Flinders University

Deep Imputation-Prediction Networks for Health Risk Prediction using Electronic Health Records

By

Lorenzo Yuxi Liu

Thesis
Submitted to Flinders University
for the degree of

Doctor of Philosophy
College of Science and Engineering

14/02/2024

Abstract

In the era of information technology advancement, data are being collected and accumulated at a dramatic pace. The ability to extract useful hidden knowledge from a large amount of data and make decisions based on this knowledge has offered significant benefits to business organizations. Predicting health risks of patients using Electronic Health Records (EHR) has attracted considerable attention in recent years, especially with the development of deep learning techniques. Health risk refers to the probability of the occurrence of a specific health outcome for a specific patient. The predicted risks can be used to assist the decision-making of healthcare specialists. EHRs contain a chronological set of medical records, and within each medical event, there is a set of clinical/medical activities. Various risk prediction models have been developed and introduced to health risk prediction on EHR data.

Predictive analytics is the process of analyzing large datasets to discover patterns and meaningful insights where machine learning and statistical methods are applied to build predictive models for decision support. However, EHR data has its unique characteristics, such as high dimensionality, sparsity, irregularity, heterogeneity, random errors, temporality, and systematic biases. It is technically challenging to apply existing risk prediction models to EHR data with a high degree of irregularity, which contains many missing values and varying time intervals between medical records. Existing studies on EHR data irregularity have been focused on the provision of deep learning-based solutions. These studies impute missing values by incorporating deep neural networks to learn variable correlations and introduce time decay mechanisms to capture the impact of varying time intervals. The complete data matrices obtained from the imputation task are used for downstream risk prediction tasks. However, the existing imputation methods lead to less than desirable prediction accuracy. Further improvements in risk prediction models are necessary before they can be adopted for real-world applications.

This thesis investigates and develops new risk prediction models for handling the irregularity of EHR data and predicting patients' health risks. We proposed compound density networks, an end-to-end, novel, and robust model to train the imputation method and prediction model simultaneously within a single framework. The purpose is to handle missing values in EHR data, enhance imputed values' reliability, and quantify their uncertainties. We then developed deep imputation-prediction networks by extending the compound density network to perform imputation and prediction in EHR data. The focus is to capture the impact of varying time intervals. We further introduced multi-task learning to perform risk prediction tasks by incorporating the imputation task as an auxiliary task while carrying out both simultaneously.

To improve imputation performance, we considered patient similarity via incorporating graph analysis techniques. We proposed contrastive learning-based imputation prediction networks, which mainly impute missing values in EHR data by exploiting similar patient information as well as patients' personal contextual information. Similar patients are generated from patient similarity computing during stratification modeling and analysis of patient graphs. We further proposed contrastive graph similarity networks, which incorporate graph contrastive learning in representation learning for EHR data. The graph similarity networks were extended to multi-graph neural networks. This enables the learning of multiple graph structures from input EHR data, which aggregates the information from similar patients to offer a richer representation of the patient and allows the extraction of patient health context for both imputation and prediction tasks. The result is an optimised graph structure that incorporates the characteristics of these graphs with attention mechanisms.

We empirically investigated the proposed deep imputation-prediction models on two tasks using the Medical Information Mart for Intensive Care Database and eICU Collaborative Research Database. These tasks are (i) multivariate clinical time series imputation and (ii) in-hospital mortality risk prediction. The empirical results indicated that our models outperform state-of-the-art imputation-prediction models by significant margins.

The strength of these models lies in their ability to present transparency and interpretability of the decision process and provide the estimation of epistemic and aleatoric uncertainties of the model decisions.

Our research work made novel contributions to the improvement of methodologies for dealing with the irregularity of EHR data in the context of health risk prediction. These methodologies are potentially applicable to other medical applications such as hospital length of stay prediction and phenotype classification.

Acknowledgements

Throughout my PhD journey, I have received a great deal of support and assistance from many wonderful people.

First and foremost, I would like to express my deepest appreciation to my supervisors for guiding me with research. My principal supervisor, Professor Shaowen Qin, has been a continuing source of support and guidance throughout. I am extremely thankful for her words of encouragement, advice, and reassurance when things are tough. I am also thankful to my co-supervisor, Dr Richard Leibbrandt, for patiently offering encouraging feedback.

I would also like to thank Professor Flora Salim and Dr Antonio Jimeno Yepes for providing generous guidance, encouragement, and collaboration.

I sincerely thank my friend Zhenhao Zhang, who is studying MPhil in bioinformatics. Thank you for supporting my research work, lending me a listening ear to vent my frustrations whenever needed, and reassuring me that my feelings were normal.

I never imagined I would be completing my PhD during a global pandemic, and this endeavor would not have been possible if not for the unfailing support from my family. A special thanks to my family for unconditional love and the countless sacrifices made for me, for constantly watching out for me, for always cheering me on and loving me. I hope I have made my family proud.

Lorenzo Yuxi Liu
Wednesday 14th February, 2024

Declaration

I, Lorenzo Yuxi Liu, declare that this thesis titled, "Deep Imputation-Prediction Networks for Health Risk Prediction using Electronic Health Records" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Lorenzo Yuxi Liu
Wednesday 14th February, 2024

Publications

Publications included in this thesis

Yuxi Liu, Shaowen Qin, Zhenhao Zhang, and Wei Shao. Compound density networks for risk prediction using electronic health records. In *2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 1078–1085. IEEE, 2022.

Yuxi Liu, Zhenhao Zhang, and Shaowen Qin. Deep imputation-prediction networks for health risk prediction using electronic health records. In *2023 International Joint Conference on Neural Networks (IJCNN)*, pages 1–9. IEEE, 2023.

Yuxi Liu, Zhenhao Zhang, Shaowen Qin, Flora D Salim, Antonio Jimeno Yepes. Contrastive Learning-based Imputation-Prediction Networks for In-hospital Mortality Risk Modeling using EHRs. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 428–443. Springer, 2023.

Submitted manuscripts included in this thesis

Yuxi Liu, Zhenhao Zhang, Shaowen Qin, Flora D Salim, Jiang Bian and Antonio Jimeno Yepes. Fine-grained Patient Similarity Measuring using Contrastive Graph Similarity Networks. In *2024 IEEE 12th International Conference on Healthcare Informatics (ICHI)*, Under Review.

Yuxi Liu, Zhenhao Zhang, Shaowen Qin, Flora D Salim. A Multi-Graph Fusion Framework for Patient Representation Learning. In *2024 IEEE 12th International Conference on Healthcare Informatics (ICHI)*, Under Review.

Yuxi Liu, Zhenhao Zhang, Shaowen Qin, Jiang Bian. Multi-Task Deep Neural Networks for Irregularly Sampled Multivariate Clinical Time Series. In *2024 IEEE 12th International Conference on Healthcare Informatics (ICHI)*, Under Review (This paper is an extended version of the publication [1]).

Other publications and submitted manuscripts during candidature

[2] **Yuxi Liu**, Zhenhao Zhang, Shaowen Qin, Flora D Salim, Antonio Jimeno Yepe,

Shen Jun, Jiang Bian. Hypergraph Convolutional Networks for Fine-grained ICU Patient Similarity Analysis and Risk Prediction. In *2024 IEEE 12th International Conference on Healthcare Informatics (ICHI)*, Under Review.

[3] **Yuxi Liu**, Zhenhao Zhang, Campbell Thompson, Richard Leibbrandt, Shaowen Qin, and Antonio Jimeno Yepes. Stacked attention-based networks for accurate and interpretable health risk prediction. In *2023 International Joint Conference on Neural Networks (IJCNN)*, pages 1-8. IEEE, 2023.

[4] **Yuxi Liu**, Zhenhao Zhang, and Shaowen Qin. Neuralhmm: A deep markov network for health risk prediction using electronic health records. In *2023 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2023.

[1] **Yuxi Liu**, Shaowen Qin, Antonio Jimeno Yepes, Wei Shao, Zhenhao Zhang, and Flora D Salim. Integrated convolutional and recurrent neural networks for health risk prediction using patient journey data with many missing values. In *2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 1658–1663. IEEE, 2022.

[5] **Yuxi Liu**, Zhenhao Zhang, Antonio Jimeno Yepes, and Flora D Salim. Modeling long-term dependencies and short-term correlations in patient journey data with temporal attention networks for health prediction. In *Proceedings of the 13th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, pages 1–10, 2022.

[6] **Yuxi Liu**, Shaowen Qin. An interpretable machine learning approach for predicting hospital length of stay and readmission. In *Advanced Data Mining and Applications: 17th International Conference, ADMA 2021, Sydney, NSW, Australia, February 2–4, 2022, Proceedings, Part I*, pages 73–85. Springer, 2022.

[7] **Yuxi Liu** and Shaowen Qin. Hospital readmission prediction via personalized feature learning and embedding: A novel deep learning framework. In *Advances and Trends in Artificial Intelligence. Theory and Practices in Artificial Intelligence: 35th International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2022, Kitakyushu, Japan, July 19–22, 2022, Proceedings*, pages 89–100. Springer, 2022.

[8] **Yuxi Liu**, Shaowen Qin, and Zhenhao Zhang. Epidemic modeling of the spatiotemporal spread of covid-19 over an intercity population mobility network. In *Advances and Trends in Artificial Intelligence. Theory and Practices in Artificial Intelligence: 35th International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2022, Kitakyushu, Japan, July 19–22, 2022, Proceedings*, pages 147–159. Springer, 2022.

Lorenzo Yuxi Liu
Wednesday 14th February, 2024

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter briefly introduces this work, including the background of this work, major challenges, research gaps, research goals, and thesis organization.

1.1 Background

Healthcare involves a process of the diagnosis, treatment, and prevention of disease, injury, and other physical and mental impairments in people. The healthcare industry, particularly in developed countries, is growing at a rapid pace. However, due to the complexity of healthcare, the healthcare industry lags behind other industries in implementing intelligent decision systems.

The past decade has seen the rapid development of digital health technologies. Digital health technologies aim to promote human health and improve healthcare systems. For example, it includes smartphone apps [9,10], wearable devices [11,12], and remote patient monitoring platforms [13,14].

Accelerating the adoption of proven digital health technologies into routine care has the potential to revolutionize human health by improving healthcare delivery and reducing hospital costs. With the adoption of digital health systems, large amounts of Electronic Health Records (EHRs) are available, but the major problem is how to translate the existing information into useful knowledge and decision support tools to guide clinical practice.

Data mining has great potential for the healthcare industry. For example, it plays a crucial role in detecting healthcare fraud and abuse [15–17], supporting drug discovery

and development [18–20], detecting the early stage of a disease [21–23], and providing clinical decision support systems [24–26], etc.

Data mining in healthcare is widely being used to predict patients' health risks [1,3,4]. Health risk refers to the probability of the occurrence of a specific health outcome for a specific patient. The predicted risks of a specific health outcome can be used to support decision-making by healthcare professionals and improve healthcare delivery. Interest in health risk prediction has been increasing, especially with the availability of a large amount of EHR data.

EHRs are an increasingly common data source for health risk prediction. EHRs contain patient health information, such as administrative and billing data, patient demographics, progress notes, vital signs, medical histories, diagnoses, medications, lab test results, etc. Integrated modeling of EHR data forms a chain of data that can be used for predictive analysis. This enables researchers to develop EHR-based prediction systems and perform health risk predictions.

Machine learning model is an important component in EHR-based prediction systems and plays a key role in predictive modeling. Machine learning models are built with machine learning algorithms, such as k-nearest neighbors, support vector machines, random forests, gradient boosting machines, and artificial neural networks trained using labeled, unlabeled, or mixed data. Examples of representative applications include disease risk prediction [27–30], in-hospital mortality risk prediction [5, 31–33], and risk-of-readmission prediction [7, 34–36].

EHR data has its own issues, such as high dimensionality, temporality, sparsity, heterogeneity, irregularity, random errors, systematic noise, random bias, etc [37]. In this thesis, we focus on the irregularity of EHR data, which contains many missing values and varying time intervals between medical records. We take the medical records of two anonymous patients from the publicly available Medical Information Mart for Intensive Care (MIMIC-III) database [38] and present these in Figure 1.1 as an example. Figure 1.1 clearly indicates the irregularity problem, including many missing values and varying time intervals between medical records. As typical machine learning models are not developed for EHR data with a high degree of irregularity, they are not suitable for EHR-based patient health risk predictions.

In this thesis, we investigate and develop new risk prediction models for handling the irregularity of EHR data and predicting patients' health risks.

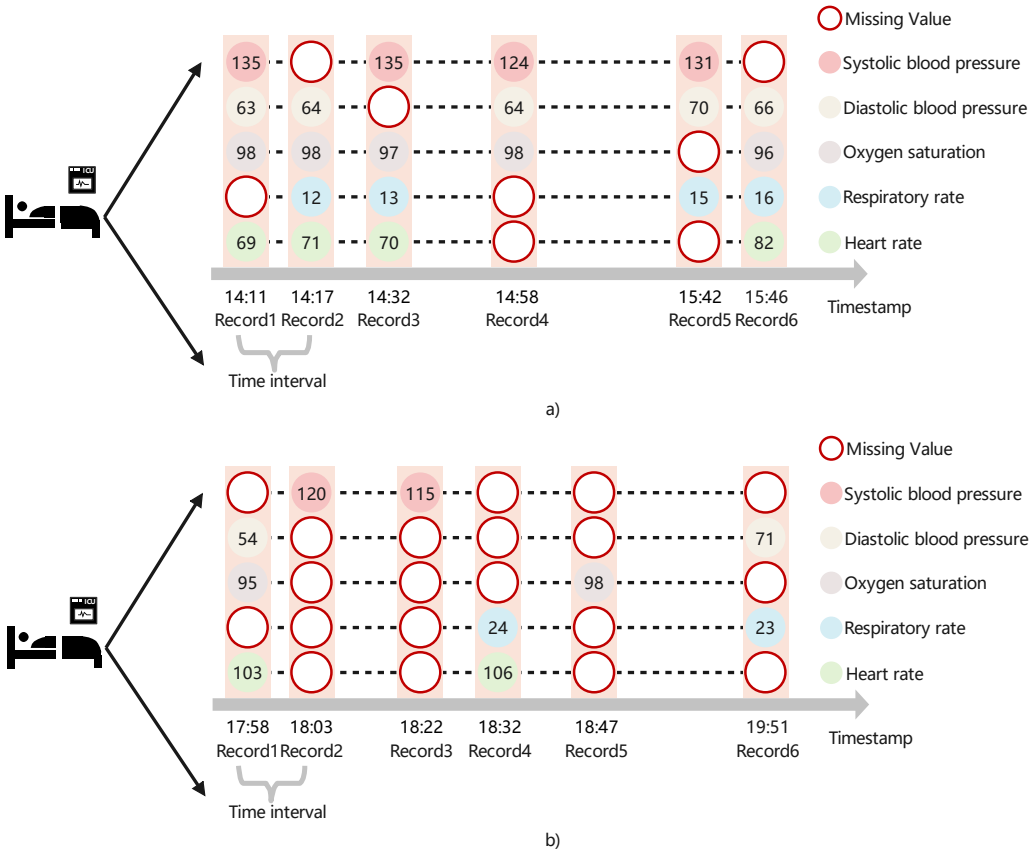

Figure 1.1: Illustration of medical records of patients A and B.

1.2 Major Challenges

1.2.1 Many missing values and Varying time intervals

With EHR datasets, missing values are likely attributed to patient symptoms and reflect treatment decisions. Different physiological variables (e.g., glucose, heart rate, and respiratory rate as shown in Figure 1.1) are examined at different times depending on a patient's health conditions. When a certain symptom disappears, corresponding variables are no longer examined. This would lead to missing values. Therefore, the patterns of missing data in patients' medical records may contain important information, e.g., different categories of missing data may reflect a typical clinical symptom, affecting both time-independent and dependent variable relationships. More accurate, meaningful, and reliable imputation of missing values would allow us to uncover such important informa-

tion that in turn can lead to a model with better prediction performance.

Recent studies focus on modeling the variable correlations in patient medical records with deep learning techniques to generate imputed values for missing values [39–47]. After obtaining the complete data matrices from the imputation task, the complete data matrices are used as input for risk prediction tasks [43, 44, 48–51]. However, not enough attention was given to the reliability of the imputed values by these approaches. Such less reliable imputed values may lead to biased prediction outcomes, especially when existing prediction models are directly applied to predict health risks.

Due to changes in a patient's underlying health condition, physiological variables being examined and the time intervals between examinations vary. Accordingly, the variation pattern of physiological variables in diverse time intervals plays a vital role in understanding a patient's underlying health condition and predicting the patient's future condition. Much of the literature focused on the provision of time decay mechanisms to take the irregular interval of examinations into consideration when imputing missing values [40, 43, 44, 46, 52]. Although previous studies have recognized the role of the time decay mechanism, research has yet to systematically evaluate the use of time decay mechanisms on the feature/variable level of inputs (i.e., physiological variables in the patient's medical record).

Besides, a patient's health status can become 'healthier', 'deteriorating', or recurrent. When predicting health outcomes, we should automatically include learning of the impact of the previous 48 hours of patient data on the prognosis (e.g., in-hospital mortality risk prediction [5, 53, 54]). If the predictive model found an association between the previous and current physiological variables, the previous physiological variables become critical indicative variables regardless of how long ago these were collected, which should be given sufficient consideration in the prediction model. However, this process is ignored by existing work.

1.2.2 Deep Neural Networks with Multi-Task Learning

Current studies have investigated three modes of imputation-prediction processing. However, there are certain drawbacks associated with the use of these modes. (i) The first is to consider imputation and prediction as two separable steps [43, 44, 46, 55–58]. Although promising prediction performance has been demonstrated, these prediction models have not attempted to learn the impact of the patterns of missing data in EHR data [39]. This may lead to suboptimal prediction performance [59]. As a better alternative, imputation and prediction can be tuned together within an end-to-end learning framework rather than

be separated into two parts. (ii) This is the second mode. Despite its efficacy, existing architectures for such modes are specifically proposed for improving risk prediction performance [39, 60–62]. When used for imputation and prediction tasks, the architecture treats both as separate optimization tasks, which essentially is not different from the first mode. (iii) The third imputation-prediction processing mode is similar to that used by the second, with the difference that the objective of the third is to simultaneously perform both imputation and prediction tasks [40, 42, 51, 63–65]. However, imputation and prediction tasks may lead to competition due to the shared parameter problem, as illustrated during multi-task learning for optimization in some studies [66–68]. This kind of optimization could also lead to suboptimal imputation and prediction results.

This work proposes to construct a single deep learning framework based on multi-task learning that performs risk prediction tasks while incorporating the imputation task as an auxiliary task. The benefit of implementing the imputation task as an auxiliary task is that such an approach can improve risk prediction performance rather than competing with it. With the construction of multi-task learning, the framework would simultaneously generate imputation and prediction results.

1.3 Research Gaps

1.3.1 Imputation of missing data in EHRs based on patients' similarities

Patient similarity is defined as the similarity between two EHR patients' journey data. The EHR patient journey data includes a series of vital sign measurements, clinical history, laboratory tests, etc. Patient similarity analysis is important for a wide range of medical applications. For example, many researchers have focused on the provision of patient similarity-based retrieval service [69, 70], and patient similarity-based model for diagnostic prediction [71] and prognostic decision support [72].

None of the existing works explicitly consider patient similarity via stratification of EHR data on the imputation task, which leads to suboptimal imputation performance. Patient stratification refers to the method of dividing a patient population into subgroups based on specific disease characteristics and symptom severity. Patients in the same subgroup generally had more similar health trajectories. This work proposes to impute missing values in patient data using information from the subgroup of similar patients rather than the entire patient population.

1.4 Research Goals

This section discusses the research goals of this thesis. At the highest level, this research aims to investigate and develop new risk prediction models for healthcare decision support.

The main focus is on classification based health risk prediction. This includes identifying high-risk individual patients based on their historical EHR data and generating corresponding risk scores or probabilities for reference by healthcare professionals.

The second focus is on the irregularity of EHR data. This includes developing new machine learning methods to address the irregularity of EHR data with the hope of integrating them into the context of classification based health risk prediction and improving the overall performance.

A summary of the goals of our research is listed as follows.

- The primary goal is to design and implement a machine learning model for classification based health risk prediction as the first step of the research. In particular, our approach needs to address missing values in EHR data for effective predictive modeling. Moreover, the handling of uncertainty quantification needs to be incorporated into the machine learning model. To handle uncertainty quantification, we focused on a mixture density network that learns about the impact of aleatoric uncertainty and epistemic uncertainty.

- The second goal is to incorporate the handling of varying time intervals into the machine learning model. To handle varying time intervals, we focused on a time-decay attention approach that captures the variation pattern of input variables at the time dimension and adaptively enhances the temporal representation of each pattern with adjustable weights. It also examines the association between input variables to identify critical indicative variables regardless of how long ago the associated event happened.

- The third goal is to develop an imputation-prediction approach that is capable of imputing missing values in patient data using information from the subgroup of similar patients rather than the entire patient population.

- The last goal is to explore the processing mode of imputation and prediction. In particular, we approached a multi-task learning problem for simultaneously generating imputation and prediction results by performing risk prediction tasks while incorporating the imputation task as an auxiliary task.

1.5 Thesis organization

The remainder of this thesis is organised into nine chapters, from Chapter 2 to Chapter 10.

Chapter 2 – Literature Review
Chapter 2 reviews and summarizes relevant literature, focusing on the application of traditional statistical, machine learning, and deep learning missing data imputation techniques.

Chapter 3 – Compound Density Networks
Chapter 3 introduces compound density networks, an integrated end-to-end approach to allow the imputation method and prediction model to be tuned together within a single framework. The proposed approach focused on the provision of accurate and reliable prediction results with EHR data. The results of this chapter have been published [62].

Chapter 4 – Attention-Based Bidirectional Recurrent Neural Networks
Chapter 4 introduces a novel deep imputation-prediction network to carry out imputation and prediction tasks with EHR. The proposed approach focused on the provision of both imputation and prediction results with EHR data. The results of this chapter have been published [73].

Chapter 5 – Contrastive Neural Networks
Chapter 5 introduces a novel contrastive learning-based imputation-prediction network to carry out imputation and prediction in EHR data. It integrates graph representation learning and contrastive learning in representation learning for EHR data. The results of this chapter have been published [74].

Chapter 6 – Contrastive Graph Similarity Networks
Chapter 6 introduces a novel contrastive graph similarity network to carry out imputation and prediction in EHR data. It incorporates graph contrastive learning in representation learning for EHR data. The results of this chapter have been submitted for publication.

Chapter 7 – Multi-Graph Neural Networks
Chapter 7 introduces a novel multi-graph neural network to carry out imputation and prediction in EHR data. It incorporates multi-graph learning in representation learning for EHR data. The results of this chapter have been submitted for publication.

Chapter 8 – Multi-Task Deep Neural Networks

Chapter 8 introduces a novel deep imputation-prediction network based on multi-task learning that performs risk prediction tasks while incorporating the imputation task as an auxiliary task. The results of this chapter have been submitted for publication.

Chapter 9 – Discussion

Chapter 9 includes a response to the research aims in relation to the research undertaken by restating the research aims and discussing the results achieved and a detailed discussion of the proposed approaches from different perspectives, such as network architecture and performance comparison.

Chapter 10 – Conclusion and Future Work

Chapter 10 summarizes the main contributions of this study. Further, the limitations of this study are discussed, and suggestions for future research are made.

Chapter 2

Literature Review

In recent years, various machine learning and statistical approaches have been developed and introduced to missing data imputation. In addition to machine learning and statistical approaches, deep neural networks such as recurrent neural networks and generative adversarial networks are the widely used architecture for missing data imputation.

In this thesis, we review previous studies from three perspectives: traditional statistical, machine learning, and deep learning techniques for missing data imputation.

This chapter presents relevant work identified through a literature review of relevant topics. The topics are the application of traditional statistical, machine learning, and deep learning missing data imputation techniques.

2.1 Traditional statistical techniques for missing data imputation

2.1.1 Simple imputation

Simple imputation is replacing missing values with the mean, median, random, or mode values of the dataset at large or some similar summary statistic (e.g., last observation carried forward and next observation carried backward) [75]. These methods are also known as single imputation. A major advantage of the simple imputation is that it requires less computational cost. When used on smaller datasets, the simple imputation is easy to ma-

nipulate and thus provide results for reference. Despite its efficacy, the simple imputation largely ignores variable correlations. Because of this, simple imputation is challenging to apply to large datasets, especially high-dimensional sparse data.

2.1.2 Hot-deck imputation

Hot-deck imputation is a simple method for dealing with missing values in a data matrix using observed values from the same matrix [76–78]. More specifically, for an object with a missing value, Hot-deck imputation aims to first find the most similar object in the data matrix and then replace the missing value with the value of the similar object. Multiple similar objects are generally found, and one is randomly selected for missing value imputation. Although the concept of this method is simple, different problems need to define different similarity criteria, which are greatly affected by subjective and random factors. The lack of a standardised outcome measure makes it difficult to interpret these imputation results with confidence.

Examples of research into the Hot-deck imputation include [76, 79–82]. The study by [79] proposes Weighted Hot-deck imputation by extending the original Hot-deck imputation. The core idea of Weighted Hot-deck imputation is to use weights in the imputation by incorporating them into the probabilities of selection for each similar object. Similar to [79], the study by [76] further examines the effects of Weighted Hot-deck imputation on bias and reports that Weighted Hot-deck imputation does not correct for bias. In another major study, [80] proposes Fractional Hot-deck imputation by replacing the missing values with a set of observed values and assigning corresponding weights to those values. Further studies by [81] and [82] extend the Fractional Hot-deck imputation [80] to multivariate missing data. Compared with [81], [82] includes making two-phase systematic sampling to improve the performance of Fractional Hot-deck imputation.

2.1.3 Expectation–maximization

The expectation-maximization imputation is built with multiple iterations, each based on two stages: expectation and maximization. The expectation stage includes estimating missing values based on the observed values, while the maximization stage includes checking whether the estimated values reach the most likely value [83].

In a study which set out to determine the efficacy of expectation-maximization, [84] compares the imputation performance of list-wise or case deletion (e.g., every case with one or more missing values is removed), mean imputation, and expectation-maximization imputation and demonstrate that expectation-maximization imputation performs better than other methods when the data missing exceeds 5%.

A key study extending expectation-maximization to non-parametric bootstrap-based expectation-maximization is that of [85], in which the complete data are hypothesised multivariate normal. However, caution must be applied with a small sample size, as the complete data might not be hypothesised multivariate normal.

In another major study, [86] employs expectation-maximization to address the problem of training Gaussian mixtures in large high-dimensional datasets with many missing values. After obtaining the complete data matrices from the imputation task, the complete data matrices are used as input for downstream classification tasks. Experimental results demonstrate that the complete data obtained by expectation-maximization imputation has a significant improvement in classification performance compared to those using other imputation methods.

2.2 Machine learning techniques for missing data imputation

Before introducing machine learning techniques for missing data imputation, we first introduce regression imputation because current machine learning techniques are inspired by its ideas. Regression imputation is also known as conditional mean imputation, where missing values are replaced with predicted values created on a regression model (e.g., a logistic regression model) if data are randomly missing. Accordingly, the core idea of regression imputation is to use all observed values to create a regression model and then predict missing values with the created regression model.

2.2.1 K-nearest neighbors

The k-nearest neighbors algorithm (KNN) has been recognised as an important supervised machine learning algorithm in the early academic community [87]. The core idea of KNN is to match a point with its closest k neighbors in a multi-dimensional space. Since KNN can be applied to continuous, discrete, ordinal, and categorical data, it is suitable for dealing with all types of missing data.

Traditionally, the KNN has been widely used for missing data imputation [88–94]. These studies focus on the provision of a standard imputation implementation of KNN and the development of KNN variants for enhanced imputation performance.

When using KNN for missing data imputation, it mainly classifies the nearest neighbors of missing values and uses these neighbors to perform imputation based on a distance metric between instances. Two of the most common methods for estimating the distance

between instances are the use of Euclidean and Hamming distances [95–99].

There are certain drawbacks associated with the use of KNN. Since the imputation process of KNN involves distance measurement between instances, a number of outliers cannot be avoided. There are a number of distances available for measuring the distance metric between instances, such as Euclidean, Manhattan, Hamming, and Weighted Hamming distances [100]. In particular, Euclidean and Manhattan distances are used for numeric attributes. Hamming and Weighted Hamming distances are used for categorical attributes. Accordingly, the imputation performance of KNN may vary depending on the distance metric used. Besides, it is challenging to directly apply KNN to categorical data without data transformation and scaling. Due to practical constraints, researchers have shown a decreased interest in the use of KNN as an imputation method.

2.2.2 Tree-based algorithms

The tree-based algorithms offer an effective way of dealing with many missing values. For example, random forest is one of the best-known tree-based models [101]. A random forest consists of many individual decision trees that operate as an ensemble. The random forest has a number of attractive features, such as capturing the non-linearity of data and handling outliers and mixed-type attributes (e.g., numerical attributes and categorical attributes).

A large and growing body of literature has demonstrated the effectiveness of tree-based algorithms on the missing data imputation [102–109].

Examples of representative tree-based imputation methods include DMI [103], SiMI [105], MissForest [104], and MD-MTS [108].

The DMI consists of an existing decision tree algorithm (e.g., C4.5 [110]) and an expectation-maximization algorithm [111]. Specifically, DMI uses the decision tree algorithm to impute categorical missing values. While for the imputation of numerical missing values, DMI uses the decision tree algorithm to identify horizontal segments of records with high correlations among the attributes first and then applies the expectation-maximization algorithm to these identified horizontal segments. Concisely, the expectation-maximization algorithm is built with multiple iterations, each based on two stages: expectation and maximization. The expectation stage estimates missing values based on the observed values, while the maximization stage checks whether the estimated values reach the most likely value.

Similar to DMI, SiMI comprises an existing decision forest algorithm (i.e., SysFor [112]) and an expectation-maximization algorithm [111]. It is also worth noting that SiMI incorporates a more practical splitting and merging approach into the framework, which is an important advantage of SiMI over DMI.

The MissForest is an iterative imputation method based on a random forest [101]. The random forest usually involves a process of multiple imputations in which unpruned classification or regression trees are carefully averaged. More specifically, the random forest comprises a series of single trees. Every single tree is built with a random sample of the training data. When using a random forest and a single decision tree for missing data imputation, the former reports significantly more imputation accuracy than the latter [102, 113–116].

The MD-MTS is particularly useful for handling missing values in multivariate clinical time series data. The MD-MTS is built with an efficient gradient-boosting decision tree (i.e., LightGBM [117]). The LightGBM is intrinsically a Gradient Boosting Machine (to be detailed later). Experimental results on the ICHI challenge 2019 dataset demonstrate the effectiveness and superiority of MD-MTS in multivariate clinical time series data imputation compared to state-of-the-art imputation methods (such as 3D-MICE [118] and BRITS [40]).

The gradient boosting machine (i.e., particularly relevant for tree-based gradient boosting machine) and random forest are ensemble learning methods that combine the outputs of single trees to perform both regression and classification tasks. The differences between the tree-based gradient boosting machine and random forest lie in how the tree is created. The former creates one tree at a time, and each new tree has more robust than the previously trained tree, while the latter uses a random sample of the data to create each tree independently.

2.2.3 Multivariate imputation by chained equations

Traditionally, Multivariate imputation by chained equations (MICE) [119] is one of the most well-known imputation methods for handling incomplete medical/clinical data. The mode of imputation processing used by MICE is to learn the distribution of observed values in order to impute missing values. Specifically, MICE imputes missing values of continuous attributes by fitting a linear regression model for the observed values. More specifically, MICE predicts the conditional mean for each missing value and randomly imputes a value from a normal distribution centered on the conditional mean.

Examples of research into the MICE architecture include [118, 120–129].

Detailed examination of the use of MICE by [126] showed that MICE reduces bias in the feature selection process compared to the basic technique that replaces the missing value with a mean, mode, median, or constant value, in addition to achieving the best imputation accuracy.

A key study comparing random forest (i.e., MissForest) and multivariate imputation

by chained equations (MICE) is that of [121]. Specifically, [121] compares the imputation performance between the use of a random forest and standard implementation of MICE and then presents a new version of MICE (also known as random forest MICE) by combining both MICE and random forest. The benefit of incorporating random forest into the MICE architecture is capturing nonlinear relations and interactions from input variables. Experimental results on the cardiovascular disease research dataset demonstrate the effectiveness and superiority of random forest MICE in multiple imputation tasks compared to using a standard implementation of MICE alone.

The study by [118] proposes 3-dimensional multiple imputations with chained equations (shorten for 3D-MICE) to impute missing values in clinical time series data. The core idea of 3D-MICE is to combine the MICE architecture with the Gaussian process to capture cross-sectional and longitudinal information from incomplete clinical time series data. Experimental results on clinical laboratory time series data demonstrate the effectiveness and superiority of 3D-MICE in the imputation task compared to using MICE and Gaussian process alone.

The study by [125] integrates single imputation and multiple imputation techniques into a hybrid approach (also known as SICE) for missing data imputation. The SICE is an extension of MICE that includes two MICE variants applied to categorical attributes and numeric attributes. Experimental results on three public medical datasets (e.g., the well-studied datasets from the UCI Machine Learning Repository) demonstrate the effectiveness and superiority of SICE in the imputation tasks compared to existing imputation methods using these datasets.

In a recent study, [128] proposes a new version of MICE (shorten for SuperMICE) for missing data imputation. The SuperMICE is built with the MICE architecture. Unlike a standard imputation implementation of MICE, the proposed SuperMICE combines an ensemble algorithm (also known as Super Learner) with the MICE architecture to estimate each missing value by predicting the conditional mean value. The Super Learner consists of a series of machine learning models, such as generalised additive models and random forests (i.e., Tree-based models). With the construction of MICE and Super Learner, SuperMICE achieves state-of-the-art imputation performance on the National Crime Victimization Survey dataset.

A more recent example of MICE-based imputation methods can be found in the work of [130]. In particular, [130] conducts a series of experiments on mortality risk prediction in emergency laparotomy in which incomplete patient health data are dealt with by combining generalised additive models and MICE.

The evidence presented in this subsection suggests a growing trend toward combining MICE with existing algorithms such as random forest as an ensemble imputation method.

2.2.4 Support vector machine

Support vector machine (SVM) is one of the popular methods for dealing with missing values [131–135]. The main goal of SVM is to determine an optimal separating hyper-plane that classifies the data points. Examples of representative SVM-based imputation methods include [132,133,135]. The study by [132] proposes an SVM regression-based method for filling in missing data. The core idea of the study [132] is to set the decision attribute as the condition attribute and the condition attribute as the decision attribute first and then predict the condition attribute values. Experimental results on the SARS dataset demonstrate the effectiveness of the proposed SVM regression-based method on the imputation task. The study by [133] utilizes least squares SVM (shorten for LS-SVM) to perform spatiotemporal traffic missing data imputation and traffic flow prediction. Experimental results show that LS-SVM significantly outperforms the traditional statistical missing data imputation techniques, such as expectation maximization imputation. The study by [135] proposes a new SVM-based method that includes making a new kernel function for addressing the problem with a large amount of missing data on the classification task. Experimental results on the four datasets from the UCI machine learning repository demonstrate the effectiveness and superiority of the proposed SVM-based method on the classification task.

2.3 Comparison of traditional statistical and machine learning missing data imputation techniques

A considerable amount of literature has been published on the comparison of traditional statistical and machine learning missing data imputation techniques.

In one well-known early study, [136] compares the imputation performance of KNN, MICE, and MissForest on the complete mammalian order Carnivora dataset. Four features are used, and their values are randomly removed as inputs for KNN, MICE, and MissForest. Extensive experimental results demonstrate that the imputation accuracy of KNN is much lower than that of MICE and MissForest.

A key study comparing the imputation performance of MissForest, mean imputation, KNN, and MICE is that of [137]. Experiments on two incomplete large medical datasets (i.e., the University of Michigan Cirrhosis Cohort and Bowel Disease Cohort) show that MissForest achieves the best imputation accuracy.

In another major study, [138] conducts a series of comparative experiments on missing data imputation. In particular, [138] validates several imputation methods, such as

mean imputation, median imputation, linear regression, KNN, and MICE, on five numeric datasets from the UCI machine learning repository. The method analysis results show that KNN achieves the best imputation accuracy.

A recent study by [139] applies six conceptually different multiple imputation methods to deal with missing values in categorical questionnaire medical data. Multiple imputation aims to deal with missing data by estimating and replacing missing values many times. The methods used in the study [139] include multiple imputation using expectation–maximization with bootstrapping, multiple imputation using multiple correspondence analysis, multiple imputation using latent class analysis, multiple hot-deck imputation, and multivariate imputation by chained equations with two different model specifications (i.e., logistic regression and random forests). Experimental results show that all the methods achieve promising imputation accuracy where the dataset contains a small missing sample. When using the dataset with 20% or more missing samples, multiple imputation using multiple correspondence analysis outperforms other imputation methods.

A more recent comparison of imputation methods can be found in the work of [140]. The study [140] focuses on the provision of time series missing data imputation. Extensive experimental results show that KNN significantly outperforms the other methods, such as mean imputation, MICE, and expectation maximization.

According to these previous studies, we can infer that the factors associated with the imputation performance of models are as follows:

- The degree of missing data could be a major factor causing the model performance differences.

- Machine learning model sensitivity could be another major factor [141, 142]. For example, MissForest is a more practical method for dealing with data with a large number of missing values [137].

- The parameterization of machine learning models could be a third factor, as the model performance highly depends on parameter settings.

The major limitation of traditional statistical and machine learning missing data imputation techniques lies in the fact that they are not suitable for the imputation of missing data in large data sets. Deep learning techniques have been proposed and proven useful in imputing missing data in large data sets. As a special network architecture, deep neural network architectures consist of layers and have been mined for many applications, such as health risk prediction [1]. We will introduce deep learning missing data imputation techniques in detail in the next section.

2.4 Deep learning techniques for missing data imputation

2.4.1 Recurrent Neural Networks

A recurrent neural network (RNN) is a type of neural network commonly used in speech recognition. The RNN has a number of attractive features, for example, the ability to capture long-term temporal dependencies and variable-length observations.

A considerable literature has grown up around the development of RNN-based imputation methods. When used for missing data imputation, RNN mainly imputes missing values by capturing long-term temporal dependencies of observed values.

RNNs addressing temporality

Preliminary work on the development of RNN-based imputation methods was undertaken by [143]. The study mainly compares the classification performance of recurrent neural networks and hidden Markov models[1] on incomplete speech datasets. Experimental results show that RNNs achieve superior both predictive and imputation accuracy.

The study by [144] utilizes a simple recurrent network (SRN) and a long short-term memory (LSTM, a variant of RNN) [145] to impute missing values in the medical examination data. The incorporated RNNs impute missing values by capturing the temporal dependencies of medical examination measurements. After obtaining the complete data matrices from the imputation task, the complete data matrices are used for early disease diagnosis. In the same vein, [146] applies LSTM to impute missing values in the time series of air pollutants and uses the outputs of LSTM, which are the complete time series of air pollutants, to make PM2.5 concentration prediction.

The study by [63] proposes an end-to-end imputation network, Residual IMPutation LSTM (shorten for RIMP-LSTM), for missing data imputation. The RIMP-LSTM includes residual units that are used to build deep neural networks. With the use of residual units, RIMP-LSTM imputes missing values by comprehensively examining the association between the previous observed values, which is an advantage over the standard LSTM. It is also worth noting that RIMP-LSTM allows the imputation and prediction to be trained together by modifying the loss function in the network architecture.

[1] An hidden Markov model is a probabilistic model that consists of a sequence of hidden states, each corresponding to an observation. The hidden states are usually not directly observable, and the purpose of HMM is to compute the sequence of hidden states given a sequence of observations.

RNNs addressing irregularity

The study by [39] proposes a deep prediction model named GRU-D to carry out a series of experiments using multivariate time series with missing values. The overall structure of GRU-D is built upon Gated recurrent units (GRU) [147]. The GRU is a variant of RNNs featured with a reset gate and an update gate, which control the flow of information between the hidden state and the current input. The GRU-D mainly incorporates the empirical mean value and the previous observation to impute missing values. Experimental results on three public datasets demonstrate the effectiveness and superiority of GRU-D in the prediction tasks compared to existing deep prediction methods.

The study by [42] proposes the use of an interpolation network and a prediction network as a deep imputation-prediction network (shorten for InterpNet). The InterpNet mainly focuses on the provision of multivariate time series data imputation. The interpolation network is an unsupervised learning network that imputes missing values in multivariate time series data. The prediction network includes Gated recurrent units that generate prediction results. Similar to the above RIMP-LSTM, InterpNet allows the imputation and prediction network to be trained together by adjusting the loss function in the network architecture. Experimental results on two public datasets demonstrate the effectiveness and superiority of InterpNet in the prediction tasks compared to existing deep prediction methods.

Although GRU-D and InterpNet have achieved promising performance in many prediction tasks, such as in-hospital mortality prediction and hospital length of stay prediction, the multivariate time series data imputation accuracy has not been reported. It is also worth noting that InterpNet and GRU-D take the irregular interval of multivariate time series data into consideration when imputing missing values. The InterpNet mainly converts observations into equally spaced ones. Despite its efficacy, the conversion process inevitably leads to information loss due to variable-length observations. The GRU-D introduces observations and corresponding timestamps into GRU to impute missing values as the decay of previous input values toward the overall mean/sampling over time (also known as the time decay mechanism). One well-known early study often cited in research on the time decay mechanism is that of T-LSTM [148]. The time decay mechanism used in the study [148] builds upon an implicit assumption that the more recent observed values are more important than previously observed values on the risk prediction tasks such as septic shock prediction [149] and in-hospital mortality prediction [150], hence, taking a monotonically way to decay the information from previous time steps. However, there are certain drawbacks when using the T-LSTM time-decay mechanism. The T-LSTM includes a time-decay mechanism without trainable parameters, which results in a fixed decay mode. This is not suitable for capturing the long-term temporal dependencies of observed values. In

contrast to T-LSTM, GRU-D adopts the T-LSTM assumption but establishes a time-decay mechanism with trainable parameters that can effectively capture the long-term temporal dependencies of observed values. The time-decay mechanism used in GRU-D continues to be used by a considerable literature on missing data imputation [40,43,44,46,47,60,65].

2.4.2 Bidirectional Recurrent Neural Networks

The bidirectional recurrent neural networks (shorten for bidirectional RNN) include two independent RNNs trained together within a single framework. The input data (e.g., sequences) are fed in normal/regular time order for one RNN and reverse time order for another RNN. The outputs of the two RNNs are combined/merged in several ways, such as the use of average, sum, multiplication, or concatenation. Therefore, bidirectional RNNs can make predictions based on the information from past and future time steps.

Examples of research into bidirectional RNN-based imputation method include [40, 52, 151–154].

The study by [40] employs a bidirectional RNN (shorten for BRITS) to impute missing values in multivariate time series data and then exploits these imputed values to predict the final imputed values. The two prediction losses are tuned together in BRITS.

The study by [151] proposes a deep imputation method (shorten for BRNN) by modeling incomplete multivariate time series data with the utilization of Bidirectional RNNs. The BRNN generates the imputed values for each variable with the last observed value or the mean values of the same variable. These imputed values are used as initial imputed values for the complete data matrix, fed into a bidirectional RNN to predict the final values (i.e., imputed values).

The study by [152] proposes the use of bidirectional RNNs to model incomplete multimodal wearable recording datasets of bio-behavioral signals. With the construction of bidirectional RNNs, the long-term temporal dependencies of observed values are captured from the forward and backward in multimodal wearable recording datasets of bio-behavioral signals. Compared with machine learning missing data imputation techniques such as KNN and MICE, the bidirectional RNNs achieves the best imputation accuracy.

The study by [52] proposes a context-aware time series imputation framework (shorten for CATSI) for handling the missing values in multivariate time series data. The CATSI framework comprises a context-aware recurrent imputation module and a cross-variable imputation module. The context-aware recurrent imputation module mainly learns from the forward and backward in multivariate time series data. The two modules are used to capture temporal information and cross-variable relations from multivariate time series data. A fusion layer in CATSI is used to integrate these two imputation outputs into the

final imputation outputs/results.

The study by [153] proposes an LSTM-based imputation-prediction network architecture (shorten for SBU-LSTM) for traffic state forecasting. The proposed network architecture comprises two key components: a bidirectional LSTM and a modified LSTM. The incorporated bidirectional LSTM captures long-term temporal dependencies of observed values from the forward and backward in spatiotemporal traffic data. The outputs of the two LSTMs are concatenated at each time step. The modified LSTM includes an imputation unit that imputes missing values in spatiotemporal traffic data. Experimental results on two real-world traffic state datasets demonstrate the effectiveness and superiority of SBU-LSTM for both missing data imputation task and traffic state prediction task.

The study by [154] proposes a new deep imputation method by modeling incomplete genotype data with the utilization of bidirectional RNNs. Experiments on two haplotype datasets show that the proposed method outperforms the existing state-of-the-art imputation approaches in genotype data imputation tasks.

2.4.3 Autoencoders and Variational Autoencoders

An autoencoder is a type of deep neural network that can be used to learn the encoding of input data in an unsupervised manner (there is no requirement for prelabeled data). The core idea of an autoencoder is to learn a low-dimensional representation of the input data by capturing the most important parts of the input data, which in turn reduces the dimensionality of the input data. An autoencoder can technically be trained with supervised learning methods (i.e., a machine learning method that learns from labeled data).

The Variational autoencoder is one of the most common methods for generating the imputed values for missing values [155]. There are two likely causes for the differences between an autoencoder and a Variational autoencoder. For the use of an autoencoder, the encoder network in the architecture maps the input data to a fixed point. While for the use of a Variational autoencoder, the encoder network in the architecture maps the input data to a normal distribution (e.g., univariate or multivariate Gaussian distribution). In addition, the Variational autoencoder includes the reconstruction loss with an additional KL divergence term.

Taken together, the Variational autoencoder has been able to generate new data points by sampling from the learned latent space, while the use of KL divergence term makes the learned distribution as close as possible to the prior distribution, which allows generating of accurate, meaningful, and reliable data samples.

In recent years, Autoencoders and Variational autoencoders, as well as their variants for the imputation of missing data, have been widely investigated [56, 65, 156–180].

In a study conducted by [168], a new deep imputation method (shorten for LSTM-AEs) is proposed for spatiotemporal time series missing data imputation. The LSTM-AEs consist of an autoencoder and an LSTM. The core idea of LSTM-AEs is to capture the diversity of missing patterns from incomplete spatiotemporal time series data and then utilize the captured spatiotemporal features to impute missing values. The captured spatiotemporal features take a lower-dimensional feature representation that retains the semantics/meanings of each feature. With the construction of an autoencoder and LSTM combination, LSTM-AEs achieves state-of-the-art imputation performance on three sensor datasets (e.g., the gas turbine data from the offshore oil Corporation).

Similar to LSTM-AEs [168], V-RIN [65] integrates a Variational autoencoder and a GRU into a single deep imputation network. The incorporated Variational autoencoder uses an encoder network to learn the distribution of patient health data and a decoder network to generate the reconstructed data distribution where the reconstructed values as the imputed values. The GRU used in V-RIN continues to be a recurrent imputation network, which aims at capturing the variation pattern of input variables at the time dimension (i.e., the vertical dimension of the input data).

A key study combining a Variational autoencoder with a Gaussian process is that of [167], in which the incomplete time series are mapped by a Variational autoencoder into a latent feature space, followed by the use of a Gaussian process to capture the temporal nature of sequential dynamic features. With the construction of the Variational autoencoder and Gaussian process, the proposed imputation method GP-VAE achieves the best imputation accuracy on three public datasets (i.e., the Healing MNIST dataset [181], the SPRITES dataset [182], the 2012 Physionet Challenge dataset [183]) compared to a standard imputation implementation of Variational autoencoder.

In another major study, [56] proposes the use of an autoencoder architecture as a deep imputation-prediction method to impute missing values in patient health data and perform patient outcome prediction (e.g., short-term and long-term mortality risk predictions). Experimental results on a real-world chronic cardiovascular disease dataset demonstrate the effectiveness and superiority of the proposed autoencoder-based imputation method for both imputation and prediction tasks.

Recent work by [177] proposes a new deep imputation method (shorten for PMIVAE) by modeling incomplete healthcare data with the utilization of the standard Variational Autoencoder and a multiple imputation procedure. The standard Variational Autoencoder only generates a single imputation result for missing values. The proposed PMIVAE generates multiple imputation results for missing values and analyzes the generated imputation results to obtain the best imputation result, thanks to the well-designed multiple imputation procedure. Experiments on multiple public medical datasets show that PMIVAE achieves

state-of-the-art imputation performance.

2.4.4 Generative Adversarial Networks

In recent years, generative adversarial network-based models have made great progress in real-world image applications such as image-to-image translation [184, 185], image generation [186, 187], image compression [188, 189], text generation [190, 191], and text-to-image generation [192, 193].

