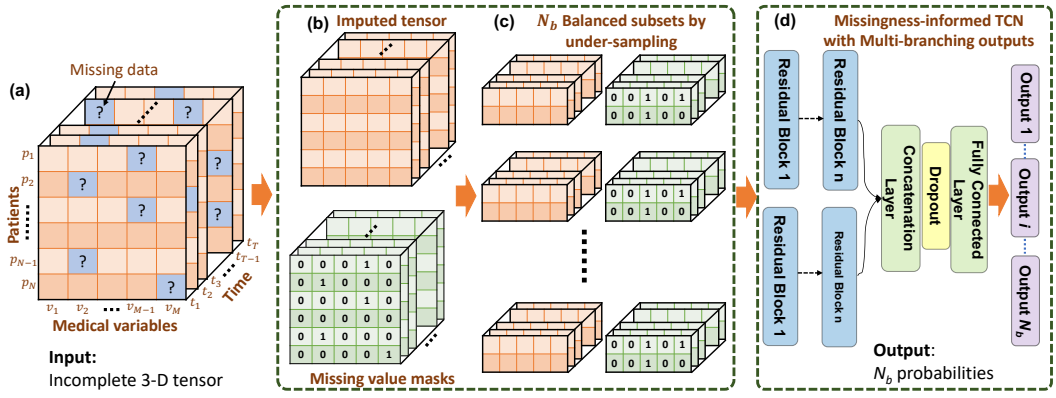


Figure C.16: (a) Multivariate longitudinal sequence with missing data; (b) Tensor completion-based data imputation; (c) Balanced subsets by under-sampling; (d) Missingness-informed TCN with multi-branching outputs.

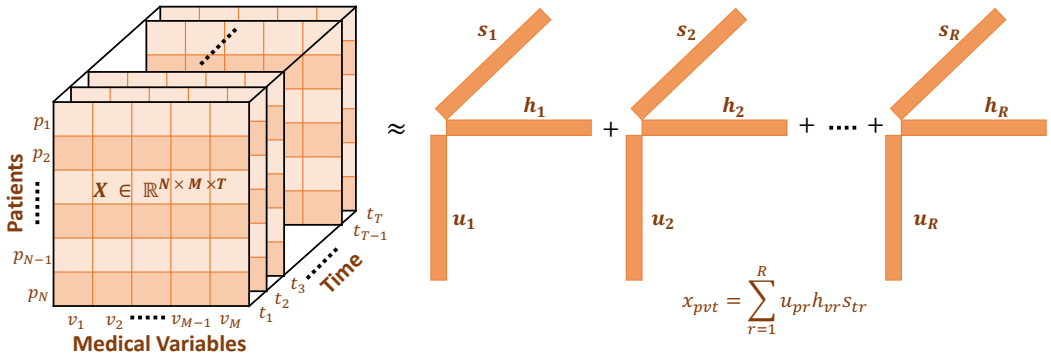


Figure C.17: Illustration of CP decomposition: $\boldsymbol{u}_r$, $\boldsymbol{h}_r$, and $\boldsymbol{s}_r$ denote the r-th column vector of matrix $\boldsymbol{U}$, $\boldsymbol{H}$, and $\boldsymbol{S}$, respectively.

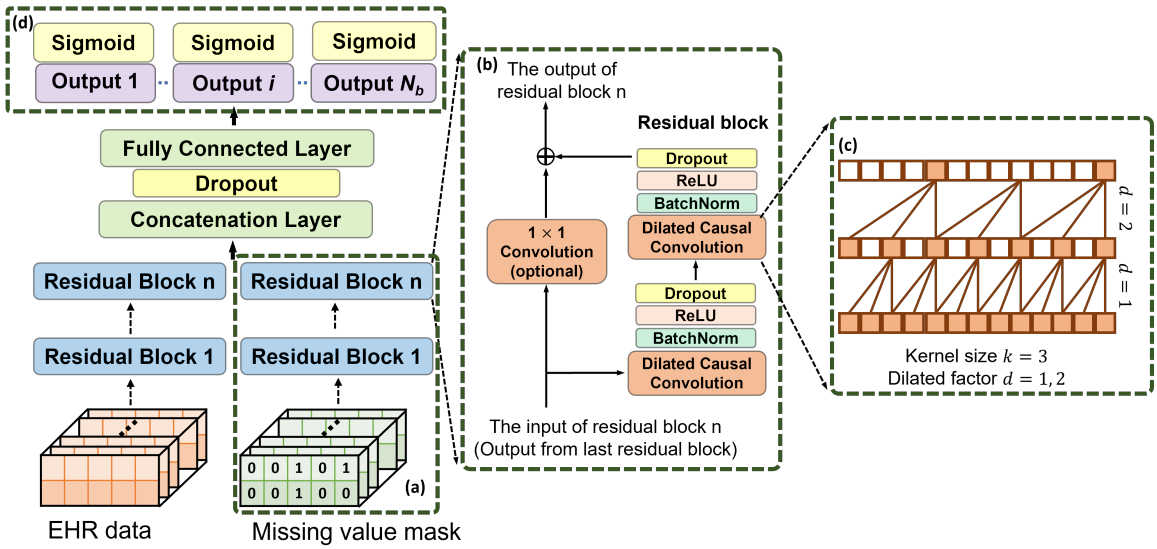


Figure C.18: Architectural detail of the missingness-informed TCN; (a) Missing value mask; (b) The architecture of a residual block; (c) The dilated causal convolution with dilation factors $d = 1, 2$ and kernel size $k_e = 3$; (d) The multi-branching output layer.

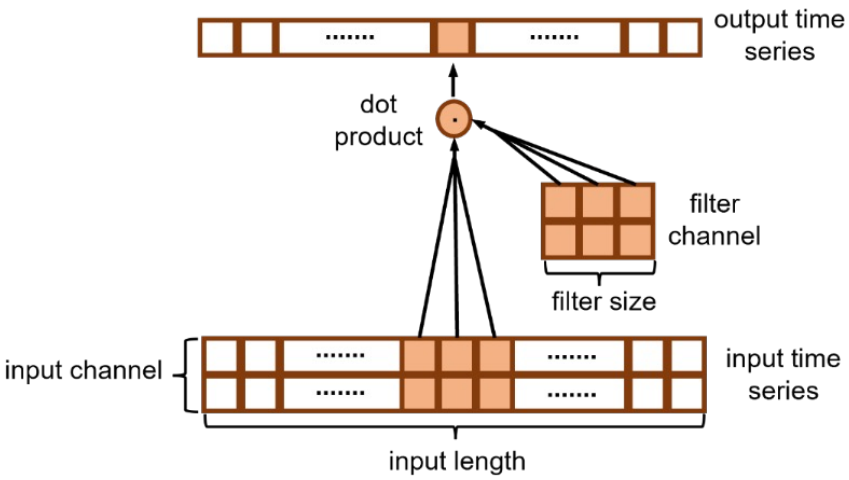


Figure C.19: Illustration of the multi-channel input in TCNs.

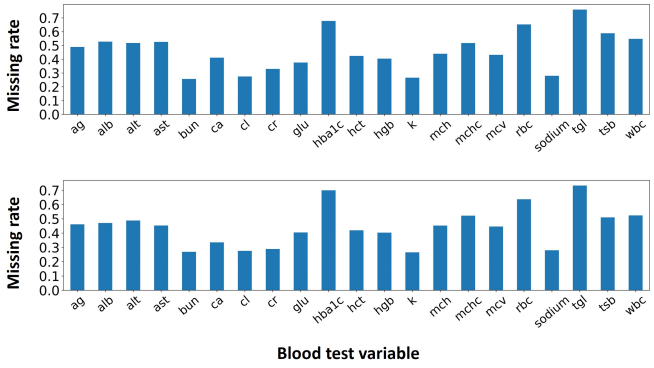


Figure C.20: Missing rates for blood test variables in the dataset from DR patients (top) and Non-DR patients (bottom).

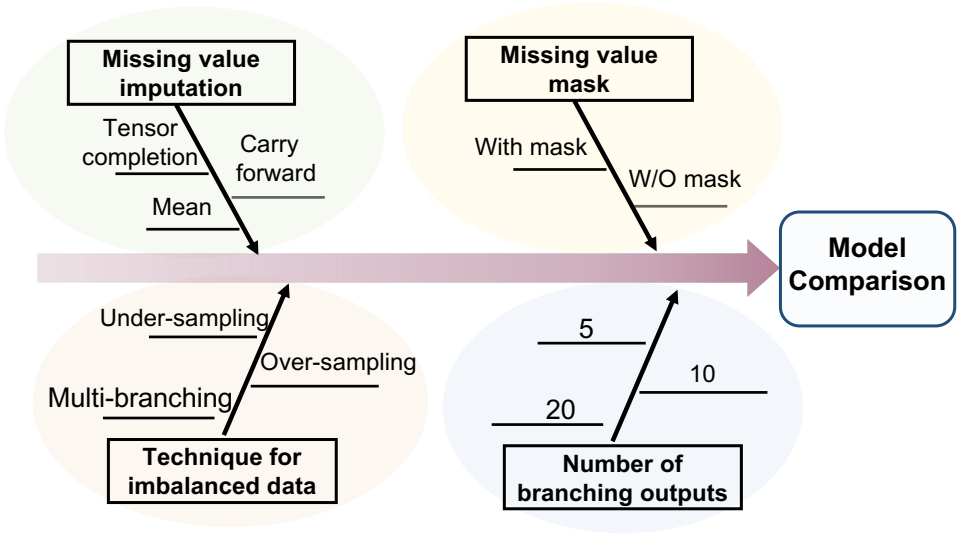


Figure C.21: Illustration of the experimental design to evaluate the performance of the proposed MB-TCN-TC method.