Generative adversarial networks (GANs) are a type of deep neural network that includes two competing neural networks (i.e., a generator and a discriminator) in a single network architecture [194]. The intuitions behind GANs can be seen as making a generator and a discriminator against each other. The generator generates fake samples from random 'noise' vectors, and the discriminator distinguishes the generator's fake samples from actual samples.

A large and growing body of literature has focused on the provision of GAN-based imputation methods. Examples of representative GAN-based imputation methods include GRUI-GAN [43], E^2GAN [44], Bi-GAN [45], conditional GAN [57], STING [46], and MBGAN [47]. These studies take the vector of actual samples, which has many missing values, use a generator to generate the corresponding imputed values and distinguish the generated imputed values from real values using a discriminator. It is worth bearing in mind that these six GAN-based imputation methods are also RNN-based. They are categorised as GAN-based imputation methods because the GAN structure is adopted.

The GRUI-GAN develops GRUI (also follows the GRU-D time-decay mechanism) to learn temporal relationships between observed values of multivariate time series data and incorporates the GAN architecture to generate complete data matrices. Despite its efficacy, GRUI-GAN is not practical since the accuracy of the generator is unstable with a random noise input, making it challenging to train the GRUI-GAN.

In order to address the difficulty of training the GRUI-GAN, E^2GAN is proposed, which uses an encoder-decoder RNN-based structure as the generator.

The Bi-GAN incorporates a bidirectional RNN into the GAN architecture. The incorporated bidirectional RNN learns from multivariate time series data in both forward and backward directions and generates the imputed values for missing values.

The overall structures of conditional GAN, STING, and MBGAN are similar to Bi-GAN. The core idea of conditional GAN is to use the observed values as "additional input" when imputing missing values. Unlike a standard implementation GAN, the conditional GAN generates the imputed values for missing values by replacing a random noise with the captured dependencies and correlations of observed values (i.e., "additional input"),

which allows the generating of accurate, meaningful, and reliable imputed values.

In addition to a bidirectional RNN, STING includes two attention-based modules, including a self-attention mechanism module and a temporal attention mechanism module, which are used to improve the quality of the generator's output (i.e., obtained from GRUs). More specifically, the self-attention mechanism module is mainly used to learn the dependencies between observed values of multivariate time series data. The intuitions behind the self-attention mechanism can be seen as allowing inputs to interact with each other and determining which ones should receive more attention [195]. The outputs are the combination of these interactions and attention scores. The temporal attention mechanism module used is similar to the GRU-D time-decay mechanism that takes into account the irregular interval of multivariate time series data when imputing missing values.

The MBGAN also includes two attention-based modules, including a multi-head self-attention module and a temporal attention mechanism module, which are used to capture the associations between observed values of multivariate time series data and take into account the adjacent timestamps of multivariate time series data when imputing missing values. Overall, the two attention-based modules used in MBGAN are essentially not different from the two attention-based modules by STING.

2.4.5 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) have revolutionised image recognition and related fields. Beginning with small networks such as AlexNet [196], and progressing towards much larger networks like GoogleNet [197], and ResNet [198], CNNs have been able to achieve performance in image recognition tasks on par with human performance.

As part of these advances, many CNN architectures have been proposed [199, 200]. Typically, these architectures consist of a combination of convolution layers of wide varieties, combined with pooling layers, and usually terminating in dense layers.

The CNN-based model is particularly well suited for extracting and learning useful feature data independently, especially in high-dimensional sparse data. When using CNNs for missing data imputation, it estimates the missing values with the extracted important information affecting correlations among features, including temporal dependence.

There is a large volume of published studies describing the role of CNN-based imputation methods. Examples of representative CNN-based imputation methods include [201–209].

In a study conducted by [203], a CNN-based network architecture is proposed to deal with incomplete health signal data. The proposed CNN-based network architecture mainly imputes missing values by capturing the nonlinear relationships from the observed values.

The study by [202] proposes a deep imputation method by modeling incomplete traffic data with the utilization of CNNs. The proposed method transforms incomplete traffic data into spatial-temporal images, then fed into the designed CNN-based context encoder to generate complete spatial-temporal images.

Similar to [202], the study [205] proposes the use of a convolutional encoder-decoder architecture to deal with incomplete traffic data. The convolutional encoder-decoder architecture comprises an encoder neural network and a decoder neural network. Both encoder and decoder neural networks are based on CNNs.

A key study of genotype imputation by [206] proposes the use of CNNs as the imputation method. In particular, the CNN-based genotype imputation method (shorten for RefRGim) is pre-trained with single nucleotide polymorphism data from the 1000 Genomes Project. The pre-trained strategy is often used as a practical way of extracting and learning rich feature representations from complex high-dimensional data such as video and 3D image [210–214].

In another major study, [204] proposes a novel denoising convolutional autoencoder to impute missing values in the accelerometer actigraphy data. Experimental results on the National Health and Nutrition Examination Survey dataset demonstrate that the proposed imputation method outperforms other imputation methods, such as mean imputation, zero-inflated Poisson regression, and Bayesian regression.

Recent work by [208] proposes a novel neural tangent kernel (shorten for NTK) based on fully connected neural networks and CNNs for incomplete data reconstruction. In a fully connected neural network, each input node is connected to each output node. A major advantage of fully connected neural networks is that no special assumptions are made about the type of input data. While for the use of a CNN, not all nodes are connected, and modules can be made based on the input, specifically for image data.

More recent applications of the CNN-based imputation method can be found in the work of [209]. Experimental results on the datasets from the UCI machine learning repository demonstrate the effectiveness and superiority of the CNN-based imputation method compared to other imputation methods such as MissForest, MICE, and GAN-based imputation method [215].

2.4.6 Attention-based Neural Networks

Attention mechanisms have revolutionised machine translation and related fields. Beginning with small network architecture [216], and progressing towards much larger network architecture (e.g., Transformer) [195], attention-based neural networks have been able to achieve promising performance in many machine translation tasks. Increased application

of Transformer-based methods has been observed across a number of research fields in recent years, such as bioinformatics [217, 218], text classification [219, 220], and medical analysis [221, 222].

Before presenting attention-based Neural Networks for missing data imputation, we detail two pioneering studies on attention mechanism [216] and [195]. The study by [216] incorporates the attention mechanism into the encoder-decoder architecture to improve the performance of the encoder-decoder model for machine translation. The idea of incorporating the attention mechanism is that such a consideration can learn a weighted combination from input sequences. The decoder then uses the most relevant parts, highlighted by corresponding attention weights, to make decisions. The pioneering work of Transformer [195] remains crucial to our wider understanding of attention mechanisms. The Transformer model is built upon the encoder-decoder architecture. The encoder in the Transformer mainly implements a multi-head self-attention mechanism, followed by a fully connected feed-forward network that includes two linear transformations with Rectified Linear Unit activation. The decoder in the Transformer implements a multi-head self-attention mechanism and a fully connected feed-forward network similar to those implemented by the encoder. It is worth noting that the multi-head mechanism implemented by the decoder leverages the queries from the previous decoder sub-layer (i.e., a total of three main sub-layers in the decoder) as well as the keys and values from the encoder, which allows the decoder to pay attention to the multiple relationships and nuances for all the words in the input text (e.g., machine translation).

Attention-based neural networks for the imputation of missing data have been investigated recently [51, 58, 61, 223–227].

The study by [223] proposes the use of the self-attention mechanism [195] to impute missing values in multivariate geo-tagged time series data. Experiments on multiple geo-tagged time series datasets show that the proposed cross-dimensional self-attention (shorten for CDSA) achieves state-of-the-art imputation performance compared to existing imputation approaches.

The study by [224] proposes a deep imputation method (shorten for AimNet) by modeling incomplete mixed data (i.e., discrete and continuous attributes) with the utilization of an attention-based framework. The mode of attention mechanism used by AimNet is comparable in complexity to that used by the multiplicative attention mechanism. More specifically, a simple dot product attention mechanism without a trainable weight matrix (e.g., an identity matrix) is employed in the context of a calculation. Experiments on the real-world datasets show that AimNet achieves the best imputation performance compared to the existing state-of-the-art imputation methods such as MICE [119], MissForest [104], and Gain [215].

The study by [58] proposes a deep imputation method (shorten for MTSIT) based on an autoencoder architecture to perform missing data imputation. The autoencoder architecture used in MTSIT includes the Transformer encoder [195] and a linear decoder, which are implemented with a joint reconstruction and imputation approach.

The study by [61] proposes the use of the multi-head attention mechanism [195] to deal with incomplete datasets from the UCI machine learning repository. The proposed imputation-prediction method (shorten for MAIN) imputes missing values first and then utilizes complete data matrixes to make downstream prediction tasks.

The study by [51] proposes a deep imputation-prediction method (shorten for MIAM) based on the self-attention mechanism [195]. The MIAM mainly focuses on the provision of multivariate clinical time series data imputation. Given multivariate clinical time series data, MIAM imputes the missing values by extracting the relationship among the observed values, missingness indicators (0 for missing and 1 for not missing), and the time interval between consecutive observed values.

Although MAIN and MIAM have achieved promising performance in many downstream tasks, such as in-hospital mortality prediction, length of stay prediction, and phenotype classification, the imputation accuracy has not been reported.

The study by [227] proposes a self-attention-based imputation method (shorten for SAITS) for time series data imputation. The core idea of SAITS is borrowed from the current masked language model [228–232]. The masked language model is trained by randomly masking a part of words in the input sequence and predicting those masked words based on the context of the non-masked words. Despite its efficacy, SAITS is not practical since the real-world datasets have an inherently high degree of missingness, making it challenging to train the SAITS.

2.4.7 Graph Neural Networks

Increased application of Graph Neural Networks (GNNs) has been observed across a number of research fields in recent years, such as social recommender systems [233], bioinformatics [234], knowledge graphs [235], drug response prediction [236], materials science and chemistry [237], and medical diagnosis and analysis [238]. GNN is a type of neural network for dealing with graph-structured data [239, 240]. When used on high-dimensional or complex data, very deep GNNs can be constructed by stacking multiple graph convolutional layers. In GNNs, each graph's convolutional layer aggregates information from neighboring nodes and edges using a message-passing strategy. At each GNN message-passing iteration, each node aggregates information from its neighborhood, and as these iterations progress, each node embedding reaches out further in the graph to ex-

tract global information. By doing so, both local and global information from the graph is taken into consideration for generating useful node and graph-level representations for various downstream predictions.

Previous studies have examined the effectiveness of GNN-based methods on missing data imputation [241–252]. Examples of representative GNN-based imputation methods include [241, 242, 244, 246, 252].

The study by [241] proposes a deep imputation method by modeling incomplete data with the utilization of a GCN autoencoder. The core idea of GCN autoencoder is to apply the idea of VAE to graph-structured data. In other words, graph-structured data is fed into an encoder to generate new graphs or reason about graphs, where more correlations are created and new edges are predicted. It is also worth mentioning that the proposed deep imputation method is trained with an adversarial loss to make the distribution of the reconstructed data as close as possible to that of the real data.

The study by [242] proposes GRAPE, a novel graph-based framework to perform feature imputation and label prediction. The core idea of GRAPE is to treat the missing data imputation as a graph representation learning (i.e., it aims to generate graph representation vectors that capture the structure of graphs effectively.), which includes making two types of nodes (i.e., the observations and features) and edges (i.e., the observed feature values) on a bipartite graph. The benefit of constructing the bipartite graph is that such a consideration can create connections between input features (i.e., based on the observations) and between the observations (i.e., based on the features). With the construction of the GRAPE framework, the feature imputation and label prediction are treated as two prediction tasks, i.e., an edge-level and a node-level.

The study by [244] proposes GRIN, a novel graph neural network architecture to carry out spatiotemporal multivariate time series missing data imputation. The core idea of GRIN is to utilize information from sensors at different locations to impute missing values in spatiotemporal multivariate time series data.

The study by [246] proposes AGRN, an adaptive graph recurrent network, to perform multivariate time series missing data imputation. The proposed AGRN is a combination of a graph convolution network and a recurrent neural network. Specifically, the following steps were taken: (i) AGRN incorporates a graph learning module to generate a graph that represents the relationships between input variables; (ii) AGRN utilizes the graph convolution module to aggregate information from neighboring nodes and edges; (iii) AGRN employs a Gated Recurrent Unit to pass temporal information from the forward and backward directions; (iv) AGRN fuses the outputs of Gated Recurrent Unit to generate complete data.

The study by [252] integrates Generative Adversarial Network (GAN) and Graph Con-

volutional Network (GCN) into an overall network architecture to carry out time series missing data imputation. Specifically, the incorporated GAN inversion is utilised to translate input (i.e., incomplete) time series data into a low-dimensional latent space. The purpose of incorporating GAN inversion [253, 254] is to generate the optimal latent variable values in the latent space of the pre-trained GAN.

Moreover, the incorporated GAN inversion is combined with a decay connection in the GCN to take the temporal irregularity of input time series data into consideration on the missing data imputation. The core idea of decay connection in the GCN is to decay the dependences between adjacent observations as the time between them increases.

It is also worth noting that the idea of pre-trained GAN is to randomly mask a part of words in the input sequence and predict those masked words based on the context of the non-masked words. Therefore, the pre-trained GAN is the same as SAITS mentioned in Section 2.2.6; they are not practical since the real-world datasets have an inherently high degree of missingness, making it challenging to train the model.

2.5 Summary

In this chapter, we comprehensively reviewed the development of missing data imputation techniques and introduced state-of-the-art methods. We discussed these methods from different angles, including traditional statistical, machine learning, and deep learning techniques for missing data imputation. We also introduced several state-of-the-art methods that are trying to perform missing data imputation and downstream prediction tasks. We also pointed out several drawbacks and problems of current techniques and will solve these problems and challenges in the following chapters.

Chapter 3

Compound Density Networks

The following publication has been incorporated into this chapter:
[62] **Yuxi Liu**, Shaowen Qin, Zhenhao Zhang, and Wei Shao. Compound density networks for risk prediction using electronic health records. In *2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 1078–1085. IEEE, 2022

3.1 Introduction

The immense accumulation of Electronic Health Records (EHRs) provides an unprecedented opportunity to develop accurate, meaningful, and reliable outcome prediction models [29, 255, 256].

However, health data in EHRs present a high degree of irregularity, due to variations of patient conditions and treatment needs. One of the notable issues is missing values, which makes accurate and reliable predictions challenging. We present an example of a patient's records from the MIMIC-III database [38] in Figure 3.1. The physician conducts/prescribes the necessary lab tests each time a patient is seen. Different physiological variables (e.g., heart rate, glucose) are examined at different times depending on the patient's symptoms [60]. When certain symptom disappears, corresponding variables are no longer examined. This results in missing values.

Machine learning algorithms have brought revolutionary changes in a wide variety of fields, such as computer vision [257], machine translation [258], and computational biology [259]. Machine learning algorithms build models based on a large amount of sample

Figure 3.1: An example of a patient's clinical records.

data, known as training data. The prediction performance of machine learning models will suffer if the training dataset has a large number of missing values. A simple solution is to remove the observations that have missing data. This, however, is not applicable to EHR data, as it means valuable information is discarded. A better strategy would be to impute missing values. Several methods currently exist for the imputation of missing values, such as Multiple Imputation, Expectation-Maximization, Nearest Neighbor and Hot Deck methods. These methods rely on variable correlations to impute missing values. Previous studies of [260–262] have demonstrated the effectiveness of these methods on EHR data.

Existing work usually separates imputation method and prediction model as two independent parts of an EHR-based machine learning system. After imputing missing values, the complete EHR data matrix is fed into existing machine learning models to make risk predictions. For example, machine learning models can be used to predict in-hospital mortality, decompensation, length-of-stay, and phenotype classification [53, 263].

However, with an EHR dataset, caution must be taken, as the missing values are likely attributed to patient symptoms (as mentioned earlier). For this reason, the imputation method and prediction model should be tuned together within a single framework rather than separated as two parts. By doing so, the prediction model is able to deal with the missing values in EHR data effectively. A recent study by [39] developed a deep prediction model named GRU-D to address this problem. The overall structure of GRU-D is built upon Gated recurrent units (GRU) [147]. The GRU-D mainly incorporates the empirical mean value and the previous observation to impute missing values. Despite its efficacy, the GRU-D suffers from notable methodological weaknesses. The empirical mean value might be biased due to the diversity of patient data, hence lacking reliability. Utilizing less reliable imputed values as part of the input is equivalent to introducing noise/error into the input. This inevitably introduces a high degree of aleatoric uncertainty into the

dataset (e.g., imputation error). Most of the existing machine learning models are sensitive to aleatoric uncertainty, that is, a few small variations in the inputs may lead to significant changes in model outputs. This might lead to biased prediction due to the cumulative effect of imputation errors.

Uncertainty quantification has a pivotal role in model optimization [264]. Uncertainty quantification allows researchers to know how confident they can be with the prediction results, which is essential to building trust in prediction models. In contrast, it is often less trustworthy when prediction results are presented without uncertainty quantification. There are two main types of uncertainties: epistemic and aleatoric [265]. Epistemic uncertainty indicates what the model does not know. It is attributed to inadequate knowledge of the model. This is the uncertainty that can be reduced by having more data. Aleatoric uncertainty is the inherent uncertainty that is part of the data generating process, such as sensor noise, record error, or missing value. This variability cannot be reduced by having more data. Current studies have used these two uncertainties as indicators of the reliability of the method [29, 266].

The study by [267] investigated the differential impact of aleatoric uncertainty and epistemic uncertainty on computer vision tasks. The approach taken in this study is a mixed methodology based on Bayesian theory and deep neural networks, known as Bayesian neural networks (BNNs). The core idea of BNNs is to replace the deterministic network's weight parameters with their probability distributions and, instead of optimizing the network weights directly, use the average of all possible weights. However, the inference of BNNs remains a major challenge and incurs a huge computational cost [268]. To address the issue, many variational inference techniques are proposed, such as stochastic variational inference [269] and sampling-based variational inference [270], which have achieved promising performance in many prediction tasks [271, 272]. Despite their efficacy, these methods still impose a tremendous burden on computational costs.

One well-known study that is often cited in research on Bayesian inference approximation is that of [273], which found that the use of dropout in deep neural networks could be regarded as an approximate Gaussian process. Their theoretical framework employs a dropout layer as a Bayesian inference approximation before every weight layer. The use of dropout as a Bayesian approximation is currently the most popular method for providing epistemic uncertainty estimation due to its low computation cost and high efficiency. Despite this, the use of dropout requires a number of repeated feed-forward calculations of deep neural networks with randomly sampled weight parameters. The resulting outputs are used to quantify the epistemic uncertainty of those deep neural networks.

In this chapter, we propose an end-to-end, novel, and robust prediction model by utilizing a Compound Density Network (*CDNet*) that consists of a Gated recurrent unit (GRU),

a Mixture Density Network (MDN), and a Regularised Attention Network (RAN). The proposed *CDNet* enables GRU and MDN to work together by iteratively leveraging the output of each other to impute missing values. The GRU is used as a latent variable model to model EHR data. The MDN is designed to sample latent variables generated by GRU. The sampling process is equivalent to exploiting the generated latent variables to model the distribution of imputed features. The MDN is built from two components: a deep neural network and a mixture of distributions. We assume the mixture of distributions comprises multiple Gaussian distributions because the imputed features are continuous. Specifically, latent variables are fed into the deep neural network. The deep neural network then provides the parameters for multiple Gaussian distributions, including their means, variances, and weights that can be used to build a Gaussian mixture distribution. The resulting Gaussian mixture distribution is a multimodal distribution that contributes to the modeling of complex patterns found in the input.

To enhance the reliability of imputed values and quantify their uncertainties, the RAN is served as a regulariser for less reliable imputed values, leading to more robust model outputs. The core idea behind RAN design is to model the attention weights as a function of the variance of Gaussian mixture distribution. When used for regularised learning, it assigns smaller weights to imputed values with large variance. The output of RAN is fed into the developed predictor network to make risk predictions. This involves making an MDN for predicting the class probability distribution. The modeling process of the MDN includes learning about the impact of aleatoric uncertainty and epistemic uncertainty. When used for quantifying epistemic uncertainty, MDN can be regarded as a sampling-free method because it does not require repeated feed-forward calculations of deep neural networks. Specifically, the MDN uses a deep neural network to provide the parameters (i.e., mean and variance) for a mixture of distributions. When properly trained, we obtain the mean and the standard deviation, which means we have the entire class probability distribution (e.g., the risk of death and no death) and, by extension, the estimate of the aleatoric and epistemic uncertainty. The resulting predicted class probability distributions are further used to estimate risk probabilities (e.g., the probability of death).

The main contributions of this paper are listed as follows: We validate *CDNet* on the in-hospital mortality prediction task from a publicly available EHR database that has a large number of missing values in the input. Our model outperforms state-of-the-art models by significant margins. We also empirically show that regularizing imputed values is a key step for superior prediction performance. Analysis of prediction uncertainty shows that our model can capture both aleatoric and epistemic uncertainties, which allows model users to gain a better understanding of the model results.

3.2 Method

We describe the proposed *CDNet* in this section. We introduce the basic notations of risk prediction tasks first. We then detail the *CDNet* architecture. Finally, we present how to use *CDNet* for risk prediction tasks.

3.2.1 Basic Notations

The EHR data consists of patients' time-ordered records. Each patient's records ensemble can be further categorised as a patient journey, termed EHR patient journey data, in the following sections. The EHR patient journey data is denoted by $X^p = [X_1^p, \cdots, X_t^p, \cdots, X_{T_p}^p] \in \mathbb{R}^{N \times T_p}$, where p is a patient and N is the number of sequential dynamic features (that occur over time, e.g., vital signs) and T_p is the number of records. For simplicity, we drop the p when it is unambiguous in the following sections.

3.2.2 Network Architecture

Our proposed network architecture comprises three key components: 1) a Gated recurrent unit, 2) a Mixture Density Network, and 3) a Regularised Attention Network, as shown in Figure 3.2. These neural network modules are trained together.

Learning feature embedding

An essential step before implementing the proposed components is to learn the embedding of sequential dynamic features. Learning feature embeddings is able to help us translate feature spaces. Specifically, an embedding layer is applied to sequential dynamic features, generating the corresponding representations. This embedding layer provides a mapping between sequential dynamic features and embedding space, allowing GRU to learn the underlying dynamics of patient journeys via lower-dimensional feature representations.

Let Z denotes learnable feature vectors, which are used as prefilled values of the patient journey X. This Z is initialised from the standard Gaussian distribution. By doing this, the X is now termed X'. Given an input X', the embedding layer can be written as:

$$X^{emb} = W^{emb} \cdot X' + b^{emb} \tag{3.1}$$

where $X^{emb} \in \mathbb{R}^{d_{emb} \times T}$ is the learned sequential dynamic feature embedding. $W^{emb} \in \mathbb{R}^{d_{emb} \times N}$ is a learnable parameter, $b^{emb} \in \mathbb{R}^{d_{emb}}$ is a bias, and d_{emb} is the dimension of the embedding space.

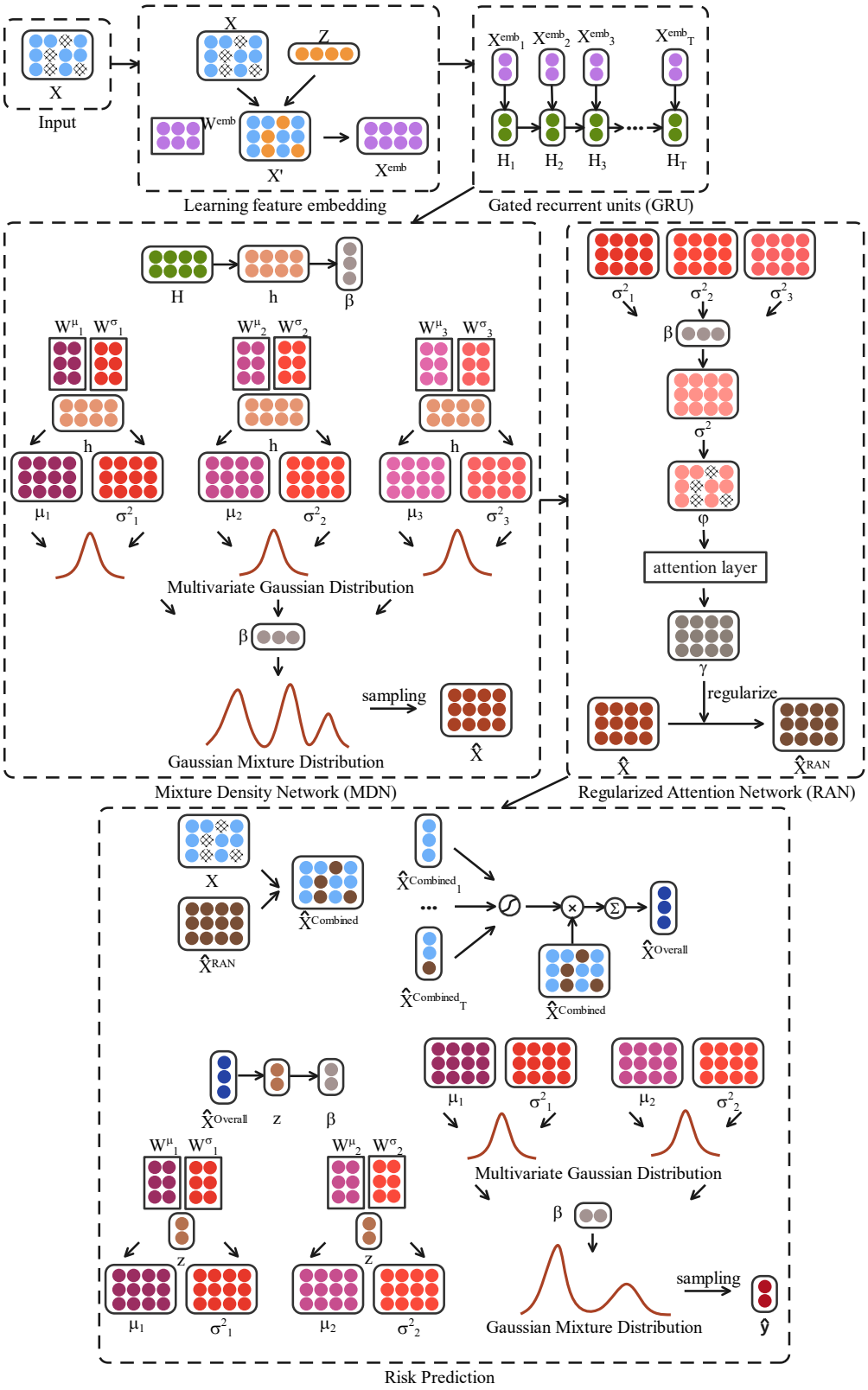

Figure 3.2: Schematic representation of the architecture and workflow of the proposed network.

Model components

Gated recurrent units (GRU)

The core idea of GRU is to exploit the degree of missingness of all EHR patient journeys to impute missing values. Due to patient-specific symptoms, the degree of missingness of sequential dynamic features may vary among patient journeys. Based on this assumption, missing values of one patient journey can be inferred from other EHR patient journeys. The inference process is achieved by employing GRU [147]. The GRU is currently the most popular method for generating latent variables from multivariate time series data. Latent variables are a transformation of the data points into a continuous lower-dimensional space. EHR patient journey data is a type of multivariate time series data with more than one time-dependent feature; each not only depends on its past values but also has some dependency on others. These dependencies must be modeled, which are used for forecasting future values. After training, the employed GRU is able to generate a series of latent variables derived from all EHR patient journeys modeling. These latent variables correspond one-to-one with sequential dynamic features.

GRU is a variant of Recurrent neural networks (RNN) that modifies the basic RNN's hidden layer. One advantage of the GRU is that it avoids the problem of the vanishing gradient suffered by an RNN. The essential nature of GRU is the gating of the hidden state. Given input $X_t^{emb} \in \mathbb{R}^{d_{emb}}$ and previous hidden state $H_{t-1} \in \mathbb{R}^g$, the current hidden state H_t can be obtained through the following steps.

Specifically, X_t^{emb} and H_{t-1} are fed into a gating mechanism. The gating mechanism, including a reset gate R_t and an update gate U_t, is to decide which of the previous information will be retained for H_t. The objective function of the gating mechanism can be written as:

$$\begin{aligned} R_t &= \sigma(W_R \cdot [H_{t-1}, X_t^{emb}] + b_R) \\ U_t &= \sigma(W_U \cdot [H_{t-1}, X_t^{emb}] + b_U) \end{aligned} \tag{3.2}$$

where $W_R \in \mathbb{R}^{g \times (d_{emb}+g)}$ and $W_U \in \mathbb{R}^{g \times (d_{emb}+g)}$ are learnable parameters. $b_R \in \mathbb{R}^g$ and $b_U \in \mathbb{R}^g$ are biases. σ is the sigmoid activation function that is used to normalize the outputs R_t and U_t in $[0, 1]$. The X_t^{emb} and the element-wise multiplication of H_{t-1} with R_t are used to generate an intermediate $\widetilde{H}_t$. H_t is obtained by the element-wise convex combinations between $\widetilde{H}_t$ and U_t. The formula can be written as:

$$\begin{aligned} \widetilde{H}_t &= tanh(W_H \cdot [R_t \odot H_{t-1}, X_t^{emb}] + h_H) \\ H_t &= U_t \odot H_{t-1} + (1 - U_t) \odot \widetilde{H}_t \end{aligned} \tag{3.3}$$

where $W_H \in \mathbb{R}^{g\times(d_{emb}+g)}$ is a learnable parameter and $h_H \in \mathbb{R}^g$ is a bias. $\odot$ denotes the element-wise multiplication.

The generated latent variables $\{H_t\}_{t=1}^T$ are used as the input of MDN.

Mixture Density Network (MDN)

The MDN is designed to sample latent variables generated by GRU. The sampling process is equivalent to exploiting the use of generated latent variables to model the distribution of imputed features. The MDN comprises a deep neural network and a mixture of distributions. Since the imputed features are continuous, we assume the mixture of distributions comprises multiple Gaussian distributions. Specifically, latent variables are fed into the deep neural network. The deep neural network then provides the parameters for multiple Gaussian distributions, including their means and variances, as well as weights that can be used to build a Gaussian mixture distribution. The Gaussian mixture distribution can be written as:

$$\begin{aligned} p(X_t|H_t) &= \sum_{k=1}^{K} \beta_k \cdot D_k(X_t|H_t) \\ D_k(X_t|H_t) &= N(\mu_k, \Sigma_k) \end{aligned} \tag{3.4}$$

where k denotes the index of the corresponding mixture component. There are up to K mixture components (i.e., distributions) per output. β denotes the mixing parameter. D is the corresponding distribution to be mixed. D is a multivariate Gaussian distribution, where μ is the mean vector and Σ is the covariance matrix with σ^2 on the diagonal and 0 otherwise.

We assume mean μ and variance σ^2 of each distribution are derived from a nonlinear combination of the inputs. A deep feed-forward network is modified to output the parameters of the Gaussian mixture distribution. A constraint we must enforce here is $\sigma^2 > 0$, i.e., the variance of Gaussian must be positive. This is implemented by employing the Exponential Linear Unit (ELU) activation with an offset [274]. Since this can technically be zero, we have added an ϵ to the modified ELU to ensure stability.

$$\begin{aligned} h &= ReLU(W^h \cdot H + b^h) \\ \beta &= softmax(W^\beta \cdot h + b^\beta) \\ \mu_k &= W_k^\mu \cdot h + b_k^\mu \\ \sigma_k^2 &= ELU(W_k^\sigma \cdot h + b_k^\sigma) + 1 + \epsilon \end{aligned} \tag{3.5}$$

where all parameters of W are projection matrices and all parameters of b are bias vectors.

ϵ is a constant term (e.g., 1×10^{-15}). β is the mixture weight of each component. We use the softmax function to keep β in the probability space.

(The sampling process of MDN) We apply Gaussian noise ξ and variance Σ_k to μ_k to obtain the predicted EHR patient journey $\hat{X}$.

$$\begin{aligned} \widetilde{X_k} &= \mu_k + \sqrt{\Sigma_k} \cdot \xi, \xi \sim \mathcal{N}(0,1) \\ \hat{X} &= \sum_{k=1}^{K} \beta_k \cdot \widetilde{X_k} \end{aligned} \tag{3.6}$$

The optimization objective of MDN is to make the predicted patient journey $\hat{X}$ as close to the real-valued patient journey X as possible. The optimization function can be written as:

$$\mathcal{L} = MSE(W^{proj} \cdot \hat{X}, X^{emb}) \tag{3.7}$$

where $MSE(\cdot)$ denotes the mean squared error. $W^{proj} \in \mathbb{R}^{d_{emb} \times N}$ is a learnable projection matrix, which translates the predicted patient journey $\hat{X}$ into the embedding space.

Regularised Attention Network (RAN)

The output of MDN is a Gaussian mixture distribution. The predicted patient journey $\hat{X}$ is obtained from the sampling of Gaussian mixture distribution. The $\hat{X}$ includes imputed values, combined with real-valued values as a complete data matrix that can be analysed by our predictor network. However, caution must be taken with imputed values, as they are inferred from the real-valued EHR patient journey data and thus are likely to be less reliable. In response to this issue, the RAN is developed to enhance the reliability of imputed values and quantify their uncertainties. The RAN contains an attention layer; its output is a set of weights. The general idea of the attention layer is to regularize the weights assigned to different patient journeys. For example, it assigns smaller weights to less reliable data.

The unreliability scores of real-valued and imputed values are defined as:

$$\varphi = \begin{cases} 0, & for\ real\ valued\ values \\ \sigma^2, & for\ imputed\ values \end{cases} \tag{3.8}$$

where $\sigma^2 = \sum_{k=1}^{K} \beta_k \cdot \sigma_k^2$ is the mixed variance of Gaussian mixture distribution that can be used to represent aleatoric uncertainty describing the unreliability of imputed values. Since the real-valued values involve no uncertainty, we set their unreliability scores to

zero.

Given the input φ, the attention layer can be written as:

$$\gamma = softmax(W_\gamma \cdot (1 - \varphi) + b_\gamma) \tag{3.9}$$

where $W_\gamma \in \mathbb{R}^{N \times N}$ is a learnable parameter and $b_\gamma \in \mathbb{R}^N$ is a bias.

The weight γ is used to regularize the predicted patient journey $\hat{X}$. The formula can be written as:

$$\hat{X}^{RAN} = ReLU(W_{RAN}(\gamma \odot \hat{X}) + b_{RAN}) \tag{3.10}$$

where $W_{RAN} \in \mathbb{R}^{N \times N}$ is a learnable parameter and $b_{RAN} \in \mathbb{R}^N$ is a bias. $\odot$ denotes the element-wise multiplication. $ReLU(\cdot)$ is an activation function.

Risk Prediction

In order to apply *CDNet* to risk prediction tasks, a predictor network is developed, which consists of an attention layer and an MDN.

The $\hat{X}^{RAN}$ (the output of RAN) includes enhanced imputed values, combined with the real-valued patient journey X as a complete data matrix that can be analysed by the predictor network. The complete data matrix is denoted by $\hat{X}^{Combined}$. Since $\hat{X}^{Combined}$ still takes the form of sequence data, it is difficult to use as the input of an MDN to obtain prediction probability distributions. In response to such an issue, the designed attention layer is used to integrate the $\hat{X}^{Combined}$ into a whole representation. The attention layer can be written as:

$$\begin{aligned} \tau &= softmax(W_\tau \cdot \hat{X}^{Combined} + b_\tau) \\ \hat{X}^{Overall} &= \sum_{t=1}^{T} \tau_t \odot \hat{X}_t^{Combined} \end{aligned} \tag{3.11}$$

where $W_\tau \in \mathbb{R}^{N \times N}$ and $b_\tau \in \mathbb{R}^N$ are learnable parameters. $\hat{X}^{Overall}$ is the weighted average of sampling a record according to its importance.

Instead of predicting a deterministic value for each patient journey, we predict the class probability distribution and moreover include aleatoric and epistemic uncertainty estimations. We model the output of every class as an MDN, generating three groups of parameters for every class: the mean $\mu_{p,k}$, the variance $\Sigma_{p,k}$, and the weights of the mixture

β_p. The process can be formalised as:

$$\begin{aligned} z_p &= ReLU(W_p^z \cdot \hat{X}_p^{Overall} + b_p^z) \\ \beta_p &= softmax(W_p^\beta \cdot z_p + b_p^\beta) \\ \mu_{p,k} &= W_{p,k}^\mu \cdot z_p + b_{p,k}^\mu \\ \sigma_{p,k}^2 &= ELU(W_{p,k}^\sigma \cdot z_p + b_{p,k}^\sigma) + 1 + \epsilon \end{aligned} \tag{3.12}$$

where all parameters of W are projection matrices and all parameters of b are bias vectors. ϵ is a constant term. p denotes the index of the corresponding patient journey. There are up to P patient journeys. k denotes the index of the corresponding mixture component. There are up to K mixture components. β_p is the mixture weight for each component of patient p's journey. $\mu_{p,k}$ is the predicted value of the k-th component of patient p's journey. $\Sigma_{p,k}$ is the variance for each coordinate $\sigma_{p,k}^2$ representing its aleatoric uncertainty. Note that for the binary classification task, both the dimensions of $\mu_{p,k}$ and $\sigma_{p,k}^2$ are set to 2. We use the softmax function to keep β_p in probability space and use the ELU function again to satisfy the positiveness constraint of the variance.

We apply Gaussian noise η and variance $\Sigma_{p,k}$ to $\mu_{p,k}$ to obtain the predicted class probability distribution for patient p's journey.

$$\begin{aligned} \tilde{y}_{p,k} &= \mu_{p,k} + \sqrt{\Sigma_{p,k}} \cdot \eta, \eta \sim \mathcal{N}(0,1) \\ \hat{y}_p &= softmax(\sum_{k=1}^{K} \beta_{p,k} \cdot \tilde{y}_{p,k}) \end{aligned} \tag{3.13}$$

where K is the number of mixture components. $\hat{y}_p$ is the prediction probability. The objective function $\mathcal{L}'$ of the risk prediction task is the average of cross-entropy:

$$\mathcal{L}' = -\frac{1}{P} \sum_{p=1}^{P} (y_p^\top \cdot log(\hat{y}_p) + (1 - y_p)^\top \cdot log(1 - \hat{y}_p)) \tag{3.14}$$

where P is the number of patient journeys. y_p is the ground-truth class/label for patient p's journey.

3.3 Experiments

3.3.1 Experimental Setup

Datasets and Tasks

We conduct the in-hospital mortality prediction experiments on the publicly available MIMIC-III[1] database [38]. MIMIC-III is one of the largest publicly available ICU databases, comprising 38,597 distinct patients and a total of 53,423 ICU stays. A total of 21,139 samples were extracted from the MIMIC-III database. We use clinical times series data (e.g., heart rate, glucose, and respiratory rate) as input [53]. The prediction tasks here are three binary classification tasks: 1) In-hospital mortality (24 hours after ICU admission): to evaluate ICU mortality based on the data from the first 24 hours after ICU admission. 2) In-hospital mortality (36 hours after ICU admission): to evaluate ICU mortality based on the data from the first 36 hours after ICU admission. 3) In-hospital mortality (48 hours after ICU admission): to evaluate ICU mortality based on the data from the first 48 hours after ICU admission. The MIMIC-III database being used has a high degree of missingness in the input. E.g., for the mortality prediction of the first 48 hours after ICU admission, the results of the statistical analysis of the input are shown in Table A.1.

Baselines

- Mean: The mean values of variables are used to impute the missing values.
- K-Nearest Neighbor (KNN): The average values of the top K most similar collections are used to impute the missing values.
- MICE: Multiple Imputation by Chained Equations (MICE) [119] uses chain equations to create multiple imputations for variables of different types.
- Simple: Simple concatenates the measurement with masking and time intervals, which are then fed into a predictor to make risk predictions [39].
- BRNN: Bidirectional-RNN (BRNN) [151] generates the imputed values for each variable with the last observed value or the mean values of the same variable. These generated values are used as initial imputed values for the complete data matrix, fed into a bidirectional RNN to predict missing values.

[1] https://mimic.physionet.org

- CATSI: CATSI [52] comprises a context-aware recurrent imputation and a cross-variable imputation, which are used to capture longitudinal information and cross-variable relations from MTS data. A fusion layer in CATSI is used to integrate the two imputation outputs into the final imputations.

- BRITS: BRITS [40] employs a bidirectional RNN to impute missing values first and then exploits these imputed values to predict the final values.

- GRU-D: GRU-D [39] is described in the introduction section. GRU-D also utilised a time decay mechanism to deal with irregular time intervals of medical events in longitudinal patient records. The time decay mechanism builds upon an implicit assumption that the more recent events are more important than previous events on patient-specific risk prediction tasks, hence, taking a monotonically way to decay the information from previous time steps for all patients (the previous medical events).

- GRU-D_{d-}: GRU-D without time decay mechanism.

- *CDNet*$_+$: *CDNet* with a time decay mechanism [39].

The outputs of Mean, KNN, MICE, Simple, BRNN, and CATSI are fed into standard GRU to make in-hospital mortality predictions.

Implementation Details & Evaluation Metrics

We perform all the baselines and *CDNet* with Python v3.9.7. We employ the following libraries: fancyimpute for KNN and MICE and PyTorch for the rest of the methods. For each task, we randomly split the datasets into training, validation, and testing sets in a 70:15:15 ratio. The validation set is used to select the best values of parameters. Training and evaluations were performed on A40 GPU from NVIDIA with 48GB of memory. We repeat all the methods ten times and report the average performance.

We use class weight in CrossEntropyLoss for a highly imbalanced dataset. This is achieved by placing an argument called 'weight' on the CrossEntropyLoss function (PyTorch).

We assess performance using the area under the receiver operating characteristic curve (AUROC) and the area under the precision-recall curve (AUPRC).

3.3.2 Performance Analysis

Table 3.1 lists the results of in-hospital mortality prediction based on the clinical times series data from the first 24, 36, and 48 hours after ICU admission, respectively. The

larger the scores of AUROC and AUPRC, the better the predictive performance of the method. Values in the parentheses denote standard deviations. These results suggest that *CDNet* significantly and consistently outperforms other baseline methods. Comparing the two results GRU-D_{d-} and GRU-D in Table 3.1 (24 hours after ICU admission), it can be seen that the use of the time decay mechanism achieves a performance improvement in AUROC by 1.86% and in AUPRC by 1.32%. Interestingly, it can be seen from the data in Table 3.1 that GRU-D_{d-} outperforms GRU-D in terms of AUROC by 1.07% (36 hours after ICU admission). In addition, significant reductions in prediction performance of $CDNet_+$ (*CDNet* with a time decay mechanism) are observed compared with *CDNet*. Taken together, these results suggest that there is high inconsistency on the effectiveness of the time decay mechanism.

Table 3.1: Performance of baselines and our approaches on in-hospital mortality prediction.

MIMIC-III/Mortality Prediction	24 hours after ICU admission		36 hours after ICU admission		48 hours after ICU admission	
Metrics	AUROC	AUPRC	AUROC	AUPRC	AUROC	AUPRC
Mean	0.6780(0.017)	0.2283(0.017)	0.6821(0.016)	0.2322(0.015)	0.6816(0.016)	0.2314(0.015)
KNN	0.7122(0.016)	0.2498(0.022)	0.7057(0.015)	0.2423(0.019)	0.7086(0.013)	0.2464(0.019)
MICE [119]	0.7089(0.019)	0.2582(0.024)	0.7113(0.020)	0.2551(0.023)	0.7058(0.018)	0.2435(0.019)
Simple	0.6821(0.012)	0.2315(0.010)	0.6806(0.012)	0.2307(0.010)	0.6791(0.013)	0.2279(0.012)
BRNN [151]	0.6735(0.011)	0.2037(0.012)	0.6704(0.010)	0.2023(0.012)	0.6732(0.011)	0.2051(0.014)
CATSI [52]	0.7042(0.011)	0.2373(0.012)	0.7024(0.013)	0.2343(0.015)	0.7057(0.012)	0.2379(0.012)
BRITS [40]	0.7463(0.010)	0.2880(0.016)	0.7445(0.009)	0.2856(0.016)	0.7447(0.009)	0.2879(0.016)
GRU-D [39]	0.7323(0.012)	0.2821(0.014)	0.7235(0.012)	0.2679(0.015)	0.7285(0.011)	0.2763(0.015)
GRU-D_{d-}	0.7137(0.011)	0.2689(0.016)	0.7342(0.015)	0.2624(0.015)	0.7244(0.011)	0.2673(0.014)
$CDNet_\alpha$	0.7536(0.008)	0.3252(0.016)	0.7502(0.011)	0.3031(0.015)	0.7546(0.008)	0.2994(0.014)
$CDNet_\beta$	0.7557(0.013)	0.3402(0.014)	0.7538(0.010)	0.3404(0.017)	0.7543(0.012)	0.3413(0.020)
CDNet	**0.7712(0.011)**	**0.3497(0.014)**	**0.7675(0.012)**	**0.3443(0.017)**	**0.7673(0.013)**	**0.3526(0.014)**
$CDNet_+$	0.7588(0.019)	0.3286(0.017)	0.7506(0.020)	0.3166(0.017)	0.7529(0.018)	0.3177(0.015)

3.3.3 Ablation Analysis

We now proceed to examine the effectiveness of different components of our *CDNet*. To this end, we conduct an ablation study on the datasets. We present two variants of *CDNet* as:

- $CDNet_\alpha$: *CDNet* without MDN and RAN.
- $CDNet_\beta$: *CDNet* with MDN without RAN.

We present the ablation study results in Table 3.1. We find that $CDNet_\beta$ outperforms $CDNet_\alpha$. Overall, $CDNet_\beta$ achieved significant performance improvements in AUPRC. These results demonstrate the effectiveness of the MDN construction. According to these results, we can also infer that $CDNet_\beta$ is more concerned with the balance of classification

than $CDNet_{\alpha}$. The superior performance of *CDNet* than $CDNet_{\beta}$ verifies the efficacy of RAN, in achieving performance improvements in AUROC and AUPRC.

3.3.4 Case study: Regularised Attention Network (RAN) Analysis

Figure 3.3 and Figure 3.4 present the results of two patient journeys (two random examples) obtained from the RAN analysis. The boxes with attention scores are imputed values. The larger the attention scores, the more reliable the imputed values. The attention scores ranging from 0.0 $\sim$ 1.0 were calculated by the RAN. The RAN takes into consideration the entire patient journey, but the images are understandably truncated for visibility. We take the first 20 records of the two patient journeys as an example for detailed discussion. Between Figure 3.3 and Figure 3.4, there is a significant difference between the degree of missingness of the two patient journeys. We can observe that less reliable imputed values are assigned lower weights in the two patient journeys. These results suggest that RAN not only can handle the different degrees of missingness of patient journey data but also has fine-grained robustness at the feature level of patient journey data.

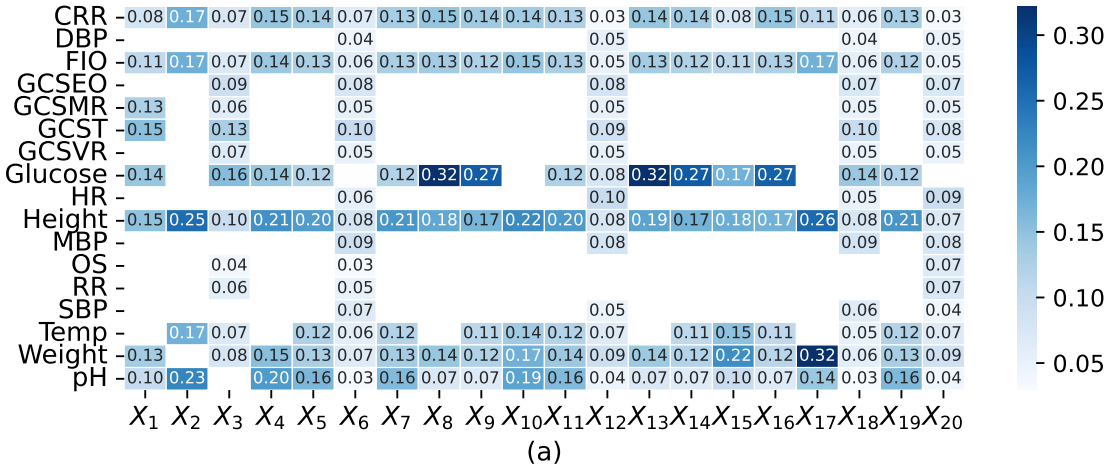

Figure 3.3: Result of Patient A.

3.3.5 Case study: Uncertainty Analysis

Figure 3.5 shows the results obtained from the epistemic uncertainty estimation of four patient journeys (four random examples). Each subgraph contains two predicted probability distributions of a patient journey, where dodger blue and dark orange histograms are derived from FFN (feed-forward network) ensembles and MDN, respectively. The MDN

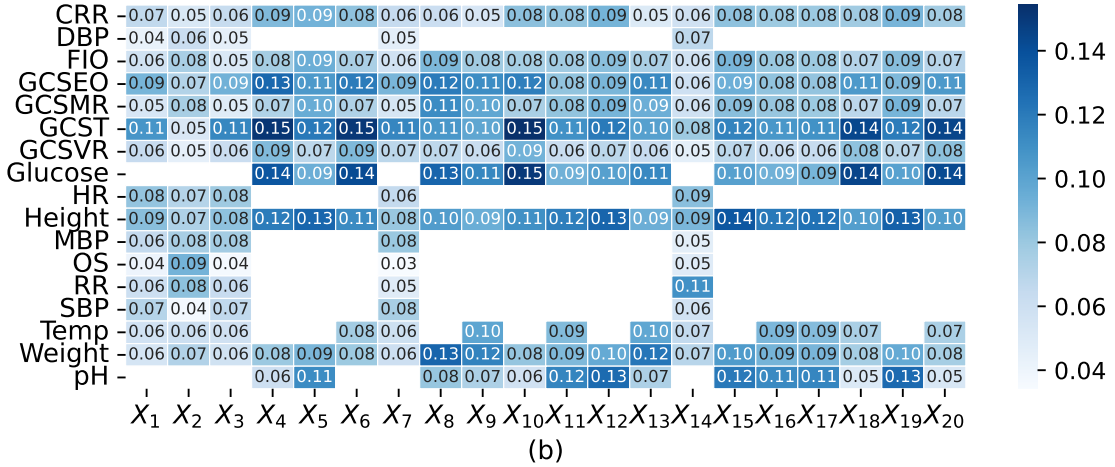

Figure 3.4: Result of Patient B.

used here is described in the subsection Risk Prediction. For MDN, we set the mixture components to 100 (K = 100). In terms of a patient journey, these 100 components would produce 100 prediction results, so that epistemic uncertainty of the prediction model can be quantified. For FFN ensembles, we replace the MDN with 100 FNNs that have different random seeds. The more the two discrete distributions (histograms) overlap, the better the ability of the model to capture epistemic uncertainty. From the data in Fig. 4, it is apparent that there are many overlaps between the two discrete distributions in each subgraph. These results suggest that our method is able to capture epistemic uncertainty similar to that of FFN ensembles.

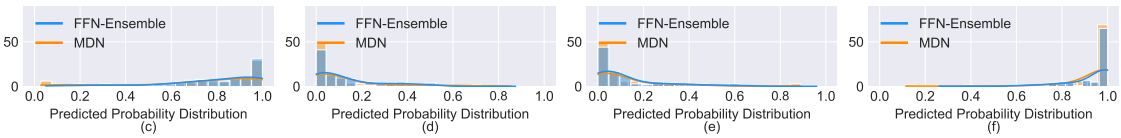

Figure 3.5: Predicted probability distribution of MDN (our method) and FFN-ensemble.

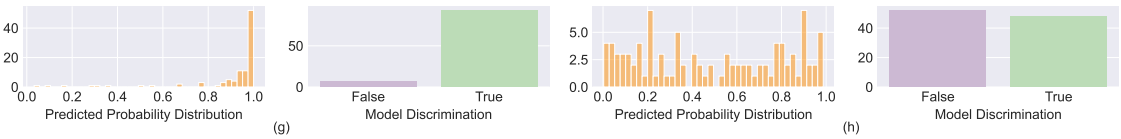

Figure 3.6: Epistemic uncertainty analysis. Two examples of the predicted probability distribution on the in-hospital mortality prediction task.

Figure 3.6 shows the results obtained from the epistemic uncertainty analysis of two patient journeys (two random examples). Figure 3.6 (g) left and Figure 3.6 (h) left compare the prediction results obtained from MDN. Figure 3.6 (g) right and Figure 3.6 (h) right

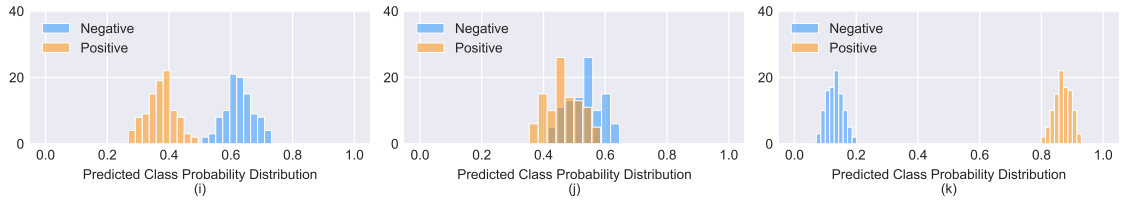

Figure 3.7: Aleatoric uncertainty analysis. Three examples of the predicted probability distribution on the in-hospital mortality prediction task.

show the corresponding model discriminations, with 'True' (likely to die) corresponding to a prediction of mortality and 'False' corresponding to the opposite (unlikely to die). From the patient in Figure 3.6 (g), the agreement among ensemble members of MDN about 'True' is high. In contrast, there is high disagreement for another patient in Figure 3.6 (h) due to epistemic uncertainty.

Figure 3.7 shows the results obtained from the aleatoric uncertainty analysis of three patient journeys (three random examples). Each subgraph contains two predicted class probability distributions of a patient journey, where dodger blue and dark orange histograms represent negative and positive classes, respectively, derived from MDN predictions. We augment MDN with 100 Gaussian noises to generate 100 class probability distributions for each of the two patient journeys. From the data in Figure 3.7 (i) and Figure 3.7 (k), we can see that the histograms corresponding to the probability distributions of the two predicted classes do not overlap. The results suggest that aleatoric uncertainty had less impact on mortality predictions in these two cases. In addition, the negative class (unlikely to die) in Figure 3.7 (i) reported significantly higher probability than the other group. Similarly, the positive class (likely to die) in Figure 3.7 (k) reported significantly higher probability than the other group. As can be seen from the data in Figure 3.7 (j), there is a large overlap between the histograms corresponding to the probability distributions of the two predicted classes. Thus, aleatoric uncertainty has a more significant impact on the mortality predictions of this patient. Although the result of model discrimination may be negative, this prediction should be taken with caution.

Chapter 4

Attention-Based Bidirectional Recurrent Neural Networks

The following publication has been incorporated into this chapter:
[73] **Yuxi Liu**, Zhenhao Zhang, and Shaowen Qin. Deep imputation-prediction networks for health risk prediction using electronic health records. In *2023 International Joint Conference on Neural Networks (IJCNN)*, pages 1–9. IEEE, 2023.

4.1 Introduction

Recent developments in deep learning techniques have stimulated interest in health risk prediction using Electronic Health Records (EHRs). Health risk refers to the probability of the occurrence of a patient's health outcome. The predicted health risks can be used to support decisions by healthcare professionals and improve healthcare delivery. Examples of successful applications include in-hospital mortality risk prediction [32], hospital readmission risk prediction [275], and disease risk prediction [30]. These studies aim to extract patient-specific contextual information from EHRs by combining deep neural networks and attention mechanisms to make risk predictions. Results obtained from these prediction models have shown improved accuracy.

Further improvements in risk prediction models are necessary before they can be adopted for real-world applications. This is challenging. One major issue is the high degree of irregularity of EHR data. Figure 4.1 shows the clinical records of two anonymous patients

from the publicly available MIMIC-III database [38]. The samples clearly indicated the irregularity problems, i.e., many missing values and varying time intervals between data points. Imputation of missing values, where the missing data are replaced with some substitute value to create a complete data matrix, offers an effective way of dealing with many missing values [118, 209, 276, 277]. Recent studies of using imputation methods for addressing missing values in EHR data include [56, 177, 278, 279].

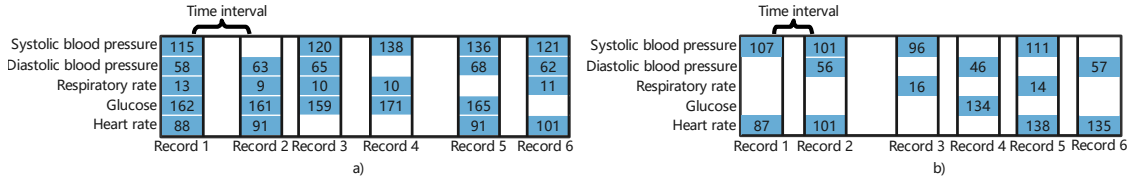

Figure 4.1: Illustration of clinical records of patients A and B.

However, with EHR datasets, missing values are likely attributed to patient symptoms and reflect treatment decisions. Different physiological variables (e.g., glucose and respiratory rate as shown in Figure 4.1) are examined at different times depending on a patient's health conditions. When a certain symptom disappears, corresponding variables are no longer examined. This would lead to missing values. Therefore, the patterns of missing data in patients' records may contain important information, e.g., different categories of missing data may reflect a typical clinical symptom, affecting both time independent and dependent variable relationships. More accurate, meaningful, and reliable imputation of missing values would allow us to uncover such important information that in turn can lead to a model with better prediction performance.

Previous studies [39, 40, 42] have combined the imputation method and prediction model in a single framework. InterpNet [42] represents such an example. InterpNet uses an interpolation network (an unsupervised network) to generate imputed values for missing values and a prediction network to generate prediction outcomes. Since it is difficult for an unsupervised network to obtain reasonable network parameters, its outputs (predicted and imputed values) are usually less than desirable. To deal with this problem, InterpNet also includes using the loss between predicted and real-valued values to train the interpolation network. However, InterpNet chooses to largely ignore the reliability of imputed values, even the relatively larger difference between imputed and real-valued values. So far, we have not found studies that offer adequate consideration of the reliability of imputed values. Such less reliable imputed values may lead to biased prediction outcomes, especially when existing prediction models are directly applied to predict health risks.

In terms of irregularity in time intervals, most EHR-based prediction research studies have focused on utilizing time-decay mechanisms to handle varying time intervals in longitudinal patient records. For example, GRU-D [39] takes into account the missing

values and time intervals when predicting the health risk of patients based on their historical EHR data. It mainly incorporates observed records and corresponding timestamps into GRU (Gated Recurrent Unit) [147] to impute missing values by the decay of previous input values toward the overall mean/sampling over time. The time-decay mechanism used in GRU-D continues to be used by a considerable amount of literature on imputation research [40, 43, 44, 46, 48].

Despite GRU-D's efficacy, it does not thoroughly consider the pattern of missing patient data that contains important information to be learned. As mentioned earlier, to capture changes in a patient's underlying health condition, physiological variables being examined and the time intervals between examinations vary. Accordingly, the variation pattern of physiological variables in diverse time intervals plays a vital role in understanding a patient's underlying health condition and predicting the patient's future condition. Besides, a patient's health status can become 'healthier', 'deteriorating', or recurrent. When predicting health outcomes, we should automatically include learning of the impact of the previous 48 hours of patient data on the prognosis (e.g., in-hospital mortality prediction [53]). If the predictive model found an association between the previous and current physiological variables, the previous physiological variables become critical indicative variables regardless of how long ago these were collected, which should be given sufficient consideration in the prediction model.

To address the aforementioned limitations, in this chapter, we propose a deep imputation-prediction network to perform imputation and prediction with EHR data. Two novel reliability-aware reconstruction (RARM) and time-decay attention (TDAM) modules are integrated into a bidirectional GRU. The bidirectional GRU learns from the longitudinal patient data in both forward and backward directions and generates the hidden state representations. The RARM translates hidden state representations into predicted and imputed values, implemented by constructing Gaussian mixture distributions and sampling these distributions.

The RARM comprises a deep neural network, a mixture of distributions, and an attention network. The deep neural network uses the hidden state representations to generate multiple Gaussian distributions, including making weights for creating Gaussian mixture distribution (i.e., a multimodal distribution). The multimodal distribution contributes to modeling complex patterns found in the input. The outputs of the sampling of multimodal distribution are predicted and imputed values. The attention network models the weights as a function of the variance of the multimodal distribution, which assigns smaller weights to imputed values with large variance and vice versa, thus enhancing the reliability of imputed values.

The TDAM comprises a time-decay mechanism and an attention network. The time-

decay mechanism incorporates three common decay functions to capture the variation pattern of input variables at the time dimension and adaptively enhances the temporal representation of each pattern with adjustable weights. The attention network examines the association between input variables. By learning the context of a given patient data, TDAM is able to identify critical indicative variables regardless of how long ago the associated event happened.

The main contributions of this paper are listed as follows: We propose a deep imputation-prediction network to perform both imputation and prediction in EHR data. It effectively handles the irregularity of EHR data, including many missing values and varying time intervals, which leads to good prediction performance. To demonstrate the efficacy of our proposed method, we conduct imputation and prediction experiments on two publicly available databases: MIMIC-III [38] and eICU [280]. The results demonstrate the imputation effectiveness and prediction superiority of our method. Further analysis of RARM and TDAM shows that our method can provide transparency and interpretability of the model decisions, which is another important advantage.

4.2 Method

In this section, we describe our proposed Deep Imputation-Prediction Network for health risk predictions. We introduce the basic notations first. We then detail the network architecture. Finally, we present how to use the Deep Imputation-Prediction Network for health risk predictions.

4.2.1 Basic Notations

In the dataset, each patient journey is a set of time-ordered clinical records, denoted by $x = \{x_1, x_2, \cdots, x_T\} \in \mathbb{R}^{N \times T}$, where T is the total number of records. Each record consists of up to N time-variant features, i.e., vital signs, denoted by $x_t = \{x_t^1, x_t^2, \cdots, x_t^N\} \in \mathbb{R}^N$, where x_t is the patient's record at time step t, and x_t^i is the i-th time-variant feature of x_t. Let s_t and s_{t+1} denote the temporal information of x_t and x_{t+1}. $\delta_{i,t}$ denotes the time interval between x_i and x_t. $\delta_{i,t} = s_t - s_i$, $t = 1, 2, \cdots, t-1$. Since each x is incomplete, we use a mask matrix $m \in \mathbb{R}^{N \times T}$ to present whether the values of x exist or not, i.e., $m_t^i = 1$, if x_t^i exists, otherwise $m_t^i = 0$.

4.2.2 Network Architecture

Our proposed network architecture comprises three key components: 1) a bidirectional GRU, 2) a reliability-aware reconstruction module, and 3) a time-decay attention module, as shown in Figure 4.2.

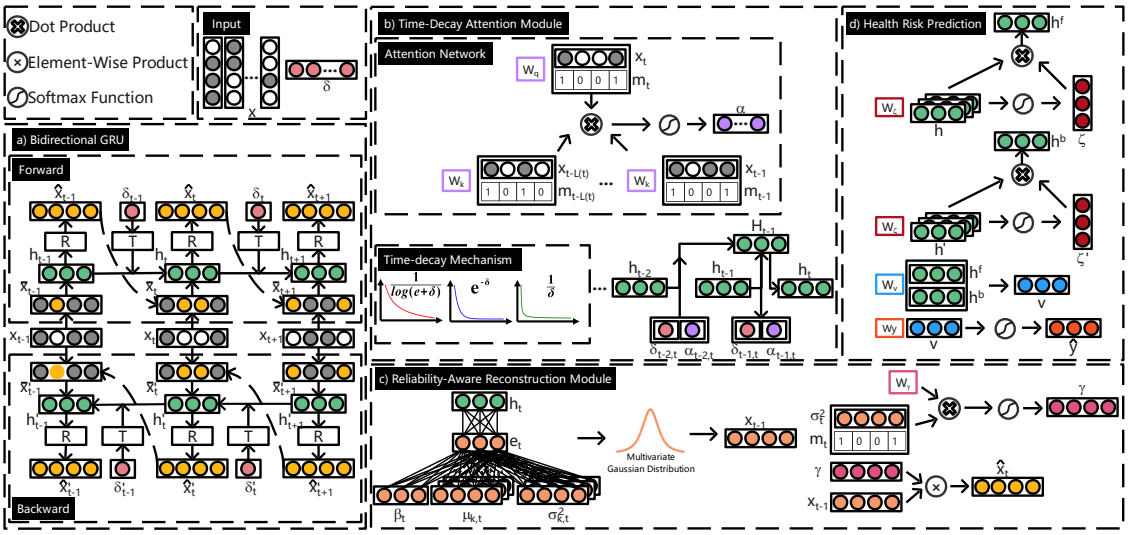

Figure 4.2: Schematic representation of the architecture and workflow of the proposed network.

Bidirectional GRU

We incorporate a bidirectional GRU into the proposed method to learn from each group of data x in both forward and backward directions and generate the hidden state representations. We take the forward direction as an example:

$$\begin{aligned} \hat{x}_{t-1} &= R(h_{t-1}), \\ \bar{x}_t &= m_t \odot x_t + (1 - m_t) \odot \hat{x}_{t-1}, \\ H_{t-1} &= T(\{\delta_{i,t}\}_{i=1}^{t-1}, h_{t-1}), \\ r_t &= \sigma(W_r \cdot [H_{t-1}, \bar{x}_t] + b_r), \\ u_t &= \sigma(W_u \cdot [H_{t-1}, \bar{x}_t] + b_u). \end{aligned} \tag{4.1}$$

The sigmoid activation function σ normalizes r_t and u_t in $[0, 1]$. $R(\cdot)$ is the designed Reliability-Aware Reconstruction Module that translates the hidden state representation h_{t-1} into the predicted $\hat{x}_{t-1}$. $T(\cdot)$ is the designed Time-Decay Attention Module that models varying time intervals. The $\bar{x}_t$ and the element-wise multiplication of H_{t-1} with r_t are used to generate an intermediate $\widetilde{h}_t$. The element-wise convex combinations between $\widetilde{h}_t$ and u_t

is used to obtain h_t as:

$$\begin{aligned} \widetilde{h_t} &= tanh(W_h \cdot [r_t \odot H_{t-1}, \bar{x}_t] + b_h), \\ h_t &= u_t \odot H_{t-1} + (1 - u_t) \cdot \widetilde{h_t}. \end{aligned} \tag{4.2}$$

The final outputs of the bidirectional GRU, including $\hat{x}$, h, $\hat{x}'$, and h'. $\hat{x}$ and h are obtained from the forward direction. $\hat{x}'$ and h' are obtained from the backward direction.

Reliability-Aware Reconstruction Module (RARM)

We design a reliability-aware reconstruction module to translate h_t into $\hat{x}_t$. It comprises a feed-forward network (FFN), a mixture of distributions (i.e., multiple Gaussian distributions because both the predicted and imputed values are continuous), and an attention network. We feed hidden state representations into the FFN to generate means and variances for multiple Gaussian distributions and, moreover, weights for making a Gaussian mixture distribution as:

$$\begin{aligned} p(\widetilde{x_t}|h_t) &= \sum_{k=1}^{K} \beta_k \cdot D_k(\widetilde{x_t}|h_t), \\ D_k(\bar{x}_t|h_t) &= \mathcal{N}(\mu_k, \Sigma_k), \end{aligned} \tag{4.3}$$

where k is the index of the corresponding mixture distribution, and each output has up to K mixture distributions. β is the mixing parameter. D is the corresponding distribution to be mixed, including the mean vector μ and the covariance matrix Σ with σ^2 on the diagonal and 0 otherwise as:

$$\begin{aligned} e_t &= W^e \cdot h_t + b^e, \\ \beta_t &= Softmax(W^\alpha \cdot e_t + b^\alpha), \\ \mu_{k,t} &= W^\mu_k \cdot e_t + b^\mu_k, \\ \sigma^2_{k,t} &= ELU(W^\sigma_k \cdot e_t + b^\sigma_k) + 1 + \epsilon, \end{aligned} \tag{4.4}$$

where β is the mixture weight of each distribution. Particularly, we augment the ELU activation [281] with an offset to keep the variance of Gaussian greater than 0, i.e., $\sigma^2 > 0$, and moreover, with a constant term ϵ (e.g., 1e-12) to ensure stability. Subsequently, we incorporate Gaussian noise ξ and variance Σ_k to μ_k. The predicted $\widetilde{x_t}$ can be written as:

$$\begin{aligned} x^s_{k,t} &= \mu_{k,t} + \sqrt{\Sigma_{k,t}} \cdot \xi, \xi \sim \mathcal{N}(0, 1), \\ \widetilde{x_t} &= \sum_{k=1}^{K} \beta_{k,t} \cdot x^s_{k,t}. \end{aligned} \tag{4.5}$$

Since imputed values are inferred from real-valued values, they are less reliable. Therefore, we design an attention network to regularize the weights assigned to imputed values as:

$$\begin{aligned} \gamma_t &= Softmax(W_\gamma \cdot (1 - (1 - m_t) \odot \sigma_t^2) + b_\gamma), \\ \hat{x}_t &= \gamma \odot \widetilde{x_t}, \end{aligned} \tag{4.6}$$

where $\sigma_t^2 = \sum_{k=1}^{K} \beta_{k,t} \cdot \sigma_{k,t}^2$ is the mixed variance of Gaussian mixture distribution. m is a mask matrix (see Section III-A). $\hat{x}_t$ is obtained by combining γ with $\bar{x}_t$.

Time-Decay Attention Module (TDAM)

We design a time-decay attention module to model varying time intervals of each patient journey x. It comprises a time-decay mechanism and an attention network. The time-decay mechanism relies on three common decay functions to capture the variation pattern of input variables in time dimensions and adaptively enhances the temporal representation of each pattern with adjustable weights as:

$$\begin{aligned} g(\delta_{i,t}) = tanh(\lambda_1 \cdot \frac{1}{log(e + w_1 \cdot \delta_{i,t})} \\ +\lambda_2 \cdot e^{-w_2 \cdot \delta_{i,t}} + \lambda_3 \cdot \frac{1}{w_3 \cdot \delta_{i,t}}), \end{aligned} \tag{4.7}$$

where w_1, w_2, and w_3 are learnable parameters. $\frac{1}{log(e+w_1 \cdot \delta_{i,t})}$ [282], $e^{-w_2 \cdot \delta_{i,t}}$ [39], and $\frac{1}{w_3 \cdot \delta_{i,t}}$ [148] are three decay functions. λ_1, λ_2, and λ_3 are learnable weights.

Meanwhile, the attention network examines the association between $\{x_i\}_{i=1}^{t-1}$ and x_t as:

$$\begin{aligned} FFN = sigmoid(W_2^L \cdot tanh(W_1^L \cdot (m_t \odot x_t) + b_1^L) + b_2^L), \\ L(t) = \lceil (t-1) \cdot FFN(m_t \odot x_t) \rceil, \\ q = W_q \cdot (m_t \cdot x_t), \\ k_i = W_k \cdot (m_t \odot x_t), i = t - L(t), \cdots, t-1, \\ \alpha_{t-L(t)}, \cdots, \alpha_{t-1} = Softmax(q \cdot k_{t-L(t)}, \cdots, q \cdot k_{t-1}), \end{aligned} \tag{4.8}$$

where FFN$(\cdot)$ is a 2-layer FFN with tanh and sigmoid activation functions. $L(t) \in [1, t-1]$ is a learnable parameter for the number of records to look backward. W_1^L, W_2^L, W_α, b_1^L, and b_2^L are learnable parameters. m is a mask matrix (see Section III-A).

The outputs of the time-decay mechanism and attention network are integrated into an

overall hidden state representation as:

$$H_{t-1} = \sum_{i=t-L(t)}^{t-1} \alpha_{i,t} \cdot g(\delta_{i,t}) \cdot h_i. \tag{4.9}$$

Health Risk Prediction

We use h and h' to make health risk predictions. Specifically, the following steps were taken: we design an attention network to generate h^f (f denotes forward) and h^b (b denotes backward) from h and h'. We integrate h^f and h^b into v. We apply a Softmax output layer to v to obtain the predicted probability $\hat{y}$:

$$\begin{aligned} \zeta = Softmax(W_\zeta \cdot h + b_\zeta), \zeta' = Softmax(W_{\zeta'} \cdot h' + b_{\zeta'}), \\ h^f = \sum_{t=1}^{T} \zeta_t \odot h_t, h^b = \sum_{t=1}^{T} \zeta'_t \odot h'_t, \\ v = W_v \cdot [h^f, h^b] + b_v, \\ \hat{y} = Softmax(W_y \cdot v + b_y). \end{aligned} \tag{4.10}$$

We calculate the loss using the cross-entropy between the ground truth y and the predicted probability $\hat{y}$. Thus, we use the average of cross entropy as the objective function of health risk prediction:

$$\mathcal{L} = -\frac{1}{P} \sum_{p=1}^{P} (y_p^\top \cdot log(\hat{y}_p) + (1 - y_p)^\top \cdot log(1 - \hat{y}_p)), \tag{4.11}$$

where P is the number of patient journeys. y_p is the ground truth class/label for the patient p's journey.

4.3 Experiments

4.3.1 Experimental Setup

Datasets

We experiment with multivariate clinical times series extracted from the MIMIC-III[1] and eICU[2] databases. The multivariate clinical times series were extracted based on literature

[1] https://mimic.physionet.org
[2] https://eicu-crd.mit.edu/

[53, 54]. Tables A.1 and A.2 provide the summary statistics for the multivariate clinical times series.

Imputation and Prediction Tasks

We conduct imputation and prediction tasks with different lengths of observation window (i.e., 24 hours and 48 hours) on both databases. We take an observation window of 48 hours as an example:

- **Multivariate clinical time series imputation and In-hospital mortality prediction (48 hours after ICU admission)** to evaluate ICU imputation and mortality accuracy based on the data from the first 48 hours after ICU admission.
- **Multivariate clinical time series imputation and In-hospital mortality prediction (48 hours after eICU admission)** to evaluate eICU imputation and mortality accuracy based on the data from the first 48 hours after eICU admission.

Baseline Approaches

We compare the proposed approach with BRITS [40], InterpNet [42], GRU-D [39], GRU-D_{t-} (without time-decay mechanism), GRUI-GAN [43], E^2GAN [44], Bi-GAN [45], and STING [46]. Since GRU-D is generated for prediction tasks, it does not include making components for obtaining imputation results. Because of this, we augment GRU-D with the BRITS regression component to obtain imputation accuracy. We feed the outputs of GRUI-GAN, E^2GAN, Bi-GAN, and STING into GRU to generate the prediction outcomes.

Implementation Details

The two EHR datasets are randomly divided into three components, including training, validation, and testing sets, for each task, in a 70:15:15 ratio. The validation set is used to obtain the best values of parameters. For training the model, we use Adam optimizer [283] with the learning rate of 1×10^{-3}, and the mini-batch size of 256. For bidirectional GRU, the dimension size of hidden state representation g is 12. For TDAM, the dimension size of W_1^L and W_2^L are 17 and 1. For RARM, the number of mixture distributions K is 7. The dimension size of W^e, W^α, W^μ, and W^σ is 24. For health risk prediction, the dimension size of W_ζ and $W_{\zeta'}$ is 1. The dimension size of W_v is 12. The dropout method is applied to the final Softmax output layer. The dropout rate is 0.3. For highly imbalanced datasets, we augment the CrossEntropyLoss function with class weight. We repeat each method ten times to obtain the mean and standard deviation of the evaluation metrics. All experiments are implemented with PyTorch 1.10.0 on A40 GPU from NVIDIA with 48GB of memory.

Evaluation Metrics

We evaluate the imputation performance of our approach with the mean absolute error (MAE) and the mean relative error (MRE) between predicted and real-valued values. Given $\hat{x}_i$ and x_i as the i-th predicted and real-valued value, as well as the total number of ground truth P_{GT}, the MAE and MRE are written as:

$$MAE = \frac{\sum_{i=1}^{P_{GT}} |\hat{x}_i - x_i|}{P_{GT}} \tag{4.12}$$

$$MRE = \frac{\sum_{i=1}^{P_{GT}} |\hat{x}_i - x_i|}{\sum_{i=1}^{P_{GT}} |x_i|} \tag{4.13}$$

We evaluate the prediction performance of our approach with the area under the receiver operating characteristic curve (AUROC) and the area under the precision-recall curve (AUPRC).

The AUROC is calculated as the area under the ROC curve. A ROC curve represents the trade-off between true positive rate (TPR) and false positive rate (FPR) across different decision thresholds. A 'decision threshold' is the number a probability is being compared with to decide if that probability should indicate the positive or negative class. The TPR and FPR are written as:

$$TPR = \frac{TP}{TP + FN} \tag{4.14}$$

$$FPR = \frac{FP}{FP + TN} \tag{4.15}$$

The AUPRC is calculated as the area under the PR curve. A PR curve represents the trade-off between precision and recall across different decision thresholds. The precision and recall are written as:

$$Precision = \frac{TP}{TP + FP}, \tag{4.16}$$

$$Recall = \frac{TP}{TP + FN}. \tag{4.17}$$

4.3.2 Performance Analysis

Table 4.1 lists the results of multivariate clinical time series imputation and physiologic decompensation prediction based on the physiological data from the first 24 hours after ICU/eICU admission. Table 4.2 lists the results of multivariate clinical time series imputation and in-hospital mortality prediction based on the physiological data from the first

48 hours after ICU/eICU admission. The larger the scores of AUROC and AUPRC, the better the predictive performance of the method. The lower the values of MAE and MRE, the better the imputation performance of the method. Values in the parentheses denote standard deviations. These results suggest that our method outperforms other baseline methods on imputation and prediction tasks.

The two tables (4.1 and 4.2) are quite revealing in several ways. First, there is a significant difference in the performance of baseline methods on both the imputation and prediction tasks. For the multivariate clinical time series imputation of MIMIC-III (24 hours after ICU admission), GRU-D outperforms E^2GAN. However, E^2GAN outperforms GRU-D in the physiologic decompensation prediction of MIMIC-III. Second, the two tables (4.1 and 4.2) show that the GAN-based imputation method resulted in higher MAE and MRE values than the RNN-based imputation method. Third, the imputation performance of baseline methods (except for Bi-GAN) over 24 hours is better than that of 48 hours. Fourth, no significant performance difference was found between GRU-D and GRU-D_{t-} on both the imputation and prediction tasks.

Table 4.1: Performance of baselines and our method on multivariate clinical time series imputation and physiologic decompensation prediction.

MIMIC-III/24 hours after ICU admission	Multivariate clinical time series imputation		Physiologic decompensation prediction	
Metrics	MAE	MRE	AUROC	AUPRC
BRITS [40]	4.6305(0.3451)	53.55%(0.0485)	0.7387(0.0093)	0.2794(0.0154)
InterpNet [42]	2.4345(0.0038)	55.82%(0.0009)	0.6476(0.0091)	0.2025(0.0118)
GRU-D [39]	3.1752(0.0151)	36.74%(0.0017)	0.7277(0.0111)	0.2785(0.0133)
GRU-D_{t-}	3.1774(0.0167)	36.76%(0.0019)	0.7103(0.0094)	0.2597(0.0075)
GRUI-GAN [43]	6.2258(0.0026)	71.97%(0.0003)	0.7188(0.0098)	0.2655(0.0104)
E^2GAN [44]	6.1391(0.0056)	70.95%(0.0007)	0.7283(0.0070)	0.2624(0.0093)
Bi-GAN [45]	5.9098(0.0454)	68.38%(0.0052)	0.7262(0.0078)	0.2630(0.0103)
STING [46]	4.6212(0.0162)	53.43%(0.0019)	0.7312(0.0083)	0.2579(0.0115)
Ours	**1.2835(0.0013)**	**15.05%(0.0002)**	**0.7507(0.0089)**	**0.2827(0.0122)**
eICU/24 hours after eICU admission	Multivariate clinical time series imputation		Physiologic decompensation prediction	
Metrics	MAE	MRE	AUROC	AUPRC
BRITS [40]	2.7905(0.2666)	36.21%(0.0234)	0.7082(0.0085)	0.2617(0.0083)
InterpNet [42]	3.1968(0.0012)	41.51%(0.0001)	0.7323(0.0045)	**0.3339(0.0100)**
GRU-D [39]	1.6043(0.0054)	20.82%(0.0007)	0.7024(0.0081)	0.2776(0.0134)
GRU-D_{t-}	1.5939(0.0067)	20.68%(0.0009)	0.6981(0.0109)	0.2668(0.0137)
GRUI-GAN [43]	5.9463(0.0057)	77.13%(0.0007)	0.7145(0.0099)	0.2996(0.0083)
E^2GAN [44]	5.7179(0.0050)	74.16%(0.0004)	0.7159(0.0101)	0.3057(0.0132)
Bi-GAN [45]	5.5418(0.0174)	71.21%(0.0024)	0.7269(0.0105)	0.2981(0.0158)
STING [46]	5.2312(0.0609)	69.85%(0.0079)	0.7268(0.0098)	0.2976(0.0108)
Ours	**1.0254(0.0005)**	**13.36%(0.0001)**	**0.7445(0.0041)**	**0.3159(0.0072)**

4.3.3 Ablation Analysis

We conduct an ablation study to examine the effectiveness of different modules of our method in imputation and prediction tasks. Four variants of our method are presented as follows:

Table 4.2: Performance of baselines and our method on multivariate clinical time series imputation and in-hospital mortality prediction.

MIMIC-III/48 hours after ICU admission	Multivariate clinical time series imputation		In-hospital mortality prediction	
Metrics	MAE	MRE	AUROC	AUPRC
BRITS [40]	5.3631(0.3804)	52.65%(0.0374)	0.7447(0.0092)	0.2879(0.0168)
InterpNet [42]	3.0316(0.0058)	37.83%(0.0007)	0.6664(0.0057)	0.2136(0.0059)
GRU-D [39]	3.6873(0.0218)	36.20%(0.0021)	0.7294(0.0097)	0.2771(0.0156)
GRU-D_{t-}	3.7043(0.0041)	36.37%(0.0004)	0.7267(0.0094)	0.2702(0.0189)
GRUI-GAN [43]	7.1359(0.0055)	70.05%(0.0005)	0.7619(0.0077)	0.3349(0.0178)
E^2GAN [44]	6.9705(0.0104)	68.43%(0.0010)	0.7652(0.0054)	**0.3599(0.0133)**
Bi-GAN [45]	5.6357(0.0244)	55.37%(0.0088)	0.7649(0.0089)	0.3343(0.0143)
STING [46]	5.1522(0.0202)	50.88%(0.0020)	0.7667(0.0106)	0.3402(0.0187)
Ours	**1.4982(0.0021)**	**14.81%(0.0002)**	**0.7828(0.0117)**	**0.3501(0.0205)**
eICU/48 hours after eICU admission	Multivariate clinical time series imputation		In-hospital mortality prediction	
Metrics	MAE	MRE	AUROC	AUPRC
BRITS [40]	4.0963(0.3359)	31.26%(0.0257)	0.7254(0.0057)	0.2573(0.0062)
InterpNet [42]	3.6726(0.0008)	28.06%(0.0003)	0.7514(0.0029)	0.2938(0.0138)
GRU-D [39]	2.8066(0.0107)	21.43%(0.0008)	0.7195(0.0111)	0.2631(0.0145)
GRU-D_{t-}	2.7735(0.0104)	21.18%(0.0008)	0.7216(0.0144)	0.2614(0.0128)
GRUI-GAN [43]	9.9809(0.0056)	76.26%(0.0002)	0.7280(0.0105)	0.2871(0.0120)
E^2GAN [44]	9.7912(0.0111)	74.70%(0.0006)	0.7294(0.0106)	0.2970(0.0133)
Bi-GAN [45]	8.1643(0.0149)	62.35%(0.0010)	0.7241(0.0089)	0.2929(0.0104)
STING [46]	8.0315(0.0466)	61.21%(0.0036)	0.7475(0.0186)	0.2838(0.0197)
Ours	**1.7047(0.0012)**	**13.08%(0.0001)**	**0.7688(0.0093)**	**0.3025(0.0092)**

Ours$_\alpha$: A variation of our method in which we omit the time-decay attention module.

Ours$_\beta$: A variation of our method in which we omit the attention network from the reliability-aware reconstruction module.

Ours$_\gamma$: A variation of our method in which we omit the attention network from the time-decay attention module.

Ours$_\delta$: A variation of our method in which we replace the time-decay attention module with the GRU-D time-decay mechanism.

The results of the ablation study can be compared in Table 4.3. Note that both Ours$_\beta$ and Ours use the same experimental settings in multivariate clinical time series imputation tasks. We find that Ours outperforms Ours$_\alpha$. It suggests that capturing the variation pattern of input variables at time dimensions and adaptively enhancing the temporal representation of each pattern with adjustable weights is helpful for improving the imputation and prediction performance. Ours outperforms Ours$_\beta$, which shows that enhancing the reliability of imputed values is practical for improving prediction performance. Ours outperforms Ours$_\gamma$, which demonstrates that identifying critical indicative features and regenerating the feature embeddings under the context of a given patient data is critical for improving the imputation and prediction performance. The superior performance of Ours compared to Ours$_\delta$ demonstrates the effectiveness of our proposed TDAM, which can capture the diversity among the variation pattern of input variables at time dimensions and thus improve the imputation and prediction performance.

Table 4.3: Ablation performance comparison.

MIMIC-III/24 hours after ICU admission	Multivariate clinical time series imputation		Physiologic decompensation prediction	
Metrics	MAE	MRE	AUROC	AUPRC
Ours_{α}	1.2990(0.0208)	15.25%(0.0025)	0.7331(0.0137)	0.2669(0.0115)
Ours_{β}	-	-	0.7449(0.0083)	0.2631(0.0106)
Ours_{γ}	1.3076(0.0146)	15.35%(0.0018)	0.7383(0.0069)	0.2665(0.0091)
Ours_{δ}	1.2964(0.0007)	15.21%(0.0001)	0.7326(0.0049)	0.2673(0.0076)
Ours	**1.2835(0.0013)**	**15.05%(0.0002)**	**0.7507(0.0089)**	**0.2827(0.0122)**
eICU/24 hours after eICU admission	Multivariate clinical time series imputation		Physiologic decompensation prediction	
Metrics	MAE	MRE	AUROC	AUPRC
Ours_{α}	1.0266(0.0007)	13.38%(0.0001)	0.7211(0.0100)	0.3011(0.0120)
Ours_{β}	-	-	0.7359(0.0088)	0.2983(0.0081)
Ours_{γ}	1.0264(0.0007)	13.37%(0.0001)	0.7363(0.0106)	0.2990(0.0108)
Ours_{δ}	1.0266(0.0006)	13.38%(0.0001)	0.7328(0.0090)	0.3083(0.0119)
Ours	**1.0254(0.0005)**	**13.36%(0.0001)**	**0.7445(0.0041)**	**0.3159(0.0072)**
MIMIC-III/48 hours after ICU admission	Multivariate clinical time series imputation		In-hospital mortality prediction	
Metrics	MAE	MRE	AUROC	AUPRC
Ours_{α}	1.5215(0.0147)	15.05%(0.0015)	0.7681(0.0059)	0.3184(0.0095)
Ours_{β}	-	-	0.7621(0.0121)	0.2944(0.0168)
Ours_{γ}	1.5246(0.0170)	15.08%(0.0017)	0.7789(0.0101)	0.3430(0.0139)
Ours_{δ}	1.5290(0.0131)	15.12%(0.0028)	0.7658(0.0065)	0.3185(0.0102)
Ours	**1.4982(0.0021)**	**14.81%(0.0002)**	**0.7828(0.0117)**	**0.3501(0.0205)**
eICU/48 hours after eICU admission	Multivariate clinical time series imputation		In-hospital mortality prediction	
Metrics	MAE	MRE	AUROC	AUPRC
Ours_{α}	1.7141(0.0142)	13.17%(0.0008)	0.7615(0.0097)	0.2995(0.0127)
Ours_{β}	-	-	0.7579(0.0087)	0.2930(0.0086)
Ours_{γ}	1.7132(0.0030)	13.14%(0.0002)	0.7542(0.0143)	0.2985(0.0160)
Ours_{δ}	1.7066(0.0021)	13.14%(0.0002)	0.7450(0.0105)	0.2902(0.0111)
Ours	**1.7047(0.0012)**	**13.08%(0.0001)**	**0.7688(0.0093)**	**0.3025(0.0092)**

4.3.4 Case Study: Visualization Analysis

We further examine the transparency and interpretability of our method with random examples selected from the MIMIC-III database (48 hours after ICU admission), which is demonstrated in Figure 4.3 and Figure 4.4. The multiple Gaussian distributions of three patient journeys obtained from the RARM analysis are presented in Figure 4.3. By querying the MIMIC-III database, we found significant differences in primary disease between patients A, B, and C. Patients B and C had diabetes mellitus, and patient A had no diabetes mellitus. As shown in Figure 4.3, RARM determines that the glucose of patients B and C are modeled as a Gaussian mixture distribution (i.e., multimodal). A possible explanation for this might be that RARM found more complex patterns in the glucose of patients B and C. Patient B also had acute respiratory failure. Comparing the three patient journey results, it can be seen that patient B's respiratory rate (RR) and oxygen saturation (OS) are modeled as multimodal distributions. In reviewing the literature [284–286], we found that there are relatively high associations between respiratory rate, oxygen saturation, and acute respiratory failure. These results corroborate the ideas of [287], which suggested that the conditional distribution should be multimodal for tasks such as structured prediction problems, forming one-to-many mappings.

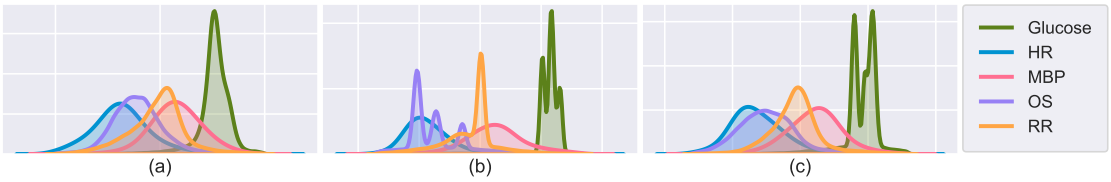

Figure 4.3: The multiple Gaussian distributions of Glucose, Heart Rate (HR), Mean blood pressure (MBP), Oxygen saturation (OS), and Respiratory rate (RR) for three patient journeys (i.e., patients A, B, and C).

The results of all patient journeys obtained from the TDAM analysis are presented in Figure 4.4. From the graph above we can see that the decay trends are similar between features such as capillary refill rate, mean blood pressure, systolic blood pressure, and weight. Data from Figure 4.4 can be compared with the data in Figure 4.2, which shows that the decay trend of the above features is consistent with $\frac{1}{\delta}$ [148]. Similarly, the decay trend of diastolic blood pressure, glasgow coma scale total, and oxygen saturation are consistent with $e^{-\delta}$ [39]. These results suggest that TDAM can effectively capture the variation pattern of input features at time dimensions and adaptively enhance the temporal representation of each pattern with adjustable weights (Section III-B3, λ_1, λ_2, and λ_3).

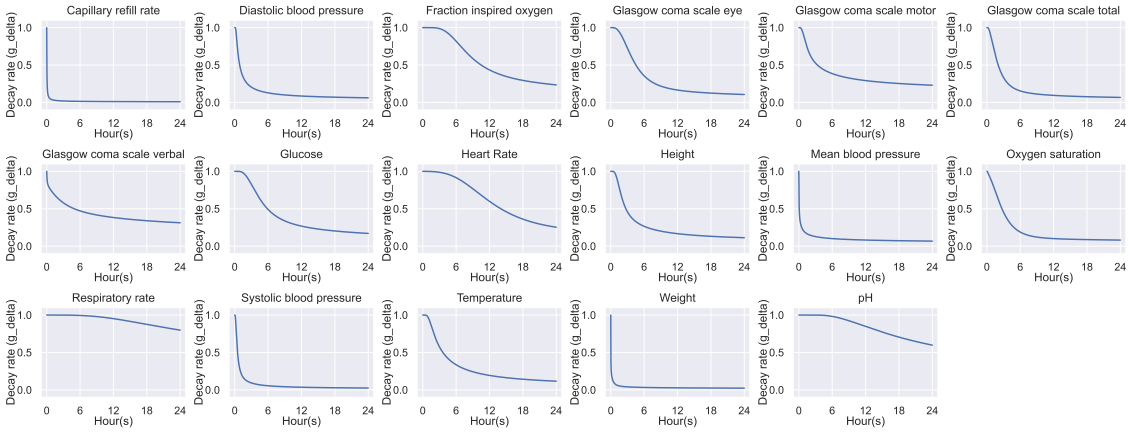

Figure 4.4: Plots of decay rate for features used from the MIMIC-III database.

Chapter 5

Contrastive Neural Networks

The following manuscript has been incorporated into this chapter:
[74] **Yuxi Liu**, Zhenhao Zhang, Shaowen Qin, Flora D Salim, Antonio Jimeno Yepes. Contrastive Learning-based Imputation-Prediction Networks for In-hospital Mortality Risk Modeling using EHRs. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 428–443. Springer, 2023.

5.1 Introduction

The broad adoption of digital healthcare systems produces a large amount of electronic health records (EHRs) data, providing us the possibility to develop predictive models and tools using machine learning techniques that would enable healthcare professionals to make better decisions and improve healthcare outcomes. One of the EHR-based risk prediction tasks is to predict the mortality risk of patients based on their historical EHR data [53, 54]. The predicted mortality risks can be used to provide early warnings when a patient's health condition is about to deteriorate so that more proactive interventions can be taken.

However, due to a high degree of irregularity in the raw EHR data, it is challenging to directly apply traditional machine learning techniques to perform predictive modeling. We take the medical records of two anonymous patients from the publicly available MIMIC-III database and present these in Figure 5.1 as an example. Figure 5.1 clearly indicates the irregularity problem, including many missing values and varying time intervals between

medical records.

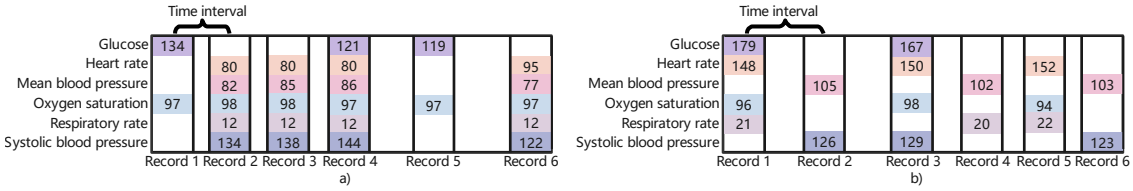

Figure 5.1: Illustration of medical records of patients A and B.

Most studies have focused on exploiting variable correlations in patient medical records to impute missing values and establishing time-decay mechanisms to take into account the effect of varying time intervals between records [39,40,43,44,46,47,60,65]. After obtaining the complete data matrices from the imputation task, the complete data matrices are used as input for downstream healthcare prediction tasks [39, 40, 43, 44, 49, 51, 56, 60, 65, 177,278]. Although these studies have achieved satisfactory imputation performance, consideration of using the information of similar patients on the imputation task, which might lead to improved imputation performance, has not yet been fully experimented. Furthermore, with imputation data, high-quality representation must be applied, as the imputation data may affect the performance of downstream healthcare prediction tasks.

Patient stratification refers to the method of dividing a patient population into subgroups based on specific disease characteristics and symptom severity. Patients in the same subgroup generally had more similar health trajectories. Therefore, we propose to impute missing values in patient data using information from the subgroup of similar patients rather than the entire patient population.

In this chapter, we propose a novel contrastive learning-based imputation-prediction network with the aim of improving in-hospital mortality prediction performance using EHR data. Missing value imputation for EHR data is done by exploiting similar patient information as well as patients' personal contextual information. Similar patients are generated from patient similarity calculation during stratification modeling and analysis of patient graphs.

Contrastive learning has been proven to be an important machine learning technique in the computer vision community [288]. In contrastive learning, representations are learned by comparing input samples. The comparisons are made on the similarity between positive pairs or dissimilarity between negative pairs. The main goal is to learn an embedding space where similar samples are put closer to each other while dissimilar samples are pushed farther apart. Contrastive learning can be applied in both supervised [289–291] and unsupervised [292–294] settings.

Motivated by the recent developments in contrastive representation learning [295–297], we integrate contrastive learning into the proposed network architecture to perform

imputation and prediction tasks. The benefit of incorporating contrastive learning into the imputation task is that such an approach can enhance patient representation learning by keeping patients of the same stratification together and pushing away patients from different stratifications. This would lead to enhanced imputation performance. The benefit of incorporating contrastive learning into the prediction task is improved predictive performance of the binary classification problem (i.e., the risk of death and no death), which is achieved by keeping the instances of a positive class closer and pushing away instances from a negative class.

The main contributions of this paper are listed as follows:

- To the best of our knowledge, this is the first attempt to consider patient similarity via stratification of EHR data on the imputation task.
- We propose a novel imputation-prediction approach to perform imputation and prediction with EHR data.
- We successfully integrate contrastive learning into the proposed network architecture to improve imputation and prediction performance.
- Extensive experiments conducted on two real-world EHR databases show that our approach outperforms all baseline approaches in imputation and prediction tasks.

5.2 Method

In this section, we describe our proposed Contrastive Learning-based Imputation-Prediction Network for imputation and prediction tasks. We introduce the basic notations first. We then detail the network architecture. Finally, we present how to use the Contrastive Learning-based Imputation-Prediction Network for imputation and prediction tasks.