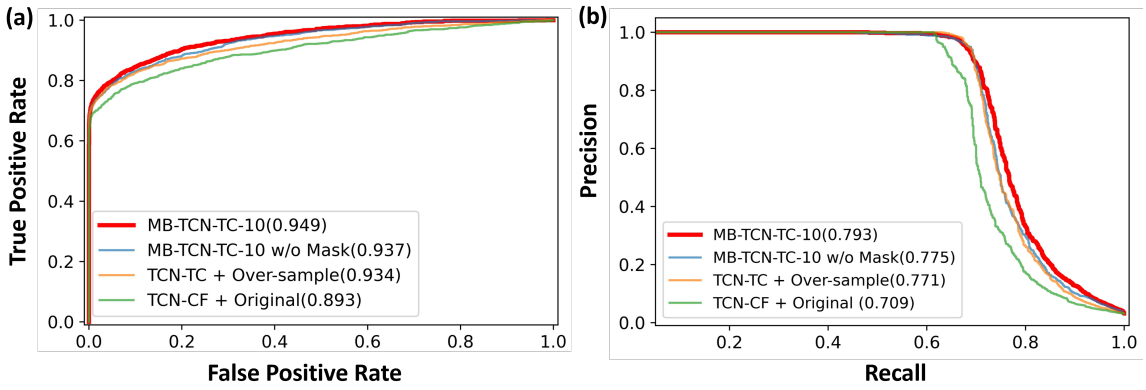


Figure C.22: (a) ROC and (b) PR curves for the proposed MB-TCN-TC with 10 branching outputs and other methods.

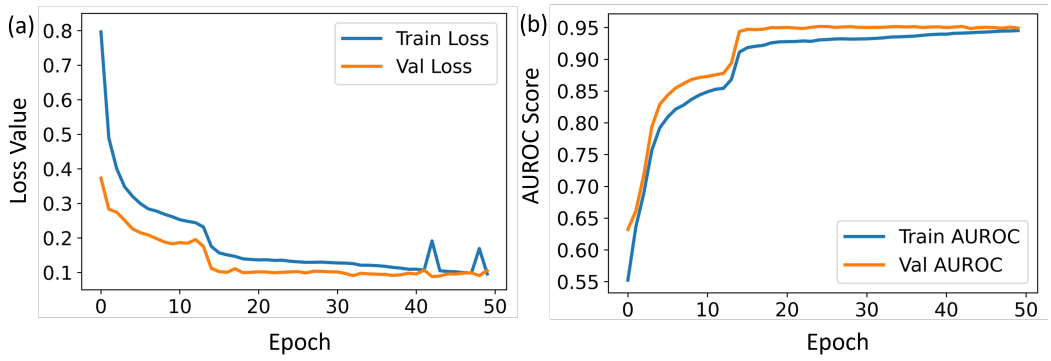


Figure C.23: The variations of training and validation (a) losses and (b) AUROC scores over epochs for our MB-TCN-TC-10 model.

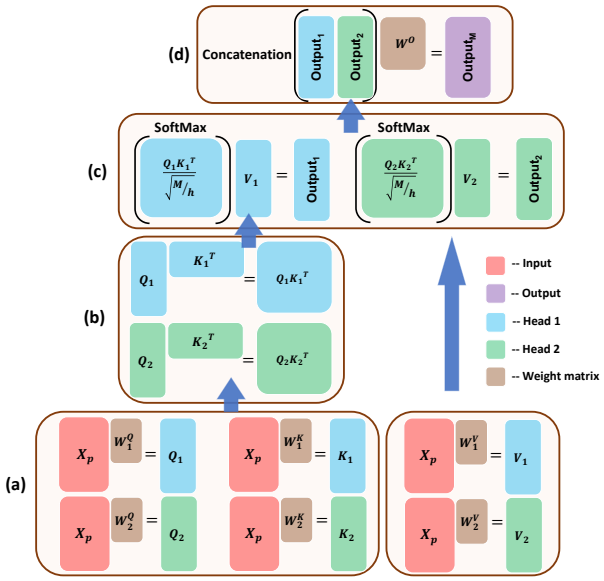


Figure C.24: Illustration of multi-head attention with 2 heads. (a) Derivation of Q, K, V matrices for each head with linear transformations of X_p: $Q_l = \mathcal{F}_Q^l(X_p) = X_p W_l^Q$, $K_l = \mathcal{F}_K^l(X_p) = X_p W_l^K$, $V_l = \mathcal{F}_V^l(X_p) = X_p W_l^V$, $l = \{1, 2\}$; (b) Compute key-query pairs for each head; (c) Compute attention weights from softmax function and further compute the output of heads 1 and 2 as the weighted sum of values; (d) The final output is the multiplication of the concatenation of all head outputs and another weight matrix W^O.

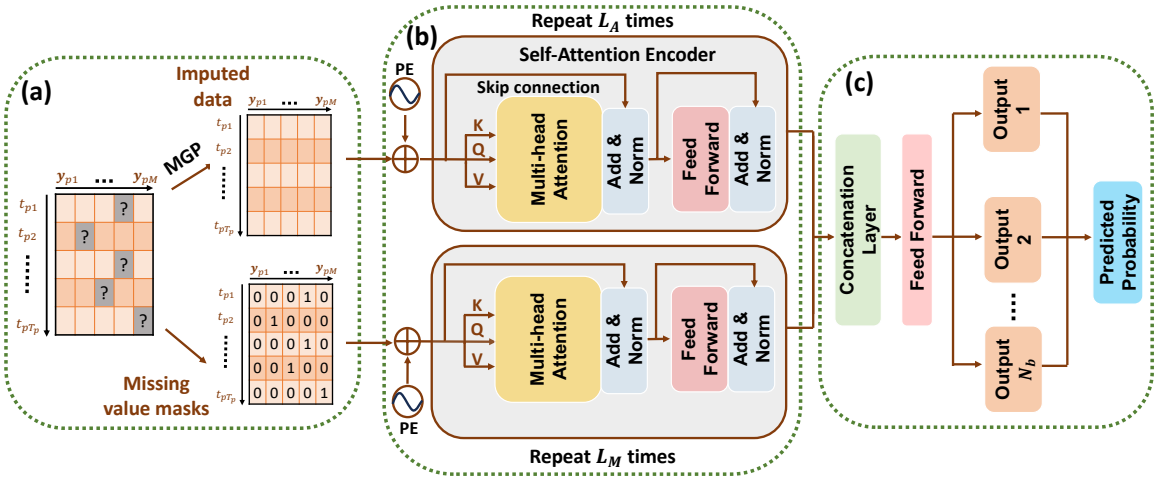


Figure C.25: The proposed MUSE-Net framework: (a) Multi-task Gaussian process for data imputation, and missing value masks generation; The imputed data and missing value masks are processed by the proposed MUSE-Net, which consist of (b) Missingness-aware Self-attention Encoder and (c) Multi-branching output layer.

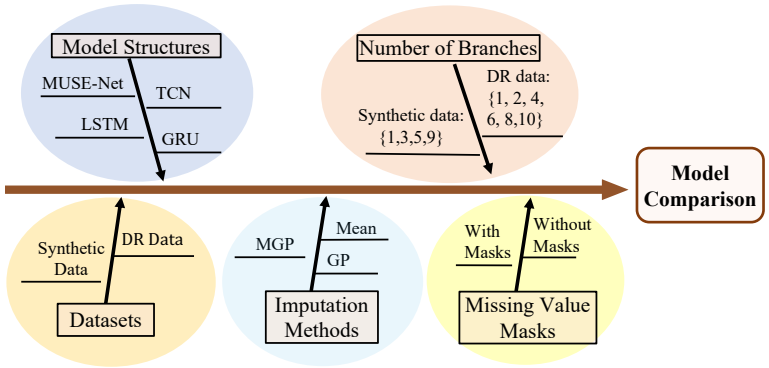


Figure C.26: Illustration of the experimental design to evaluate the performance of the proposed MUSE-Net.

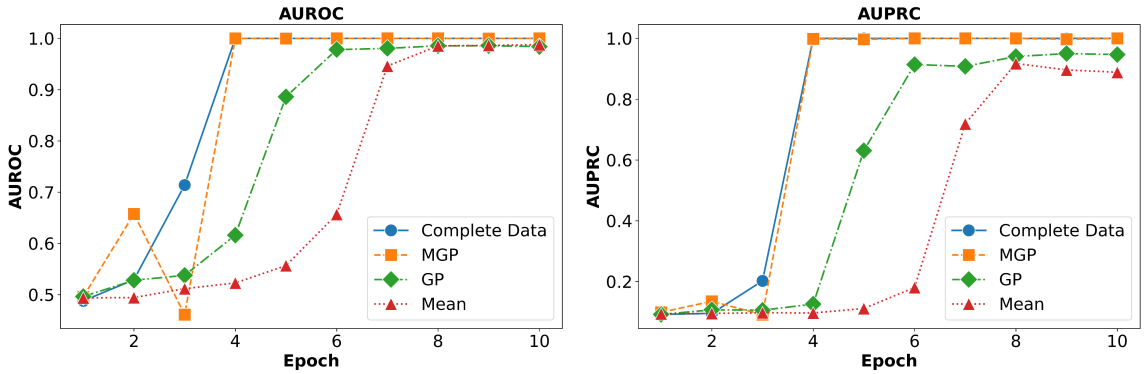


Figure C.27: AUROC and AUPRC scores across epochs for MUSE-Net-9 utilizing different imputation methods on the validation set of the simulated data.