5.2.1 Basic Notations

We represent a multivariate time series X with up to N variables of length T as a set of observed triplets, i.e., $X = \{(f_i, v_i, t_i)\}_{i=1}^{N}$. An observed triplet is represented as a (f, v, t), where $f \in F$ is the variable/feature, $v \in \mathbb{R}^T$ is the observed value, and $t \in \mathbb{R}^T$ is the time. We incorporate a masking vector m_i to represent missing values in v_i as:

$$m_{i,t} = \begin{cases} 1, & if\ v_{i,t}\ is\ observed \\ 0, & otherwise \end{cases} \tag{5.1}$$

Let $\delta \in \mathbb{R}^{N\times T}$, $\delta^{(l)} \in \mathbb{R}^{N\times T}$, and $\delta^{(n)} \in \mathbb{R}^{N\times T}$ denote three time interval matrices. δ_t is the time interval between the current time t and the last time $t-1$. $\delta^{(l)}_{i,t}$ is the time interval between the current time t and the time where the i-th variable is observed the last time. $\delta^{(n)}_{i,t}$ is the time interval between the current time t and the time where the i-th variable is observed next time. $\delta^{(l)}_{i,t}$ and $\delta^{(n)}_{i,t}$ can be written as:

$$\delta^{(l)}_{i,t} = \begin{cases} \delta_{i,t}, & if\ m_{i,t-1} = 1 \\ \delta_{i,t} + \delta^{(l)}_{i,t-1}, & otherwise \end{cases} \tag{5.2}$$

$$\delta^{(n)}_{i,t} = \begin{cases} \delta_{i,t+1}, & if\ m_{i,t+1} = 1 \\ \delta_{i,t+1} + \delta^{(n)}_{i,t+1}, & otherwise \end{cases} \tag{5.3}$$

Let $v^{(l)}$ and $v^{(n)}$ denote two neighboring value matrices, the observed values of the last time and next time. $v^{(l)}$ and $v^{(n)}$ can be written as:

$$v^{(l)}_{i,t} = \begin{cases} v_{i,t-1}, & if\ m_{i,t-1} = 1 \\ v^{(l)}_{i,t-1}, & otherwise \end{cases} \tag{5.4}$$

$$v^{(n)}_{i,t} = \begin{cases} v_{i,t+1}, & if\ m_{i,t+1} = 1 \\ v^{(n)}_{i,t+1}, & otherwise \end{cases} \tag{5.5}$$

where $v^{(l)}_{i,t}$ and $v^{(n)}_{i,t}$ are the values of the i-th variable of $v^{(l)}_t$ and $v^{(n)}_t$.

Let $D = \{(X_p, y_p)\}^P_{p=1}$ denote the EHR dataset with up to P labeled samples. The p-th sample contains a multivariate time series X_p consisting of the physiological variables, and a binary label of in-hospital mortality $y_p \in \{0, 1\}$. Let $X_{base} \in \mathbb{R}^g$ denote the patient-specific characteristics (i.e., age, sex, ethnicity, admission diagnosis) with up to g dimension.

5.2.2 Network Architecture

The architecture of the proposed network is shown in Figure 5.2.

Personalised Patient Representation Learning

Given an input multivariate time series/a single patient data $X = \{(f_i, v_i, t_i)\}^N_{i=1}$, the embedding for the i-th triplet $e_i \in \mathbb{R}^d$ is generated by aggregating the feature embedding $e^{(f)}_i \in \mathbb{R}^d$, the value embedding $e^{(v)}_i \in \mathbb{R}^{d\times T}$, and the time interval embedding $e^{(t)}_i \in \mathbb{R}^{d\times T}$.

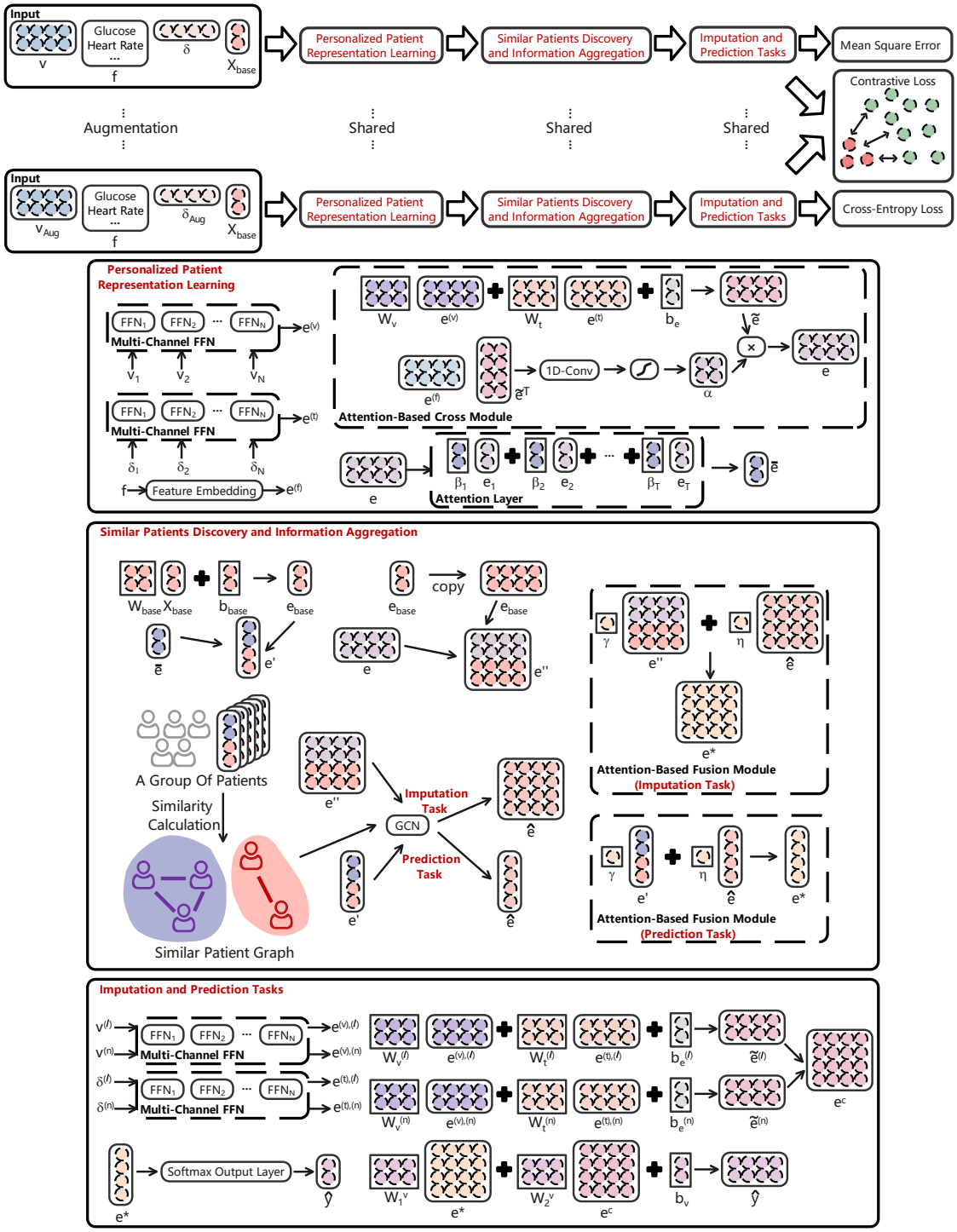

Figure 5.2: Schematic representation of the architecture and workflow of the proposed network.

The feature embedding is similar to the word embedding, which allows features with similar meanings to have a similar representation. Particularly, the value embedding and time interval embedding are obtained by separately implementing a multi-channel feed-forward neural network (FFN) as:

$$\begin{aligned} e_{i,1}^{(v)}, \cdots, e_{i,T}^{(v)} &= FFN_i^{(v)}(v_{i,1}, \cdots, v_{i,T}), \\ e_{i,1}^{(t)}, \cdots, e_{i,T}^{(t)} &= FFN_i^{(t)}(\delta_{i,1}, \cdots, \delta_{i,T}). \end{aligned} \tag{5.6}$$

Through the processes above, we are able to obtain $e^{(f)} \in \mathbb{R}^{Nd}$, $e^{(v)} \in \mathbb{R}^{Nd \times T}$, and $e^{(t)} \in \mathbb{R}^{Nd \times T}$, which are fed into the attention-based cross module to generate an overall representation. Note that $e^{(f)} \in \mathbb{R}^{Nd}$ is expanded into $e^{(f)} \in \mathbb{R}^{Nd \times T}$. Specifically, we design the attention-based cross module to generate a cross-attention matrix as:

$$\begin{aligned} \tilde{e} &= W_v \cdot e^{(v)} + W_t \cdot e^{(t)} + b_e, \\ E &= ScaledDot(e^{(f)}, \tilde{e}) = \frac{e^{(f)} \cdot \tilde{e}^{\top}}{\sqrt{d}}, \end{aligned} \tag{5.7}$$

where $E \in \mathbb{R}^{Nd \times Nd}$ is the cross-attention matrix that corresponds to the scaled-dot similarity. We then apply a 1D convolutional layer to the cross-attention matrix E as:

$$\alpha = Softmax(Conv(E)), \tag{5.8}$$

where $Conv$ is the 1D convolutional layer and α is the cross-attention score matrix. We integrate α and $\tilde{e}$ into a weighted representation e as:

$$e = \alpha \odot \tilde{e}. \tag{5.9}$$

Given a batch of patients, the embedding for them can be written as:

$$e = [e_1, e_2, \cdots, e_B] \in \mathbb{R}^{B \times Nd \times T}, \tag{5.10}$$

where B is the batch size. Since e still takes the form of sequence data, we design an attention layer to generate a series of attention weights $(\beta_1, \beta_2, \cdots, \beta_T)$ and reweight these weights to produce an overall feature representation as:

$$\begin{aligned} \beta &= Softmax(e \cdot W_e + b_e), \\ \bar{e} &= \sum_{t=1}^{T} \beta_t \odot e_t, \end{aligned} \tag{5.11}$$

where $\bar{e} \in \mathbb{R}^{B \times Nd}$ is the new generated patient representation.

Similar Patients Discovery and Information Aggregation

Before conducting patient similarity calculation, we encode $X_{base} \in \mathbb{R}^{g}$ as $e_{base} \in \mathbb{R}^{d_g}$ and concatenate e_{base} with $\bar{e}$ as:

$$\begin{aligned} e_{base} &= W_{base} \cdot X_{base} + b_{base}, \\ e' &= Concate(\bar{e}, e_{base}), \end{aligned} \tag{5.12}$$

where Concate is the concatenation operation.

For the batch of patient representations, the pairwise similarities that correspond to any two patient representations can be calculated as:

$$\Lambda = sim(e', e') = \frac{e' \cdot e'}{(Nd + d_g)^2}, \tag{5.13}$$

where $sim(\cdot)$ is the measure of cosine similarity and $\Lambda \in \mathbb{R}^{B \times B}$ is the patient similarity matrix.

Moreover, we incorporate a learnable threshold φ into the patient similarity calculation to filter out similarities below the threshold. The similarity matrix can be rewritten as:

$$\Lambda' = \begin{cases} \Lambda, & if\ \Lambda > \varphi \\ 0, & otherwise \end{cases} \tag{5.14}$$

We take into account the batch of patients' representations as a graph to aggregate the information from similar patients, where the similarity matrix Λ' is the graph adjacency matrix. We apply graph convolutional layers to enhance the representation learning as:

$$\begin{aligned} \hat{e} &= [\hat{e}_1, \hat{e}_2, \cdots, \hat{e}_B]^\top = GCN(e', \Lambda') \\ &= ReLU(\Lambda' ReLU(\Lambda' \cdot e' W_1^e) \cdot W_2^e), \end{aligned} \tag{5.15}$$

where $\hat{e}$ is the aggregated auxiliary information from similar patients. A note of caution is due here since we ignore the bias term. We replace e' in Eq. (15) with e'' for the imputation task. By doing so, the output of graph convolutional layers can take the form of sequence data. Particularly, e'' is obtained by concatenating e and e_{base}, where $e_{base} \in \mathbb{R}^{d_g}$ is expanded into $e_{base} \in \mathbb{R}^{d_g \times T}$.

Through the processes above, we are able to generate e'/e'' and $\hat{e}$ representations for the batch of patients. The e'/e'' refers to the patient themselves. For an incomplete patient p (i.e., the patient data has many missing values), we generate the missing value representa-

tions with $\hat{e}$. For a complete patient, we augment e'/e'' with $\hat{e}$ to enhance the representation learning.

We design an attention-based fusion module to refine both e'/e'' (the two representations used in prediction and imputation tasks) and $\hat{e}$. Since imputation and prediction tasks involve the same process of modeling, we take the prediction task as an example. The two weights $\gamma \in \mathbb{R}^B$ and $\eta \in \mathbb{R}^B$ are incorporated to determine the importance of e' and $\hat{e}$, obtained by implementing fully connected layers as:

$$\begin{aligned} \gamma &= Sigmoid(e' \cdot W_\gamma + b_\gamma), \\ \eta &= Sigmoid(\hat{e} \cdot W_\eta + b_\eta). \end{aligned} \tag{5.16}$$

A note of caution is due here since we keep the sum of γ and η must be 1, i.e., $\gamma + \eta = 1$. We achieve this constraint by combining $\gamma = \frac{\gamma}{\gamma+\eta}$ and $\eta = 1 - \gamma$. The final representation e^* is obtained by calculating $\gamma \cdot e' + \eta \cdot \hat{e}$.

Contrastive Learning

We integrate contrastive learning into the proposed network architecture to perform imputation and prediction tasks. For the prediction task, we augment the standard cross-entropy loss with the supervised contrastive loss [289]. We treat the patient representations with the same label as the positive pairs and the patient representations with different labels as the negative pairs. For the imputation task, we augment the standard mean squared error loss with the unsupervised contrastive loss [298]. We treat a single patient representation and its augmented representations as positive pairs and the other patient representations within a batch and their augmented representations as negative pairs. The formula can be written as:

$$\begin{aligned} \mathcal{L}_{SC} &= -\sum_{i=1}^{B} \frac{1}{B_{y_i}} log \frac{\sum_{j=1}^{B} \mathbb{1}_{[y_i=y_j]} exp(sim(e_i^*, e_j^*)/\tau)}{\sum_{k=1}^{B} \mathbb{1}_{[k\neq i]} exp(sim(e_i^*, e_k^*)/\tau)}, \\ \mathcal{L}_{UC} &= -log \frac{exp(sim(e_i^*, e_j^*)/\tau)}{\sum_{k=1}^{2B} \mathbb{1}_{[k\neq i]} exp(sim(e_i^*, e_k^*)/\tau)}, \end{aligned} \tag{5.17}$$

where B represents the batch size; $\mathbb{1}_{[\cdot]}$ represents an indicator function; $sim(\cdot)$ represents the cosine similarity measure; τ represents a hyper-parameter that is used to control the strength of penalties on negative pairs; B_{y_i} is the number of samples with the same label in each batch.

Imputation and Prediction Tasks

For the prediction task, we feed e^* into a softmax output layer to obtain the predicted $\hat{y}$ as:

$$\hat{y} = Softmax(W_y \cdot e^* + b_y). \tag{5.18}$$

The objective loss is the summation of cross-entropy loss and the supervised contrastive loss with a scaling parameter λ to control the contribution of each loss as:

$$\begin{aligned} \mathcal{L}_{CE} &= -\frac{1}{P}\sum_{p=1}^{P}(y_p^\top \cdot log(\hat{y}_p) + (1-y_p)^\top \cdot log(1-\hat{y}_p)), \\ \mathcal{L} &= \lambda \cdot \mathcal{L}_{CE} + (1-\lambda) \cdot \mathcal{L}_{SC}. \end{aligned} \tag{5.19}$$

For the imputation task, we take the neighboring observed values (of each patient) as inputs to incorporate patient-specific contextual information. The process of embedding used by $v^{(l)}$ and $v^{(n)}$ can be written as:

$$\begin{aligned} e_i^{(v),(l)} = FFN_i^{(v),(l)}(v_i^{(l)}), e_i^{(t),(l)} &= FFN_i^{(t),(l)}(\delta_i^{(l)}), \\ e_i^{(v),(n)} = FFN_i^{(v),(n)}(v_i^{(n)}), e_i^{(t),(n)} &= FFN_i^{(t),(n)}(\delta_i^{(n)}), \\ \tilde{e}^{(l)} &= W_v^{(l)} \cdot e^{(v),(l)} + W_t^{(l)} \cdot e^{(t),(l)} + b_e^{(l)}, \\ \tilde{e}^{(n)} &= W_n^{(v)} \cdot e^{(v),(n)} + W_t^{(n)} \cdot e^{(t),(n)} + b_e^{(n)}, \\ e^c &= Concate(\tilde{e}^{(l)}, \tilde{e}^{(n)}), \end{aligned} \tag{5.20}$$

where $\tilde{e}^{(l)}$ and $\tilde{e}^{(n)}$ are the representations of $v^{(l)}$ and $v^{(n)}$ after embedding. The embedding matrix e^c is obtained by concatenating $\tilde{e}^{(l)}$ and $\tilde{e}^{(n)}$.

Given the final representation e^* and the embedding matrix e^c, we use a fully connected layer to impute missing values as:

$$\hat{v} = e^* \cdot W_1^v + e^c \cdot W_2^v + b_v. \tag{5.21}$$

The objective loss is the summation of the mean square error and the unsupervised contrastive loss with a scaling parameter λ to control the contribution of each loss as:

$$\begin{aligned} \mathcal{L}_{MSE} &= \frac{1}{P}\sum_{p=1}^{P}(m_p \odot v_p - m_p \odot \hat{v}_p)^2, \\ \mathcal{L} &= \lambda \cdot \mathcal{L}_{MSE} + (1-\lambda) \cdot \mathcal{L}_{UC}. \end{aligned} \tag{5.22}$$

5.3 Experiments

5.3.1 Experimental Setup

Datasets and Tasks

We validate our approach[1] on the MIMIC-III[2] Database and eICU[3] Database. The 21,139 and 38,056 samples/patients were taken from the two databases. Detailed information on the two databases can be found in the literature [38] and [280]. Table 5.1 presents the summary statistics for the MIMIC-III and eICU features used.

For the MIMIC-III database, we evaluate multivariate clinical time series imputation and in-hospital mortality accuracy based on the data from the first 24/48 hours after ICU admission. Similarly, for the eICU database, we evaluate multivariate clinical time series imputation and in-hospital mortality accuracy based on the data from the first 24/48 hours after eICU admission.

Baseline Approaches

We compare our approach with GRU-D [39], BRITS [40], GRUI-GAN [43], E^{2}GAN [44], E^{2}GAN-RF [50], STING [46], MTSIT [58], and MIAM [51] (see related work section). We feed the output of GRUI-GAN, E^{2}GAN, E^{2}GAN-RF, STING, and MTSIT into GRU to estimate in-hospital mortality risk probabilities. Moreover, the regression component used in BRITS is integrated into GRU-D and MIAM to obtain imputation accuracy.

Implementation Details

We implement all approaches with PyTorch 1.11.0 and conduct experiments on A40 GPU from NVIDIA with 48GB of memory. We randomly use 70%, 15%, and 15% of the dataset as training, validation, and testing sets. We train the proposed approach using an Adam optimizer [283] with a learning rate of 0.0023 and a mini-batch size of 256. For Personalised patient representation learning, the dimension size d is 3. For similar patients discovery and information aggregation, the initial value of φ is 0.56, and the dimension size of W_1^e and W_2^e are 34 and 55. For contrastive learning, the value of τ is 0.07. The dropout method is applied to the final Softmax output layer for the prediction task, and the dropout rate is 0.1. For the imputation task, the dimension size of $W_v^{(l)}$, $W_t^{(l)}$, $W_v^{(n)}$, and

[1]The implementation code is available at https://github.com/liulab1356/CL-ImpPreNet
[2]https://mimic.physionet.org
[3]https://eicu-crd.mit.edu/

Table 5.1: MIMIC-III and eICU features used for multivariate clinical time series imputation and in-hospital mortality prediction 48 hours after ICU admission.

MIMIC-III Feature	Data Type	Missingness (%)
Capillary refill rate	categorical	99.78
Diastolic blood pressure	continuous	30.90
Fraction inspired oxygen	continuous	94.33
Glasgow coma scale eye	categorical	82.84
Glasgow coma scale motor	categorical	81.74
Glasgow coma scale total	categorical	89.16
Glasgow coma scale verbal	categorical	81.72
Glucose	continuous	83.04
Heart Rate	continuous	27.43
Height	continuous	99.77
Mean blood pressure	continuous	31.38
Oxygen saturation	continuous	26.86
Respiratory rate	continuous	26.80
Systolic blood pressure	continuous	30.87
Temperature	continuous	78.06
Weight	continuous	97.89
pH	continuous	91.56
Age	continuous	0.00
Admission diagnosis	categorical	0.00
Ethnicity	categorical	0.00
Gender	categorical	0.00
eICU Feature	**Type**	**Missingness (%)**
Diastolic blood pressure	continuous	33.80
Fraction inspired oxygen	continuous	98.14
Glasgow coma scale eye	categorical	83.42
Glasgow coma scale motor	categorical	83.43
Glasgow coma scale total	categorical	81.70
Glasgow coma scale verbal	categorical	83.54
Glucose	continuous	83.89
Heart Rate	continuous	27.45
Height	continuous	99.19
Mean arterial pressure	continuous	96.53
Oxygen saturation	continuous	38.12
Respiratory rate	continuous	33.11
Systolic blood pressure	continuous	33.80
Temperature	continuous	76.35
Weight	continuous	98.65
pH	continuous	97.91
Age	continuous	0.00
Admission diagnosis	categorical	0.00
Ethnicity	categorical	0.00
Gender	categorical	0.00

$W_t^{(n)}$ are 28. For a fair comparison, the hyper-parameter of the proposed model (i.e., τ) was fine-tuned by a grid-searching strategy.

The performance of contrastive learning heavily relies on data augmentation. We augment the observed value v with random time shifts and reversion. For example, given the observed value $v = [v_1, v_2, \cdots, v_T]$, we are able to obtain $v_{shift} = [v_{1+n}, v_{2+n}, \cdots, v_{T+n}]$ and $v_{reverse} = [v_T, v_{T-1}, \cdots, v_1]$ from random time shift and reversion, and n is the number of data points to shift.

Evaluation Metrics

We use the mean absolute error (MAE) and the mean relative error (MRE) between predicted and real-valued values as the evaluation metrics for imputation performance. We use the area under the receiver operating characteristic curve (AUROC) and the area under the precision-recall curve (AUPRC) as the evaluation metrics for prediction performance. We report the mean and standard deviation of the evaluation metrics after repeating all the approaches ten times.

5.3.2 Performance Analysis

Table 5.2 presents the experimental results of all approaches on imputation and prediction tasks from MIMIC-III and eICU databases. The larger the scores of AUROC and AUPRC, the better the predictive performance of the method. The lower the values of MAE and MRE, the better the imputation performance of the method. Values in the parentheses denote standard deviations. Together these results suggest that our approach achieves the best performance in both imputation and prediction tasks. For example, for the multivariate clinical time series imputation of MIMIC-III (24 hours after ICU admission), the MAE and MRE of Ours are 0.3563 and 8.16%, smaller than 0.3988 and 38.44% achieved by the best baseline (i.e., MTSIT). For the in-hospital mortality prediction of MIMIC-III (24 hours after ICU admission), the AUROC and AUPRC of Ours are 0.8533 and 0.4752, larger than 0.8461 and 0.4513 achieved by the best baseline (i.e., GRU-D).

Similarly, for the multivariate clinical time series imputation of eICU (24 hours after ICU admission), the MAE and MRE of Ours are 0.5365 and 7.02%, smaller than 1.1726 and 15.35% achieved by the best baseline (i.e., MIAM). For the in-hospital mortality prediction of eICU (24 hours after ICU admission), the AUROC and AUPRC of Ours are 0.7626 and 0.3388, larger than 0.7455 and 0.3178 achieved by the best baseline (i.e., GRU-D).

As Table 5.2 shows, the RNN-based approach (i.e., GRU-D and BRITS) outperforms the GAN-based approach (i.e., GRUI-GAN, E^{2}GAN, E^{2}GAN-RF, and STING) in the im-

putation task. From the prediction results of the MIMIC-III database, we can see that the transformer-based approaches (i.e., MTSIT and MIAM) resulted in lower values of AUROC and AUPRC. From the prediction results of the eICU database, no significant difference between the transformer-based approach and other approaches was evident.

5.3.3 Ablation Analysis

We conduct an ablation study to examine the effectiveness of different components of our method in imputation and prediction tasks. We present two variants of our approach as follows:

- Ours$_{\alpha}$: A variation of our approach that does not perform graph analysis-based patient stratification modeling.
- Ours$_{\beta}$: A variation of our approach in which we omit the contrastive learning component.

All implementations of Ours$_{\alpha}$ and Ours$_{\beta}$ can be found in the aforementioned Github repository.

The results of the ablation study can be compared in Table 5.2. We find that Ours outperforms its variants Ours$_{\alpha}$ and Ours$_{\beta}$. Overall, these results confirm the effectiveness of the network construction with enhanced imputation and prediction performance.

Table 5.2: Performance of our approaches with other baselines on multivariate clinical time series imputation and in-hospital mortality prediction.

MIMIC-III/24 hours after ICU admission	Multivariate clinical time series imputation		In-hospital mortality prediction	
Metrics	MAE	MRE	AUROC	AUPRC
GRU-D [39]	1.3134(0.0509)	87.33%(0.0341)	0.8461(0.0051)	0.4513(0.0124)
BRITS [40]	1.3211(0.0923)	87.92%(0.0611)	0.8432(0.0040)	0.4193(0.0144)
GRUI-GAN [43]	1.6083(0.0043)	107.20%(0.0029)	0.8324(0.0077)	0.4209(0.0280)
E^2GAN [44]	1.5885(0.0045)	105.86%(0.0032)	0.8377(0.0083)	0.4295(0.0137)
E^2GAN-RF [50]	1.4362(0.0031)	101.09%(0.0027)	0.8430(0.0065)	0.4328(0.0101)
STING [46]	1.5018(0.0082)	102.53%(0.0047)	0.8344(0.0126)	0.4431(0.0158)
MTSIT [58]	0.3988(0.0671)	38.44%(0.0647)	0.8029(0.0117)	0.4150(0.0165)
MIAM [51]	1.1391(0.0001)	75.65%(0.0001)	0.8140(0.0044)	0.4162(0.0079)
Ours	**0.3563(0.0375)**	**8.16%(0.0086)**	**0.8533(0.0119)**	**0.4752(0.0223)**
Ours$_\alpha$	0.3833(0.0389)	8.78%(0.0089)	0.8398(0.0064)	0.4555(0.0139)
Ours$_\beta$	0.4125(0.0319)	8.95%(0.0077)	0.8417(0.0059)	0.4489(0.0182)
eICU/24 hours after eICU admission	Multivariate clinical time series imputation		In-hospital mortality prediction	
Metrics	MAE	MRE	AUROC	AUPRC
GRU-D [39]	3.9791(0.2008)	52.11%(0.0262)	0.7455(0.0107)	0.3178(0.0190)
BRITS [40]	3.6879(0.3782)	48.30%(0.0726)	0.7139(0.0101)	0.2511(0.0111)
GRUI-GAN [43]	9.1031(0.0130)	119.29%(0.0016)	0.7298(0.0094)	0.3013(0.0141)
E^2GAN [44]	7.5746(0.0141)	99.20%(0.0018)	0.7317(0.0155)	0.2973(0.0253)
E^2GAN-RF [50]	6.7108(0.0127)	90.38%(0.0015)	0.7402(0.0131)	0.3045(0.0227)
STING [46]	7.1447(0.0651)	93.56%(0.0083)	0.7197(0.0154)	0.2873(0.0182)
MTSIT [58]	1.6192(0.1064)	21.20%(0.0139)	0.7215(0.0071)	0.2992(0.0115)
MIAM [51]	1.1726(0.3103)	15.35%(0.0406)	0.7262(0.0179)	0.2659(0.0148)
Ours	**0.5365(0.0612)**	**7.02%(0.0079)**	**0.7626(0.0117)**	**0.3388(0.0211)**
Ours$_\alpha$	0.6792(0.0716)	8.89%(0.0093)	0.7501(0.0143)	0.3325(0.0151)
Ours$_\beta$	0.5923(0.0514)	7.75%(0.0067)	0.7533(0.0104)	0.3303(0.0175)
MIMIC-III/48 hours after ICU admission	Multivariate clinical time series imputation		In-hospital mortality prediction	
Metrics	MAE	MRE	AUROC	AUPRC
GRU-D [39]	1.4535(0.0806)	86.47%(0.0482)	0.8746(0.0026)	0.5143(0.0077)
BRITS [40]	1.3802(0.1295)	82.21%(0.0768)	0.8564(0.0040)	0.4445(0.0189)
GRUI-GAN [43]	1.7523(0.0030)	104.50%(0.0018)	0.8681(0.0077)	0.5123(0.0166)
E^2GAN [44]	1.7436(0.0036)	103.98%(0.0022)	0.8705(0.0043)	0.5091(0.0120)
E^2GAN-RF [50]	1.6122(0.0027)	102.34%(0.0017)	0.8736(0.0031)	0.5186(0.0095)
STING [46]	1.6831(0.0068)	100.46%(0.0035)	0.8668(0.0123)	0.5232(0.0236)
MTSIT [58]	0.4503(0.0465)	30.42%(0.0314)	0.8171(0.0114)	0.4308(0.0189)
MIAM [51]	1.3158(0.0003)	78.20%(0.0002)	0.8327(0.0024)	0.4460(0.0061)
Ours	**0.4396(0.0588)**	**6.23%(0.0073)**	**0.8831(0.0149)**	**0.5328(0.0347)**
Ours$_\alpha$	0.7096(0.0532)	8.85%(0.0066)	0.8671(0.0093)	0.5161(0.0151)
Ours$_\beta$	0.5786(0.0429)	7.47%(0.0056)	0.8709(0.0073)	0.5114(0.0176)
eICU/48 hours after eICU admission	Multivariate clinical time series imputation		In-hospital mortality prediction	
Metrics	MAE	MRE	AUROC	AUPRC
GRU-D [39]	5.8071(0.2132)	44.53%(0.0164)	0.7767(0.0141)	0.3210(0.0182)
BRITS [40]	5.5546(0.5497)	42.59%(0.0421)	0.7285(0.0114)	0.2510(0.0097)
GRUI-GAN [43]	14.0750(0.0301)	107.96%(0.0021)	0.7531(0.0167)	0.2897(0.0201)
E^2GAN [44]	12.9694(0.0195)	99.47%(0.0015)	0.7605(0.0063)	0.3014(0.0137)
E^2GAN-RF [50]	11.8138(0.0161)	91.52%(0.0011)	0.7763(0.0057)	0.3101(0.0125)
STING [46]	12.0962(0.0806)	92.79%(0.0062)	0.7453(0.0182)	0.2805(0.0190)
MTSIT [58]	2.8150(0.2105)	21.58%(0.0161)	0.7418(0.0091)	0.3078(0.0120)
MIAM [51]	2.1146(0.4012)	16.23%(0.0414)	0.7574(0.0127)	0.2776(0.0105)
Ours	**0.9412(0.0930)**	**7.21%(0.0071)**	**0.7907(0.0123)**	**0.3417(0.0217)**
Ours$_\alpha$	1.1099(0.1064)	8.51%(0.0081)	0.7732(0.0100)	0.3311(0.0265)
Ours$_\beta$	0.9930(0.0817)	7.61%(0.0062)	0.7790(0.0117)	0.3335(0.0178)

Chapter 6

Contrastive Graph Similarity Networks

The following manuscript has been incorporated into this chapter:
Yuxi Liu, Zhenhao Zhang, Shaowen Qin, Flora D Salim, Jiang Bian and Antonio Jimeno Yepes. Fine-grained Patient Similarity Measuring using Contrastive Graph Similarity Networks. In *2024 IEEE 12th International Conference on Healthcare Informatics (ICHI)*, Under Review

6.1 Introduction

With the adoption of digital health systems, large amounts of Electronic Health Records (EHRs) are available, but the major problem is how to translate the existing information into useful knowledge and decision-support tools to guide clinical practice. Data mining is the process of analyzing large datasets to discover patterns and meaningful insights. Specifically, it involves a process of analysis in which data scientists employ machine learning and statistical methods to build predictive models for decision support.

Various machine learning methods have been developed for clinical and research applications using EHR data, such as clinical risk prediction [291, 299–302], phenotype analysis [148, 303–306], disease prediction and progression [307–311].

However, EHR data are often incomplete. For example, vital sign measurements are gathered from multiple sources at various time points during an ICU stay [53, 54]. A patient's vital signs are measured as indicators of their health status. When a patient's condition deteriorates, or new symptoms appear, the corresponding vital signs are more

frequently measured and recorded [312]. Accordingly, this results in the creation of multiple incomplete patient data, where missing values need to be filled with plausible values through imputation. As typical machine learning methods are not developed for EHR data with such irregularity, they are not directly applicable to EHR data analysis. A general practice is to impute missing values in EHR data to address this issue. Existing studies construct deep imputation methods by combining recurrent neural networks or generative adversarial networks with attention mechanisms and model the variable associations in EHR data to impute missing values [40, 46, 47, 57, 313].

Although existing studies have demonstrated promising performance, the similarity between samples/patients has not been fully taken into consideration in the imputation task. Patient similarity analysis aims to classify patients into medically relevant groups likely to have similar health outcomes or temporal experiences [314]. In real clinical reasoning scenarios, it is a general practice to utilize data from similar patients to generate hypotheses and make decisions (i.e., precision medicine [314–316]). Accordingly, we argue that the missing values in each patient's data could be handled by aggregating the information from similar patients. Since no set criteria are available, a new challenge we face is how to calculate the similarity between patients in a large EHR dataset.

In this paper, we propose a novel Contrastive Graph Similarity Network to simultaneously perform imputation and prediction with EHR data. The core idea of our method is borrowed from Graph Contrastive Learning (GCL). The GCL is a self-supervised graph learning technique that exploits the structure of graphs with data augmentation techniques for contrastive learning to create different views [317–319]. To this end, we construct multiple patient-patient similarity graphs using vital signs and demographics as well as diagnosis and procedure codes as relational information and then aggregate the information from similar patients to generate rich patient representations (Figure 6.1a). To further put similar patients closer and push dissimilar patients apart, we construct positive and negative sample pairs in contrastive learning using the generated patient representations. We arbitrarily select a node as an anchor (Figure 6.1b). Positive samples for an anchor are defined as (i) the same nodes as the anchor in different views, (ii) the nodes connected to the anchor within the same view, and (iii) the nodes connected to the anchor from different views. The remaining samples are negative. For the imputation task, we construct sample pairs by pairing a positive (or negative) sample with the anchor. For the prediction task, we repeat the above sample pairing process with the constraint that the pairs must be formed between samples with the same binary label. This process is repeated for all nodes. We design a composite loss for imputation and prediction, where two hyper-parameters are used to control the ratio between imputation loss and prediction loss to minimize the overall loss.

The main contributions of this paper are listed as follows:

- We propose a novel Contrastive Graph Similarity Network to simultaneously perform imputation and prediction with EHR data.
- To the best of our knowledge, this is the first attempt that uses a tailored Contrastive Graph Similarity Network for similarity calculation among patients in ICUs.
- We evaluate our method against competing baselines on real-world EHR databases, and the results demonstrate the effectiveness and superiority of our method in ICU mortality risk prediction and clinical time series imputation.

6.2 Method

In this section, we describe our proposed Contrastive Graph Similarity Network for imputation and prediction tasks. We first introduce the basic notations. We then detail the network architecture. Finally, we present how to use the Contrastive Graph Similarity Network for imputation and prediction tasks.

6.2.1 Basic Notations

Let $D = \{(X_i, Y_i)\}_{i=1}^{|\mathcal{P}|}$ represent the EHR dataset with up to $|\mathcal{P}|$ samples/patients. X contains multivariate time series data $X^{(t)}$ and static data $X^{(s)}$. Particularly, $X^{(t)}$ contains a series of vital sign measurements (e.g., oxygen saturation, fraction inspired oxygen, and temperature), and $X^{(s)}$ contains demographics (i.e., age, sex, and ethnicity) as well as diagnosis and procedure codes (i.e., unique medical codes). Y represents target labels for the benchmarks/tasks. We represent the elements in $X^{(t)}$ using $(x_1^{(t)}, \cdots, x_T^{(t)}) \in \mathbb{R}^{d \times T}$, where T is the number of time steps and d is the number of vital signs. We represent missing values in $X^{(t)}$ using a mask matrix $M \in \mathbb{R}^{d \times T}$.

6.2.2 Network Architecture

Figure 6.1 displays the overview of the proposed Contrastive Graph Similarity Network.

Learning patient representation with learnable graph augmentation

Let $\mathcal{G} = \{\mathcal{P}, \mathcal{E}\}$ represent a patient-patient similarity graph. $\mathcal{P}$ is a set of nodes, where each node represents a patient. $\mathcal{E}$ is a set of edges that connects patients. The adjacency matrix

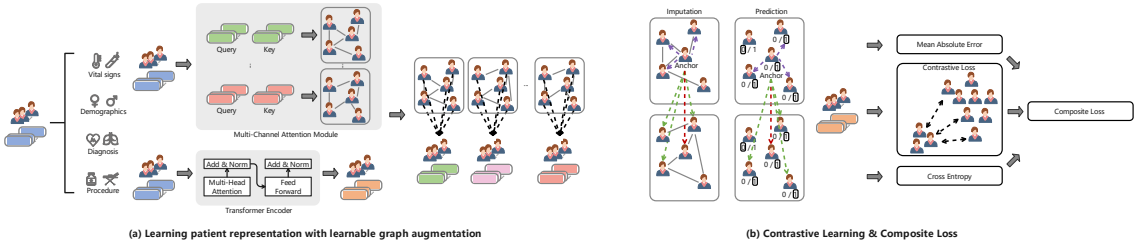

Figure 6.1: Schematic representation of the architecture and workflow of the proposed network.

A represents the causal connections between patients. For example, $A_{i,j}$ is 1 if patients i and j are connected, and 0 otherwise.

Now, we feed X into a multi-channel attention module to generate up to N adjacency matrices. The multi-channel attention module has up to N channels, where each channel has an attention layer. Specifically, the following steps were taken: (i) we apply adaptive average pooling to X (i.e., at the horizontal dimension) to generate a new feature representation $\bar{X}$. (ii) we apply a linear transformation to $\bar{X}$ to generate query and key vectors. (iii) we take the dot product between query and key vectors and then apply the Softmax function to obtain a set of attention weight matrices. We formulate the above process as:

$$\begin{aligned} Q_i = W_i^Q \cdot \bar{X}, K_i = W_i^K \cdot \bar{X}, \\ A_i = Softmax(Q_i \cdot K_i^\top), i \in \{1, 2, \cdots, N\}, \end{aligned} \tag{6.1}$$

where all W are learnable weight matrices. Q and K are query and key vectors. $\{A_i\}_{i=1}^N$ is a set of attention weight matrices, where each attention weight matrix corresponds to a patient-patient similarity graph. We introduce a learnable threshold φ to those matrices to generate binary matrices as adjacency matrices to consider the information from similar patients.

Next, we feed X into the Transformer encoder [195] to generate a rich feature representation. To be specific, we apply a linear transformation to X to generate Q', K', and V' and then take the dot product between Q' and K' and apply the Softmax function to obtain attention scores on V'. We formulate the above process as:

$$\begin{aligned} Q' = W_{Q'} \cdot X, K' = W_{K'} \cdot X, V' = W_{V'} \cdot X, \\ \alpha_j = Softmax(Q'_j \cdot K'^\top_j), \\ head_j = \alpha_j \cdot V'_j, j \in \{1, 2, \cdots, L\}, \\ Z = (head_1 || \cdots || head_L) \cdot W_O, \end{aligned} \tag{6.2}$$

where all W are learnable weight matrices. $||$ is the concatenation operator. L is the number of heads. Subsequently, we feed Z into a normalised layer with the residual connection

[198], followed by a feed-forward network (FFN). In the same vein, we feed the output of FFN into a normalization layer with the residual connection again to generate a rich feature representation $\tilde{Z}$ as:

$$\begin{aligned} Z' &= norm(X + Z), \\ Z_{FFN} &= ReLU(Z' \cdot W_1) \cdot W_2, \\ \tilde{Z} &= norm(Z' + Z_{FFN}), \end{aligned} \tag{6.3}$$

where all W are learnable parameters, $norm(\cdot)$ is the batch normalization, and $ReLU(\cdot)$ is the rectified linear activation function.

Last, we combine $\tilde{Z}$ with $\{A_i\}_{i=1}^{N}$ to aggregate the information from similar patients as:

$$\hat{Z}^{(i)} = A_i \cdot \tilde{Z}, i \in \{1, 2, \cdots, N\}. \tag{6.4}$$

Subsequently, we apply adaptive average pooling to $\hat{Z}$ to generate the patient representation $\bar{Z}$.

Contrastive Learning

To further put similar patients closer and push dissimilar patients apart, we construct positive and negative sample pairs in contrastive learning using the generated patient representations $\{\bar{Z}^{(i)}\}_{i=1}^{N}$. We arbitrarily select a node as an anchor (as shown in Figure 6.1b) and treat each graph as a view. Positive samples for an anchor are defined as (i) the same nodes as the anchor in different views, (ii) the nodes connected to the anchor within the same view, and (iii) the nodes connected to the anchor from different views. The remaining samples are negative. For the imputation task, we construct sample pairs by pairing a positive (or negative) sample with the anchor. For the prediction task, we repeat the above sample pairing process with the constraint that the pairs must be formed between samples with the same binary label. This process is repeated for all nodes.

Contrastive Imputation Loss We select $\bar{Z}_p^{(1)}$ as an anchor. The contrastive loss be-

tween $\bar{Z}^{(1)}$ and $\bar{Z}^{(2)}$ can be calculated as:

$$\begin{aligned}\mathcal{L}_{CL}^{(Imp)}(\bar{Z}_p^{(1)}) = \\ -\frac{1}{2|\mathcal{N}(p)|+1} log \frac{exp(sim(\bar{Z}_p^{(1)}, \bar{Z}_p^{(2)})/\tau)}{exp(sim(\bar{Z}_j^{(1)}, \bar{Z}_p^{(2)})/\tau)} \\ \frac{+\sum_{k\in\mathcal{N}(p)}(exp(sim(\bar{Z}_p^{(1)}, \bar{Z}_k^{(1)})/\tau)}{+\sum_{k\neq p}(exp(sim(\bar{Z}_p^{(1)}, \bar{Z}_k^{(1)})/\tau)} \\ \frac{+exp(sim(\bar{Z}_p^{(1)}, \bar{Z}_p^{(2)}/\tau)))}{+exp(sim(\bar{Z}_p^{(1)}, \bar{Z}_p^{(2)}/\tau)))},\end{aligned} \tag{6.5}$$

where $\mathcal{N}(p)$ is a set of neighbors in $\bar{Z}_p^{(1)}$. $sim(\cdot)$ is the dot product operation. τ is a temperature parameter that controls the strength of penalties on negative pairs. Since the two views are symmetric, we select $\bar{Z}_p^{(2)}$ as an anchor again. The contrastive loss $\mathcal{L}_{CL}^{Imp}(\bar{Z}_p^{(2)})$ can be calculated in the way as Eq. (5). Accordingly, the contrastive loss between $\bar{Z}^{(1)}$ and $\bar{Z}^{(2)}$ can be calculated as:

$$\begin{aligned}\mathcal{L}_{CL}^{(Imp)}(\bar{Z}^{(1)}, \bar{Z}^{(2)}) = \\ \frac{1}{2|\mathcal{P}|}\sum_{p=1}^{|\mathcal{P}|}(\mathcal{L}_{CL}^{(Imp)}(\bar{Z}_p^{(1)}) + \mathcal{L}_{CL}^{(Imp)}(\bar{Z}_p^{(2)})).\end{aligned} \tag{6.6}$$

Through the processes above, we have been able to calculate the contrastive loss between $\bar{Z}^{(1)}$ and $\bar{Z}^{(2)}$. Since $\{\bar{Z}^{(i)}\}_{i=1}^{N}$ is set of patient representations, we arbitrarily select one and then calculate the contrastive loss between it and the others. Accordingly, we utilize the total contrastive loss as the contrastive imputation loss:

$$\mathcal{L}_{CL}^{(Imp)} = \frac{1}{N}\sum_{n=1,n\neq l}^{N}\mathcal{L}_{CL}^{(Imp)}(\bar{Z}^{(l)}, \bar{Z}^{(n)}). \tag{6.7}$$

Contrastive Prediction Loss We select $\bar{Z}_p^{(1)}$ as an anchor. The contrastive loss between $\bar{Z}^{(1)}$ and $\bar{Z}^{(2)}$ can be calculated as:

$$\begin{aligned}\mathcal{L}_{CL}^{(Pre)}(\bar{Z}_p^{(1)}) = \\ -\frac{1}{2N_{Y_p}+1} log \frac{exp(sim(\bar{Z}_p^{(1)}, \bar{Z}_p^{(2)})/\tau)}{exp(sim(\bar{Z}_p^{(1)}, \bar{Z}_p^{(2)})/\tau)}, \\ \frac{+\sum_{k=1}^{|P|}1_{[Y_p=Y_k]}(exp(sim(\bar{Z}_p^{(1)}, \bar{Z}_k^{(1)})/\tau)}{+\sum_{k=1}^{|P|}1_{[k\neq p]}(exp(sim(\bar{Z}_p^{(1)}, \bar{Z}_k^{(1)})/\tau)}, \\ \frac{+exp(sim(\bar{Z}_p^{(1)}, \bar{Z}_k^{(2)}/\tau)))}{+exp(sim(\bar{Z}_p^{(1)}, \bar{Z}_k^{(2)}/\tau)))},\end{aligned} \tag{6.8}$$

where Y_p is the label of node p (i.e., patient p). N_{Y_p} is the number of nodes with the same label as node p. $sim(\cdot)$ is the cosine similarity. $1_{[\cdot]}$ is an indicator function. Similar to the imputation task, we utilize the total contrastive loss as the contrastive prediction loss:

$$\begin{aligned} &\mathcal{L}_{CL}^{(Pre)}(\bar{Z}^{(1)}, \bar{Z}^{(2)}) = \\ &\frac{1}{2|\mathcal{P}|}\sum_{p=1}^{|\mathcal{P}|}(\mathcal{L}_{CL}^{(Pre)}(\bar{Z}_p^{(1)}) + \mathcal{L}_{CL}^{(Pre)}(\bar{Z}_p^{(2)})), \\ &\mathcal{L}_{CL}^{(Pre)} = \frac{1}{N}\sum_{n=1,n\neq l}^{N}\mathcal{L}_{CL}^{(Pre)}(\bar{Z}^{(l)}, \bar{Z}^{(n)}). \end{aligned} \tag{6.9}$$

Composite Loss

Since the imputation task can be viewed as a regression task, we employ the mean absolute error (MAE) as the objective function between the original X and predicted $\hat{X}$ of each patient as:

$$\begin{aligned} \hat{X} &= W_r \cdot \tilde{Z} + b_r, \\ \mathcal{L}_{MAE}^{(Imp)} &= \frac{1}{|\mathcal{P}|}\sum_{p=1}^{|\mathcal{P}|}|X_p \odot M_p - \hat{X}_p \odot M_p|, \end{aligned} \tag{6.10}$$

where W_r is a learnable parameter and b_r is a bias. Accordingly, the imputation loss $\mathcal{L}^{(Imp)}$ is the summation of the MAE and the contrastive imputation loss as:

$$\mathcal{L}^{(Imp)} = \lambda_{MAE} \cdot \mathcal{L}_{MAE}^{(Imp)} + (1 - \lambda_{MAE}) \cdot \mathcal{L}_{CL}^{(Imp)}, \tag{6.11}$$

where λ_{MAE} is a scaling parameter used to make the trade-off between the MAE and the contrastive imputation loss.

In order to perform prediction tasks, we employ the cross entropy (CE) as the objective function between the target label Y and predicted label $\hat{Y}$ of each patient as:

$$\begin{aligned} \hat{Y} &= Softmax(Z^* \cdot W_c + b_c), \\ \mathcal{L}_{CE}^{(Pre)} &= -\frac{1}{|\mathcal{P}|}\sum_{p=1}^{|\mathcal{P}|}(Y_p^\top \cdot log(\hat{Y}_p) \\ &\quad +(1 - Y_p)^\top \cdot log(1 - \hat{Y}_p)), \end{aligned} \tag{6.12}$$

where Z^* is the pooled representation obtained by applying adaptive average pooling to $\tilde{Z}$ before feeding into the Softmax output layer. W_c is a learnable parameter and b_c is a bias. Accordingly, the prediction loss $\mathcal{L}^{(Pre)}$ is the summation of the CE and the contrastive

prediction loss as:

$$\mathcal{L}^{(Pre)} = \lambda_{CE} \cdot \mathcal{L}_{CE}^{(Pre)} + (1 - \lambda_{CE}) \cdot \mathcal{L}_{CL}^{(Pre)}, \tag{6.13}$$

where λ_{CE} is a scaling parameter used to make the trade-off between the CE and the contrastive prediction loss.

We design a composite loss for imputation and prediction, where two scaling parameters are used to make the trade-off between imputation loss and prediction loss as:

$$\mathcal{L} = \lambda^{(Pre)} \cdot \mathcal{L}^{(Pre)} + \lambda^{(Imp)} \cdot \mathcal{L}^{(Imp)}. \tag{6.14}$$

where $\lambda^{(Pre)}$ and $\lambda^{(Imp)}$ are scaling parameters.

6.3 Experiments

6.3.1 Experimental Setup

Datasets and Baselines

We conduct extensive experiments on the MIMIC-III[1] and eICU[2] databases with ICU mortality risk prediction and clinical time series imputation. The details of the two databases are described in the literature [38, 280]. We extract vital sign measurements (e.g., oxygen saturation, fraction inspired oxygen, and temperature), demographics (i.e., age, sex, and ethnicity), as well as diagnosis and procedure codes (i.e., unique medical codes) from the two databases. We provide mortality risk assessments for ICU patients using the data from the first 24 and 48 hours after admission [53, 54]. For the MIMIC-III database, the sample size is 17,886, where the Positive (likely to die)/Negative (unlikely to die) ratio is 1:6.59. For the eICU database, the sample size is 36,670, where the Positive/Negative ratio is 1:7.49.

We evaluate the performance of our method against deep imputation methods [39, 40, 46,47,57,58,313]. For the imputation task, we replace the fully connected layer in GRU-D with a regression layer. For the prediction task, we replace the linear decoder in MTSIT with a Softmax output layer; we feed the output of Conditional GAN, STING, MBGAN, and SA-EDGAN into GRU to estimate ICU mortality risk probabilities.

The source code of our method and data extraction are released at the Github repository[3].

[1] https://mimic.physionet.org
[2] https://eicu-crd.mit.edu/
[3] https://github.com/LZlab01/CGSNet

Table 6.1 presents the summary statistics for the vital signs and demographics used.

Implementations & Evaluations

The two EHR datasets are extracted from the MIMIC-III and eICU databases. Each EHR dataset is randomly split into the training, validation, and testing set in a 0.7:0.15:0.15 ratio. The Adam optimizer [283] (with an initial learning rate of 0.001 and a batch size of 256) is applied to train the proposed method. For the MIMIC-III dataset, the number of channels in the multi-channel attention module is 2, and the dimension size of W_i^Q and W_i^K are 17; the number of heads in the Transformer encoder is 4, the number of layers is 1, and the dimension size of W_Q, W_K and W_V is 24; the temperature parameter τ is 0.6; the scaling parameters λ_{MAE} and λ_{CE} are 0.8; the scaling parameters $\lambda^{(Imp)}$ and $\lambda^{(Pre)}$ are 0.5 and 0.7, respectively. For the ICU dataset, the number of channels in the multi-channel attention module is 4, and the dimension size of W_i^Q and W_i^K are 16; the number of heads in the Transformer encoder is 2, the number of layers is 1, and the dimension size of W_Q, W_K and W_V is 26; the temperature parameter τ is 0.5; the scaling parameters λ_{MAE} and λ_{CE} are 0.95; the scaling parameters $\lambda^{(Imp)}$ and $\lambda^{(Pre)}$ are 0.9 and 0.7, respectively. For the ICU patient deterioration prediction, the dropout method is also employed for the Softmax output layer, and the dropout rates of MIMIC-III and eICU are 0.1 and 0.2, respectively. For a fair comparison, the hyper-parameter of the proposed model (i.e., τ) was fine-tuned by a grid-searching strategy. All experiments run with PyTorch 1.11.0 on an NVIDIA RTX A5000 GPU. We adopt the mean absolute error (MAE), the mean relative error (MRE), the receiver operating characteristic curve (AUROC), and the area under the precision-recall curve (AUPRC) to evaluate imputation and prediction performance. We repeat each experiment ten times and report the average performance.

6.3.2 Performance Analysis

We report the result of ICU mortality risk prediction and clinical time series imputation in Tables 6.2 and 6.3. From the table below, we can see that our method reports more AUROC and AUPRC scores and lower MAE and MRE scores than the baselines. For instance, from the data in Table 6.2, our method reaches the highest AUROC and AUPRC scores with 0.8967 and 0.5863 and the lowest MAE and MRE scores with 1.9012 and 0.2248. Similarly, from the data in Table 6.3, our method reaches the highest AUROC and AUPRC scores with 0.8889 and 0.5637 and the lowest MAE and MRE scores with 1.1728 and 0.2548.

Besides, there was no significant prediction performance difference between RNN-based methods (i.e., GRU-D, Brits) and GAN-based methods (i.e., Conditional GAN,

Table 6.1: MIMIC-III and eICU vital signs and demographics used for clinical time series imputation and ICU mortality risk prediction 48 hours after admission.

MIMIC-III Feature	Data Type	Missingness (%)
Capillary refill rate	categorical	99.78
Diastolic blood pressure	continuous	30.90
Fraction inspired oxygen	continuous	94.33
Glasgow coma scale eye	categorical	82.84
Glasgow coma scale motor	categorical	81.74
Glasgow coma scale total	categorical	89.16
Glasgow coma scale verbal	categorical	81.72
Glucose	continuous	83.04
Heart Rate	continuous	27.43
Height	continuous	99.77
Mean blood pressure	continuous	31.38
Oxygen saturation	continuous	26.86
Respiratory rate	continuous	26.80
Systolic blood pressure	continuous	30.87
Temperature	continuous	78.06
Weight	continuous	97.89
pH	continuous	91.56
Age	continuous	0.00
Ethnicity	categorical	0.00
Gender	categorical	0.00
eICU Feature	**Type**	**Missingness (%)**
Diastolic blood pressure	continuous	33.80
Fraction inspired oxygen	continuous	98.14
Glasgow coma scale eye	categorical	83.42
Glasgow coma scale motor	categorical	83.43
Glasgow coma scale total	categorical	81.70
Glasgow coma scale verbal	categorical	83.54
Glucose	continuous	83.89
Heart Rate	continuous	27.45
Height	continuous	99.19
Mean arterial pressure	continuous	96.53
Oxygen saturation	continuous	38.12
Respiratory rate	continuous	33.11
Systolic blood pressure	continuous	33.80
Temperature	continuous	76.35
Weight	continuous	98.65
pH	continuous	97.91
Age	continuous	0.00
Ethnicity	categorical	0.00
Gender	categorical	0.00

STING, MBGAN, SA-EDGAN). What is interesting about the data is in the two tables that STING achieves poor imputation performance compared to other baseline methods. For example, for clinical time series imputation (i.e., 48 hours after eICU admission), STING achieves the highest MAE and MRE scores with 7.2456 and 0.6355, which is an increase over the best baseline Brits by 4.6432 and 0.4369.

The most striking result from the data comparison is that MTSIT resulted in the lowest MAE and MRE scores among the baselines (except for 48 hours after eICU admission). For example, for clinical time series imputation (i.e., 24 hours after eICU admission), MTSIT achieves the lowest MAE and MRE scores with 1.7775 and 0.2307. These results suggest that MTSIT is the most competitive baseline method.

6.3.3 Visualization Analysis

Now, we make a comparison between the proposed method and its variants that change parts of the contrastive learning module. Doing such a comparison can allow us to understand how the contrastive learning module makes decisions. The results obtained from the visualization analysis of the contrastive learning module can be compared in Figure 6.2. The experimental data were gathered 48 hours after admission (MIMIC-III). As shown in Figure 6.2, positive represents the patient who died and negative represents the patient who did not die. From Figure 6.2a to Figure 6.2d, we omit the contrastive learning component; $\lambda^{(Imp)}$ is greater than $\lambda^{(Pre)}$; $\lambda^{(Pre)}$ is greater than $\lambda^{(Imp)}$; $\lambda^{(Imp)}$ is equal to $\lambda^{(Pre)}$. The two scaling parameters $\lambda^{(Imp)}$ and $\lambda^{(Pre)}$ are used to make the trade-off between imputation loss and prediction loss. Looking at Figure 6.2a, the instances from the positive and negative classes are scattered. Compared with the instances in Figure 6.2a, the instances in Figure 6.2b are clustered together, and each cluster has instances from both positive and negative classes. From the data in Figure 6.2c, we can see that the instances from the positive and negative classes are clustered together towards two distinguishable clusters. From the data in Figure 6.2d, we can see that the instances from the positive and negative classes in each cluster towards two distinguishable sub-clusters. These results are in agreement with our expectations. It is also important to bear in mind the possible bias in these responses, as those overlapping samples with different labels are illustrated in Figure 6.2c. This result seems to be consistent with other research, which found that samples with different labels in graphs can also be connected (also known as heterophily in graphs [320–322]). Further work is needed to fully consider homophily and heterophily in graphs on the development of the patient-patient similarity graph.

Table 6.2: Performance Comparison (48 hours after admission).

MIMIC-III	ICU Mortality Risk Prediction		Clinical Time Series Imputation	
Metrics	AUROC	AUPRC	MAE	MRE
GRU-D [39]	0.8820(0.0097)	0.5568(0.0189)	2.6698(0.3745)	0.3158(0.0443)
Brits [40]	0.8805(0.0017)	0.5524(0.0133)	2.3192(0.2674)	0.2743(0.0316)
Conditional GAN [57]	0.8762(0.0074)	0.5458(0.0214)	3.2563(0.1382)	0.3852(0.0281)
STING [46]	0.8824(0.0043)	0.5349(0.0285)	6.3998(0.1918)	0.7570(0.0227)
MBGAN [47]	0.8705(0.0045)	0.5611(0.0199)	2.7752(0.0764)	0.3346(0.0104)
SA-EDGAN [313]	0.8785(0.0063)	0.5585(0.0178)	2.4288(0.0489)	0.2873(0.0057)
MTSIT [58]	0.8735(0.0032)	0.5352(0.0171)	2.0814(0.1265)	0.2461(0.0149)
Our	**0.8967(0.0038)**	**0.5863(0.0098)**	**1.9012(0.1517)**	**0.2248(0.0179)**
eICU	ICU Mortality Risk Prediction		Clinical Time Series Imputation	
Metrics	AUROC	AUPRC	MAE	MRE
GRU-D [39]	0.8276(0.0054)	0.4275(0.0144)	2.6695(0.2664)	0.2038(0.0132)
Brits [40]	0.8163(0.0122)	0.4248(0.0127)	2.6024(0.1399)	0.1986(0.0107)
Conditional GAN [57]	0.8219(0.0093)	0.4033(0.0217)	3.8645(0.2085)	0.2949(0.0179)
STING [46]	0.8270(0.0070)	0.4084(0.0185)	7.2456(0.3626)	0.6355(0.0213)
MBGAN [47]	0.8235(0.0095)	0.4025(0.0169)	3.1270(0.1762)	0.2386(0.0158)
SA-EDGAN [313]	0.8267(0.0072)	0.4190(0.0165)	2.9355(0.2238)	0.2241(0.0171
MTSIT [58]	0.8044(0.0025)	0.3921(0.0446)	2.6124(0.1918)	0.1994(0.0145)
Our	**0.8420(0.0036)**	**0.4457(0.0231)**	**2.4924(0.1880)**	**0.1902(0.0143)**

Table 6.3: Performance Comparison (24 hours after admission).

MIMIC-III	ICU Mortality Risk Prediction		Clinical Time Series Imputation	
Metrics	AUROC	AUPRC	MAE	MRE
GRU-D [39]	0.8821(0.0087)	0.5526(0.0282)	1.6365(0.2514)	0.3612(0.0546)
Brits [40]	0.8816(0.0015)	0.5543(0.0097)	1.7512(0.5390)	0.3805(0.1171)
Conditional GAN [57]	0.8757(0.0061)	0.5374(0.0198)	2.1139(0.1017)	0.4594(0.0216)
STING [46]	0.8784(0.0069)	0.5177(0.0220)	3.9430(0.3774)	0.8568(0.0820)
MBGAN [47]	0.8778(0.0073)	0.5539(0.0287)	1.8159(0.1025)	0.3947(0.0192)
SA-EDGAN [313]	0.8819(0.0098)	0.5448(0.0514)	1.7396(0.1081)	0.3781(0.0234)
MTSIT [58]	0.8775(0.0035)	0.5425(0.0143)	1.6224(0.3226)	0.3526(0.0701)
Our	**0.8889(0.0036)**	**0.5637(0.0073)**	**1.1728(0.1210)**	**0.2548(0.0262)**
eICU	ICU Mortality Risk Prediction		Clinical Time Series Imputation	
Metrics	AUROC	AUPRC	MAE	MRE
GRU-D [39]	0.8166(0.0198)	0.4618(0.0266)	2.2304(0.3413)	0.2894(0.0449)
Brits [40]	0.8194(0.0077)	0.4601(0.0179)	2.0333(0.5573)	0.2638(0.0723)
Conditional GAN [57]	0.8168(0.0083)	0.4371(0.0175)	2.8693(0.1328)	0.3724(0.0143)
STING [46]	0.8226(0.0061)	0.4355(0.0121)	5.9237(0.5266)	0.7680(0.0685)
MBGAN [47]	0.8193(0.0053)	0.4554(0.0198)	2.2592(0.0981)	0.2933(0.0107)
SA-EDGAN [313]	0.8214(0.0049)	0.4466(0.0256)	2.0350(0.1014)	0.2642(0.0132)
MTSIT [58]	0.8138(0.0031)	0.4322(0.0529)	1.7775(0.0435)	0.2307(0.0055)
Our	**0.8385(0.0035)**	**0.4746(0.0137)**	**1.6671(0.0814)**	**0.2162(0.0105)**

Figure 6.2: The t-SNE plot of the feature representation $\tilde{Z}$. (a) w/o contrastive learning module; (b) $\lambda^{(Imp)}$ is greater than $\lambda^{(Pre)}$; (c) $\lambda^{(Pre)}$ is greater than $\lambda^{(Imp)}$; (d) $\lambda^{(Imp)}$ is equal to $\lambda^{(Pre)}$.

Chapter 7

Multi-Graph Neural Networks

The following publication has been incorporated into this chapter:
Yuxi Liu, Zhenhao Zhang, Shaowen Qin, Flora D Salim. A Multi-Graph Fusion Framework for Patient Representation Learning. In *2024 IEEE 12th International Conference on Healthcare Informatics (ICHI)*, Under Review

7.1 Introduction

Electronic health records (EHRs) represent a digital version of a patient's medical history generated from clinical routine care. A popular focus in deep learning using EHR data is patient representation learning, which mainly learns a dense mathematical representation of individual patients using deep neural networks [1, 5, 323–331]. Based on the outcome of patient representation learning, patient similarity can be computed and, further, information from similar patients can be aggregated for clinical risk prediction (e.g., disease risk prediction and mortality risk prediction) [324–326].

Previous studies have focused on patient representation learning from a single graph view. However, in real clinical reasoning scenarios, it is a common practice to use information from different patient-level features [332] (e.g., demographics, vital signs, diagnoses, procedures, and lab tests) to represent a patient health context, which naturally results in a rich representation with multiple graphs generated from the patient-level features.

Motivated by the recent developments in graph representation learning [333–338], a strategy to build a patient-patient similarity graph is adopted to exploit multivariate clini-

cal time series, demographics, and diagnoses as relational information form multi-graphs. To this end, we propose a novel Multi-Graph Fusion Framework for patient representation learning (Figure 7.1), which learns multiple graph structures from input patient-level features and, in turn, generates an optimal graph structure that incorporates the characteristics of these graphs with attention mechanisms. Based on the foundation established by multi-graph representation learning, we aggregate the information from similar patients to offer a rich representation of the patient, which allows extraction of patient health context for missing data imputation and clinical risk prediction.

The main contributions of this paper are listed as follows:

- We propose a novel Multi-Graph Fusion Framework for patient representation learning. To the best of our knowledge, this is the first attempt to consider multivariate clinical time series, demographics, and diagnoses as patient health context in multi-graph representation learning.

- We conduct extensive experiments on two real-world EHR databases with multivariate clinical time series imputation and in-hospital mortality risk prediction tasks, and the results demonstrate the effectiveness and superiority of our method in comparison to all baselines.

7.2 Method

In this section, we describe our proposed Multi-Graph Fusion Framework for patient representation learning. We first introduce the basic notations. We then detail the network architecture. Finally, we present how to use the Multi-Graph Fusion Framework for imputation and prediction tasks.

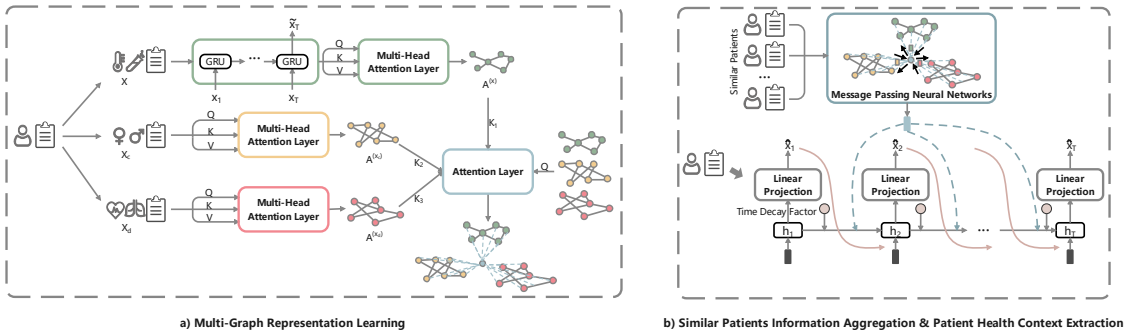

Figure 7.1: Schematic representation of the architecture and workflow of the proposed network.

7.2.1 Basic Notations

EHRs contain longitudinal data that are generated from clinical routine care. Each patient has a multivariate clinical time series with up to K physiological variables of length T, i.e., $X = \{x_1, x_2, \cdots, x_T\} \in \mathbb{R}^{K \times T}$, where $x_t = \{x_t^1, x_t^2, \cdots, x_t^K\} \in \mathbb{R}^K$ is the t-th record and x_t^k is the value of the k-th variable of x_t.

We represent the presence of the data using a masking matrix $M = \{m_1, m_2, \cdots, m_T\} \in \mathbb{R}^{K \times T}$ as:

$$m_t^k = \begin{cases} 1, & if\ x_t^k\ is\ presence \\ 0, & if\ x_t^k\ is\ absence \end{cases}. \tag{7.1}$$

Let $\tau \in \mathbb{R}$ represent the time stamp when the t-th record is obtained. We represent the time interval between two consecutive records using a time interval vector $\Delta = \{\delta_1, \delta_2, \cdots, \delta_T\} \in \mathbb{R}^T$. The elements of Δ are represented as:

$$\delta_t = \begin{cases} \tau_{t+1} - \tau_t, & if\ t < T \\ 0, & if\ t = T \end{cases}. \tag{7.2}$$

EHR data also contains demographics and diagnoses. Let $X_c \in \mathbb{R}^C$ represent the demographics (i.e., age, sex, ethnicity) with up to C dimensions. Let $X_d \in \{0, 1\}^D$ represent the previous ICD-9 diagnosis codes with up to D dimensions.

7.2.2 Network Architecture

The architecture of the proposed network is shown in Figure 7.1.

Multi-Graph Representation Learning

Graph representation learning has attracted considerable attention, both scholarly and popular. Graph representation learning aims to generate graph representation vectors that effectively capture the structure and features of high-dimensional sparse graph data. The Graph Convolutional Network (GCN) is currently the most popular method for learning representations of graph-structured data [339–342]. The GCN is a neural network architecture that exploits the graph structure and aggregates node information from the neighborhoods in a convolutional fashion [298]. Recently, GCN-based models have made great progress in various real-world applications such as text classification [343], social recommendation [344], chemical-gene interaction [345], drug response prediction [236], and medical diagnosis and analysis [238].

More recent attention has focused on the provision of multi-graph representation learning. Multi-graph representation learning aims to generate a consistent representation by exploiting the complementary information of multiple graphs [346]. Representative multi-graph representation learning applications include entity linkage identification [347], drug discovery [348], semi-supervised classification problem [346], urban region profiling [349], and gene-disease association prediction [350].

Motivated by these successful applications, we propose a strategy to build a patient-patient similarity graph (see Figure 7.1a), which exploits multivariate clinical time series (i.e., X), demographics (i.e., X_c), and ICD-9 diagnosis codes (i.e., X_d) as relational information. The intuitions behind our strategy can be explained as seeing the need to specify x, x_c, x_d in the form of a multi-graph.

We define the target graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ and $\mathcal{E}$ are the set of nodes and edges. The mathematical representation of the graph $\mathcal{G}$ is the adjacency matrix $A \in \mathbb{R}^{P \times P}$ (to be detailed later), where P is the number of nodes that correspond to the number of patients. A_{ij} is equal to 1 if and only if nodes i and j are connected.

We feed X, X_c, and X_d into a multi-head attention layer [195] to generate three adjacency matrices. Towards this, we take X as an example for a detailed description. First, we feed X into Gated Recurrent Units [147] to generate a series of hidden representations, i.e., $\tilde{x}_1, \tilde{x}_2, \cdots, \tilde{x}_T = GRU(x_1, x_2, \cdots, x_T)$. Then, we take $\tilde{x}_T$ (i.e., the hidden representation at time T) as input for the multi-head attention layer (Note that we remove the subscript T in the following steps). The multi-head attention layer has up to N heads. We take $head_i$ as an example. For $head_i$, $\tilde{x}$ is linearly transformed to generate the query, key, and value vectors as:

$$\begin{aligned} Q_i &= \tilde{x} \cdot W_i^Q, \\ K_i &= \tilde{x} \cdot W_i^K, \end{aligned} \tag{7.3}$$

where $Q_i \in \mathbb{R}^{d_k}$ and $K_i \in \mathbb{R}^{d_k}$ are query and key vectors. W_i^Q and W_i^K are learnable projection matrices. The $head_i$ can be rewritten as:

$$head_i = Softmax(\frac{Q_i \cdot K_i^\top}{\sqrt{d_k}}). \tag{7.4}$$

We concatenate all heads and use a linear transformation to project it back to the original space as:

$$A^{(X)} = (head_1 || head || \cdots || head_N) \cdot W^O, \tag{7.5}$$

where W^O is a learnable projection matrix. $||$ is the concatenation operator. $A^{(X)}$ is an

attention-weight matrix with patient relations. In the same vein, X_c and X_d are used as inputs for the multi-head attention layer in order to generate $A^{(X_c)}$ and $A^{(X_d)}$.

Next we design an attention layer to adaptively fuse $A^{(X)}$, $A^{(X_c)}$, and $A^{(X_d)}$ into an overall representation A as:

$$\begin{aligned}
\tilde{A} &= A^{(X)} || A^{(X_c)} || A^{(X_d)}, \\
Q_A &= \tilde{A} \cdot W_A^Q, \\
K_{A,1} &= A^{(X)} \cdot W_{A,1}^K, \\
K_{A,2} &= A^{(X_c)} \cdot W_{A,2}^K, \\
K_{A,3} &= A^{(X_d)} \cdot W_{A,3}^K, \\
\alpha_1, \alpha_2, \alpha_3 &= Softmax(\frac{Q_A \cdot K_{A,1}}{\sqrt{d_K}}, \frac{Q_A \cdot K_{A,2}}{\sqrt{d_K}}, \frac{Q_A \cdot K_{A,3}}{\sqrt{d_K}}), \\
A &= \alpha_1 \odot A^{(X)} + \alpha_2 \odot A^{(X_c)} + \alpha_2 \odot A^{(X_d)},
\end{aligned} \tag{7.6}$$

where all W are learnable projection matrices. $\odot$ and $||$ are the Hadamard product and the concatenation operator, respectively.

Similar Patients Information Aggregation & Patient Health Context Extraction

We combine Message Passing Neural Networks and Gated Recurrent Units to aggregate the information from similar patients and extract patient health context from rich patient representations (see Figure 7.1b). Given $x_{i,t}^{(l-1)} \in \mathbb{R}^K$ that represents the features of node i (i.e., the i-th patient) in layer (l-1), Message Passing Neural Networks can be written as:

$$x_{i,t}^{(l)} = \Theta_{\mathcal{G}}(x_{i,t}^{(l-1)}) = \gamma^{(l)}(x_{i,t}^{(l-1)}, \bigoplus_{j \in \mathcal{N}(i)} \varphi^{(l)}(x_{i,t}^{(l-1)}, x_{j,t}^{(l-1)})), \tag{7.7}$$

where $\mathcal{N}(i)$ is the set of neighbors of the i-th node in $\mathcal{G}$. $\gamma^{(l)}$ and $\varphi^{(l)}$ are update and message functions. $\bigoplus$ is an aggregation function.

Based on the foundation established by Message Passing Neural Networks, we implement Gated Recurrent Units as:

$$\begin{aligned}
r_{i,t} &= \sigma(\Theta_{\mathcal{G}}([x'_{i,t} || m_{i,t} || \hat{h}_{i,t-1}], W_r)), \\
u_{i,t} &= \sigma(\Theta_{\mathcal{G}}([x'_{i,t} || m_{i,t} || \hat{h}_{i,t-1}], W_u)), \\
\tilde{h}_{i,t} &= tanh(\Theta_{\mathcal{G}}([x'_{i,t} || m_{i,t} || r_{i,t} \odot \hat{h}_{i,t-1}], W_h)), \\
h_{i,t} &= u_{i,t} \odot h_{i,t-1} + (1 - u_{i,t}) \odot \tilde{h}_{i,t},
\end{aligned} \tag{7.8}$$

where $r_{i,t}$ and $u_{i,t}$ are reset and update gates. $x'_{i,t}$ is a refined vector of the i-th node at time

t, i.e., a combination of actual value (i.e., x_t) and predicted value (i.e., $\hat{x}_t$) as:

$$\begin{aligned} \hat{x}_t &= W_x \cdot h_{t-1} + b_x, \\ x'_t &= m_t \odot x_t + (1 - m_t) \odot \hat{x}_t. \end{aligned} \tag{7.9}$$

Moreover, $\hat{h}_{i,t}$ is the refined hidden representation of the i-th node at time t, obtained by decaying the hidden state h_{t-1} [39] as:

$$\begin{aligned} \eta_t &= exp\{-max(0, W_\eta \cdot \delta_t + b_\eta)\}, \\ \hat{h}_{t-1} &= \eta_t \odot h_{t-1}, \end{aligned} \tag{7.10}$$

where W_η and b_η are learnable parameters. η is a time decay factor.

Missing Data Imputation and Clinical Risk Prediction

For the imputation task, we feed the hidden state representation h into a fully connected layer to generate the predicted $\hat{X}$ as:

$$\hat{X} = W_y \cdot h + b_y. \tag{7.11}$$

Subsequently, the objective loss is the mean absolute error as:

$$\mathcal{L}_{MAE} = \frac{1}{P} \sum_{i=1}^{P} |M_i \odot X_i - M_i \odot \hat{X}_i|. \tag{7.12}$$

For the prediction task, we feed the last hidden state representation h_T into a Softmax output layer to generate the predicted $\hat{y}$ as:

$$\hat{y} = Softmax(W_y \cdot h_T + b_y). \tag{7.13}$$

Subsequently, the objective loss is the cross-entropy loss as:

$$\mathcal{L}_{CE} = -\frac{1}{P} \sum_{i=1}^{P} (y_i^\top \cdot log(\hat{y}_i) + (1 - y_i)^\top \cdot log(1 - \hat{y}_i)). \tag{7.14}$$

7.3 Experiments

7.3.1 Experimental Setup

Datasets, Tasks, and Evaluation Metrics

Our experiments are carried out on the MIMIC-III[1] Database [38] and eICU[2] Database [280]. We extract 21,105 and 27,390 patients/samples from the MIMIC-III and eICU databases, where the Positive (likely to die)/Negative (unlikely to die) ratio is 1:6.56 and 1:6.15, respectively. We extract multivariate clinical time series (i.e., a series of physiological variables), demographics (i.e., age, sex, ethnicity), and ICD-9 diagnosis codes (i.e., unique ICD codes) from the two databases. Tables A.1 and A.2 show multivariate clinical time series data on the two databases. The multivariate clinical time series imputation is a regression task with the mean absolute error (MAE) and the mean relative error (MRE) between the original and predicted target multivariate clinical time series being the primary evaluation metrics. In-hospital mortality prediction is defined as predicting the mortality risk of patients based on the data from the first 48 hours after ICU/eICU admission. Physiologic decline/decompensation prediction is defined as predicting the mortality risk of patients based on the data from the first 24 hours after ICU/eICU admission. These are binary classification tasks, with the area under the receiver operating characteristic curve (AUROC) and the area under the precision-recall curve (AUPRC) being the primary evaluation metrics. Taken together, we carry out in-hospital mortality prediction using the data from the first 24 hours and 48 hours after ICU/eICU admission.

Baselines

We compare our approach with Recurrent Neural Networks GRU-D [39] and BRITS [40], Generative Adversarial Networks GRUI-GAN [43] and E^{2}GAN [44], Attention-based Neural Networks MIAM [51] and MTSIT [58], Graph Neural Networks GCT [351], AGRN [246], and GRIN [244]. For the imputation task, we replace the fully connected layer in GRU-D with a regression layer [40]. Similarly, we replace the Softmax output layer in GCT with a regression layer [40]. For the prediction task, we replace the linear decoder in MTSIT with a Softmax output layer; for AGRN and GRIN, we take the last hidden state in their GRUs as input for a Softmax output layer. We also provide a variant of our approach (i.e., Our$_\alpha$), which specifies input patient-level features in the form of a single graph.

[1] https://mimic.physionet.org
[2] https://eicu-crd.mit.edu/

Implementations

All approaches are implemented with PyTorch 1.10.0 on an Nvidia A40 GPU. The twp EHR datasets are randomly divided into three parts, using 70% for training, 15% for validation, and 15% for testing. The Adam optimizer [283] (i.e., with a learning rate of 0.0016 and a batch size of 256) is employed to train the proposed approach. For multivariate clinical time series, the dimension size of W^Q and W^K is 19. For demographics, the dimension size of W^Q and W^K is 2. For ICD-9 diagnosis codes, the dimension size of W^Q and W^K is 17. The dimension size of W_A^Q, $W_{A,1}^K$, $W_{A,2}^K$, $W_{A,3}^K$ is 139. The dropout method is applied to the Softmax output layer (i.e., with a dropout rate of 0.3) for the prediction task. All approaches are repeated ten times, and the average values with standard deviation for each evaluation metric are reported. The source code of our method and data extraction are released at the Github repository[3].

7.3.2 Performance Analysis

Tables 7.1 and 7.2 compare the performance of our approach and baselines on the MIMIC-III and eICU databases. The larger the scores of AUROC and AUPRC, the better the predictive performance of the method. The lower the values of MAE and MRE, the better the imputation performance of the method. Values in the parentheses denote standard deviations. Together, these results indicate that our approach consistently achieves the best performance in terms of MAE, MRE, AUROC, and AUPRC scores. For example, as can be seen from Table 7.1 (below), our method achieves the highest AUROC and AUPRC with 0.8379 and 0.4623 and the lowest MAE and MRE with 0.7962 and 0.1825. Similarly, from Table 7.2, our method achieves the highest AUROC and AUPRC with 0.8622 and 0.5103 and the lowest MAE and MRE with 1.3039 and 0.1627. The superior performance of our approach than the variant of our approach Our$_\alpha$ confirms the effectiveness of constructing multi-graph representation learning in improving the performance.

For multivariate clinical time series imputation (i.e., 24 hours after ICU/eICU admission), the best baseline is given by the MIAM. For in-hospital mortality prediction (i.e., 24 hours after ICU/eICU admission), the best baseline is given by the GRIN. For multivariate clinical time series imputation (i.e., 48 hours after ICU admission), the best baseline is given by the GRIN. For multivariate clinical time series imputation (i.e., 48 hours after eICU admission), the best baseline is given by the GCT. For in-hospital mortality prediction (i.e., 48 hours after ICU admission), the best baseline is given by the GRU-D. For in-hospital mortality prediction (i.e., 48 hours after eICU admission), the best baseline is given by the GRIN. These results suggest that GRIN is the most competitive baseline

[3]https://github.com/Le1328/Model

method.

Furthermore, the prediction performance of all approaches improved significantly as the prediction window from the first 24 hours to the first 48 hours after ICU/eICU admission. For example, GRU-D achieves an AUROC of 0.8521 and an AUPRC of 0.4720 based on the data from the first 48 hours after ICU admission, which is a significant improvement over an AUROC of 0.8259 and an AUPRC of 0.4329 based on the data from the first 24 hours after ICU admission. Similarly, GRU-D achieves an AUROC of 0.8027 and an AUPRC of 0.3949 based on the data from the first 48 hours after eICU admission, which is a significant improvement over an AUROC of 0.7682 and an AUPRC of 0.3409 based on the data from the first 24 hours after eICU admission.

Table 7.1: Imputation and prediction results on the two databases (24 hours after ICU/eICU admission).

MIMIC-III/24 hours after ICU admission	Multivariate Clinical Time Series Imputation		In-hospital Mortality Prediction	
Metrics	MAE	MRE	AUROC	AUPRC
GRU-D [39]	1.8554(0.1774)	0.4254(0.0406)	0.8259(0.0113)	0.4329(0.0242)
BRITS [40]	1.7887(0.2928)	0.4101(0.0671)	0.8066(0.0010)	0.4298(0.0044)
GRUI-GAN [43]	1.1536(0.0024)	1.1425(0.0022)	0.7848(0.0699)	0.3666(0.0855)
E^2GAN [44]	1.1386(0.0046)	1.1277(0.0042)	0.8061(0.0269)	0.3763(0.0210)
MIAM [51]	0.8109(0.0063)	0.1859(0.0014)	0.8121(0.0041)	0.4068(0.0065)
MTSIT [58]	0.8917(0.0672)	0.2044(0.0154)	0.7899(0.0274)	0.3684(0.0341)
GCT [351]	1.1355(0.3137)	0.2603(0.0719)	0.8232(0.0077)	0.4305(0.0175)
AGRN [246]	0.8339(0.0331)	0.1912(0.0076)	0.6822(0.0017)	0.2257(0.0032)
GRIN [244]	0.8216(0.0259)	0.1883(0.0059)	0.8306(0.0053)	0.4224(0.0179)
Our_{α}	1.0045(0.2895)	0.2303(0.0663)	0.8008(0.0338)	0.3512(0.0591)
Our	**0.7962(0.1171)**	**0.1825(0.0255)**	**0.8379(0.0104)**	**0.4623(0.0403)**
eICU/24 hours after eICU admission	Multivariate Clinical Time Series Imputation		In-hospital Mortality Prediction	
Metrics	MAE	MRE	AUROC	AUPRC
GRU-D [39]	1.2804(0.1452)	0.1666(0.0189)	0.7682(0.0184)	0.3409(0.0222)
BRITS [40]	1.6927(0.3193)	0.2205(0.0414)	0.7624(0.0036)	0.3279(0.0070)
GRUI-GAN [43]	4.0085(0.0194)	1.2726(0.0031)	0.7511(0.0393)	0.3385(0.0352)
E^2GAN [44]	3.8742(0.0163)	1.2311(0.0047)	0.7567(0.0235)	0.3286(0.0297)
MIAM [51]	1.0534(0.0961)	0.1377(0.0125)	0.7389(0.0096)	0.3155(0.0106)
MTSIT [58]	1.5627(0.1366)	0.2032(0.0179)	0.7540(0.0043)	0.3473(0.0159)
GCT [351]	1.0902(0.1293)	0.1425(0.0168)	0.7419(0.0114)	0.3208(0.0123)
AGRN [246]	1.2368(0.0553)	0.1610(0.0073)	0.6885(0.0025)	0.2587(0.0029)
GRIN [244]	1.1298(0.0725)	0.1466(0.0094)	0.7687(0.0064)	0.3390(0.0224)
Our_{α}	1.7136(0.8482)	0.2234(0.1107)	0.7635(0.0077)	0.3552(0.0199)
Our	**1.0498(0.0137)**	**0.1374(0.0053)**	**0.7845(0.0094)**	**0.3761(0.0237)**

Table 7.2: Imputation and prediction results on the two databases (48 hours after ICU/eICU admission).

MIMIC-III/48 hours after ICU admission	Multivariate Clinical Time Series Imputation		In-hospital Mortality Prediction	
Metrics	MAE	MRE	AUROC	AUPRC
GRU-D [39]	2.1163(0.1304)	0.2640(0.0162)	0.8521(0.0087)	0.4720(0.0224)
BRITS [40]	1.9234(0.4344)	0.2400(0.0542)	0.8105(0.0016)	0.4333(0.0043)
GRUI-GAN [43]	1.5349(0.0011)	1.0741(0.0008)	0.8324(0.0513)	0.4041(0.0766)
E^2GAN [44]	1.5139(0.0085)	1.0593(0.0058)	0.8035(0.0421)	0.4413(0.0511)
MIAM [51]	1.3226(0.0041)	0.1650(0.0005)	0.8381(0.0075)	0.4350(0.0164)
MTSIT [58]	1.9679(0.2400)	0.2455(0.0299)	0.8207(0.0117)	0.4220(0.0238)
GCT [351]	1.3868(0.2330)	0.1730(0.0290)	0.8410(0.0162)	0.4659(0.0230)
AGRN [246]	1.3420(0.0228)	0.1674(0.0025)	0.7270(0.0041)	0.2659(0.0071)
GRIN [244]	1.3101(0.0627)	0.1634(0.0078)	0.8462(0.0109)	0.4773(0.0247)
Our_{α}	1.4367(0.1719)	0.1792(0.0214)	0.8208(0.0080)	0.4073(0.0477)
Our	**1.3039(0.0872)**	**0.1627(0.0108)**	**0.8622(0.0125)**	**0.5103(0.0326)**
eICU/48 hours after eICU admission	Multivariate Clinical Time Series Imputation		In-hospital Mortality Prediction	
Metrics	MAE	MRE	AUROC	AUPRC
GRU-D [39]	1.8661(0.0510)	0.1419(0.0038)	0.8027(0.0144)	0.3949(0.0245)
BRITS [40]	2.5136(0.4963)	0.1911(0.0377)	0.7968(0.0029)	0.3650(0.0094)
GRUI-GAN [43]	6.0778(0.0095)	1.1291(0.0006)	0.7750(0.0343)	0.3667(0.0352)
E^2GAN [44]	5.9437(0.0166)	1.1023(0.0041)	0.7829(0.0323)	0.3686(0.0382)
MIAM [51]	1.4796(0.1245)	0.1126(0.0095)	0.7404(0.0375)	0.3082(0.0325)
MTSIT [58]	2.5308(0.1942)	0.1923(0.0148)	0.7768(0.0038)	0.3983(0.0154)
GCT [351]	1.4624(0.0377)	0.1108(0.0026)	0.7686(0.0162)	0.3565(0.0137)
AGRN [246]	1.9260(0.0357)	0.1466(0.0026)	0.7267(0.0022)	0.3151(0.0046)
GRIN [244]	1.5053(0.1072)	0.1146(0.0081)	0.8180(0.0063)	0.4094(0.0188)
Our_{α}	1.9849(0.0371)	0.1511(0.0029)	0.7930(0.0117)	0.3869(0.0257)
Our	**1.4287(0.0958)**	**0.1092(0.0075)**	**0.8265(0.0138)**	**0.4355(0.0226)**

Chapter 8

Multi-Task Deep Neural Networks

The following manuscript has been incorporated into this chapter:
Yuxi Liu, Zhenhao Zhang, Shaowen Qin, Jiang Bian. Multi-Task Deep Neural Networks for Irregularly Sampled Multivariate Clinical Time Series. In *2024 IEEE 12th International Conference on Healthcare Informatics (ICHI)*, Under Review (This paper is an extended version of the publication [1])

8.1 Introduction

Digital health systems are widely available and being integrated into routine healthcare operations, resulting in growth in electronic health records (EHRs) data. With advances in data processing tools and methods, there has been an increased interest in establishing health risk prediction models as a key instrument in clinical decision support [352–354].

However, EHR data has its unique characteristics, such as high dimensionality, sparsity, irregularity, temporality, bias, etc [299]. It is technically challenging to apply traditional machine learning or statistical models to such data. The high degree of irregularity, including many missing values and varying time intervals, needs to be dealt with when establishing predictive models. EHR data irregularity is a natural consequence of health care provision, as every patient is different. For example, patients are more likely to be examined by healthcare specialists when changes in their health status or treatment decisions occur, hence the intervals between physiological variables are often irregular (as shown in Figure 8.1) [60, 355]. Additionally, it can be seen from Figure 8.1 that the data miss-

ingness of patient B is significantly higher than that of patient A. The variation of missing data patterns adds another layer of complexity, which would affect the performance of downstream risk prediction [49, 356].

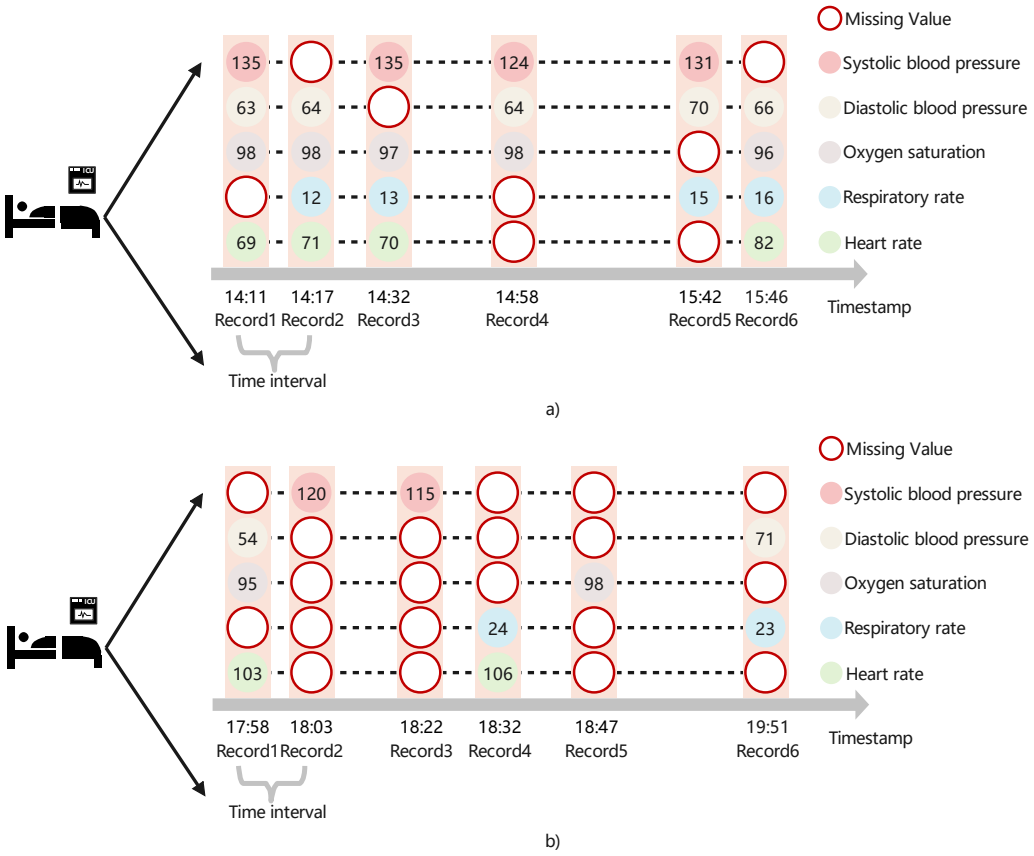

Figure 8.1: Illustration of irregular multivariate clinical time series.

Most of the previous research studies with EHR data have been focused on the provision of deep learning-based solutions [39, 40, 43, 44, 46, 60]. These studies mainly impute missing values by incorporating recurrent neural networks to learn variable correlations and introduce time-decay mechanisms to take the effect of varying time intervals into account. The complete data matrices obtained from the imputation task are used for downstream risk prediction tasks.

There are three modes of imputation-prediction processing, each has its drawbacks. The first is to consider imputation and prediction as two separable steps [43, 44, 46, 56–58]. Although promising prediction performance has been demonstrated, these prediction models have not attempted to learn the impact of the patterns of missing data in EHR data. This may lead to suboptimal prediction performance. As a better alternative, imputation and prediction can be tuned together within an end-to-end learning framework rather than be

separated into two parts. This is the second mode. Despite its efficacy, existing architectures for such modes are specifically proposed for improving risk prediction performance [39, 60–62]. When used for imputation and prediction tasks, the architecture treats both as separate optimization tasks, which essentially is not different from the first mode. The third imputation-prediction processing mode is similar to that used by the second, with the difference that the objective of the third is to simultaneously perform both imputation and prediction tasks [40, 42, 51, 63–65]. However, imputation and prediction tasks may lead to competition due to the shared parameter problem, as illustrated during multi-task learning for optimization in some studies [66–68]. This kind of optimization could also lead to suboptimal imputation and prediction results.

In this chapter, we propose to construct a single deep learning framework based on multi-task learning that performs the risk prediction task while incorporating the imputation task as an auxiliary task. The benefit of implementing the imputation task as an auxiliary task is that such an approach can improve risk prediction performance rather than competing with it. It is a novel deep imputation-prediction network in which imputation and prediction tasks are implemented with an auxiliary network and a main network, respectively (as shown in Figure 8.2). The intuition behind our network architecture is that the direction of information flow is from the auxiliary network to the main network only. By doing so, the forward pass of the main network depends on the auxiliary network, while the inference of the auxiliary network does not depend on the main network. Therefore, imputation and prediction tasks can be implemented simultaneously within a single deep learning framework without competition.

The main contributions of this paper are listed as follows:

- We propose a novel imputation-prediction method to simultaneously carry out imputation and prediction tasks using irregularly sampled multivariate clinical time series.
- To the best of our knowledge, this is the first research to perform risk prediction tasks by incorporating the imputation task as an auxiliary task while carrying out both simultaneously.
- Experiments on data from two real-world EHR databases using our proposed method demonstrates superior prediction and imputation accuracy.

8.2 Method

In this section, we propose a Multi-Task Deep Neural Network, which performs the risk prediction task while incorporating the imputation task as an auxiliary task. We first in-

troduce the basic notations. We then detail the network architecture. Finally, we present how to use the Multi-Task Deep Neural Network for imputation and prediction tasks.

8.2.1 Basic Notations

We represent a multivariate clinical time series with up to K physiological variables as $X = \{x_1, x_2, \cdots, x_T\} \in \mathbb{R}^{K \times T}$, where T is the number of medical records. For example, $x_t = \{x_t^1, x_t^2, \cdots, x_t^K\} \in \mathbb{R}^K$ is the t-th medical record and x_t^k is the value of the k-th physiological variable of x_t.

Since X can be incomplete, we represent the missing values in x_t^k by introducing a masking vector M_t^k as:

$$M_t^k = \begin{cases} 1, & if\ x_t^k\ is\ observed \\ 0, & otherwise \end{cases} . \tag{8.1}$$

Let s_t represent the timestamp when the t-th medical record is obtained, and Δ_t represent the time interval for each physiological variable since its last medical record. The Δ_t^k can be written as:

$$\Delta_t^k = \begin{cases} s_t - s_{t-1} + \Delta_{t-1}^k, & t > 1, M_{t-1}^k = 0 \\ s_t - s_{t-1}, & t > 1, M_{t-1}^k = 1 \\ 0, & t = 1 \end{cases} . \tag{8.2}$$

Let $D = \{(X_n, Y_n^{(I)}, Y_n^{(P)}) | n = 1, 2, \cdots, N\}$ represent an EHR dataset with up to N multivariate clinical time series. Each has two target labels $Y_n^{(I)}$ and $Y_n^{(P)}$, which are used for imputation and prediction tasks.

8.2.2 Network Architecture

In this section, we propose a new deep imputation-prediction network by modeling irregularly sampled multivariate clinical time series with the utilization of the convolutional and residual recurrent components (As shown in Figure 8.2). The benefit of integrating the convolutional and residual recurrent components is capturing the long-term dependencies and short-term correlations of multivariate clinical time series data, which leads to good representation learning [1, 5, 357]. Moreover, we incorporate the most commonly used time decay mechanisms [39, 60, 150] into the proposed network architecture to deal with varying time intervals. We give a detailed description of the proposed network architecture in the following subsections.