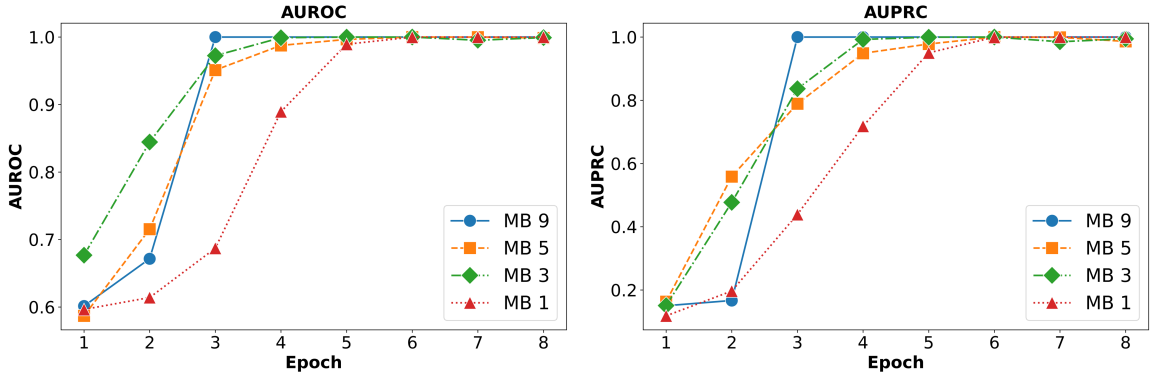


Figure C.28: AUROC and AUPRC scores across epochs for MUSE-Net with different numbers of MB outputs on the validation set of the simulated data imputed by MGP.

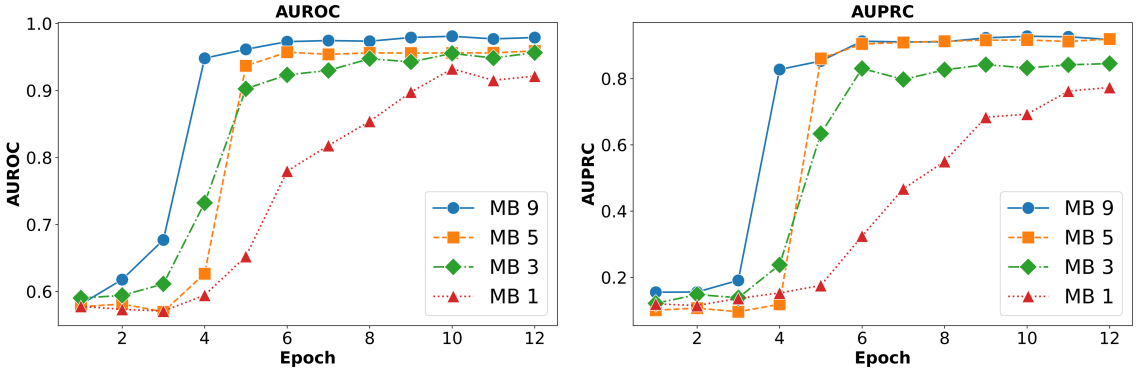


Figure C.29: AUROC and AUPRC scores across epochs for MUSE-Net with mean imputation and different numbers of MB outputs on the validation set of the simulated data.

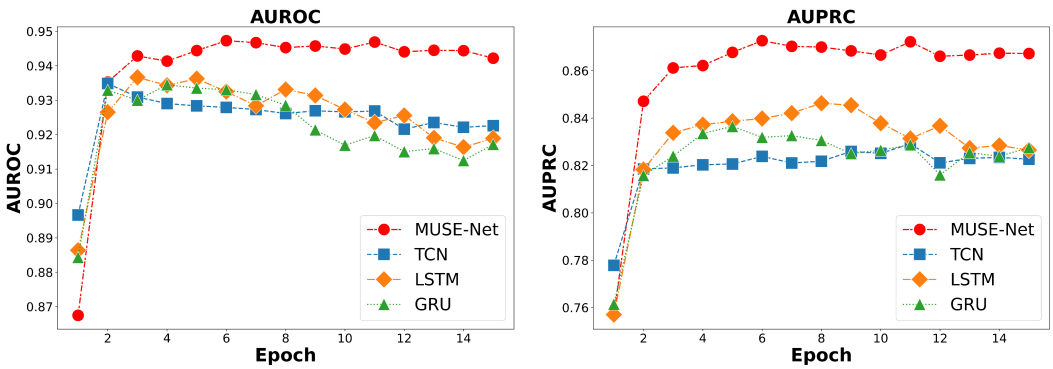


Figure C.30: AUROC and AUPRC across 15 epochs for our MUSE-Net and other benchmarks on the DR validation set.

Appendix D

Tables

Table D.1: The comparison of AUROC and AUPRC between the proposed MB-TCN method with existing deep learning-based methods.