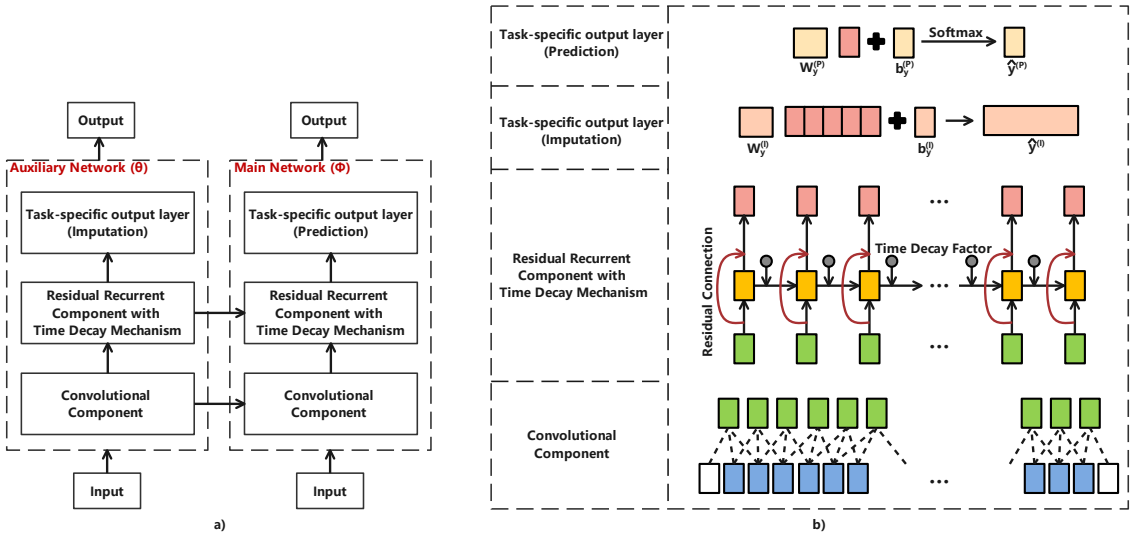

Figure 8.2: Schematic representation of the architecture and workflow of the proposed network.

Convolutional Component

Given multivariate clinical time series X, we first construct a learnable variable to carry out prefilling operations. Let ψ represent a learnable variable, which is initialised as $\tilde{X} = M \cdot X + (1 - M) \cdot \psi$. We then apply the zero vector padding to $\tilde{X}$ by embedding a zero vector before the first record of $\tilde{X}$ and after the last record of $\tilde{X}$. We finally feed $\tilde{X}$ into a convolutional component.

In particular, a combination of up to K kernels $\{W_k\}_{k=1}^{K}$ is applied to the corresponding K variables. For example, $\tilde{x}^k_{t:t+l-1}$ represents the concatenation of k-th variable of different records $\{\tilde{x}^k_t, \tilde{x}^k_{t+1}, \cdots, \tilde{x}^k_{t+l-1}\}$. A kernel $W_k \in \mathbb{R}^l$ is applied to the window of $\tilde{x}^k_{t:t+l-1}$ to generate a new latent variable $v^k_t \in \mathbb{R}$ with a rectified linear unit (ReLU) activation function as:

$$v^k_t = ReLU(\tilde{x}^k_{t:t+l-1} \cdot W_k + b_k), \tag{8.3}$$

where $ReLU(x) = max(x, 0)$ and $b_k \in \mathbb{R}$ is a bias. In a follow-up step, W_k is implemented as a sliding window in order to generate a latent vector $v^k = \{v^k_1, v^k_2, \cdots, v^k_T\}$. The final representation of $\tilde{X}$ can be $v \in \mathbb{R}^{K \times T}$ based on concatenating all of those latent vectors.

Residual Recurrent Component

The residual recurrent component is built upon GRU [147]. The GRU is a variant of RNN that is characterised by the reset gate r_t and the update gate u_t, which decide the information from the previous hidden state h_{t-1} should be updated or reset the previous hidden state

h_{t-1} whenever needed.

Given the final representation v_t obtained from the convolutional component, GRU generates h_t by the use of a linear combination of the previous hidden state h_{t-1} and the candidate state $\tilde{h}_t$ as:

$$\begin{aligned} h_t = GRU(v_t) = u_t \odot \tilde{h}_t + (1 - u_t) \odot h_{t-1}, \\ u_t = \sigma(W_u^1 \cdot h_{t-1} + W_u^2 \cdot v_t + b_u), \\ \tilde{h}_t = tanh(W_h^1 \cdot (r_t \odot h_{t-1}) + W_h^2 \cdot v_t + b_h), \\ r_t = \sigma(W_r^1 \cdot h_{t-1} + W_r^2 \cdot v_t + b_r), \end{aligned} \tag{8.4}$$

where all W and b are learnable parameters, $\odot$ is the element-wise multiplication, and σ is the sigmoid function. The u_t controls the information from the previous hidden state h_{t-1} and the candidate state $\tilde{h}_t$. Note that $\tilde{h}_t$ is computed in the way as a standard implementation of RNN. The r_t decides the proper amount of information from the previous hidden state h_{t-1} that contributes to $\tilde{h}_t$ generation.

Inspired by the residual connection [198], we forward an identity mapping of the GRU input to its output side as $h'_t = ResGRU(v_t) = GRU(v_t)+v_t$. By doing so, the corresponding residual block is only required to capture the difference between input and output, which in turn simplifies the overall training process by reducing the number of epochs required for the model to converge.

Time Decay Mechanism

To capture the impact of varying time intervals, competitive time decay mechanisms that fit a deep imputation-prediction network are sought and critically reviewed. Collectively, we separately incorporate three types of time decay mechanisms [39,60,150] into the proposed network architecture to test their efficacy on imputation and prediction performance.

Specifically, we augment the residual recurrent component with time decay mechanisms [39, 60] respectively. The mathematical formulations for [39, 60] are as:

$$f_1(\Delta_t) = exp\{-max(0, W_\gamma \cdot \Delta_t + b_\gamma)\}, \tag{8.5}$$

where W_γ and b_γ are learnable parameters.

$$\begin{aligned} f_2(\Delta_t) = \frac{1}{log(e + \Delta_t)}, \\ f_3(\Delta_t) = e^{-\Delta_t}, \\ f_4(\Delta_t) = \frac{1}{\Delta_t}, \end{aligned} \tag{8.6}$$

where $f_2(\cdot)$, $f_3(\cdot)$, and $f_4(\cdot)$ are three types of decay functions that describe the process of reducing weight by a consistent percentage rate over a period of time.

The above $f(\cdot)$ functions are integrated into the GRU architecture that contribute hidden state representation generation. Therefore, $\hat{h}_{t-1}$ can be written as $f(\Delta_t) \odot h_{t-1}$. Subsequently, Eq. (4) can be rewritten as:

$$\begin{aligned} h_t = GRU(v_t) = u_t \odot \tilde{h}_t + (1 - u_t) \odot \hat{h}_{t-1}, \\ u_t = \sigma(W_u^1 \cdot \hat{h}_{t-1} + W_u^2 \cdot v_t + b_u), \\ \tilde{h}_t = tanh(W_h^1 \cdot (r_t \odot \hat{h}_{t-1}) + W_h^2 \cdot v_t + b_h), \\ r_t = \sigma(W_r^1 \cdot \hat{h}_{t-1} + W_r^2 \cdot v_t + b_r). \end{aligned} \tag{8.7}$$

Compared with time decay mechanisms in [39] and [60], [150] also takes the similarity between medical records into consideration on the time decay mechanism. In other words, if the similarity between two medical records is significant, the importance of the previous one should be slightly decayed. This is achieved by combining the attention function [195] and the decay function $\frac{1}{log(e+\Delta_t)}$. The mathematical formulations for [150] are as:

$$\begin{aligned} Q_T^k = W_Q^k \cdot \tilde{x}_T^k, \\ K_t^k = W_K^k \cdot \tilde{x}_t^k, \\ \eta_t^k = tanh(\frac{Q_T^k \cdot K_t^k}{\beta_k \cdot log(e + (1 - \sigma \cdot (Q_T^k \cdot K_t^k)) \cdot \Delta_t)}), \\ \alpha = Softmax(\eta), \\ \tilde{X}' = \alpha \odot \tilde{X}. \end{aligned} \tag{8.8}$$

Multi-Task Learning for Imputation and Prediction Tasks

Multi-task learning refers to a single shared machine learning model that performs multiple target tasks simultaneously. As mentioned in the introduction section, the mode of imputation-prediction processing used by [40,42,51,63–65] are based on multi-task learning with deep neural networks.

Multi-task learning with deep neural networks can be done based on either hard or soft parameter sharing of hidden layers [358]. The hard parameter sharing method allows target tasks to share parameters from a series of hidden layers, while the soft parameter sharing method allows each target task to have its own backbone with its own parameters. Previous studies suggest that multiple target tasks lead to competition regardless of the hard or soft parameter sharing methods [359].

In response to the competition, we construct different optimizers for imputation and

prediction tasks and then perform the risk prediction task by incorporating the imputation task as an auxiliary task. As Figure 8.2 shows, an auxiliary network and a main network are developed and introduced to the imputation and prediction tasks. The key aspect of our network architecture is that the direction of information flow is from the auxiliary network to the main network only. Accordingly, the forward pass of the main network depends on the auxiliary network, while the inference of the auxiliary network does not depend on the main network. Because of this, imputation and prediction tasks can be implemented simultaneously within a single deep learning framework without competition.

Now we define the objective functions for the imputation and prediction tasks. Given the final representation h', we utilize a fully connected layer to impute missing values as:

$$\hat{y}^{(I)} = W_y^{(I)} \cdot h' + b_y^{(I)}. \tag{8.9}$$

The objective function of the imputation task is the mean square error as:

$$\mathcal{L}^{(I)} = \frac{1}{N} \sum_{n=1}^{N} (M_n \odot \hat{y}_n^{(I)} - M_n \odot Y_n^{(I)})^2. \tag{8.10}$$

For the risk prediction task, we utilize h'_T as input for a Softmax output layer in order to obtain the predicted $\hat{y}^{(p)}$ as:

$$\hat{y}^{(P)} = Softmax(W_y^{(P)} \cdot h'_T + b_y^{(P)}). \tag{8.11}$$

The objective function of the risk prediction task is the average of cross-entropy with a constraint L-infinity norm $\| \cdot \|_\infty$ as:

$$\begin{aligned} \mathcal{L} = -\frac{1}{N} \sum_{n=1}^{N} ((Y_n^{(P)})^\top \cdot log(\hat{y}_n^{(P)}) + (1 - Y_n^{(P)})^\top \cdot log(1 - \hat{y}_n^{(P)})), \\ \mathcal{L}^{(P)} = \mathcal{L} + \lambda \cdot \|\theta - \phi\|_\infty, \\ \|\theta - \phi\|_\infty = \lim_{p \to \infty} (\sum_j |\theta_j - \phi_j|^p)^{\frac{1}{p}}, \end{aligned} \tag{8.12}$$

where λ is a scaling parameter to control the contribution of cross-entropy and constraint, $\|\theta - \phi\|_\infty$ is the distance between the auxiliary network parameter $\{\theta_j\}_{j=1}^J$ and the main network parameter $\{\phi_j\}_{j=1}^J$, and J is the number of shared layers in the network architecture.

8.3 Experiments

8.3.1 Experimental Setup

Datasets, Tasks, and Evaluation Metrics

We validate the performance of our model[1] on two real-world EHR databases, i.e., MIMIC-III[2]) Database and eICU[3] Database. We extract 21,105 and 36,670 patients/samples from the MIMIC-III and eICU databases, where the Positive (likely to die)/Negative (unlikely to die) ratio is 1:6.56 and 1:7.49, respectively. Detailed information on the two databases can be found in the literature [38,280]. We conduct multivariate clinical time series imputation and in-hospital mortality prediction experiments on the two databases. The multivariate clinical time series data are selected on the basis of [53, 54]. Tables A.1 and A.2 show multivariate clinical time series data on both databases.

To validate the imputation performance of our model, we use mean absolute error (MAE) and mean relative error (MRE) between predicted and actual values as the primary evaluation metrics. Given the n-th actual and predicted values x_n and $\hat{x}_n$, as well as the total number of ground truth N, we define MAE and MRE as:

$$\begin{aligned} MAE &= \frac{\sum_{n=1}^{N} |\hat{x}_n - x_n|}{N}, \\ MRE &= \frac{\sum_{n=1}^{N} |\hat{x}_n - x_n|}{\sum_{n=1}^{N} |x_n|}. \end{aligned} \tag{8.13}$$

According to the literature [53, 54], in-hospital mortality risk prediction is defined as predicting the mortality risk of patients based on the data from the first 48 hours after admission. This is a binary classification task with the area under the receiver operating characteristic curve (AUROC) and the area under the precision-recall curve (AUPRC) being the primary evaluation metrics. It is worth noting that the physiologic decline prediction is defined as predicting the mortality risk of patients based on the data from the first 24 hours after admission [53, 54]. Taken together, two mortality prediction tasks are carried out on each database based on the data from the first 24 and 48 hours after admission.

[1] The implementation code is available at https://github.com/zxc0160/Model
[2] https://mimic.physionet.org
[3] https://eicu-crd.mit.edu/

Baseline Methods

To demonstrate the effectiveness of our method, we use GRU-D [39], BRITS [40], V-RIN [65], GRUI-GAN [43], E^{2}GAN [44], STING [46], MTSIT [58], MIAM [51] as baseline methods for comparison.

Besides, we present six variants of our method as follows:

Ours$_\alpha$: A variation of our method in which we incorporate the time decay mechanism [39] into the residual recurrent component.

Ours$_\beta$: A variation of our method in which we incorporate the time decay mechanism [60] (i.e., the first row of Eq. (6)) into the residual recurrent component.

Ours$_\gamma$: A variation of our method in which we incorporate the time decay mechanism [60] (i.e., the second row of Eq. (6)) into the residual recurrent component.

Ours$_\delta$: A variation of our method in which we incorporate the time decay mechanism [60] (i.e., the third row of Eq. (6)) into the residual recurrent component.

Ours$_\epsilon$: A variation of our method in which we incorporate the time decay mechanism [150] into the network architecture.

Ours$_\varepsilon$: A variation of our method that does not perform any time decay mechanism.

Implementation Details

The training was done in a machine equipped with a CPU: AMD EPYC 7543, 80GB RAM, and a GPU: NVIDIA A40 with 48GB of memory using Pytorch 1.10.0. For training the model, we used Adam optimizer [283] with the mini-batch of 256 patients. We randomly split 70%, 15%, and 15% of the dataset into training, validation, and testing sets. We chose the best one from the model's performance on the validation set.

For the MIMIC-III database, the number of physiological variables K is 17. For the convolutional component, the kernel size is 3 and the stride is 1. For the residual recurrent component, the dimension of hidden variables is 17. For multi-task learning, the scaling parameter λ is 0.002, and the learning rates for the imputation and prediction optimizers are 0.0065 and 0.0034, respectively. We also applied the dropout method to the imputation and prediction tasks; the dropout rate is 0.3 and 0.1, respectively. For the eICU database, the number of physiological variables K is 16. For the convolutional component, the kernel size is 3 and the stride is 1. For the residual recurrent component, the dimension of hidden variables is 16. For multi-task learning, the scaling parameter λ is 0.0013, and the learning rates for the imputation and prediction optimizers are 0.0077 and 0.0022, respectively. We also applied the dropout method to the imputation and prediction tasks; the dropout rate is 0.3 and 0.2, respectively. For a fair comparison, the hyper-parameter of the proposed model (i.e., λ) was fine-tuned by a grid-searching strategy.

Since GRU-D is proposed for the risk prediction task, the regression component [40] was integrated into its network architecture to generate imputation results. We replaced the linear decoder of MTSIT with a Softmax output layer to generate prediction results. For a fair comparison, we used the complete data matrices imputed by GRUI-GAN, E^{2}GAN, STING as input to GRU to generate prediction results.

8.3.2 Performance Analysis

Tables 8.1, 8.2, 8.3, and 8.4 present the imputation and prediction results obtained from all methods. The larger the scores of AUROC and AUPRC, the better the predictive performance of the method. The lower the values of MAE and MRE, the better the imputation performance of the method. Values in the parentheses denote standard deviations. Overall, these results suggest that our proposed method achieves the best imputation and prediction accuracy. For example, it can be seen from Table 8.1 that Ours$_\alpha$ reported significantly more AUROC and AUPRC scores than the best baseline method V-RIN and Ours$_\delta$ reported significantly less MAE and MRE values than the best baseline method MTSIT.

Besides, it can be seen from the data in Table 8.2 that V-RIN is the best baseline method and significantly outperforms other baseline methods in both imputation and prediction tasks. The interesting aspect of Table 8.3 is that MTSIT achieves the best imputation performance and the worst prediction performance. From the data in Table 8.4, it is apparent that the prediction performance among the baseline methods GRU-D, BRITS, MTSIT, and MIAM is very close.

Comparing the prediction results across the four tables, it can be seen that Ours$_\alpha$ significantly and consistently outperforms Ours$_\varepsilon$ (i.e., without any time decay mechanism). These results suggest that the time decay mechanism [39] plays an important role in addressing varying time intervals of multivariate time series data, which leads to good prediction performance. Ours$_\alpha$ also consistently outperforms other baseline methods in the risk prediction task. For example, it can be seen from Table 8.2 that Ours$_\alpha$ reported more AUROC and AUPRC scores than the best baseline method V-RIN and also variant method Ours$_\epsilon$. These results suggest that the time decay mechanism [39] is particularly well suited for improving the downstream risk prediction performance of our network architecture.

The most interesting aspect of the four tables is the imputation results of the variant methods (i.e., Ours$_\beta$, Ours$_\gamma$, Ours$_\delta$). From the data in Tables 8.1 and 8.3, we can see that Ours$_\delta$ resulted in the lowest value of MAE and MRE. From the data in Tables 8.2 and 8.4, we can see that Ours$_\beta$ and Ours$_\gamma$ resulted in the lowest value of MAE and MRE, respectively. These results suggest that the time decay mechanism [60] helps to improve the imputation performance of our network architecture. In addition, the three variant

methods (Ours_{β}, Ours_{γ}, Ours_{δ}) consistently outperform $\text{Ours}_{\varepsilon}$ (i.e., without any time decay mechanism). These results suggest that capturing the effect of varying time intervals can help improve imputation performance.

Comparing the results of three types of time decay mechanisms [39, 60, 150], it can be seen that the incorporating of the two types of time decay mechanisms [39, 60] into the multi-task deep neural network could improve the imputation performance, but only the time decay mechanism [39] could improve the prediction performance. Contrary to expectations, the incorporating of time decay mechanism [150] into the multi-task deep neural network has failed to improve imputation and prediction performance. This result is in line with the literature [62] finding, which showed there is high inconsistency in the effectiveness of the time decay mechanism. This conflicting experimental result could be associated with the structure of deep neural networks. Further investigations are needed to confirm and validate this finding.

Table 8.1: Performance of all methods on multivariate clinical time series imputation and in-hospital mortality prediction.

Method	Multivariate Clinical time series imputation		In-hospital mortality prediction	
MIMIC-III/48 hours after ICU admission	MAE	MRE	AUROC	AUPRC
GRU-D [39]	3.6873(0.0218)	36.20%(0.0021)	0.7294(0.0097)	0.2771(0.0156)
BRITS [40]	5.3631(0.3804)	52.65%(0.0374)	0.7447(0.0092)	0.2879(0.0168)
V-RIN [65]	3.1522(0.0080)	31.15%(0.0010)	0.7758(0.0003)	0.3244(0.0010)
GRUI-GAN [43]	7.1359(0.0055)	70.05%(0.0005)	0.7619(0.0077)	0.3349(0.0178)
E^2GAN [44]	6.9705(0.0104)	68.43%(0.0010)	0.7652(0.0054)	0.3599(0.0133)
STING [46]	5.1522(0.0202)	50.88%(0.0020)	0.7667(0.0106)	0.3402(0.0187)
MTSIT [58]	1.6965(0.1114)	21.16%(0.0139)	0.6841(0.0171)	0.2584(0.0182)
MIAM [51]	2.0941(0.0596)	26.13%(0.0074)	0.7192(0.0158)	0.2600(0.0111)
Ours_{α}	0.6017(0.0289)	7.50%(0.0036)	**0.8031(0.0045)**	**0.3800(0.0126)**
Ours_{β}	0.4246(0.0600)	5.29%(0.0074)	0.7420(0.0435)	0.2982(0.0438)
Ours_{γ}	0.4080(0.0531)	5.09%(0.0066)	0.7331(0.0108)	0.2689(0.0145)
Ours_{δ}	**0.3743(0.0565)**	**4.66%(0.0070)**	0.7093(0.0258)	0.2547(0.0271)
Ours_{ϵ}	1.3653(0.0232)	17.03%(0.0028)	0.7754(0.0085)	0.3621(0.0176)
$\text{Ours}_{\varepsilon}$	0.8525(0.0396)	10.63%(0.0049)	0.7789(0.0077)	0.3553(0.0145)

Table 8.2: Performance of all methods on multivariate clinical time series imputation and in-hospital mortality prediction.

Method	Multivariate Clinical time series imputation		In-hospital mortality prediction	
eICU/48 hours after ICU admission	MAE	MRE	AUROC	AUPRC
GRU-D [39]	2.8066(0.0107)	21.43%(0.0008)	0.7195(0.0111)	0.2631(0.0145)
BRITS [40]	4.0963(0.3359)	31.26%(0.0257)	0.7254(0.0057)	0.2573(0.0062)
V-RIN [65]	1.8357(0.1097)	14.01%(0.0083)	0.7846(0.0139)	0.3373(0.0117)
GRUI-GAN [43]	9.9809(0.0056)	76.26%(0.0002)	0.7280(0.0105)	0.2871(0.0120)
E^2GAN [44]	9.7912(0.0111)	74.70%(0.0006)	0.7294(0.0106)	0.2970(0.0133)
STING [46]	8.0315(0.0466)	61.21%(0.0036)	0.7475(0.0186)	0.2838(0.0197)
MTSIT [58]	2.8713(0.1357)	21.92%(0.0103)	0.7237(0.0042)	0.2952(0.0103)
MIAM [51]	2.2828(0.1288)	17.42%(0.0098)	0.7222(0.0099)	0.2513(0.0087)
Ours$_\alpha$	0.8846(0.0697)	6.75%(0.0053)	**0.7984(0.0055)**	**0.3510(0.0075)**
Ours$_\beta$	**0.5852(0.0787)**	**4.46%(0.0060)**	0.7169(0.0155)	0.2650(0.0168)
Ours$_\gamma$	0.5858(0.0988)	4.47%(0.0075)	0.7166(0.0054)	0.2510(0.0047)
Ours$_\delta$	0.6011(0.1219)	4.58%(0.0093)	0.7028(0.0068)	0.2533(0.0060)
Ours$_\epsilon$	1.8689(0.0655)	14.26%(0.0050)	0.7577(0.0048)	0.2994(0.0054)
Ours$_\varepsilon$	1.1397(0.0918)	8.70%(0.0066)	0.7477(0.0074)	0.2861(0.0150)

Table 8.3: Performance of all methods on multivariate clinical time series imputation and in-hospital mortality prediction.

Method	Multivariate Clinical time series imputation		In-hospital mortality prediction	
MIMIC-III/24 hours after ICU admission	MAE	MRE	AUROC	AUPRC
GRU-D [39]	3.1752(0.0151)	36.74%(0.0017)	0.7277(0.0111)	0.2785(0.0133)
BRITS [40]	4.6305(0.3451)	53.55%(0.0485)	0.7387(0.0093)	0.2794(0.0154)
V-RIN [65]	2.8958(0.0040)	33.68%(0.0009)	0.7183(0.0144)	0.2776(0.1261)
GRUI-GAN [43]	6.2258(0.0026)	71.97%(0.0003)	0.7188(0.0098)	0.2655(0.0104)
E^2GAN [44]	6.1391(0.0056)	70.95%(0.0007)	0.7283(0.0070)	0.2624(0.0093)
STING [46]	4.6212(0.0162)	53.43%(0.0019)	0.7312(0.0083)	0.2579(0.0115)
MTSIT [58]	1.1495(0.0861)	26.35%(0.0197)	0.6459(0.0111)	0.2049(0.0131)
MIAM [51]	1.2760(0.0506)	29.26%(0.0116)	0.6845(0.0152)	0.2215(0.0128)
Ours$_\alpha$	0.3763(0.0202)	8.62%(0.0046)	**0.7491(0.0061)**	**0.2917(0.0095)**
Ours$_\beta$	0.2394(0.0311)	5.48%(0.0071)	0.6546(0.0339)	0.1940(0.0247)
Ours$_\gamma$	0.2412(0.0309)	5.52%(0.0070)	0.6521(0.0075)	0.2043(0.0039)
Ours$_\delta$	**0.2363(0.0316)**	**5.41%(0.0072)**	0.6541(0.0078)	0.2030(0.0068)
Ours$_\epsilon$	0.7881(0.0109)	18.07%(0.0025)	0.7143(0.0080)	0.2794(0.0131)
Ours$_\varepsilon$	0.4826(0.0162)	11.06%(0.0037)	0.7015(0.0105)	0.2620(0.0114)

Table 8.4: Performance of all methods on multivariate clinical time series imputation and in-hospital mortality prediction.

Method	Multivariate Clinical time series imputation		In-hospital mortality prediction	
eICU/24 hours after ICU admission	MAE	MRE	AUROC	AUPRC
GRU-D [39]	1.6043(0.0054)	20.82%(0.0007)	0.7024(0.0081)	0.2776(0.0134)
BRITS [40]	2.7905(0.2666)	36.21%(0.0234)	0.7082(0.0085)	0.2617(0.0083)
V-RIN [65]	1.1811(0.0745)	15.32%(0.0096)	0.7431(0.0070)	0.3071(0.0037)
GRUI-GAN [43]	5.9463(0.0057)	77.13%(0.0007)	0.7145(0.0099)	0.2996(0.0083)
E^2GAN [44]	5.7179(0.0050)	74.16%(0.0004)	0.7159(0.0101)	0.3057(0.0132)
STING [46]	5.2312(0.0609)	69.85%(0.0079)	0.7268(0.0098)	0.2976(0.0108)
MTSIT [58]	1.8202(0.1014)	23.62%(0.0131)	0.7067(0.0034)	0.2895(0.0106)
MIAM [51]	1.3827(0.0903)	17.94%(0.0117)	0.7084(0.0086)	0.2586(0.0076)
Ours$_\alpha$	0.5384(0.0454)	6.98%(0.0059)	**0.7679(0.0019)**	**0.3603(0.0105)**
Ours$_\beta$	0.3846(0.0687)	4.99%(0.0089)	0.6817(0.0145)	0.2341(0.0133)
Ours$_\gamma$	**0.3603(0.0561)**	**4.67%(0.0072)**	0.6766(0.0065)	0.2242(0.0086)
Ours$_\delta$	0.3913(0.0718)	5.07%(0.0093)	0.6785(0.0048)	0.2323(0.0040)
Ours$_\epsilon$	1.0752(0.0413)	13.95%(0.0053)	0.7422(0.0033)	0.3005(0.0077)
Ours$_\varepsilon$	0.7439(0.0472)	9.65%(0.0061)	0.7223(0.0075)	0.2847(0.0095)

Chapter 9

Discussion

This chapter presents a discussion of the research undertaken. In the following sections, we will include a response to the research aims in relation to the research undertaken by restating the research aims and discussing the results achieved. In addition, we will discuss the proposed approaches from different perspectives, including network architecture comparison, performance comparison, time-decay mechanism, transparency, interpretability, and reliability of the model decisions.

9.1 Confirmation of Research Aims

As stated in Chapter 1, the aim of this research was to investigate and develop new risk prediction models for healthcare decision support. One of the complexities of such an aim is the irregularity of EHR data, including many missing values and varying time intervals between medical records. With an aim that held such complexities, the proposed approach needed to be able to handle the irregularity of EHR data and predict patients' health risks.

Even though the ability to provide such a risk prediction model was the aim of the research, various objectives were set out to confirm it.

The first objective was to design and implement a machine learning model for classification based health risk prediction. In particular, missing values in EHR data need to be addressed for effective predictive modeling. As demonstrated in Chapter 3, the proposed Compound Density Network addressed this issue by imputing missing values as a patient health context was given. After generating imputed values for missing values, it

was demonstrated that the proposed Compound Density Network can also enhance the reliability of imputed values and quantify their uncertainties and thus improve prediction performance. This was illustrated in Figure 3.3 and Figure 3.4 when used for regularised learning, and it assigns smaller weights to imputed values with large variance and larger weights to imputed values with small variance.

The second objective was to incorporate the handling of varying time intervals between medical records into the Compound Density Network. This objective was achieved by extending the Compound Density Network into the Attention-based Bidirectional Recurrent Neural Network, as illustrated in Chapter 4. After presenting several case studies, it was demonstrated that the proposed Attention-based Bidirectional Recurrent Neural Network could capture the variation pattern of input variables at the time dimension and adaptively enhance the temporal representation of each pattern with adjustable weights (As shown in Figure 4.4). Besides, the proposed Attention-based Bidirectional Recurrent Neural Network demonstrated its ability to model more complex patterns as a clinical context was given. For example, this was illustrated in Figure 4.3 when used for modeling clinical feature distributions, and the glucose of patients with diabetes was modeled as a Gaussian mixture distribution.

The third objective was capable of imputing missing values in patient data using information from the subgroup of similar patients rather than the entire patient population. This objective was achieved with the development of three graph neural networks, including a Contrastive Neural Network, a Contrastive Graph Similarity Network, and a Multi-Graph Neural Network, as illustrated in Chapter 5, Chapter 6, and Chapter 7. After introducing these graph neural networks into the multivariate clinical time series imputation task, it was demonstrated that they outperform state-of-the-art baseline approaches by significant margins. For example, this was illustrated in Tables 5.2 when comparing the imputation results of the Contrastive Neural Network with those of the baseline approaches.

The last objective was to explore the processing mode of imputation and prediction. As demonstrated in Chapter 8, three modes of imputation-prediction processing were investigated for this work, including (i) considering imputation and prediction as two separable steps, (ii) training imputation and prediction within an end-to-end learning framework, and (iii) simultaneously performing both imputation and prediction tasks. Moreover, a Multi-Task Deep Neural Network was proposed, which performs the risk prediction task while incorporating the imputation task as an auxiliary task. This was achieved through constructing a single deep learning framework based on multi-task learning. It enabled the direction of information flow only from the auxiliary network to the main network. When the Multi-Task Deep Neural Network was applied to multivariate clinical time series imputation and in-hospital mortality risk prediction, it outperformed state-of-the-art

baseline approaches by significant margins.

9.2 Network Architecture Comparison

The proposed imputation-prediction approaches are developed based on modular deep neural networks. Modular deep learning has emerged as a promising solution to challenges from real-world EHR data [5, 29, 150, 360, 361]. In a modular deep learning framework, computation units are implemented as autonomous parameter-efficient modules, and information flow is conditionally passed among modules and subsequently aggregated. The benefit of constructing the modular deep learning framework is that such an approach can easily develop and introduce new neural network modules into the network architecture to support hidden knowledge and information extraction and inference; consequently, it is easy to extend the network architecture to a wide range of medical applications such as drug discovery and phenotype analysis.

As demonstrated in Figure 3.2, the Compound Density Network is a combination of three modules, including a Gated recurrent unit, a Mixture Density Network, and a Regularised Attention Network. These neural network modules are stacked and trained together. The Attention-based Bidirectional Recurrent Neural Network also consists of three modules, as shown in Figure 4.2, including a bidirectional Gated recurrent unit, a reliability-aware reconstruction, and a time-decay attention, where reliability-aware reconstruction and time-decay attention modules are integrated into the bidirectional Gated recurrent unit. Accordingly, the three modules used in the Attention-based Bidirectional Recurrent Neural Network are not directly stacked together, which is significantly different from the Compound Density Network. The multi-task deep neural network consists of a convolutional neural network module and a recurrent neural network module adjusted together by implementing the residual connection [198]. The convolutional neural network and recurrent neural network modules are integrated into a main network and an auxiliary network, respectively, and the direction of information flow is from the auxiliary network to the main network only, as illustrated in Figure 8.2. Accordingly, the forward pass of the main network depends on the auxiliary network, while the inference of the auxiliary network does not depend on the main network.

It is worth noting that the above three approaches all contain the recurrent neural network, especially the Compound Density Network and Attention-based Bidirectional Recurrent Neural Network, which are based on the recurrent neural network as the main backbone. Accordingly, they are categorised as recurrent neural network-based imputation approaches with similar network architectures to other approaches in this family, such as GRU-D [39], BRITS [40], TBM [55], BRNN [151], and InterpNet [42] described in

Chapter 2.

Different from the aforementioned recurrent neural network-based imputation approaches, the proposed contrastive neural network, contrastive graph similarity network, and multi-graph neural network are developed based on graph representation learning. In other words, the core idea of these approaches is to incorporate graph representation learning in representation learning for EHR data. In terms of network components, contrastive neural network and contrastive graph similarity network are more similar because they all include making a contrastive learning module for enhanced patient similarity calculations, as illustrated in Figure 5.2 and Figure 6.1. In contrast, the multi-graph neural network can be categorised as a type of Graph Recurrent Neural Network that combines graph neural network and recurrent neural network [246, 250, 362]. As typical graph representation learning usually involves learning the graphical structure of input data, as shown in Figure 7.1, these graph neural network-based approaches are more complex than those recurrent neural network-based approaches.

9.3 Performance Comparison

We compared the performance of the proposed deep imputation-prediction networks with Recurrent Neural Networks (i.e., GRU-D [39], BRITS [40], InterpNet [42], V-RIN [65]), Generative Adversarial Networks (i.e., GRUI-GAN [43], E^{2}GAN [44], Conditional GAN [57], Bi-GAN [45], STING [46], E^{2}GAN-RF [50], MBGAN [47], SA-EDGAN [313]), Attention-based Neural Networks (i.e., MTSIT [58], MIAM [51]), and Graph Neural Networks (i.e., GCT [351], AGRN [246], and GRIN [244]). From the results demonstrated in Chapter 3 to Chapter 8, we have demonstrated the effectiveness and superiority of our proposed deep imputation-prediction approaches on multivariate clinical time series imputation and in-hospital mortality risk prediction.

From the results demonstrated in Tables 4.1 and 4.2, we can see that Recurrent Neural Networks (i.e., GRU-D and InterpNet) resulted in the lowest value of MAE and MRE scores compared with Generative Adversarial Networks (i.e., GRUI-GAN, E^{2}GAN, Bi-GAN, and STING). Similarly, from the results demonstrated in Table 5.2, we can see that Recurrent Neural Networks (i.e., GRU-D and BRITS) resulted in the lowest value of MAE and MRE scores compared with Generative Adversarial Networks (i.e., GRUI-GAN, E^{2}GAN, E^{2}GAN-RF, and STING). These results suggest that Recurrent Neural Networks outperformed Generative Adversarial Networks in terms of multivariate clinical time series imputation on the MIMIC-III and eICU databases. It is difficult to explain this result, but it may be related to the quality of the input data, as the inputs from the two databases have high missing ratios. These findings might help us to better understand the character-

istics of deep neural networks, especially those Generative Adversarial Networks that may be more sensitive to the missing ratio of the input data.

From the results demonstrated in Table 5.2, we can see that Attention-based Neural Networks (i.e., MTSIT and MIAM) resulted in the lowest value of MAE and MRE scores compared with Recurrent Neural Networks and Generative Adversarial Networks. Additionally, the performance superiority of MTSIT and MIAM in the eICU database is more significant than that in the MIMIC-III database. Besides, from the results demonstrated in Tables 6.2 and 6.3, we can see that MIAM is a strong baseline approach in terms of multivariate clinical time series imputation. According to these results, we can demonstrate that Attention-based Neural Networks outperform Recurrent Neural Networks and Generative Adversarial Networks in terms of multivariate clinical time series imputation on the MIMIC-III and eICU databases.

From the results demonstrated in Tables 6.2, 6.3, 7.1 and 7.2, no significant differences were found in the imputation performance of the baseline approaches (except for most Generative Adversarial Networks, i.e., GRUI-GAN, E^2GAN, Conditional GAN, and STING) in the MIMIC-III and eICU databases. This result may be explained in part by the association of multivariate clinical time series, demographics, and ICD-9 diagnosis codes [3]. Therefore, demographics and ICD-9 diagnosis codes are influenced factors that have a considerable (positive) impact on the imputation performance of multivariate clinical time series. Moreover, we can see that Generative Adversarial Networks resulted in a higher value of MAE and MRE scores in terms of multivariate clinical time series imputation on the eICU database. This also accords with our earlier observations, which showed that Generative Adversarial Networks achieved suboptimal imputation performance compared with Recurrent Neural Networks and Attention-based Neural Networks.

From the results demonstrated in Chapter 3 to Chapter 8, we have demonstrated that there is high inconsistency in the imputation and prediction effectiveness of the baseline approaches. This was illustrated in Tables 8.1, 8.2, 8.3, and 8.4 when comparing the imputation and prediction results of the baseline approaches. For example, MTSIT achieved the best imputation performance and suboptimal prediction performance. Similarly, from the results demonstrated in Tables 7.1 and 7.2, MIAM, GRIN, and GCT achieved the best imputation performance and suboptimal prediction performance. This provides some explanation as to why the Multi-Task Deep Neural Network needs to be proposed. In other words, imputation and prediction can be implemented as a multi-task learning problem, where the imputation task is an auxiliary task to improve risk prediction performance. Overall, the proposed Multi-Task Deep Neural Network simultaneously achieved the best imputation and prediction performance.

9.4 Time-decay Mechanism

The novelty of the Attention-based Bidirectional Recurrent Neural Network is its time-decay attention module's ability to incorporate three decay functions to capture the variation pattern of input variables at the time dimension and adaptively enhance the temporal representation of each pattern with adjustable weights. As well it examines the association between input variables to identify critical indicative variables regardless of how long ago the associated event happened. This allows the Attention-based Bidirectional Recurrent Neural Network to work for real-world applications. By applying the time-decay attention module to all patient journeys, the decay trends of clinical features are consistent with the three decay functions instead of a linear process, as seen in literature [39]. Further studies, which take more decay functions into account, will need to be undertaken.

9.5 Transparency and Interpretability

The novelty of the proposed imputation-prediction approaches lies in their ability to present transparency and interpretability of the model decisions. A predictive model's transparency generally means its ability to export a relationship map between inputs and outputs [363]. The relationship map allows users to understand, validate, and edit the findings extracted by the predictive model. Model interpretability refers to the ability to understand learning processes without knowing the results [364]. As demonstrated in Figure 3.3 and Figure 3.4, the regularised attention network module of the Compound Density Network could export a relationship map between inputs and outputs, where less reliable imputed values are assigned lower weights and vice versa. The reliability-aware reconstruction module of the Attention-based Bidirectional Recurrent Neural Network could model complex patterns of input clinical features. As demonstrated in Figure 4.3, the reliability-aware reconstruction module could model the input clinical feature as a Gaussian mixture distribution (i.e., multimodal distribution) instead of an unimodal distribution. These results further support the idea of [4, 287], which found the conditional distribution should be multimodal for tasks such as structured prediction problems, forming one-to-many mappings. The contrastive learning module of the contrastive graph similarity network could flexibly group similar patients based on the imputation and prediction tasks, as illustrated in Figure 6.2.

9.6 Epistemic Uncertainty and Aleatoric Uncertainty

The novelty of the proposed Compound Density Network lies in its ability to present the reliability of the model decisions. This was demonstrated in Figure 3.5, Figure 3.6 and Figure 3.7, the mixture density network module of the Compound Density Network could capture and quantify the impact of epistemic/model uncertainty and aleatoric uncertainty on the in-hospital mortality risk prediction. These results suggest that epistemic uncertainty and aleatoric uncertainty have a considerable impact on model decisions. There are similarities between the attitudes expressed by epistemic uncertainty and aleatoric uncertainty analysis in this study and those described by [255]. The success of this demonstration provides real-world validation of foundational work that is believed to be a first of its kind for analyzing epistemic uncertainty and aleatoric uncertainty in the context of multivariate clinical time series imputation and in-hospital mortality risk prediction. With such an approach for analyzing epistemic uncertainty and aleatoric uncertainty in a real-world application, specifically for users (e.g., healthcare professionals), they can be used to build trust in prediction models, which may help facilitate and support the collaboration between users and developers. Further experimental investigations are needed to examine the effects of epistemic uncertainty and aleatoric uncertainty on more medical applications. For example, the effects of epistemic uncertainty and aleatoric uncertainty on the mortality risk in patients with a particular disease, such as Congestive Heart Failure (CHF) [365] and Diabetes [366] will need to be undertaken. Considerably modeling work will have to be conducted in order to reduce the effects of epistemic uncertainty, such as incorporating more observations and reducing the number of parameters of neural networks.

Chapter 10

Conclusion

In the following sections, we will discuss our research findings in relation to each chapter, highlight corresponding contributions, acknowledge the limitations, and propose future research.

10.1 Summary of the thesis findings and contributions

In Chapter 3, we have presented a novel approach of integrated training and regularizing a deep learning model with the aim of predicting patient health risk using EHRs with a large number of missing values. We validated the proposed *CDNet* on the in-hospital mortality risk prediction tasks using the publicly available MIMIC-III database that has a large degree of missingness in the input. Extensive experimental results demonstrated that *CDNet* significantly outperformed existing approaches. The ablation experiments proved that regularizing imputed values is a key factor for performance improvements. Further analysis of prediction uncertainty proved that our model could capture both aleatoric and epistemic uncertainties, which allows model users to know how reliable the results are.

In Chapter 4, we have presented a novel deep imputation-prediction network with the aim of improving the prediction of patient health risks using EHR data. We integrated two novel modules, including time-decay attention and reliability-aware reconstruction, in a bidirectional GRU that performs missing data imputation and health risk prediction

tasks. We evaluated the efficacy of our approach with the publicly available MIMIC-III and eICU databases, proving the competitiveness and superiority of our approach in multivariate clinical time series imputation and in-hospital mortality risk prediction compared with baseline approaches. Moreover, several case studies are presented to show the transparency and interpretability of the model decisions.

In Chapter 5, we have presented a novel contrastive learning-based imputation-prediction network to perform imputation and prediction with EHR data. The proposed approach explicitly considers patient similarity by stratification of EHR data and successfully integrates contrastive learning into the network architecture. We empirically show that the proposed approach outperforms all the baseline approaches by conducting multivariate clinical time series imputation and in-hospital mortality risk prediction on the publicly available MIMIC-III and eICU databases. The ablation experiments confirmed the effectiveness of the network construction with enhanced imputation and prediction performance.

In Chapter 6, we have presented a novel Contrastive Graph Similarity Network, which focuses on the provision of missing data imputation and health risk prediction. The proposed approach explicitly calculates the similarity between patients and then aggregates the information from similar patients to impute missing values. We evaluated our approach against competing baseline approaches on the publicly available MIMIC-III and eICU databases, and the results demonstrated the effectiveness and superiority of our approach in multivariate clinical time series imputation and in-hospital mortality risk prediction. Further, the model analysis results confirmed the effectiveness of constructing the contrastive learning module in patient similarity calculating.

In Chapter 7, we have presented a novel Multi-Graph Fusion Framework for patient representation learning. We demonstrated the effectiveness of our approach with extensive experiments on the publicly available MIMIC-III and eICU databases with multivariate clinical time series imputation and in-hospital mortality risk prediction, and the results indicated that our approach outperforms all baseline approaches.

In Chapter 8, we have presented a novel deep imputation-prediction network based on multi-task learning, which performs risk prediction tasks by incorporating the imputation task as an auxiliary task. We experimentally demonstrated that the proposed approach achieves the best imputation and prediction accuracy by conducting multivariate clinical time series imputation and in-hospital mortality risk prediction on the publicly available MIMIC-III and eICU databases. Moreover, we empirically demonstrated that the incorporation of time decay mechanisms is a key factor for superior imputation and prediction performance.

10.2 Limitations and Future Works

10.2.1 Network Architecture Optimization

Although the proposed deep imputation-prediction networks have achieved promising performance in multivariate clinical time series imputation and in-hospital mortality prediction tasks, both in performance and computing efficiency, there is a need to find the optimal network architecture; consequently, it is very crucial to optimize these network architectures. In future investigations, it might be possible to reduce the complexity of neural network architecture in which the hyper-parameters and hidden layers as well as the number of nodes can be tuned.

Training deep neural network models is computationally expensive, especially when building very deep neural network architectures. A further study with more focus on the efficiency of the model is therefore suggested. For example, the training time of the model should be taken into consideration in the imputation and prediction tasks.

10.2.2 Use of Both Structured and Unstructured Data

The MIMIC-III and eICU are large medical databases comprising all information relating to patients admitted to intensive care units. This work mainly uses structured data (e.g., multivariate clinical times series and ICD-9 diagnosis codes) as input for the imputation and prediction tasks. This may lead to sub-optimal prediction performance. Further investigation and experimentation into more informative details, such as free text diagnosis (i.e., unstructured data), are strongly recommended. In terms of directions for future research, further work could expand deep imputation-prediction networks into a multi-modal fusion network architecture to handle both structured and unstructured data.

10.2.3 Secondary Healthcare Applications

In our case studies, the experimental data were selected on the basis of [53, 54], i.e., the data from the first 24 and 48 hours after ICU admission. However, due to variations in patient conditions and treatment needs, real-time mortality risk scores should be generated for patients, which also contribute to more proactive intervention generation. In terms of directions for future research, deep imputation-prediction networks could be reconstructed and introduced into real-time decision-making.

This work has only taken into account the irregularity of EHR data in the context of in-hospital mortality risk prediction. Since deep neural network architectures are a result

of intuition and trial-and-error, the performance of other case studies would allow us to verify the robustness of the proposed deep imputation-prediction networks. The same experimental setup is likely to be effective for other case studies, such as regression-based hospital length of stay prediction, binary classification-based readmission prediction, and multi-classification-based phenotype prediction. Further studies regarding the comparison of imputed values under different prediction tasks would be worthwhile.

Further experiments could also be conducted to determine the safety and efficacy of the proposed deep imputation-prediction networks on other publicly available EHR databases, such as the Medical Information Mart for Intensive Care IV (MIMIC-IV) database [367] and Medical Information Mart for Intensive Care IV Emergency Department (MIMIC-IV-ED) database [368].

10.2.4 Individual Fairness on Similarity Computing

Some of the distribution of patient-specific characteristics, such as age, sex, and ethnicity, are imbalanced, which, although considered, is not thoroughly analyzed when computing patient similarity in the proposed Contrastive Neural Network, Contrastive Graph Similarity Network, and Multi-Graph Neural Network. These characteristics are sensitive attributes that may lead to bias in imputation and prediction results. In addition, there might still be many "unseen" attributes that could significantly affect the model training process. For example, the missing rates among vital signs vary significantly, ranging from less than 30% (e.g., diastolic blood pressure and heart rate) to exceeding 90% (e.g., capillary refill rate and pH), causing concerns about the fairness of the patient similarity model. Therefore, effort should be made to develop patient models that minimize unfairness in the future.

10.2.5 Transfer Learning, Few-shot Learning, and Zero-shot Learning

The proposed deep imputation-prediction networks may not be applicable to all learning settings. For example, this includes transfer learning, few-shot learning, and zero-shot learning required for real application scenarios. Transfer learning aims to transfer knowledge obtained from one task to another related task to improve performance. For example, mortality prediction windows are set to 24 hours and 48 hours after admission. Accordingly, the complete data matrix obtained from the former task can be extracted as input to the latter task. Few-shot learning aims to make predictions for new classes based on just a few examples of labeled data, while for zero-shot learning, there is no labeled data

available for new classes. Considerably more work will need to be done to incorporate these deep imputation-prediction networks into transfer learning, few-shot learning, and zero-shot learning settings.

10.2.6 Explainable Neural Network Architecture

Despite its great success, significant barriers remain when it comes to the adoption of deep imputation-prediction networks for real-world applications. A key issue is that deep interpolation prediction networks remain "black-box" in terms of their architectures. In other words, these "black-box" models are built from data by algorithms, which means humans, even those who design them, cannot understand how the models combine input features to make decisions. Therefore, the learning process of these models is non-transparent or partially transparent, which leads to difficulties in the interpretation of results. Further research should be carried out to generate more interpretable representations (i.e., feature vectors or matrices derived from deep learning models that can be further visualized), which link to available information and lead to better user acceptance.

10.3 Summary

The aim of this thesis was to investigate and develop new risk prediction models for addressing the irregularity of EHR data and predicting patients' health risks. This aim was addressed by proposing six deep imputation-prediction models using a combination of real-world EHR databases and case studies. The strength of these models lies in their ability to present transparency and interpretability of the model decision process and provide the estimation of epistemic and aleatoric uncertainties of the model decisions. This research made novel contributions to the improvement of methodologies for handling irregularity of EHR data in the context of health risk prediction. These methodologies are potentially applicable to other medical applications such as hospital length of stay prediction and phenotype classification.

Bibliography

[1] Yuxi Liu, Shaowen Qin, Antonio Jimeno Yepes, Wei Shao, Zhenhao Zhang, and Flora D Salim. Integrated convolutional and recurrent neural networks for health risk prediction using patient journey data with many missing values. In *2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 1658–1663. IEEE, 2022.

[2] Yuxi Liu, Zhenhao Zhang, Shaowen Qin, Flora D Salim, Antonio Jimeno Yepes, and Jun Shen. Hypergraph convolutional networks for fine-grained icu patient similarity analysis and risk prediction. *arXiv preprint arXiv:2308.12575*, 2023.

[3] Yuxi Liu, Zhenhao Zhang, Campbell Thompson, Richard Leibbrandt, Shaowen Qin, and Antonio Jimeno Yepes. Stacked attention-based networks for accurate and interpretable health risk prediction. In *2023 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2023.

[4] Yuxi Liu, Zhenhao Zhang, and Shaowen Qin. Neuralhmm: A deep markov network for health risk prediction using electronic health records. In *2023 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2023.

[5] Yuxi Liu, Zhenhao Zhang, Antonio Jimeno Yepes, and Flora D Salim. Modeling long-term dependencies and short-term correlations in patient journey data with temporal attention networks for health prediction. In *Proceedings of the 13th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, pages 1–10, 2022.

[6] Yuxi Liu and Shaowen Qin. An interpretable machine learning approach for predicting hospital length of stay and readmission. In *Advanced Data Mining and Ap-*

plications: 17th International Conference, ADMA 2021, Sydney, NSW, Australia, February 2–4, 2022, Proceedings, Part I, pages 73–85. Springer, 2022.

[7] Yuxi Liu and Shaowen Qin. Hospital readmission prediction via personalized feature learning and embedding: A novel deep learning framework. In *Advances and Trends in Artificial Intelligence. Theory and Practices in Artificial Intelligence: 35th International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2022, Kitakyushu, Japan, July 19–22, 2022, Proceedings*, pages 89–100. Springer, 2022.

[8] Yuxi Liu, Shaowen Qin, and Zhenhao Zhang. Epidemic modeling of the spatiotemporal spread of covid-19 over an intercity population mobility network. In *Advances and Trends in Artificial Intelligence. Theory and Practices in Artificial Intelligence: 35th International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2022, Kitakyushu, Japan, July 19–22, 2022, Proceedings*, pages 147–159. Springer, 2022.

[9] Jay A Pandit, Jennifer M Radin, Giorgio Quer, and Eric J Topol. Smartphone apps in the covid-19 pandemic. *Nature Biotechnology*, 40(7):1013–1022, 2022.

[10] Muhammad Nazrul Islam, Iyolita Islam, Kazi Md Munim, and AKM Najmul Islam. A review on the mobile applications developed for covid-19: an exploratory analysis. *Ieee Access*, 8:145601–145610, 2020.

[11] Sheikh MA Iqbal, Imadeldin Mahgoub, E Du, Mary Ann Leavitt, and Waseem Asghar. Advances in healthcare wearable devices. *NPJ Flexible Electronics*, 5(1):9, 2021.

[12] Kyeonghye Guk, Gaon Han, Jaewoo Lim, Keunwon Jeong, Taejoon Kang, Eun-Kyung Lim, and Juyeon Jung. Evolution of wearable devices with real-time disease monitoring for personalized healthcare. *Nanomaterials*, 9(6):813, 2019.

[13] Malcolm Clarke, Joost de Folter, Vivek Verma, and Hulya Gokalp. Interoperable end-to-end remote patient monitoring platform based on ieee 11073 phd and zigbee health care profile. *IEEE Transactions on Biomedical Engineering*, 65(5):1014–1025, 2017.

[14] Farrukh M Koraishy and Rajeev Rohatgi. Telenephrology: an emerging platform for delivering renal health care. *American Journal of Kidney Diseases*, 76(3):417–426, 2020.

[15] Wan-Shiou Yang and San-Yih Hwang. A process-mining framework for the detection of healthcare fraud and abuse. *Expert Systems with Applications*, 31(1):56–68, 2006.

[16] Hossein Joudaki, Arash Rashidian, Behrouz Minaei-Bidgoli, Mahmood Mahmoodi, Bijan Geraili, Mahdi Nasiri, and Mohammad Arab. Using data mining to detect health care fraud and abuse: a review of literature. *Global journal of health science*, 7(1):194, 2015.

[17] Hossein Joudaki, Arash Rashidian, Behrouz Minaei-Bidgoli, Mahmood Mahmoodi, Bijan Geraili, Mahdi Nasiri, and Mohammad Arab. Improving fraud and abuse detection in general physician claims: a data mining study. *International journal of health policy and management*, 5(3):165, 2016.

[18] Michael P Mazanetz, Robert J Marmon, Catherine BT Reisser, and Inaki Morao. Drug discovery applications for knime: an open source data mining platform. *Current topics in medicinal chemistry*, 12(18):1965–1979, 2012.

[19] Snezana Agatonovic-Kustrin and David Morton. Data mining in drug discovery and design. In *Artificial neural network for drug design, delivery and disposition*, pages 181–193. Elsevier, 2016.

[20] Jessica Vamathevan, Dominic Clark, Paul Czodrowski, Ian Dunham, Edgardo Ferran, George Lee, Bin Li, Anant Madabhushi, Parantu Shah, Michaela Spitzer, et al. Applications of machine learning in drug discovery and development. *Nature reviews Drug discovery*, 18(6):463–477, 2019.

[21] Divya Chauhan and Varun Jaiswal. An efficient data mining classification approach for detecting lung cancer disease. In *2016 International Conference on Communication and Electronics Systems (ICCES)*, pages 1–8. IEEE, 2016.

[22] Shaicy P Shaji et al. Predictionand diagnosis of heart disease patients using data mining technique. In *2019 international conference on communication and signal processing (ICCSP)*, pages 0848–0852. IEEE, 2019.

[23] Ana Pinto, Diana Ferreira, Cristiana Neto, António Abelha, and José Machado. Data mining to predict early stage chronic kidney disease. *Procedia Computer Science*, 177:562–567, 2020.

[24] Syed Umar Amin, Kavita Agarwal, and Rizwan Beg. Data mining in clinical decision support systems for diagnosis, prediction and treatment of heart disease. *In-*

ternational Journal of Advanced Research in Computer Engineering & Technology (IJARCET), 2(1):218–223, 2013.

[25] Lu-Yen A Chen and Tonks N Fawcett. Using data mining strategies in clinical decision making: a literature review. *CIN: computers, informatics, nursing*, 34(10):448–454, 2016.

[26] Bunyamin Ozaydin, J Michael Hardin, and David C Chhieng. Data mining and clinical decision support systems. *Clinical Decision Support Systems: Theory and Practice*, pages 45–68, 2016.

[27] Alexandros C Dimopoulos, Mara Nikolaidou, Francisco Félix Caballero, Worrawat Engchuan, Albert Sanchez-Niubo, Holger Arndt, José Luis Ayuso-Mateos, Josep Maria Haro, Somnath Chatterji, Ekavi N Georgousopoulou, et al. Machine learning methodologies versus cardiovascular risk scores, in predicting disease risk. *BMC medical research methodology*, 18:1–11, 2018.

[28] Fenglong Ma, Jing Gao, Qiuling Suo, Quanzeng You, Jing Zhou, and Aidong Zhang. Risk prediction on electronic health records with prior medical knowledge. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1910–1919, 2018.

[29] Xianli Zhang, Buyue Qian, Shilei Cao, Yang Li, Hang Chen, Yefeng Zheng, and Ian Davidson. Inprem: An interpretable and trustworthy predictive model for healthcare. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 450–460, 2020.

[30] Junyu Luo, Muchao Ye, Cao Xiao, and Fenglong Ma. Hitanet: Hierarchical time-aware attention networks for risk prediction on electronic health records. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 647–656, 2020.

[31] Tomohisa Seki, Yoshimasa Kawazoe, and Kazuhiko Ohe. Machine learning-based prediction of in-hospital mortality using admission laboratory data: A retrospective, single-site study using electronic health record data. *PloS one*, 16(2):e0246640, 2021.

[32] Yilmazcan Ozyurt, Mathias Kraus, Tobias Hatt, and Stefan Feuerriegel. Attdmm: an attentive deep markov model for risk scoring in intensive care units. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 3452–3462, 2021.

[33] Fan Yang, Jian Zhang, Wanyi Chen, Yongxuan Lai, Ying Wang, and Quan Zou. Deepmpm: a mortality risk prediction model using longitudinal ehr data. *BMC bioinformatics*, 23(1):423, 2022.

[34] Guodong Du, Jia Zhang, Zhiming Luo, Fenglong Ma, Lei Ma, and Shaozi Li. Joint imbalanced classification and feature selection for hospital readmissions. *Knowledge-Based Systems*, 200:106020, 2020.

[35] Guodong Du, Jia Zhang, Fenglong Ma, Min Zhao, Yaojin Lin, and Shaozi Li. Towards graph-based class-imbalance learning for hospital readmission. *Expert Systems with Applications*, 176:114791, 2021.

[36] Katherine Shi, Vy Ho, Joanna J Song, Katelyn Bechler, and Jonathan H Chen. Predicting unplanned 7-day intensive care unit readmissions with machine learning models for improved discharge risk assessment. In *AMIA Annual Symposium Proceedings*, volume 2022, page 446. American Medical Informatics Association, 2022.

[37] Yu Cheng, Fei Wang, Ping Zhang, and Jianying Hu. Risk prediction with electronic health records: A deep learning approach. In *Proceedings of the 2016 SIAM International Conference on Data Mining*, pages 432–440. SIAM, 2016.

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

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

[40] Wei Cao, Dong Wang, Jian Li, Hao Zhou, Lei Li, and Yitan Li. Brits: Bidirectional recurrent imputation for time series. *Advances in neural information processing systems*, 31, 2018.

[41] Jinsung Yoon, William R Zame, and Mihaela van der Schaar. Estimating missing data in temporal data streams using multi-directional recurrent neural networks. *IEEE Transactions on Biomedical Engineering*, 66(5):1477–1490, 2018.

[42] Satya Narayan Shukla and Benjamin M Marlin. Interpolation-prediction networks for irregularly sampled time series. *arXiv preprint arXiv:1909.07782*, 2019.

[43] Yonghong Luo, Xiangrui Cai, Ying Zhang, Jun Xu, et al. Multivariate time series imputation with generative adversarial networks. *Advances in neural information processing systems*, 31, 2018.

[44] Yonghong Luo, Ying Zhang, Xiangrui Cai, and Xiaojie Yuan. E2gan: End-to-end generative adversarial network for multivariate time series imputation. In *Proceedings of the 28th international joint conference on artificial intelligence*, pages 3094–3100. AAAI Press, 2019.

[45] Mehak Gupta, Thao-Ly T Phan, H Timothy Bunnell, and Rahmatollah Beheshti. Concurrent imputation and prediction on ehr data using bi-directional gans: Bi-gans for ehr imputation and prediction. In *Proceedings of the 12th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics*, pages 1–9, 2021.

[46] Eunkyu Oh, Taehun Kim, Yunhu Ji, and Sushil Khyalia. Sting: Self-attention based time-series imputation networks using gan. In *2021 IEEE International Conference on Data Mining (ICDM)*, pages 1264–1269. IEEE, 2021.

[47] Qingjian Ni and Xuehan Cao. Mbgan: An improved generative adversarial network with multi-head self-attention and bidirectional rnn for time series imputation. *Engineering Applications of Artificial Intelligence*, 115:105232, 2022.

[48] Xiaoye Miao, Yangyang Wu, Jun Wang, Yunjun Gao, Xudong Mao, and Jianwei Yin. Generative semi-supervised learning for multivariate time series imputation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 8983–8991, 2021.

[49] Zhenkun Shi, Sen Wang, Lin Yue, Lixin Pang, Xianglin Zuo, Wanli Zuo, and Xue Li. Deep dynamic imputation of clinical time series for mortality prediction. *Information Sciences*, 579:607–622, 2021.

[50] Ying Zhang, Baohang Zhou, Xiangrui Cai, Wenya Guo, Xiaoke Ding, and Xiaojie Yuan. Missing value imputation in multivariate time series with end-to-end generative adversarial networks. *Information Sciences*, 551:67–82, 2021.

[51] Yurim Lee, Eunji Jun, Jaehun Choi, and Heung-Il Suk. Multi-view integrative attention-based deep representation learning for irregular clinical time-series data. *IEEE Journal of Biomedical and Health Informatics*, 26(8):4270–4280, 2022.

[52] Kejing Yin and William K Cheung. Context-aware imputation for clinical time series. In *2019 IEEE International Conference on Healthcare Informatics (ICHI)*, pages 1–3. IEEE, 2019.

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

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

[55] Yeo-Jin Kim and Min Chi. Temporal belief memory: Imputing missing data during rnn training. In *In Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI-2018)*, 2018.

[56] Da Xu, Jessica Qiuhua Sheng, Paul Jen-Hwa Hu, Ting-Shuo Huang, and Chih-Chin Hsu. A deep learning–based unsupervised method to impute missing values in patient records for improved management of cardiovascular patients. *IEEE Journal of Biomedical and Health Informatics*, 25(6):2260–2272, 2020.

[57] Shuo Yang, Minjing Dong, Yunhe Wang, and Chang Xu. Adversarial recurrent time series imputation. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[58] A Yarkın Yıldız, Emirhan Koç, and Aykut Koç. Multivariate time series imputation with transformers. *IEEE Signal Processing Letters*, 29:2517–2521, 2022.

[59] Brian J Wells, Kevin M Chagin, Amy S Nowacki, and Michael W Kattan. Strategies for handling missing data in electronic health record derived data. *Egems*, 1(3), 2013.

[60] Qingxiong Tan, Mang Ye, Baoyao Yang, Siqi Liu, Andy Jinhua Ma, Terry Cheuk-Fung Yip, Grace Lai-Hung Wong, and PongChi Yuen. Data-gru: Dual-attention time-aware gated recurrent unit for irregular multivariate time series. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 930–937, 2020.

[61] Spyridon Mouselinos, Kyriakos Polymenakos, Antonis Nikitakis, and Konstantinos Kyriakopoulos. Main: multihead-attention imputation networks. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2021.

[62] Yuxi Liu, Shaowen Qin, Zhenhao Zhang, and Wei Shao. Compound density networks for risk prediction using electronic health records. In *2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 1078–1085. IEEE, 2022.

[63] Lifeng Shen, Qianli Ma, and Sen Li. End-to-end time series imputation via residual short paths. In *Asian conference on machine learning*, pages 248–263. PMLR, 2018.

[64] Ruoxi Yu, Yali Zheng, Ruikai Zhang, Yuqi Jiang, and Carmen CY Poon. Using a multi-task recurrent neural network with attention mechanisms to predict hospital mortality of patients. *IEEE journal of biomedical and health informatics*, 24(2):486–492, 2019.

[65] Ahmad Wisnu Mulyadi, Eunji Jun, and Heung-Il Suk. Uncertainty-aware variational-recurrent imputation network for clinical time series. *IEEE Transactions on Cybernetics*, 52(9):9684–9694, 2021.

[66] Chris Fifty, Ehsan Amid, Zhe Zhao, Tianhe Yu, Rohan Anil, and Chelsea Finn. Efficiently identifying task groupings for multi-task learning. *Advances in Neural Information Processing Systems*, 34:27503–27516, 2021.

[67] David Mueller, Nicholas Andrews, and Mark Dredze. Do text-to-text multi-task learners suffer from task conflict? *arXiv preprint arXiv:2212.06645*, 2022.

[68] Gencer Sumbul and Begüm Demir. Plasticity-stability preserving multi-task learning for remote sensing image retrieval. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–16, 2022.

[69] Stefania Montani, Luigi Portinale, Giorgio Leonardi, Riccardo Bellazzi, and Roberto Bellazzi. Case-based retrieval to support the treatment of end stage renal failure patients. *Artificial Intelligence in Medicine*, 37(1):31–42, 2006.

[70] Mohammed Saeed and Roger Mark. A novel method for the efficient retrieval of similar multiparameter physiologic time series using wavelet-based symbolic representations. In *AMIA Annual Symposium Proceedings*, volume 2006, page 679. American Medical Informatics Association, 2006.

[71] Jimeng Sun, Daby Sow, Jianying Hu, and Shahram Ebadollahi. A system for mining temporal physiological data streams for advanced prognostic decision support. In *2010 IEEE International Conference on Data Mining*, pages 1061–1066. IEEE, 2010.

[72] Zheng Jia, Xian Zeng, Huilong Duan, Xudong Lu, and Haomin Li. A patient-similarity-based model for diagnostic prediction. *International journal of medical informatics*, 135:104073, 2020.

[73] Yuxi Liu, Zhenhao Zhang, and Shaowen Qin. Deep imputation-prediction networks for health risk prediction using electronic health records. In *2023 International Joint Conference on Neural Networks (IJCNN)*, pages 1–9. IEEE, 2023.

[74] Yuxi Liu, Zhenhao Zhang, Shaowen Qin, Flora D Salim, and Antonio Jimeno Yepes. Contrastive learning-based imputation-prediction networks for in-hospital mortality risk modeling using ehrs. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 428–443. Springer, 2023.

[75] Zhongheng Zhang. Missing data imputation: focusing on single imputation. *Annals of translational medicine*, 4(1), 2016.

[76] Rebecca R Andridge and Roderick J Little. The use of sample weights in hot deck imputation. *Journal of Official Statistics*, 25(1):21, 2009.

[77] Rebecca R Andridge and Roderick JA Little. A review of hot deck imputation for survey non-response. *International statistical review*, 78(1):40–64, 2010.

[78] Dieter William Joenssen and Udo Bankhofer. Hot deck methods for imputing missing data: the effects of limiting donor usage. In *Machine Learning and Data Mining in Pattern Recognition: 8th International Conference, MLDM 2012, Berlin, Germany, July 13-20, 2012. Proceedings 8*, pages 63–75. Springer, 2012.

[79] Adrian Mander and David Clayton. Weighted hotdeck imputation. *Statistical Software Components, Boston College Department of Economics, http://fmwww. bc. edu/repec/bocode/w/whotdeck. pdf*, 2003.

[80] Jae Kwang Kim and Wayne Fuller. Fractional hot deck imputation. *Biometrika*, 91(3):559–578, 2004.

[81] Jae Kwang Kim and Wayne Fuller. Hot deck imputation for multivariate missing data. In *Proceedings 59th ISI world statistics congress*, pages 25–30, 2013.

[82] Jongho Im, Jae-Kwang Kim, and Wayne A Fuller. Two-phase sampling approach to fractional hot deck imputation. In *Proceedings of the Survey Research Methods Section*, pages 1030–1043, 2015.

[83] Wei-Chao Lin and Chih-Fong Tsai. Missing value imputation: a review and analysis of the literature (2006–2017). *Artificial Intelligence Review*, 53:1487–1509, 2020.

[84] Leah H Rubin, Katie Witkiewitz, Justin St Andre, and Steve Reilly. Methods for handling missing data in the behavioral neurosciences: Don't throw the baby rat out

with the bath water. *Journal of Undergraduate Neuroscience Education*, 5(2):A71, 2007.

[85] James Honaker, Gary King, and Matthew Blackwell. Amelia ii: A program for missing data. *Journal of statistical software*, 45:1–47, 2011.

[86] Olivier Delalleau, Aaron Courville, and Yoshua Bengio. Efficient em training of gaussian mixtures with missing data. *arXiv preprint arXiv:1209.0521*, 2012.

[87] Aman Kataria and MD Singh. A review of data classification using k-nearest neighbour algorithm. *International Journal of Emerging Technology and Advanced Engineering*, 3(6):354–360, 2013.

[88] Olga Troyanskaya, Michael Cantor, Gavin Sherlock, Pat Brown, Trevor Hastie, Robert Tibshirani, David Botstein, and Russ B Altman. Missing value estimation methods for dna microarrays. *Bioinformatics*, 17(6):520–525, 2001.

[89] Jason Van Hulse and Taghi M Khoshgoftaar. Incomplete-case nearest neighbor imputation in software measurement data. *Information Sciences*, 259:596–610, 2014.

[90] Lorenzo Beretta and Alessandro Santaniello. Nearest neighbor imputation algorithms: a critical evaluation. *BMC medical informatics and decision making*, 16(3):197–208, 2016.

[91] Jianglin Huang, Jacky Wai Keung, Federica Sarro, Yan-Fu Li, Yuen-Tak Yu, WK Chan, and Hongyi Sun. Cross-validation based k nearest neighbor imputation for software quality datasets: an empirical study. *Journal of Systems and Software*, 132:226–252, 2017.

[92] Justin Y Lee and Mark P Styczynski. Ns-knn: a modified k-nearest neighbors approach for imputing metabolomics data. *Metabolomics*, 14:1–12, 2018.

[93] Ching-Hsue Cheng, Chia-Pang Chan, and Yu-Jheng Sheu. A novel purity-based k nearest neighbors imputation method and its application in financial distress prediction. *Engineering Applications of Artificial Intelligence*, 81:283–299, 2019.

[94] Miriam Seoane Santos, Pedro Henriques Abreu, Szymon Wilk, and João Santos. How distance metrics influence missing data imputation with k-nearest neighbours. *Pattern Recognition Letters*, 136:111–119, 2020.

[95] Xiaoxin Zhang, Yuping Zhao, and Li Zou. Optimum tcm codes design for gaussian channels by considering both euclidean and hamming distances. In *2009 IEEE International Conference on Communications*, pages 1–5. IEEE, 2009.

[96] Thangasamy Jeyapoovan and M Murugan. Surface roughness classification using image processing. *Measurement*, 46(7):2065–2072, 2013.

[97] Siddarth Hari, Foteini Agrafioti, and Dimitrios Hatzinakos. Design of a hamming-distance classifier for ecg biometrics. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3009–3012. IEEE, 2013.

[98] Kittipong Chomboon, Pasapitch Chujai, Pongsakorn Teerarassamee, Kittisak Kerdprasop, and Nittaya Kerdprasop. An empirical study of distance metrics for k-nearest neighbor algorithm. In *Proceedings of the 3rd international conference on industrial application engineering*, volume 2, 2015.

[99] Zishun Liu, Zhenxi Li, Juyong Zhang, and Ligang Liu. Euclidean and hamming embedding for image patch description with convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 72–78, 2016.

[100] Haneen Arafat Abu Alfeilat, Ahmad BA Hassanat, Omar Lasassmeh, Ahmad S Tarawneh, Mahmoud Bashir Alhasanat, Hamzeh S Eyal Salman, and VB Surya Prasath. Effects of distance measure choice on k-nearest neighbor classifier performance: a review. *Big data*, 7(4):221–248, 2019.

[101] Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001.

[102] Adam Pantanowitz and Tshilidzi Marwala. Missing data imputation through the use of the random forest algorithm. In *Advances in computational intelligence*, pages 53–62. Springer, 2009.

[103] Md Geaur Rahman and Md Zahidul Islam. A decision tree-based missing value imputation technique for data pre-processing. In *The 9th Australasian Data Mining Conference: AusDM 2011*, pages 41–50. Australian Computer Society Inc, 2011.

[104] Daniel J Stekhoven and Peter Bühlmann. Missforest—non-parametric missing value imputation for mixed-type data. *Bioinformatics*, 28(1):112–118, 2012.

[105] Md Geaur Rahman and Md Zahidul Islam. Missing value imputation using decision trees and decision forests by splitting and merging records: Two novel techniques. *Knowledge-Based Systems*, 53:51–65, 2013.

[106] Pasha Khosravi, Antonio Vergari, YooJung Choi, Yitao Liang, and Guy Van den Broeck. Handling missing data in decision trees: A probabilistic approach. *arXiv preprint arXiv:2006.16341*, 2020.

[107] Shangzhi Hong and Henry S Lynn. Accuracy of random-forest-based imputation of missing data in the presence of non-normality, non-linearity, and interaction. *BMC medical research methodology*, 20(1):1–12, 2020.

[108] Xiao Xu, Xiaoshuang Liu, Yanni Kang, Xian Xu, Junmei Wang, Yuyao Sun, Quanhe Chen, Xiaoyu Jia, Xinyue Ma, Xiaoyan Meng, et al. A multi-directional approach for missing value estimation in multivariate time series clinical data. *Journal of Healthcare Informatics Research*, 4(4):365–382, 2020.

[109] Xinmeng Zhang, Chao Yan, Cheng Gao, Bradley A Malin, and You Chen. Predicting missing values in medical data via xgboost regression. *Journal of healthcare informatics research*, 4:383–394, 2020.

[110] J Ross Quinlan. Improved use of continuous attributes in c4. 5. *Journal of artificial intelligence research*, 4:77–90, 1996.

[111] Tapio Schneider. Analysis of incomplete climate data: Estimation of mean values and covariance matrices and imputation of missing values. *Journal of climate*, 14(5):853–871, 2001.

[112] Zahidul Islam and Helen Giggins. Knowledge discovery through sysfor: a systematically developed forest of multiple decision trees. In *Proceedings of the Ninth Australasian Data Mining Conference-Volume 121*, pages 195–204, 2011.

[113] Elena M Hernández-Pereira, Diego Álvarez-Estévez, and Vicente Moret-Bonillo. Automatic classification of respiratory patterns involving missing data imputation techniques. *Biosystems Engineering*, 138:65–76, 2015.

[114] Utkarsh Mital, Dipankar Dwivedi, James B Brown, Boris Faybishenko, Scott L Painter, and Carl I Steefel. Sequential imputation of missing spatio-temporal precipitation data using random forests. *Frontiers in Water*, 2:20, 2020.

[115] Runhai Feng, Dario Grana, and Niels Balling. Imputation of missing well log data by random forest and its uncertainty analysis. *Computers & Geosciences*, 152:104763, 2021.

[116] Xin Jing, Jungang Luo, Jingmin Wang, Ganggang Zuo, and Na Wei. A multi-imputation method to deal with hydro-meteorological missing values by integrating chain equations and random forest. *Water Resources Management*, 36(4):1159–1173, 2022.

[117] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30, 2017.

[118] Yuan Luo, Peter Szolovits, Anand S Dighe, and Jason M Baron. 3d-mice: integration of cross-sectional and longitudinal imputation for multi-analyte longitudinal clinical data. *Journal of the American Medical Informatics Association*, 25(6):645–653, 2018.

[119] Stef Van Buuren and Karin Groothuis-Oudshoorn. mice: Multivariate imputation by chained equations in r. *Journal of statistical software*, 45:1–67, 2011.

[120] Melissa J Azur, Elizabeth A Stuart, Constantine Frangakis, and Philip J Leaf. Multiple imputation by chained equations: what is it and how does it work? *International journal of methods in psychiatric research*, 20(1):40–49, 2011.

[121] Anoop D Shah, Jonathan W Bartlett, James Carpenter, Owen Nicholas, and Harry Hemingway. Comparison of random forest and parametric imputation models for imputing missing data using mice: a caliber study. *American journal of epidemiology*, 179(6):764–774, 2014.

[122] Zhongheng Zhang. Multiple imputation with multivariate imputation by chained equation (mice) package. *Annals of translational medicine*, 4(2), 2016.

[123] Matthieu Resche-Rigon and Ian R White. Multiple imputation by chained equations for systematically and sporadically missing multilevel data. *Statistical methods in medical research*, 27(6):1634–1649, 2018.

[124] Lauren J Beesley and Jeremy MG Taylor. A stacked approach for chained equations multiple imputation incorporating the substantive model. *Biometrics*, 77(4):1342–1354, 2021.

[125] Shahidul Islam Khan and Abu Sayed Md Latiful Hoque. Sice: an improved missing data imputation technique. *Journal of big Data*, 7(1):1–21, 2020.

[126] Maritza Mera-Gaona, Ursula Neumann, Rubiel Vargas-Canas, and Diego M López. Evaluating the impact of multivariate imputation by mice in feature selection. *Plos one*, 16(7):e0254720, 2021.

[127] Jiang Li, Xiaowei S Yan, Durgesh Chaudhary, Venkatesh Avula, Satish Mudiganti, Hannah Husby, Shima Shahjouei, Ardavan Afshar, Walter F Stewart, Mohammed Yeasin, et al. Imputation of missing values for electronic health record laboratory data. *NPJ digital medicine*, 4(1):147, 2021.

[128] Hannah S Laqueur, Aaron B Shev, and Rose MC Kagawa. Supermice: An ensemble machine learning approach to multiple imputation by chained equations. *American journal of epidemiology*, 191(3):516–525, 2022.

[129] Dianbo Liu, Won-Yong Shin, Eli Sprecher, Kathleen Conroy, Omar Santiago, Gal Wachtel, and Mauricio Santillana. Machine learning approaches to predicting no-shows in pediatric medical appointment. *NPJ digital medicine*, 5(1):50, 2022.

[130] Jakob F Mathiszig-Lee, Finneas JR Catling, S Ramani Moonesinghe, and Stephen J Brett. Highlighting uncertainty in clinical risk prediction using a model of emergency laparotomy mortality risk. *npj Digital Medicine*, 5(1):70, 2022.

[131] Kristiaan Pelckmans, Jos De Brabanter, Johan AK Suykens, and Bart De Moor. Handling missing values in support vector machine classifiers. *Neural Networks*, 18(5-6):684–692, 2005.

[132] Feng Honghai, Chen Guoshun, Yin Cheng, Yang Bingru, and Chen Yumei. A svm regression based approach to filling in missing values. In *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, pages 581–587. Springer, 2005.

[133] Yang Zhang and Yuncai Liu. Data imputation using least squares support vector machines in urban arterial streets. *IEEE Signal Processing Letters*, 16(5):414–417, 2009.

[134] Banghua Yang, Davy Janssens, Da Ruan, Mario Cools, Tom Bellemans, and Geert Wets. A data imputation method with support vector machines for activity-based transportation models. In *Foundations of Intelligent Systems: Proceedings of the Sixth International Conference on Intelligent Systems and Knowledge Engineering, Shanghai, China, Dec 2011 (ISKE2011)*, pages 249–257. Springer, 2012.

[135] Mengeheng Zhu and Hong Shi. A novel support vector machine algorithm for missing data. In *Proceedings of the 2nd International Conference on Innovation in Artificial Intelligence*, pages 48–53, 2018.

[136] Caterina Penone, Ana D Davidson, Kevin T Shoemaker, Moreno Di Marco, Carlo Rondinini, Thomas M Brooks, Bruce E Young, Catherine H Graham, and Gabriel C Costa. Imputation of missing data in life-history trait datasets: Which approach performs the best? *Methods in Ecology and Evolution*, 5(9):961–970, 2014.

[137] Akbar K Waljee, Ashin Mukherjee, Amit G Singal, Yiwei Zhang, Jeffrey Warren, Ulysses Balis, Jorge Marrero, Ji Zhu, and Peter DR Higgins. Comparison of imputation methods for missing laboratory data in medicine. *BMJ open*, 3(8):e002847, 2013.

[138] Anil Jadhav, Dhanya Pramod, and Krishnan Ramanathan. Comparison of performance of data imputation methods for numeric dataset. *Applied Artificial Intelligence*, 33(10):913–933, 2019.

[139] Marianne Riksheim Stavseth, Thomas Clausen, and Jo Røislien. How handling missing data may impact conclusions: A comparison of six different imputation methods for categorical questionnaire data. *SAGE open medicine*, 7:2050312118822912, 2019.

[140] Hyun Ahn, Kyunghee Sun, and K Kim. Comparison of missing data imputation methods in time series forecasting. *Computers, Materials & Continua*, 70(1):767–779, 2022.

[141] Abhinav Atla, Rahul Tada, Victor Sheng, and Naveen Singireddy. Sensitivity of different machine learning algorithms to noise. *Journal of Computing Sciences in Colleges*, 26(5):96–103, 2011.

[142] Markus J Ankenbrand, Liliia Shainberg, Michael Hock, David Lohr, and Laura M Schreiber. Sensitivity analysis for interpretation of machine learning based segmentation models in cardiac mri. *BMC Medical Imaging*, 21:1–8, 2021.

[143] Shahla Parveen and Phil Green. Speech recognition with missing data using recurrent neural nets. *Advances in Neural Information Processing Systems*, 14, 2001.

[144] Han-Gyu Kim, Gil-Jin Jang, Ho-Jin Choi, Minho Kim, Young-Won Kim, and Jaehun Choi. Recurrent neural networks with missing information imputation for medical examination data prediction. In *2017 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pages 317–323. IEEE, 2017.

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

[146] Hongwu Yuan, Guoming Xu, Zijian Yao, Ji Jia, and Yiwen Zhang. Imputation of missing data in time series for air pollutants using long short-term memory recurrent neural networks. In *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*, pages 1293–1300, 2018.

[147] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[148] Inci M Baytas, Cao Xiao, Xi Zhang, Fei Wang, Anil K Jain, and Jiayu Zhou. Patient subtyping via time-aware lstm networks. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 65–74, 2017.

[149] Yuan Zhang. Attain: Attention-based time-aware lstm networks for disease progression modeling. In *In Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI-2019), pp. 4369-4375, Macao, China.*, 2019.

[150] Liantao Ma, Chaohe Zhang, Yasha Wang, Wenjie Ruan, Jiangtao Wang, Wen Tang, Xinyu Ma, Xin Gao, and Junyi Gao. Concare: Personalized clinical feature embedding via capturing the healthcare context. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 833–840, 2020.

[151] Qiuling Suo, Liuyi Yao, Guangxu Xun, Jianhui Sun, and Aidong Zhang. Recurrent imputation for multivariate time series with missing values. In *2019 IEEE International Conference on Healthcare Informatics (ICHI)*, pages 1–3. IEEE, 2019.

[152] Tiantian Feng and Shrikanth Narayanan. Imputing missing data in large-scale multivariate biomedical wearable recordings using bidirectional recurrent neural networks with temporal activation regularization. In *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 2529–2534. IEEE, 2019.

[153] Zhiyong Cui, Ruimin Ke, Ziyuan Pu, and Yinhai Wang. Stacked bidirectional and unidirectional lstm recurrent neural network for forecasting network-wide traffic state with missing values. *Transportation Research Part C: Emerging Technologies*, 118:102674, 2020.

[154] Kaname Kojima, Shu Tadaka, Fumiki Katsuoka, Gen Tamiya, Masayuki Yamamoto, and Kengo Kinoshita. A genotype imputation method for de-identified haplotype reference information by using recurrent neural network. *PLoS Computational Biology*, 16(10):e1008207, 2020.

[155] Oleg Ivanov, Michael Figurnov, and Dmitry Vetrov. Variational autoencoder with arbitrary conditioning. *arXiv preprint arXiv:1806.02382*, 2018.

[156] Luan Tran, Xiaoming Liu, Jiayu Zhou, and Rong Jin. Missing modalities imputation via cascaded residual autoencoder. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1405–1414, 2017.

[157] Natasha Jaques, Sara Taylor, Akane Sano, and Rosalind Picard. Multimodal autoencoder: A deep learning approach to filling in missing sensor data and enabling better mood prediction. In *2017 Seventh International Conference on Affective Computing and Intelligent Interaction (ACII)*, pages 202–208. IEEE, 2017.

[158] Yao Jia, Chongyu Zhou, and Mehul Motani. Spatio-temporal autoencoder for feature learning in patient data with missing observations. In *2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 886–890. IEEE, 2017.

[159] Divyanshu Talwar, Aanchal Mongia, Debarka Sengupta, and Angshul Majumdar. Autoimpute: Autoencoder based imputation of single-cell rna-seq data. *Scientific reports*, 8(1):1–11, 2018.

[160] Lovedeep Gondara and Ke Wang. Mida: Multiple imputation using denoising autoencoders. In *Advances in Knowledge Discovery and Data Mining: 22nd Pacific-Asia Conference, PAKDD 2018, Melbourne, VIC, Australia, June 3-6, 2018, Proceedings, Part III 22*, pages 260–272. Springer, 2018.

[161] John T McCoy, Steve Kroon, and Lidia Auret. Variational autoencoders for missing data imputation with application to a simulated milling circuit. *IFAC-PapersOnLine*, 51(21):141–146, 2018.

[162] Xiaochen Lai, Xia Wu, Liyong Zhang, Wei Lu, and Chongquan Zhong. Imputations of missing values using a tracking-removed autoencoder trained with incomplete data. *Neurocomputing*, 366:54–65, 2019.

[163] Guillem Boquet, Jose Lopez Vicario, Antoni Morell, and Javier Serrano. Missing data in traffic estimation: A variational autoencoder imputation method. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2882–2886. IEEE, 2019.

[164] Adrian V Dalca, John Guttag, and Mert R Sabuncu. Unsupervised data imputation via variational inference of deep subspaces. *arXiv preprint arXiv:1903.03503*, 2019.

[165] Savvas Kinalis, Finn Cilius Nielsen, Ole Winther, and Frederik Otzen Bagger. Deconvolution of autoencoders to learn biological regulatory modules from single cell mrna sequencing data. *BMC bioinformatics*, 20:1–9, 2019.

[166] Ruimin Xie, Nabil Magbool Jan, Kuangrong Hao, Lei Chen, and Biao Huang. Supervised variational autoencoders for soft sensor modeling with missing data. *IEEE Transactions on Industrial Informatics*, 16(4):2820–2828, 2019.

[167] Vincent Fortuin, Dmitry Baranchuk, Gunnar Rätsch, and Stephan Mandt. Gp-vae: Deep probabilistic time series imputation. In *International conference on artificial intelligence and statistics*, pages 1651–1661. PMLR, 2020.

[168] Dong Li, Linhao Li, Xianling Li, Zhiwu Ke, and Qinghua Hu. Smoothed lstm-ae: A spatio-temporal deep model for multiple time-series missing imputation. *Neurocomputing*, 411:351–363, 2020.

[169] Alfredo Nazabal, Pablo M Olmos, Zoubin Ghahramani, and Isabel Valera. Handling incomplete heterogeneous data using vaes. *Pattern Recognition*, 107:107501, 2020.

[170] Chao Ma, Sebastian Tschiatschek, Richard Turner, José Miguel Hernández-Lobato, and Cheng Zhang. Vaem: a deep generative model for heterogeneous mixed type data. *Advances in Neural Information Processing Systems*, 33:11237–11247, 2020.

[171] Ricardo Cardoso Pereira, Pedro Henriques Abreu, and Pedro Pereira Rodrigues. Vae-bridge: Variational autoencoder filter for bayesian ridge imputation of missing data. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE, 2020.

[172] Seunghyoung Ryu, Minsoo Kim, and Hongseok Kim. Denoising autoencoder-based missing value imputation for smart meters. *IEEE Access*, 8:40656–40666, 2020.

[173] Yeping Lina Qiu, Hong Zheng, and Olivier Gevaert. Genomic data imputation with variational auto-encoders. *GigaScience*, 9(8):giaa082, 2020.

[174] Adrián Sánchez-Morales, José-Luis Sancho-Gómez, and Aníbal R Figueiras-Vidal. Complete autoencoders for classification with missing values. *Neural Computing and Applications*, 33:1951–1957, 2021.

[175] Ignacio Peis, Chao Ma, and José Miguel Hernández-Lobato. Missing data imputation and acquisition with deep hierarchical models and hamiltonian monte carlo. *arXiv preprint arXiv:2202.04599*, 2022.

[176] Zhuofu Pan, Yalin Wang, Kai Wang, Hongtian Chen, Chunhua Yang, and Weihua Gui. Imputation of missing values in time series using an adaptive-learned median-filled deep autoencoder. *IEEE Transactions on Cybernetics*, 2022.

[177] Ricardo Cardoso Pereira, Pedro Henriques Abreu, and Pedro Pereira Rodrigues. Partial multiple imputation with variational autoencoders: Tackling not at randomness in healthcare data. *IEEE Journal of Biomedical and Health Informatics*, 26(8):4218–4227, 2022.

[178] Jin-Hong Du, Zhanrui Cai, and Kathryn Roeder. Robust probabilistic modeling for single-cell multimodal mosaic integration and imputation via scvaeit. *Proceedings of the National Academy of Sciences*, 119(49):e2214414119, 2022.

[179] Siqi Chen, Xuhua Yan, Ruiqing Zheng, and Min Li. Bubble: a fast single-cell rna-seq imputation using an autoencoder constrained by bulk rna-seq data. *Briefings in Bioinformatics*, 24(1):bbac580, 2023.

[180] Konstantinos Psychogyios, Loukas Ilias, Christos Ntanos, and Dimitris Askounis. Missing value imputation methods for electronic health records. *IEEE Access*, 11:21562–21574, 2023.

[181] Rahul G Krishnan, Uri Shalit, and David Sontag. Deep kalman filters. *arXiv preprint arXiv:1511.05121*, 2015.

[182] Li Yingzhen and Stephan Mandt. Disentangled sequential autoencoder. In *International Conference on Machine Learning*, pages 5670–5679. PMLR, 2018.

[183] Ikaro Silva, George Moody, Daniel J Scott, Leo A Celi, and Roger G Mark. Predicting in-hospital mortality of icu patients: The physionet/computing in cardiology challenge 2012. In *2012 Computing in Cardiology*, pages 245–248. IEEE, 2012.

[184] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8789–8797, 2018.

[185] Kanghyeok Ko, Taesun Yeom, and Minhyeok Lee. Superstargan: Generative adversarial networks for image-to-image translation in large-scale domains. *Neural Networks*, 162:330–339, 2023.

[186] Piotr Teterwak, Aaron Sarna, Dilip Krishnan, Aaron Maschinot, David Belanger, Ce Liu, and William T Freeman. Boundless: Generative adversarial networks for

image extension. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10521–10530, 2019.

[187] He-Liang Huang, Yuxuan Du, Ming Gong, Youwei Zhao, Yulin Wu, Chaoyue Wang, Shaowei Li, Futian Liang, Jin Lin, Yu Xu, et al. Experimental quantum generative adversarial networks for image generation. *Physical Review Applied*, 16(2):024051, 2021.

[188] Eirikur Agustsson, Michael Tschannen, Fabian Mentzer, Radu Timofte, and Luc Van Gool. Generative adversarial networks for extreme learned image compression. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 221–231, 2019.

[189] Jingkuan Song, Tao He, Lianli Gao, Xing Xu, Alan Hanjalic, and Heng Tao Shen. Unified binary generative adversarial network for image retrieval and compression. *International Journal of Computer Vision*, 128:2243–2264, 2020.

[190] Weili Nie, Nina Narodytska, and Ankit Patel. Relgan: Relational generative adversarial networks for text generation. In *International conference on learning representations*, 2018.

[191] Yang Yang, Xiaodong Dan, Xuesong Qiu, and Zhipeng Gao. Fggan: Feature-guiding generative adversarial networks for text generation. *IEEE Access*, 8:105217–105225, 2020.

[192] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1316–1324, 2018.

[193] Minfeng Zhu, Pingbo Pan, Wei Chen, and Yi Yang. Dm-gan: Dynamic memory generative adversarial networks for text-to-image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5802–5810, 2019.

[194] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[195] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[196] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[197] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

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

[199] Asifullah Khan, Anabia Sohail, Umme Zahoora, and Aqsa Saeed Qureshi. A survey of the recent architectures of deep convolutional neural networks. *Artificial intelligence review*, 53:5455–5516, 2020.

[200] Laith Alzubaidi, Jinglan Zhang, Amjad J Humaidi, Ayad Al-Dujaili, Ye Duan, Omran Al-Shamma, José Santamaría, Mohammed A Fadhel, Muthana Al-Amidie, and Laith Farhan. Review of deep learning: Concepts, cnn architectures, challenges, applications, future directions. *Journal of big Data*, 8:1–74, 2021.

[201] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9446–9454, 2018.

[202] Yifan Zhuang, Ruimin Ke, and Yinhai Wang. Innovative method for traffic data imputation based on convolutional neural network. *IET Intelligent Transport Systems*, 13(4):605–613, 2019.

[203] Gao Fan, Jun Li, and Hong Hao. Lost data recovery for structural health monitoring based on convolutional neural networks. *Structural Control and Health Monitoring*, 26(10):e2433, 2019.

[204] Jong-Hwan Jang, Junggu Choi, Hyun Woong Roh, Sang Joon Son, Chang Hyung Hong, Eun Young Kim, Tae Young Kim, Dukyong Yoon, et al. Deep learning approach for imputation of missing values in actigraphy data: Algorithm development study. *JMIR mHealth and uHealth*, 8(7):e16113, 2020.

[205] Ouafa Benkraouda, Bilal Thonnam Thodi, Hwasoo Yeo, Monica Menendez, and Saif Eddin Jabari. Traffic data imputation using deep convolutional neural networks. *IEEE Access*, 8:104740–104752, 2020.

[206] Shuo Shi, Qiheng Qian, Shuhuan Yu, Qi Wang, Jinyue Wang, Jingyao Zeng, Zhenglin Du, and Jingfa Xiao. Refrgim: an intelligent reference panel reconstruction method for genotype imputation with convolutional neural networks. *Briefings in Bioinformatics*, 22(6):bbab326, 2021.

[207] Marcin Przewięźlikowski, Marek Śmieja, Łukasz Struski, and Jacek Tabor. Misconv: Convolutional neural networks for missing data. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2060–2069, 2022.

[208] Adityanarayanan Radhakrishnan, George Stefanakis, Mikhail Belkin, and Caroline Uhler. Simple, fast, and flexible framework for matrix completion with infinite width neural networks. *Proceedings of the National Academy of Sciences*, 119(16):e2115064119, 2022.

[209] Hufsa Khan, Xizhao Wang, and Han Liu. Handling missing data through deep convolutional neural network. *Information Sciences*, 595:278–293, 2022.

[210] Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. Strategies for pre-training graph neural networks. *arXiv preprint arXiv:1905.12265*, 2019.

[211] Runpei Dong, Zekun Qi, Linfeng Zhang, Junbo Zhang, Jianjian Sun, Zheng Ge, Li Yi, and Kaisheng Ma. Autoencoders as cross-modal teachers: Can pretrained 2d image transformers help 3d representation learning? *arXiv preprint arXiv:2212.08320*, 2022.

[212] Renrui Zhang, Liuhui Wang, Yu Qiao, Peng Gao, and Hongsheng Li. Learning 3d representations from 2d pre-trained models via image-to-point masked autoencoders. *arXiv preprint arXiv:2212.06785*, 2022.

[213] Yuchong Sun, Hongwei Xue, Ruihua Song, Bei Liu, Huan Yang, and Jianlong Fu. Long-form video-language pre-training with multimodal temporal contrastive learning. *arXiv preprint arXiv:2210.06031*, 2022.

[214] Fanyi Xiao, Kaustav Kundu, Joseph Tighe, and Davide Modolo. Hierarchical self-supervised representation learning for movie understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9727–9736, 2022.

[215] Jinsung Yoon, James Jordon, and Mihaela Schaar. Gain: Missing data imputation using generative adversarial nets. In *International conference on machine learning*, pages 5689–5698. PMLR, 2018.

[216] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[217] Likun Jiang, Changzhi Jiang, Xinyu Yu, Rao Fu, Shuting Jin, and Xiangrong Liu. Deeptta: a transformer-based model for predicting cancer drug response. *Briefings in bioinformatics*, 23(3):bbac100, 2022.

[218] Shuang Zhang, Rui Fan, Yuti Liu, Shuang Chen, Qiao Liu, and Wanwen Zeng. Applications of transformer-based language models in bioinformatics: a survey. *Bioinformatics Advances*, 3(1):vbad001, 2023.

[219] Jiong Zhang, Wei-Cheng Chang, Hsiang-Fu Yu, and Inderjit Dhillon. Fast multi-resolution transformer fine-tuning for extreme multi-label text classification. *Advances in Neural Information Processing Systems*, 34:7267–7280, 2021.

[220] Jibing Gong, Zhiyong Teng, Qi Teng, Hekai Zhang, Linfeng Du, Shuai Chen, Md Zakirul Alam Bhuiyan, Jianhua Li, Mingsheng Liu, and Hongyuan Ma. Hierarchical graph transformer-based deep learning model for large-scale multi-label text classification. *IEEE Access*, 8:30885–30896, 2020.

[221] Yikuan Li, Mohammad Mamouei, Gholamreza Salimi-Khorshidi, Shishir Rao, Abdelaali Hassaine, Dexter Canoy, Thomas Lukasiewicz, and Kazem Rahimi. Hibehrt: Hierarchical transformer-based model for accurate prediction of clinical events using multimodal longitudinal electronic health records. *IEEE journal of biomedical and health informatics*, 27(2):1106–1117, 2022.

[222] Shishir Rao, Yikuan Li, Rema Ramakrishnan, Abdelaali Hassaine, Dexter Canoy, John G Cleland, Thomas Lukasiewicz, Gholamreza Salimi-Khorshidi, and Kazem Rahimi. An explainable transformer-based deep learning model for the prediction of incident heart failure. *IEEE Journal of Biomedical and Health Informatics*, 2022.

[223] Jiawei Ma, Zheng Shou, Alireza Zareian, Hassan Mansour, Anthony Vetro, and Shih-Fu Chang. Cdsa: cross-dimensional self-attention for multivariate, geo-tagged time series imputation. *arXiv preprint arXiv:1905.09904*, 2019.

[224] Richard Wu, Aoqian Zhang, Ihab Ilyas, and Theodoros Rekatsinas. Attention-based learning for missing data imputation in holoclean. *Proceedings of Machine Learning and Systems*, 2:307–325, 2020.