	MB-AVE	MB-CM	MB-VO	Pure Model	Over-sampling	Under-sampling	Under-sampling DTW-HC
AUROC (TCN)	0.892	0.891	0.885	0.812	0.884	0.782	0.831
AUROC (LSTM)	0.878	0.879	0.878	0.831	0.872	0.834	0.824
AUPRC (TCN)	0.527	0.525	0.499	0.449	0.413	0.242	0.447
AUPRC (LSTM)	0.460	0.460	0.458	0.431	0.414	0.400	0.436

Table D.2: The comparison of AUROC and AUPRC scores of MB-TCN-AVE with missing-value mask and other imputation methods

	MB-TCN-AVE	Carry-forward imputation	Mean imputation
AUROC	0.892	0.821	0.821
AUPRC	0.527	0.397	0.389

Table D.3: The comparison of missing-value rate of vital signs between samples with ESTD$\geq$ 0.3 and others in the test dataset

	HR	O2Sat	Temp	SBP	MAP	DBP	Resp
std$\geq$ 0.3	0.1825	0.2134	0.7264	0.2113	0.2295	0.2118	0.2766
std$<$ 0.3	0.1381	0.1580	0.6686	0.1558	0.1661	0.1561	0.2248

Table D.4: The comparison of AUROC and AUPRC between the proposed MB-TCN method with existing literature.

Authors	Methods	AUROC	AUPRC
Wang and Yao	MB-TCN-AVE	0.892	0.527
Morrill et al.Morrill et al. (2019), 2019	Signature-based regression	0.868	-
Du et al.Du et al. (2019), 2019	Gradient boosted decision tree	0.851	0.116
Zabihi et al.Zabihi et al. (2019), 2019	XGBoost	0.833	-
Yang et al. Yang et al. (2019), 2019	XGBoost	0.847	-
Singh et al. Singh et al. (2019), 2019	XGBoost	0.838	-

Table D.5: The comparison of AUROC and AUPRC between MB-TCN and MB-TCN+DTW-HC

	MB-AVE + DTW-HC	MB-CM + DTW-HC	MB-VO + DTW-HC	MB-AVE	MB-CM	MB-VO
AUROC	0.900	0.900	0.895	0.892	0.891	0.885
AUPRC	0.560	0.559	0.522	0.527	0.525	0.499

Table D.6: The comparison of AUROC and AUPRC for the MB-TCN trained by dataset with different sample lengths

	Full length	60-hours	40-hours
AUROC	0.900	0.887	0.872
AUPRC	0.560	0.545	0.487

Table D.7: The comparison of AUROC and AUPRC scores of our MadeGAN and other models in the first-level anomaly detection.

	AUROC	AUPRC
MadeGAN	0.954	0.936
BeatGAN	0.941	0.922
MemAE	0.927	0.902
AE	0.906	0.886

Table D.8: The comparison of point metrics of MadeGAN and other methods in the first-level anomaly detection.

	Recall	Precision	f-score	Accuracy	Balanced-Accuracy
MadeGAN	0.954	0.856	0.902	0.885	0.913
BeatGAN	0.944	0.826	0.881	0.884	0.892
AE	0.894	0.760	0.821	0.784	0.834
MemAE	0.886	0.782	0.831	0.800	0.844

Table D.9: The comparison of AUROC and AUPRC between MB-Conv1D-Discrim and other approaches in the second-level classification

	S Class		V Class		F Class	
	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC
MB-Conv1D-Discrim	0.988	0.989	0.988	0.987	0.997	0.988
MB-Conv1D-Encoder	0.963	0.955	0.960	0.938	0.952	0.899
MB-Conv1D-Gene	0.983	0.981	0.985	0.938	0.952	0.953
MB-Conv1D	0.987	0.987	0.984	0.975	0.981	0.939
Conv1D-Discrim	0.985	0.981	0.984	0.987	0.963	0.911
Conv1D-Encoder	0.974	0.956	0.979	0.986	0.952	0.719
Conv1D-Gene	0.979	0.961	0.980	0.981	0.959	0.828
Conv1D	0.976	0.953	0.974	0.983	0.924	0.534

Table D.10: Confusion matrix on the test dataset from the second-level MB-Conv1d-Discrim method.

		Predicted labels		
		S	V	F
True labels	S	299	4	0
	V	4	687	10
	F	0	1	80

Table D.11: The comparison of point metrics between MB-Conv1d-Discrim and other approaches in the second-level classification

	Recall	Precision	f-score	Accuracy
MB-Conv1d-Discrim	0.976	0.976	0.975	0.976
MB-Conv1d-Encoder	0.881	0.881	0.879	0.891
MB-Conv1d-Generator	0.913	0.933	0.922	0.928
MB-Conv1d	0.931	0.936	0.934	0.939
Conv1d-Discrim	0.955	0.947	0.951	0.973
Conv1d-Encoder	0.786	0.869	0.819	0.917
Conv1d-Generator	0.888	0.877	0.881	0.934
Conv1d	0.752	0.801	0.770	0.896

Table D.12: The comparison of AUROC and AUPRC between MadeGAN and existing anomaly detection models.

Authors	Methods	AUROC	AUPRC
Wang et al.	MadeGAN	0.950	0.922
(Zhou et al., 2019), 2019	BeatGAN	0.945	0.911
(Shin et al., 2020), 2019	AnoGAN	0.948	-

Table D.13: The comparison of performance scores between our MB-Conv1d-Discrim and existing classification models.

Authors	AUROC	AUPRC	Recall	Precision	$f-$score	Accuracy
Wang et al. (2023)	0.989	0.984	0.964	0.967	0.965	0.967
Li and Zhou (2016)	-	-	0.647	0.475	-	0.946
Acharya et al. (2017)	-	-	0.960	0.915	-	0.940
Mousavi et al. (2021)	-	-	0.831	0.923	-	0.987
Kachuee et al. (2018)	-	-	-	-	-	0.934

Table D.14: The comparison of the proposed MadeGAN and other models in 1st-level anomaly detection for the long-term ST database

	AUROC	AUPRC	Recall	Precision	f-score	Accuracy
MadeGAN	0.984 (0.001)	0.986 (0.002)	0.956 (0.004)	0.954 (0.005)	0.947 (0.004)	0.955 (0.003)
AE	0.956 (0.004)	0.964 (0.001)	0.897 (0.01)	0.919 (0.007)	0.898 (0.004)	0.908 (0.004)
MemAE	0.965 (0.003)	0.965 (0.003)	0.904 (0.018)	0.940 (0.019)	0.912 (0.016)	0.921 (0.015)
BeatGAN	0.972 (0.002)	0.973 (0.002)	0.944 (0.014)	0.927 (0.015)	0.930 (0.004)	0.938 (0.004)

Table D.15: The comparison of MB-Conv1d-Discrim and other models in 2nd-level multi-classification for the long-term ST database

	AUROC	AUPRC	Recall	Precisio	f-score	Accuracy
MB-Conv1d-Discrim	0.991 (0.002)	0.953 (0.003)	0.917 (0.005)	0.955 (0.003)	0.920 (0.003)	0.972 (0.003)
MB-Conv1d-Encoder	0.957 (0.002)	0.801 (0.005)	0.800 (0.004)	0.758 (0.009)	0.770 (0.007)	0.905 (0.006)
MB-Conv1d-Generator	0.962 (0.001)	0.849 (0.004)	0.799 (0.001)	0.842 (0.002)	0.818 (0.003)	0.925 (0.002)
MB-Conv1d	0.987 (0.003)	0.912 (0.02)	0.876 (0.02)	0.953 (0.004)	0.920 (0.003)	0.972 (0.003)
Conv1d-Discrim	0.988 (0.001)	0.943 (0.007)	0.909 (0.032)	0.933 (0.019)	0.918 (0.17)	0.963 (0.17)
Conv1d-Encoder	0.956 (0.006)	0.803 (0.032)	0.837 (0.063)	0.690 (0.021)	0.715 (0.029)	0.932 (0.004)
Conv1d-Generator	0.954 (0.007)	0.826 (0.022)	0.847 (0.038)	0.747 (0.069)	0.765 (0.061)	0.916 (0.01)
Conv1d	0.969 (0.007)	0.884 (0.013)	0.888 (0.006)	0.850 (0.029)	0.866 (0.015)	0.968 (0.013)

Table D.16: Existing machine learning methods for DR prediction using EHRs and their limitations

	Authors	Methodology	Limitations
1	Semeraro et al. (2011)	Cox Proportional Hazard Model	1). Linearity assumption not generally valid for real-world EHRs 2). Not applicable to model medical data with missing values and imbalanced issue.
2	Ogunyemi and Kermah (2015)	Ensemble Learning	1). Not able to model longitudinal EHRs. 2). Not able to address missing values and imbalanced issues in longitudinal EHRs.
3	Piri et al. (2017)	Logistic Regression Random Forest Neural Network	
4	Wang et al. (2021)	Random Forest	
5	Chen et al. (2022)	Temporal Neural Network	Not able to exploit the multi-dimensional correlation for missing value imputation.
6	Wang et al. (2023a)	Bayesian Finite Mixture Model	1). Not able to model longitudinal EHRs. 2). Not able to exploit the multi-dimensional correlation for missing value imputation.

Table D.17: The comparison of performance scores between our MB-TCN-TC with 10 branching outputs and other methods.

Model Name	True Positive	False Positive	True Negative	False Negative	Recall	Precision	F1 Score
MB-TCN -TC-10	1,880	722	79,558	680	0.734	0.723	0.728
MB-TCN -TC-10 w/o Mask	1,852	794	79,486	708	0.723	0.699	0.711
TCN-TC + Over-sample	1,844	822	79,458	716	0.720	0.691	0.705
TCN-CF + Original	1,800	1,540	78,740	760	0.703	0.539	0.610

Table D.18: The performance comparison in AUROC and AUPRC between our MB-TCN-TC model and other imputation methods with and w/o missing value mask. Note that all the network models contain an MB layer with 10 branching outputs.

Model: MB-TCN-10	Tensor Decomposition		Carry Forward imputation		Mean imputation	
	Mask	w/o Mask	Mask	w/o Mask	Mask	w/o Mask
AUROC	0.949	0.937	0.925	0.915	0.916	0.920
AUPRC	0.793	0.775	0.734	0.724	0.726	0.727

Table D.19: The comparison between MB-TCN-TC-10 and other methods in handling the imbalanced data issue.

Model	MB-TCN-TC-10	TCN-TC + under-sampling	TCN-TC + over-sampling	TCN + original data
AUROC	0.949	0.933	0.934	0.923
AUPRC	0.793	0.756	0.771	0.772

Table D.20: Performance comparison between our MB-TCN-TC-10 method and traditional machine learning methods.

Model Name	True Positive	False Positive	True Negative	False Negative	AUROC	AUPRC	Recall	Precision	F1 Score
MB-TCN -TC-10	1,880	722	79,558	680	0.949	0.793	0.734	0.723	0.728
Original TCN	1,800	1,540	78,740	760	0.893	0.709	0.703	0.539	0.610
Random forest	1,805	2,382	77,898	755	0.907	0.728	0.705	0.431	0.535
XGBoost	1,813	2,345	77,935	747	0.915	0.734	0.708	0.436	0.540
Neural network	1,840	4,730	75,550	720	0.900	0.716	0.719	0.280	0.404
Logistic regression	1,772	2,669	77,611	788	0.882	0.709	0.692	0.399	0.507

Table D.21: The variation of AUROC and AUPRC generated by MB-TCN-TC with different numbers of MB outputs.

Model	MB-TCN-TC-5		MB-TCN-TC-10		MB-TCN-TC-20	
	Mask	w/o Mask	Mask	w/o Mask	Mask	w/o Mask
AUROC	0.948	0.941	0.949	0.944	0.934	0.931
AUPRC	0.791	0.780	0.793	0.787	0.771	0.760

Table D.22: Architectures of MUSE-Net and other benchmarks for DR dataset.

	Number of model parameters	Number of layers (blocks)	Number of outputs in MB layer	Optimizer
MUSE-Net	4,282	2	10	Adam with decoupled weight decay (AdamW) Loshchilov and Hutter (2017)
GRU	4,333			
LSTM	4,686			
TCN	4,396			

Table D.23: AUROC and AUPRC scores for MUSE-Net and other benchmarks on the DR test set.

	MUSE-Net	TCN	LSTM	GRU
AUROC	0.949	0.928	0.919	0.932
AUPRC	0.883	0.839	0.825	0.844

Table D.24: The comparison of AUROC and AUPRC between our MUSE-Net-10 with MGP imputation and other imputation methods on the DR test set.

Model: MUSE-Net	MGP imputation		GP imputation		Mean imputation	
	Mask	w/o Mask	Mask	w/o Mask	Mask	w/o Mask
AUROC	0.949	0.910	0.938	0.909	0.917	0.901
AUPRC	0.883	0.814	0.833	0.799	0.819	0.770

Table D.25: AUROC and AUPRC scores for MUSE-Net+MGP with different MB numbers in the DR test set.

	MB 10	MB 8	MB 6	MB 4	MB 2	MB 1
AUROC	0.949	0.948	0.947	0.947	0.940	0.935
AUPRC	0.883	0.884	0.878	0.879	0.870	0.860

Vita

Zekai Wang received a B.E. in Civil Engineering from Wuhan University of Technology, China; an M.S. in Applied Statistics and Data Analytics from Southern Methodist University, USA; and an M.S. in Industrial Engineering and Management from Oklahoma State University, USA. He is currently a Ph.D. candidate in the Department of Industrial Systems Engineering at the University of Tennessee, Knoxville. His research focuses on advanced data analytics, deep learning, and health informatics.

Zekai has been recognized with several prestigious awards, including the Best Poster Award at the 2022 IISE QCRE Session Best Student Poster Competition, the Best Healthcare Automation Paper Award at the 2022 IEEE 18th International Conference on Automation Science and Engineering, and third place in the 2021 INFORMS Poster Competition. He is a student member of IISE, INFORMS, and IEEE. After graduation, Zekai aims to pursue a career in healthcare research and innovation, with a particular focus on data-driven methods for decision-making and systems diagnosis. Additionally, he is passionate about contributing to education by mentoring and inspiring the next generation of scholars and professionals.