[225] Parikshit Bansal, Prathamesh Deshpande, and Sunita Sarawagi. Missing value imputation on multidimensional time series. *arXiv preprint arXiv:2103.01600*, 2021.

[226] George Zerveas, Srideepika Jayaraman, Dhaval Patel, Anuradha Bhamidipaty, and Carsten Eickhoff. A transformer-based framework for multivariate time series representation learning. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 2114–2124, 2021.

[227] Wenjie Du, David Côté, and Yan Liu. Saits: Self-attention-based imputation for time series. *Expert Systems with Applications*, 219:119619, 2023.

[228] Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. Mask-predict: Parallel decoding of conditional masked language models. *arXiv preprint arXiv:1904.09324*, 2019.

[229] Hangbo Bao, Li Dong, Furu Wei, Wenhui Wang, Nan Yang, Xiaodong Liu, Yu Wang, Jianfeng Gao, Songhao Piao, Ming Zhou, et al. Unilmv2: Pseudo-masked language models for unified language model pre-training. In *International conference on machine learning*, pages 642–652. PMLR, 2020.

[230] Kornraphop Kawintiranon and Lisa Singh. Knowledge enhanced masked language model for stance detection. In *Proceedings of the 2021 conference of the north american chapter of the association for computational linguistics: human language technologies*, pages 4725–4735, 2021.

[231] Elad Ben Zaken, Shauli Ravfogel, and Yoav Goldberg. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. *arXiv preprint arXiv:2106.10199*, 2021.

[232] Yue Guo, Yi Yang, and Ahmed Abbasi. Auto-debias: Debiasing masked language models with automated biased prompts. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1012–1023, 2022.

[233] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. Graph neural networks for social recommendation. In *The world wide web conference*, pages 417–426, 2019.

[234] Xiao-Meng Zhang, Li Liang, Lin Liu, and Ming-Jing Tang. Graph neural networks and their current applications in bioinformatics. *Frontiers in genetics*, 12:690049, 2021.

[235] Jiapeng Wu, Meng Cao, Jackie Chi Kit Cheung, and William L Hamilton. Temp: Temporal message passing for temporal knowledge graph completion. *arXiv preprint arXiv:2010.03526*, 2020.

[236] Tuan Nguyen, Giang TT Nguyen, Thin Nguyen, and Duc-Hau Le. Graph convolutional networks for drug response prediction. *IEEE/ACM transactions on computational biology and bioinformatics*, 19(1):146–154, 2021.

[237] Patrick Reiser, Marlen Neubert, André Eberhard, Luca Torresi, Chen Zhou, Chen Shao, Houssam Metni, Clint van Hoesel, Henrik Schopmans, Timo Sommer, et al. Graph neural networks for materials science and chemistry. *Communications Materials*, 3(1):93, 2022.

[238] David Ahmedt-Aristizabal, Mohammad Ali Armin, Simon Denman, Clinton Fookes, and Lars Petersson. Graph-based deep learning for medical diagnosis and analysis: past, present and future. *Sensors*, 21(14):4758, 2021.

[239] Qi Zhang, Jianlong Chang, Gaofeng Meng, Shibiao Xu, Shiming Xiang, and Chunhong Pan. Learning graph structure via graph convolutional networks. *Pattern Recognition*, 95:308–318, 2019.

[240] Tirtharaj Dash, Ashwin Srinivasan, and Lovekesh Vig. Incorporating symbolic domain knowledge into graph neural networks. *Machine Learning*, 110(7):1609–1636, 2021.

[241] Indro Spinelli, Simone Scardapane, and Aurelio Uncini. Missing data imputation with adversarially-trained graph convolutional networks. *Neural Networks*, 129:249–260, 2020.

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

[243] David Gordon, Panayiotis Petousis, Henry Zheng, Davina Zamanzadeh, and Alex AT Bui. Tsi-gnn: Extending graph neural networks to handle missing data in temporal settings. *Frontiers in big Data*, 4:693869, 2021.

[244] Andrea Cini, Ivan Marisca, and Cesare Alippi. Filling the g_ap_s: Multivariate time series imputation by graph neural networks. *arXiv preprint arXiv:2108.00298*, 2021.

[245] Hibiki Taguchi, Xin Liu, and Tsuyoshi Murata. Graph convolutional networks for graphs containing missing features. *Future Generation Computer Systems*, 117:155–168, 2021.

[246] Yakun Chen, Zihao Li, Chao Yang, Xianzhi Wang, Guodong Long, and Guandong Xu. Adaptive graph recurrent network for multivariate time series imputation. In *International Conference on Neural Information Processing*, pages 64–73. Springer, 2022.

[247] Buliao Huang, Yunhui Zhu, Muhammad Usman, Xiren Zhou, and Huanhuan Chen. Graph neural networks for missing value classification in a task-driven metric space. *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[248] Caizheng Liu, Guangfan Cui, and Shenghua Liu. Cgcnimp: a causal graph convolutional network for multivariate time series imputation. *PeerJ Computer Science*, 8:e966, 2022.

[249] Jiajun Zhong, Ning Gui, and Weiwei Ye. Data imputation with iterative graph reconstruction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 11399–11407, 2023.

[250] Xiangjie Kong, Wenfeng Zhou, Guojiang Shen, Wenyi Zhang, Nali Liu, and Yao Yang. Dynamic graph convolutional recurrent imputation network for spatiotemporal traffic missing data. *Knowledge-Based Systems*, 261:110188, 2023.

[251] Jingwei Zuo, Karine Zeitouni, Yehia Taher, and Sandra Garcia-Rodriguez. Graph convolutional networks for traffic forecasting with missing values. *Data Mining and Knowledge Discovery*, 37(2):913–947, 2023.

[252] Longfei Xu, Lingyu Xu, and Jie Yu. Time series imputation with gan inversion and decay connection. *Information Sciences*, page 119234, 2023.

[253] Antonia Creswell and Anil Anthony Bharath. Inverting the generator of a generative adversarial network. *IEEE transactions on neural networks and learning systems*, 30(7):1967–1974, 2018.

[254] Jiapeng Zhu, Yujun Shen, Deli Zhao, and Bolei Zhou. In-domain gan inversion for real image editing. In *European conference on computer vision*, pages 592–608. Springer, 2020.

[255] Michael W Dusenberry, Dustin Tran, Edward Choi, Jonas Kemp, Jeremy Nixon, Ghassen Jerfel, Katherine Heller, and Andrew M Dai. Analyzing the role of model uncertainty for electronic health records. In *Proceedings of the ACM Conference on Health, Inference, and Learning*, pages 204–213, 2020.

[256] Chacha Chen, Junjie Liang, Fenglong Ma, Lucas Glass, Jimeng Sun, and Cao Xiao. Unite: Uncertainty-based health risk prediction leveraging multi-sourced data. In *Proceedings of the Web Conference 2021*, pages 217–226, 2021.

[257] Asharul Islam Khan and Salim Al-Habsi. Machine learning in computer vision. *Procedia Computer Science*, 167:1444–1451, 2020.

[258] Shashi Pal Singh, Ajai Kumar, Hemant Darbari, Lenali Singh, Anshika Rastogi, and Shikha Jain. Machine translation using deep learning: An overview. In *2017 international conference on computer, communications and electronics (comptelix)*, pages 162–167. IEEE, 2017.

[259] Adi L Tarca, Vincent J Carey, Xue-wen Chen, Roberto Romero, and Sorin Drăghici. Machine learning and its applications to biology. *PLoS computational biology*, 3(6):e116, 2007.

[260] Dimitris Bertsimas, Agni Orfanoudaki, and Colin Pawlowski. Imputation of clinical covariates in time series. *Machine Learning*, 110(1):185–248, 2021.

[261] Catherine A Welch, Irene Petersen, Jonathan W Bartlett, Ian R White, Louise Marston, Richard W Morris, Irwin Nazareth, Kate Walters, and James Carpenter. Evaluation of two-fold fully conditional specification multiple imputation for longitudinal electronic health record data. *Statistics in medicine*, 33(21):3725–3737, 2014.

[262] Mark Bounthavong, Jonathan H Watanabe, and Kevin M Sullivan. Approach to addressing missing data for electronic medical records and pharmacy claims data research. *Pharmacotherapy: The Journal of Human Pharmacology and Drug Therapy*, 35(4):380–387, 2015.

[263] Seyedmostafa Sheikhalishahi, Vevake Balaraman, and Venet Osmani. Benchmarking machine learning models on eicu critical care dataset. *arXiv preprint arXiv:1910.00964*, 2019.

[264] Moloud Abdar, Farhad Pourpanah, Sadiq Hussain, Dana Rezazadegan, Li Liu, Mohammad Ghavamzadeh, Paul Fieguth, Xiaochun Cao, Abbas Khosravi, U Rajendra

Acharya, et al. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fusion*, 76:243–297, 2021.

[265] Eyke Hüllermeier and Willem Waegeman. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine Learning*, 110(3):457–506, 2021.

[266] Guijie Li, Zhenzhou Lu, Luyi Li, and Bo Ren. Aleatory and epistemic uncertainties analysis based on non-probabilistic reliability and its kriging solution. *Applied Mathematical Modelling*, 40(9-10):5703–5716, 2016.

[267] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *arXiv preprint arXiv:1703.04977*, 2017.

[268] Raanan Y Rohekar, Shami Nisimov, Yaniv Gurwicz, Guy Koren, and Gal Novik. Constructing deep neural networks by bayesian network structure learning. *Advances in Neural Information Processing Systems*, 31, 2018.

[269] Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. Stochastic variational inference. *Journal of Machine Learning Research*, 2013.

[270] Justin Domke and Daniel R Sheldon. Importance weighting and variational inference. *Advances in neural information processing systems*, 31, 2018.

[271] Akash Kumar Dhaka, Alejandro Catalina, Michael R Andersen, Måns Magnusson, Jonathan Huggins, and Aki Vehtari. Robust, accurate stochastic optimization for variational inference. *Advances in Neural Information Processing Systems*, 33:10961–10973, 2020.

[272] Luigi Acerbi. Variational bayesian monte carlo. *Advances in Neural Information Processing Systems*, 31, 2018.

[273] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059, 2016.

[274] Axel Brando Guillaumes. *Mixture density networks for distribution and uncertainty estimation*. PhD thesis, Universitat Politècnica de Catalunya. Facultat d'Informàtica de Barcelona, 2017.

[275] Ziheng Chen, Chaojie Lai, and Jiangtao Ren. Hospital readmission prediction based on long-term and short-term information fusion. *Applied Soft Computing*, 96:106690, 2020.

[276] James Honaker and Gary King. What to do about missing values in time-series cross-section data. *American journal of political science*, 54(2):561–581, 2010.

[277] Hyun Kang. The prevention and handling of the missing data. *Korean journal of anesthesiology*, 64(5):402, 2013.

[278] Niamh McCombe, Shuo Liu, Xuemei Ding, Girijesh Prasad, Magda Bucholc, David P Finn, Stephen Todd, Paula L McClean, and KongFatt Wong-Lin. Practical strategies for extreme missing data imputation in dementia diagnosis. *IEEE journal of biomedical and health informatics*, 26(2):818–827, 2021.

[279] Yuan Luo. Evaluating the state of the art in missing data imputation for clinical data. *Briefings in Bioinformatics*, 23(1):bbab489, 2022.

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

[281] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 2015.

[282] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. Unsupervised learning of video representations using lstms. In *International conference on machine learning*, pages 843–852. PMLR, 2015.

[283] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[284] James Schneider and Todd Sweberg. Acute respiratory failure. *Critical Care Clinics*, 29(2):167–183, 2013.

[285] Chih-Cheng Lai, Mei-I Sung, Hsiao-Hua Liu, Chin-Ming Chen, Shyh-Ren Chiang, Wei-Lun Liu, Chien-Ming Chao, Chung-Han Ho, Shih-Feng Weng, Shu-Chen Hsing, et al. The ratio of partial pressure arterial oxygen and fraction of inspired oxygen 1 day after acute respiratory distress syndrome onset can predict the outcomes of involving patients. *Medicine*, 95(14), 2016.

[286] Chih-Cheng Lai, Kuei-Ling Tseng, Chung-Han Ho, Shyh-Ren Chiang, Chin-Ming Chen, Khee-Siang Chan, Chien-Ming Chao, Shu-Chen Hsing, and Kuo-Chen Cheng. Prognosis of patients with acute respiratory failure and prolonged intensive care unit stay. *Journal of Thoracic Disease*, 11(5):2051, 2019.

[287] Charlie Tang and Russ R Salakhutdinov. Learning stochastic feedforward neural networks. *Advances in Neural Information Processing Systems*, 26, 2013.

[288] Phuc H Le-Khac, Graham Healy, and Alan F Smeaton. Contrastive representation learning: A framework and review. *Ieee Access*, 8:193907–193934, 2020.

[289] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673, 2020.

[290] Wenguan Wang, Tianfei Zhou, Fisher Yu, Jifeng Dai, Ender Konukoglu, and Luc Van Gool. Exploring cross-image pixel contrast for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7303–7313, 2021.

[291] Chengxi Zang and Fei Wang. Scehr: Supervised contrastive learning for clinical risk prediction using electronic health records. In *Proceedings. IEEE International Conference on Data Mining*, volume 2021, pages 857–866, 2021.

[292] Mingkun Li, Chun-Guang Li, and Jun Guo. Cluster-guided asymmetric contrastive learning for unsupervised person re-identification. *IEEE Transactions on Image Processing*, 31:3606–3617, 2022.

[293] Jiacheng Li, Jingbo Shang, and Julian McAuley. Uctopic: Unsupervised contrastive learning for phrase representations and topic mining. *arXiv preprint arXiv:2202.13469*, 2022.

[294] Bo Pang, Yizhuo Li, Yifan Zhang, Gao Peng, Jiajun Tang, Kaiwen Zha, Jiefeng Li, and Cewu Lu. Unsupervised representation for semantic segmentation by implicit cycle-attention contrastive learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 2044–2052, 2022.

[295] Chuanguang Yang, Zhulin An, Linhang Cai, and Yongjun Xu. Mutual contrastive learning for visual representation learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 3045–3053, 2022.

[296] Xin Yuan, Zhe Lin, Jason Kuen, Jianming Zhang, Yilin Wang, Michael Maire, Ajinkya Kale, and Baldo Faieta. Multimodal contrastive training for visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6995–7004, 2021.

[297] Yingheng Wang, Yaosen Min, Xin Chen, and Ji Wu. Multi-view graph contrastive representation learning for drug-drug interaction prediction. In *Proceedings of the Web Conference 2021*, pages 2921–2933, 2021.

[298] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[299] Xi Sheryl Zhang, Fengyi Tang, Hiroko H Dodge, Jiayu Zhou, and Fei Wang. Metapred: Meta-learning for clinical risk prediction with limited patient electronic health records. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2487–2495, 2019.

[300] Stephen R Pfohl, Agata Foryciarz, and Nigam H Shah. An empirical characterization of fair machine learning for clinical risk prediction. *Journal of biomedical informatics*, 113:103621, 2021.

[301] Yuyang Xu, Haochao Ying, Siyi Qian, Fuzhen Zhuang, Xiao Zhang, Deqing Wang, Jian Wu, and Hui Xiong. Time-aware context-gated graph attention network for clinical risk prediction. *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[302] Thai-Hoang Pham, Changchang Yin, Laxmi Mehta, Xueru Zhang, and Ping Zhang. A fair and interpretable network for clinical risk prediction: a regularized multi-view multi-task learning approach. *Knowledge and Information Systems*, 65(4):1487–1521, 2023.

[303] Runpu Chen, Le Yang, Steve Goodison, and Yijun Sun. Deep-learning approach to identifying cancer subtypes using high-dimensional genomic data. *Bioinformatics*, 36(5):1476–1483, 2020.

[304] Changchang Yin, Ruoqi Liu, Dongdong Zhang, and Ping Zhang. Identifying sepsis subphenotypes via time-aware multi-modal auto-encoder. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 862–872, 2020.

[305] Changchang Yin, Sayoko E Moroi, and Ping Zhang. Predicting age-related macular degeneration progression with contrastive attention and time-aware lstm. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4402–4412, 2022.

[306] Anant Dadu, Vipul Satone, Rachneet Kaur, Sayed Hadi Hashemi, Hampton Leonard, Hirotaka Iwaki, Mary B Makarious, Kimberley J Billingsley, Sara Bandres-Ciga, Lana J Sargent, et al. Identification and prediction of parkinson's disease subtypes and progression using machine learning in two cohorts. *npj Parkinson's Disease*, 8(1):172, 2022.

[307] Shahadat Uddin, Arif Khan, Md Ekramul Hossain, and Mohammad Ali Moni. Comparing different supervised machine learning algorithms for disease prediction. *BMC medical informatics and decision making*, 19(1):1–16, 2019.

[308] Md Mamun Ali, Bikash Kumar Paul, Kawsar Ahmed, Francis M Bui, Julian MW Quinn, and Mohammad Ali Moni. Heart disease prediction using supervised machine learning algorithms: Performance analysis and comparison. *Computers in Biology and Medicine*, 136:104672, 2021.

[309] Ahmed Alaa and Mihaela van der Schaar. Attentive state-space modeling of disease progression. 2019.

[310] Leibo Liu, Oscar Perez-Concha, Anthony Nguyen, Vicki Bennett, and Louisa Jorm. Hierarchical label-wise attention transformer model for explainable icd coding. *Journal of biomedical informatics*, 133:104161, 2022.

[311] Xu Ye, Meng Xiao, Zhiyuan Ning, Weiwei Dai, Wenjuan Cui, Yi Du, and Yuanchun Zhou. Needed: Introducing hierarchical transformer to eye diseases diagnosis. In *Proceedings of the 2023 SIAM International Conference on Data Mining (SDM)*, pages 667–675. SIAM, 2023.

[312] Idar Johan Brekke, Lars Håland Puntervoll, Peter Bank Pedersen, John Kellett, and Mikkel Brabrand. The value of vital sign trends in predicting and monitoring clinical deterioration: A systematic review. *PloS one*, 14(1):e0210875, 2019.

[313] Jingqi Zhao, Chuitian Rong, Chunbin Lin, and Xin Dang. Multivariate time series data imputation using attention-based mechanism. *Neurocomputing*, 542:126238, 2023.

[314] Sherry-Ann Brown. Patient similarity: emerging concepts in systems and precision medicine. *Frontiers in physiology*, 7:561, 2016.

[315] Enea Parimbelli, Simone Marini, Lucia Sacchi, and Riccardo Bellazzi. Patient similarity for precision medicine: A systematic review. *Journal of biomedical informatics*, 83:87–96, 2018.

[316] Attila A Seyhan and Claudio Carini. Are innovation and new technologies in precision medicine paving a new era in patients centric care? *Journal of translational medicine*, 17:1–28, 2019.

[317] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. *Advances in neural information processing systems*, 33:5812–5823, 2020.

[318] Yuning You, Tianlong Chen, Yang Shen, and Zhangyang Wang. Graph contrastive learning automated. In *International Conference on Machine Learning*, pages 12121–12132. PMLR, 2021.

[319] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. Graph contrastive learning with adaptive augmentation. In *Proceedings of the Web Conference 2021*, pages 2069–2080, 2021.

[320] Sudhanshu Chanpuriya and Cameron Musco. Simplified graph convolution with heterophily. *Advances in Neural Information Processing Systems*, 35:27184–27197, 2022.

[321] Sitao Luan, Chenqing Hua, Qincheng Lu, Jiaqi Zhu, Mingde Zhao, Shuyuan Zhang, Xiao-Wen Chang, and Doina Precup. Revisiting heterophily for graph neural networks. *Advances in neural information processing systems*, 35:1362–1375, 2022.

[322] Yizhen Zheng, He Zhang, Vincent Lee, Yu Zheng, Xiao Wang, and Shirui Pan. Finding the missing-half: Graph complementary learning for homophily-prone and heterophily-prone graphs. *arXiv preprint arXiv:2306.07608*, 2023.

[323] Zihao Zhu, Changchang Yin, Buyue Qian, Yu Cheng, Jishang Wei, and Fei Wang. Measuring patient similarities via a deep architecture with medical concept embedding. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pages 749–758. IEEE, 2016.

[324] Madhumita Sushil, Simon Šuster, Kim Luyckx, and Walter Daelemans. Patient representation learning and interpretable evaluation using clinical notes. *Journal of biomedical informatics*, 84:103–113, 2018.

[325] Qiuling Suo, Fenglong Ma, Ye Yuan, Mengdi Huai, Weida Zhong, Jing Gao, and Aidong Zhang. Deep patient similarity learning for personalized healthcare. *IEEE transactions on nanobioscience*, 17(3):219–227, 2018.

[326] Chaohe Zhang, Xin Gao, Liantao Ma, Yasha Wang, Jiangtao Wang, and Wen Tang. Grasp: generic framework for health status representation learning based on incorporating knowledge from similar patients. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 715–723, 2021.

[327] Liqi Lei, Yangming Zhou, Jie Zhai, Le Zhang, Zhijia Fang, Ping He, and Ju Gao. An effective patient representation learning for time-series prediction tasks based on ehrs. In *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 885–892. IEEE, 2018.

[328] Tong Ruan, Liqi Lei, Yangming Zhou, Jie Zhai, Le Zhang, Ping He, and Ju Gao. Representation learning for clinical time series prediction tasks in electronic health records. *BMC medical informatics and decision making*, 19(8):1–14, 2019.

[329] Isotta Landi, Benjamin S Glicksberg, Hao-Chih Lee, Sarah Cherng, Giulia Landi, Matteo Danieletto, Joel T Dudley, Cesare Furlanello, and Riccardo Miotto. Deep representation learning of electronic health records to unlock patient stratification at scale. *NPJ digital medicine*, 3(1):1–11, 2020.

[330] Yuqi Si, Jingcheng Du, Zhao Li, Xiaoqian Jiang, Timothy Miller, Fei Wang, W Jim Zheng, and Kirk Roberts. Deep representation learning of patient data from electronic health records (ehr): A systematic review. *Journal of biomedical informatics*, 115:103671, 2021.

[331] Menglin Lu, Yujie Zhang, Suixia Zhang, Hanrui Shi, and Zhengxing Huang. Knowledge-aware patient representation learning for multiple disease subtypes. *Journal of Biomedical Informatics*, 138:104292, 2023.

[332] Vera Ehrenstein, Hadi Kharrazi, Harold Lehmann, and Casey Overby Taylor. Obtaining data from electronic health records. In *Tools and technologies for registry interoperability, registries for evaluating patient outcomes: A user's guide, 3rd edition, Addendum 2 [Internet]*. Agency for Healthcare Research and Quality (US), 2019.

[333] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

[334] Cheng Zheng, Bo Zong, Wei Cheng, Dongjin Song, Jingchao Ni, Wenchao Yu, Haifeng Chen, and Wei Wang. Robust graph representation learning via neural sparsification. In *International Conference on Machine Learning*, pages 11458–11468. PMLR, 2020.

[335] Emma Rocheteau, Catherine Tong, Petar Veličković, Nicholas Lane, and Pietro Liò. Predicting patient outcomes with graph representation learning. *arXiv preprint arXiv:2101.03940*, 2021.

[336] Amara Tariq, Siyi Tang, Hifza Sakhi, Leo Anthony Celi, Janice M Newsome, Daniel L Rubin, Hari Trivedi, Judy Wawira Gichoy, Bhavik Patel, and Imon Banerjee. Graph-based fusion modeling and explanation fordisease trajectory prediction. In *AMIA Annual Symposium Proceedings*, volume 2022, page 1052. American Medical Informatics Association, 2022.

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

[338] Yujie Mo, Liang Peng, Jie Xu, Xiaoshuang Shi, and Xiaofeng Zhu. Simple unsupervised graph representation learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 7797–7805, 2022.

[339] Wenbing Huang, Tong Zhang, Yu Rong, and Junzhou Huang. Adaptive sampling towards fast graph representation learning. *Advances in neural information processing systems*, 31, 2018.

[340] Nima Dehmamy, Albert-László Barabási, and Rose Yu. Understanding the representation power of graph neural networks in learning graph topology. *Advances in Neural Information Processing Systems*, 32, 2019.

[341] Xing Li, Wei Wei, Xiangnan Feng, Xue Liu, and Zhiming Zheng. Representation learning of graphs using graph convolutional multilayer networks based on motifs. *Neurocomputing*, 464:218–226, 2021.

[342] Chao Gao, Junyou Zhu, Fan Zhang, Zhen Wang, and Xuelong Li. A novel representation learning for dynamic graphs based on graph convolutional networks. *IEEE Transactions on Cybernetics*, 2022.

[343] Liang Yao, Chengsheng Mao, and Yuan Luo. Graph convolutional networks for text classification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 7370–7377, 2019.

[344] Le Wu, Peijie Sun, Richang Hong, Yanjie Fu, Xiting Wang, and Meng Wang. Socialgcn: An efficient graph convolutional network based model for social recommendation. *arXiv preprint arXiv:1811.02815*, 2018.

[345] Wei Wang, Xi Yang, Chengkun Wu, and Canqun Yang. Cginet: graph convolutional network-based model for identifying chemical-gene interaction in an integrated multi-relational graph. *BMC bioinformatics*, 21(1):1–17, 2020.

[346] Bo Jiang, Si Chen, Beibei Wang, and Bin Luo. Mglnn: Semi-supervised learning via multiple graph cooperative learning neural networks. *Neural Networks*, 153:204–214, 2022.

[347] Rakshit Trivedi, Bunyamin Sisman, Jun Ma, Christos Faloutsos, Hongyuan Zha, and Xin Luna Dong. Linknbed: Multi-graph representation learning with entity linkage. *arXiv preprint arXiv:1807.08447*, 2018.

[348] Bo-Wei Zhao, Zhu-Hong You, Leon Wong, Ping Zhang, Hao-Yuan Li, and Lei Wang. Mgrl: predicting drug-disease associations based on multi-graph representation learning. *Frontiers in Genetics*, 12:657182, 2021.

[349] Yan Luo, Fu-lai Chung, and Kai Chen. Urban region profiling via multi-graph representation learning. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 4294–4298, 2022.

[350] Ziyang Wang, Yaowen Gu, Si Zheng, Lin Yang, and Jiao Li. Mgrel: A multi-graph representation learning-based ensemble learning method for gene-disease association prediction. *Computers in Biology and Medicine*, 155:106642, 2023.

[351] Edward Choi, Zhen Xu, Yujia Li, Michael Dusenberry, Gerardo Flores, Emily Xue, and Andrew Dai. Learning the graphical structure of electronic health records with graph convolutional transformer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 606–613, 2020.

[352] Artuur M Leeuwenberg and Ewoud Schuit. Prediction models for covid-19 clinical decision making. *The Lancet Digital Health*, 2(10):e496–e497, 2020.

[353] Videha Sharma, Ibrahim Ali, Sabine van der Veer, Glen Martin, John Ainsworth, and Titus Augustine. Adoption of clinical risk prediction tools is limited by a lack of integration with electronic health records. *BMJ Health & Care Informatics*, 28(1), 2021.

[354] Sandra Eloranta and Magnus Boman. Predictive models for clinical decision making: Deep dives in practical machine learning. *Journal of Internal Medicine*, 292(2):278–295, 2022.

[355] Rimma Pivovarov. *Electronic health record summarization over heterogeneous and irregularly sampled clinical data*. Columbia University, 2016.

[356] Md Kamrul Hasan, Md Ashraful Alam, Shidhartho Roy, Aishwariya Dutta, Md Tasnim Jawad, and Sunanda Das. Missing value imputation affects the performance of machine learning: A review and analysis of the literature (2010–2021). *Informatics in Medicine Unlocked*, 27:100799, 2021.

[357] Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long- and short-term temporal patterns with deep neural networks. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pages 95–104, 2018.

[358] Sebastian Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017.

[359] Trevor Standley, Amir Zamir, Dawn Chen, Leonidas Guibas, Jitendra Malik, and Silvio Savarese. Which tasks should be learned together in multi-task learning? In *International Conference on Machine Learning*, pages 9120–9132. PMLR, 2020.

[360] Muchao Ye, Junyu Luo, Cao Xiao, and Fenglong Ma. Lsan: Modeling long-term dependencies and short-term correlations with hierarchical attention for risk prediction. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 1753–1762, 2020.

[361] Liantao Ma, Junyi Gao, Yasha Wang, Chaohe Zhang, Jiangtao Wang, Wenjie Ruan, Wen Tang, Xin Gao, and Xinyu Ma. Adacare: Explainable clinical health status representation learning via scale-adaptive feature extraction and recalibration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 825–832, 2020.

[362] Lei Bai, Lina Yao, Can Li, Xianzhi Wang, and Can Wang. Adaptive graph convolutional recurrent network for traffic forecasting. *Advances in neural information processing systems*, 33:17804–17815, 2020.

[363] Christoph Molnar. *Interpretable machine learning*. Lulu. com, 2020.

[364] Nishant Jain and Prasanta K Jana. Xrrf: An explainable reasonably randomised forest algorithm for classification and regression problems. *Information Sciences*, 2022.

[365] Didi Han, Fengshuo Xu, Luming Zhang, Rui Yang, Shuai Zheng, Tao Huang, Haiyan Yin, and Jun Lyu. Early prediction of in-hospital mortality in patients with congestive heart failure in intensive care unit: a retrospective observational cohort study. *BMJ Open*, 12(7):e059761, 2022.

[366] Rajsavi S Anand, Paul Stey, Sukrit Jain, Dustin R Biron, Harikrishna Bhatt, Kristina Monteiro, Edward Feller, Megan L Ranney, Indra Neil Sarkar, and Elizabeth S Chen. Predicting mortality in diabetic icu patients using machine learning and severity indices. *AMIA summits on translational science proceedings*, 2018:310, 2018.

[367] Alistair EW Johnson, Lucas Bulgarelli, Lu Shen, Alvin Gayles, Ayad Shammout, Steven Horng, Tom J Pollard, Sicheng Hao, Benjamin Moody, Brian Gow, et al. Mimic-iv, a freely accessible electronic health record dataset. *Scientific data*, 10(1):1, 2023.

[368] Feng Xie, Jun Zhou, Jin Wee Lee, Mingrui Tan, Siqi Li, Logasan S Rajnthern, Marcel Lucas Chee, Bibhas Chakraborty, An-Kwok Ian Wong, Alon Dagan, et al. Benchmarking emergency department prediction models with machine learning and public electronic health records. *Scientific data*, 9(1):1–12, 2022.

[369] Ahmed M Alaa, Thomas Bolton, Emanuele Di Angelantonio, James HF Rudd, and Mihaela Van der Schaar. Cardiovascular disease risk prediction using automated machine learning: A prospective study of 423,604 uk biobank participants. *PloS one*, 14(5):e0213653, 2019.

[370] Xianli Zhang, Buyue Qian, Yang Li, Changchang Yin, Xudong Wang, and Qinghua Zheng. Knowrisk: an interpretable knowledge-guided model for disease risk prediction. In *2019 IEEE International Conference on Data Mining (ICDM)*, pages 1492–1497. IEEE, 2019.

[371] Andrew Ward, Ashish Sarraju, Sukyung Chung, Jiang Li, Robert Harrington, Paul Heidenreich, Latha Palaniappan, David Scheinker, and Fatima Rodriguez. Machine learning and atherosclerotic cardiovascular disease risk prediction in a multi-ethnic population. *NPJ digital medicine*, 3(1):125, 2020.

[372] Kiyana Zolfaghar, Naren Meadem, Ankur Teredesai, Senjuti Basu Roy, Si-Chi Chin, and Brian Muckian. Big data solutions for predicting risk-of-readmission for congestive heart failure patients. In *2013 IEEE international conference on big data*, pages 64–71. IEEE, 2013.

[373] Senjuti Basu Roy, Ankur Teredesai, Kiyana Zolfaghar, Rui Liu, David Hazel, Stacey Newman, and Albert Marinez. Dynamic hierarchical classification for patient risk-of-readmission. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1691–1700, 2015.

[374] Elham Mahmoudi, Neil Kamdar, Noa Kim, Gabriella Gonzales, Karandeep Singh, and Akbar K Waljee. Use of electronic medical records in development and validation of risk prediction models of hospital readmission: systematic review. *bmj*, 369, 2020.

[375] Johanna AAG Damen, Lotty Hooft, Ewoud Schuit, Thomas PA Debray, Gary S Collins, Ioanna Tzoulaki, Camille M Lassale, George CM Siontis, Virginia Chiocchia, Corran Roberts, et al. Prediction models for cardiovascular disease risk in the general population: systematic review. *bmj*, 353, 2016.

[376] Xavier Rossello, Jannick AN Dorresteijn, Arne Janssen, Ekaterini Lambrinou, Martijn Scherrenberg, Eric Bonnefoy-Cudraz, Mark Cobain, Massimo F Piepoli, Frank LJ Visseren, Paul Dendale, et al. Risk prediction tools in cardiovascular disease prevention: a report from the esc prevention of cvd programme led by the european association of preventive cardiology (eapc) in collaboration with the acute cardiovascular care association (acca) and the association of cardiovascular nursing and allied professions (acnap). *European journal of preventive cardiology*, 26(14):1534–1544, 2019.

[377] Farshad Farzadfar. Cardiovascular disease risk prediction models: challenges and perspectives. *The Lancet Global Health*, 7(10):e1288–e1289, 2019.

[378] Kazem Rahimi, Derrick Bennett, Nathalie Conrad, Timothy M Williams, Joyee Basu, Jeremy Dwight, Mark Woodward, Anushka Patel, John McMurray, and Stephen MacMahon. Risk prediction in patients with heart failure: a systematic review and analysis. *JACC: Heart Failure*, 2(5):440–446, 2014.

[379] Hong Yang, Kazuaki Negishi, Petr Otahal, and Thomas H Marwick. Clinical prediction of incident heart failure risk: a systematic review and meta-analysis. *Open heart*, 2(1):e000222, 2015.

[380] Gian Luca Di Tanna, Heidi Wirtz, Karen L Burrows, and Gary Globe. Evaluating risk prediction models for adults with heart failure: A systematic literature review. *PLoS One*, 15(1):e0224135, 2020.

[381] Gary S Collins, Susan Mallett, Omar Omar, and Ly-Mee Yu. Developing risk prediction models for type 2 diabetes: a systematic review of methodology and reporting. *BMC medicine*, 9(1):1–14, 2011.

[382] Zidian Xie, Olga Nikolayeva, Jiebo Luo, and Dongmei Li. Peer reviewed: building risk prediction models for type 2 diabetes using machine learning techniques. *Preventing chronic disease*, 16, 2019.

[383] Elias Dritsas and Maria Trigka. Data-driven machine-learning methods for diabetes risk prediction. *Sensors*, 22(14):5304, 2022.

[384] Cao Xiao, Tengfei Ma, Adji B Dieng, David M Blei, and Fei Wang. Readmission prediction via deep contextual embedding of clinical concepts. *PloS one*, 13(4):e0195024, 2018.

[385] Romain Pirracchio, Maya L Petersen, Marco Carone, Matthieu Resche Rigon, Sylvie Chevret, and Mark J van der Laan. Mortality prediction in intensive care units with the super icu learner algorithm (sicula): a population-based study. *The Lancet Respiratory Medicine*, 3(1):42–52, 2015.

[386] Aya Awad, Mohamed Bader-El-Den, James McNicholas, and Jim Briggs. Early hospital mortality prediction of intensive care unit patients using an ensemble learning approach. *International journal of medical informatics*, 108:185–195, 2017.

[387] Sakyajit Bhattacharya, Vaibhav Rajan, and Harsh Shrivastava. Icu mortality prediction: a classification algorithm for imbalanced datasets. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

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

[389] Wendong Ge, Jin-Won Huh, Yu Rang Park, Jae-Ho Lee, Young-Hak Kim, and Alexander Turchin. An interpretable icu mortality prediction model based on logistic regression and recurrent neural networks with lstm units. In *AMIA Annual Symposium Proceedings*, volume 2018, page 460. American Medical Informatics Association, 2018.

[390] Ning Liu, Pan Lu, Wei Zhang, and Jianyong Wang. Knowledge-aware deep dual networks for text-based mortality prediction. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, pages 1406–1417. IEEE, 2019.

[391] Flávio Monteiro, Fernando Meloni, José Augusto Baranauskas, and Alessandra Alaniz Macedo. Prediction of mortality in intensive care units: a multivariate feature selection. *Journal of Biomedical Informatics*, 107:103456, 2020.

[392] Aya Awad, Mohamed Bader-El-Den, James McNicholas, Jim Briggs, and Yasser El-Sonbaty. Predicting hospital mortality for intensive care unit patients: time-series analysis. *Health informatics journal*, 26(2):1043–1059, 2020.

[393] Hans-Christian Thorsen-Meyer, Annelaura B Nielsen, Anna P Nielsen, Benjamin Skov Kaas-Hansen, Palle Toft, Jens Schierbeck, Thomas Strøm, Piotr J Chmura, Marc Heimann, Lars Dybdahl, et al. Dynamic and explainable machine learning prediction of mortality in patients in the intensive care unit: a retrospective study of high-frequency data in electronic patient records. *The Lancet Digital Health*, 2(4):e179–e191, 2020.

[394] Fuhai Li, Hui Xin, Jidong Zhang, Mingqiang Fu, Jingmin Zhou, and Zhexun Lian. Prediction model of in-hospital mortality in intensive care unit patients with heart failure: machine learning-based, retrospective analysis of the mimic-iii database. *BMJ open*, 11(7):e044779, 2021.

[395] Julian Theis, William L Galanter, Andrew D Boyd, and Houshang Darabi. Improving the in-hospital mortality prediction of diabetes icu patients using a process mining/deep learning architecture. *IEEE Journal of Biomedical and Health Informatics*, 26(1):388–399, 2021.

[396] Chonghui Guo, Mucan Liu, and Menglin Lu. A dynamic ensemble learning algorithm based on k-means for icu mortality prediction. *Applied Soft Computing*, 103:107166, 2021.

[397] Zina M Ibrahim, Daniel Bean, Thomas Searle, Linglong Qian, Honghan Wu, Anthony Shek, Zeljko Kraljevic, James Galloway, Sam Norton, James TH Teo, et al. A knowledge distillation ensemble framework for predicting short-and long-term hospitalization outcomes from electronic health records data. *IEEE Journal of Biomedical and Health Informatics*, 26(1):423–435, 2021.

[398] Jonathan Montomoli, Luca Romeo, Sara Moccia, Michele Bernardini, Lucia Migliorelli, Daniele Berardini, Abele Donati, Andrea Carsetti, Maria Grazia Bocci, Pedro David Wendel Garcia, et al. Machine learning using the extreme gradient boosting (xgboost) algorithm predicts 5-day delta of sofa score at icu admission in covid-19 patients. *Journal of Intensive Medicine*, 1(02):110–116, 2021.

[399] Min Hyuk Choi, Dokyun Kim, Eui Jun Choi, Yeo Jin Jung, Yong Jun Choi, Jae Hwa Cho, and Seok Hoon Jeong. Mortality prediction of patients in intensive care units using machine learning algorithms based on electronic health records. *Scientific reports*, 12(1):7180, 2022.

[400] Shinya Iwase, Taka-aki Nakada, Tadanaga Shimada, Takehiko Oami, Takashi Shimazui, Nozomi Takahashi, Jun Yamabe, Yasuo Yamao, and Eiryo Kawakami. Pre-

diction algorithm for icu mortality and length of stay using machine learning. *Scientific Reports*, 12(1):12912, 2022.

[401] Alaleh Azhir, Soheila Talebi, Louis-Henri Merino, Yikuan Li, Thomas Lukasiewicz, Edgar Argulian, Jagat Narula, and Borislava Mihaylova. Behrtday: Dynamic mortality risk prediction using time-variant covid-19 patient specific trajectories. In *AMIA Annual Symposium Proceedings*, volume 2022, page 120. American Medical Informatics Association, 2022.

[402] William Caicedo-Torres and Jairo Gutierrez. Iseeu2: Visually interpretable mortality prediction inside the icu using deep learning and free-text medical notes. *Expert Systems with Applications*, 202:117190, 2022.

[403] Jean-Roger Le Gall, Philippe Loirat, Annick Alperovitch, Paul Glaser, Claude Granthil, Daniel Mathieu, Philippe Mercier, Remi Thomas, and Daniel Villers. A simplified acute physiology score for icu patients. *Critical care medicine*, 12(11):975–977, 1984.

[404] William A Knaus, Jack E Zimmerman, Douglas P Wagner, Elizabeth A Draper, and Diane E Lawrence. Apache—acute physiology and chronic health evaluation: a physiologically based classification system. *Critical care medicine*, 9(8):591–597, 1981.

[405] Jean-Roger Le Gall, Stanley Lemeshow, and Fabienne Saulnier. A new simplified acute physiology score (saps ii) based on a european/north american multicenter study. *Jama*, 270(24):2957–2963, 1993.

[406] William A Knaus, Elizabeth A Draper, Douglas P Wagner, and Jack E Zimmerman. Apache ii: a severity of disease classification system. *Critical care medicine*, 13(10):818–829, 1985.

[407] Edward Choi, Mohammad Taha Bahadori, Joshua A Kulas, Andy Schuetz, Walter F Stewart, and Jimeng Sun. Retain: An interpretable predictive model for healthcare using reverse time attention mechanism. *arXiv preprint arXiv:1608.05745*, 2016.

Appendices

Appendix A

A.1 MIMIC-III and eICU Databases

The EHRs being used are two real-world large medical databases, the Medical Information Mart for Intensive Care (MIMIC-III[1]) Database and eICU[2] Collaborative Research Database.

MIMIC-III is one of the largest publicly available intensive care unit (ICU) databases, comprising 38,597 distinct patients and a total of 53,423 ICU stays. eICU is a multi-center intensive care unit database comprising medical records of 200,859 patients collected from 208 critical care units in the United States between 2014 and 2015. These two databases contain all information relating to patients admitted to intensive care units. Detailed information on the two databases can be found in the literature [38, 280].

A total of 21,139 samples were taken from the MIMIC-III database [53]. A total of 38,056 samples were taken from the eICU database [54]. For the MIMIC-III database, a total of 17 physiological variables (e.g., systolic blood pressure, diastolic blood pressure, respiratory rate, heart rate. As shown in Figure 1.1) were selected on the basis of the literature [53]. Similarly, for the eICU database, a total of 16 physiological variables were selected on the basis of the literature [54]. The tables A.1 and A.2 below illustrate the missingness proportion of different categories of physiological variables. The selected physiological variables are a subset of the MIMIC-III and eICU databases for a wide range

[1] https://mimic.physionet.org
[2] https://eicu-crd.mit.edu/

of benchmark tasks, such as modeling the risk of physiologic decline and in-hospital mortality (to be detailed in the next section).

The characteristics of physiological variables are listed as follows:

- Each physiological variable can be considered as a sequential dynamic feature that records an independent observation, while a set of features with the same timestamp can represent the patient's health status at that time during an ICU stay.
- For each patient, all timestamped sequential dynamic features form a chain of data providing the context of the entire duration of hospitalization (i.e., one patient journey), which should be taken into consideration in a holistic manner for predictive modeling.
- Each patient journey data can also be considered one multivariate clinical time series data with more than one time-dependent variable (i.e., a series of physiological variables).
- Each variable not only depends on its past values (i.e., long-term dependencies) but also has some dependency on other variables (i.e., present interdependencies) [3].

A.2 EHR-based Prediction Tasks

Based on the publicly available MIMIC-III [38] and eICU [280] databases, researchers have created benchmark datasets and proposed four benchmarks/tasks [53, 54]. Specifically, the proposed four tasks include modeling the risk of physiologic decline (i) and in-hospital mortality (ii), and estimating hospital length of stay (iii), as well as classifying phenotype (iv). More specifically, physiologic decline prediction and in-hospital mortality prediction are binary classification tasks, hospital length of stay prediction is a regression task, and phenotype classification is a multi-class classification task.

In this section, we focus mainly on predicting the mortality risk of patients based on their historical EHR data. According to the literature [53, 54], in-hospital mortality risk prediction is defined as predicting the mortality risk of patients based on the data from the first 48 hours after ICU/eICU admission.

It is also worth noting that the physiologic decline prediction is defined as predicting the mortality risk of patients based on the data from the first 24 hours after ICU/eICU admission. Together, these two risk prediction tasks can be seen as predicting patients' "long-term" and "short-term" mortality risks.

Outside the above four benchmarks/tasks, researchers have investigated disease risk prediction [369–371] and risk-of-readmission prediction [372–374]. Overall, these dis-

ease risk prediction and risk-of-readmission prediction are also classification-based risk predictions. Examples of research into disease risk prediction include cardiovascular disease risk prediction [375–377], heart failure risk prediction [378–380], and diabetes risk prediction [381–383]. Representative risk-of-readmission prediction methods include recurrent neural network-based method [384], statistical learning method [34], and graph-based method [35].

Table A.1: The 17 physiological variables selected from the MIMIC-III database.

Feature	Data Type	Missingness (%)
Capillary refill rate	categorical	99.78
Diastolic blood pressure	continuous	30.90
Fraction inspired oxygen	continuous	94.33
Glasgow coma scale eye	categorical	82.84
Glasgow coma scale motor	categorical	81.74
Glasgow coma scale total	categorical	89.16
Glasgow coma scale verbal	categorical	81.72
Glucose	continuous	83.04
Heart Rate	continuous	27.43
Height	continuous	99.77
Mean blood pressure	continuous	31.38
Oxygen saturation	continuous	26.86
Respiratory rate	continuous	26.80
Systolic blood pressure	continuous	30.87
Temperature	continuous	78.06
Weight	continuous	97.89
pH	continuous	91.56

Table A.2: The 16 physiological variables selected from the eICU database.

Feature	Data Type	Missingness (%)
Diastolic blood pressure	continuous	33.80
Fraction inspired oxygen	continuous	98.14
Glasgow coma scale eye	categorical	83.42
Glasgow coma scale motor	categorical	83.43
Glasgow coma scale total	categorical	81.70
Glasgow coma scale verbal	categorical	83.54
Glucose	continuous	83.89
Heart Rate	continuous	27.45
Height	continuous	99.19
Mean arterial pressure	continuous	96.53
Oxygen saturation	continuous	38.12
Respiratory rate	continuous	33.11
Systolic blood pressure	continuous	33.80
Temperature	continuous	76.35
Weight	continuous	98.65
pH	continuous	97.91

Appendix B

B.1 ICU mortality risk prediction

A considerable literature has been published around the theme of ICU mortality risk prediction [5, 32, 33, 39, 64, 150, 385–402]. Representative ICU mortality risk prediction models include [5, 32, 39, 150, 385, 386, 389]. It is worth mentioning that the traditional SAPS [403] and APACHE [404] scores, as well as their variants SAPS II [405] and APACHE II [406] scores, are mainly used for assessing the severity of the health condition as defined by the probability of patient mortality.

The study by [385] introduced the Super Learner Algorithm (SICULA) to predict mortality risk for ICU patients. The SICULA is an ensemble machine-learning framework that comprises a series of traditional machine-learning models, such as generalised linear models. Experimental results on the MIMIC-II dataset demonstrate that SICULA outperforms the traditional SAPS-II and APACHE-II scores.

Similarly, the study by [386] proposed an ensemble machine learning framework (EMPICU Random Forest) to predict the mortality risk of patients based on data from the first 24 hours and 48 hours after ICU admission. Experimental results on the MIMIC-II dataset demonstrate that EMPICU Random Forest outperforms the traditional SAPS-I and APACHE-II scores, random forests, decision trees, etc.

The ICU-LSTM [389] is proposed to take both sequential and non-sequential features as inputs for ICU mortality risk prediction. The former refers to vital signs, while the latter refers to the previous ICD-10 diagnosis codes. ICU-LSTM is built with the Long

short-term memory (LSTM) units [145]. Experimental results on the Asan Medical Center (AMC) ICU dataset demonstrate that ICU-LSTM outperforms the traditional logistic regression model.

The study by [39] proposed GRU-D to model the long-term temporal dependencies in multivariate clinical time series and utilize the decay mechanism to learn the impact of varying time intervals. GRU-D [39] is built upon Gated Recurrent Unit (GRU) [147]. The GRU is a variant of recurrent neural networks featuring a reset gate and an update gate, which control the flow of information between the hidden state and the current input. Experimental results on the MIMIC-III and PhysioNet datasets demonstrate that GRU-D achieves superior performance over the state-of-the-art models on mortality risk prediction.

The study by [150] proposes a deep learning predictive framework (shorten for ConCare) based on stacked recurrent neural networks. The ConCare mainly utilizes a multi-channel GRU architecture to model long-term temporal dependencies of multivariate clinical time series data [53]. With the construction of multi-channel GRU architecture, each univariate time series is modeled by a standard implementation of GRU. Each univariate time series is a time series that contains a single clinical variable recorded sequentially over time increments. Experimental results on the MIMIC-III dataset demonstrate the effectiveness and superiority of ConCare in the mortality risk prediction compared to the existing deep prediction methods such as RETAIN [407] and T-LSTM [148].

The study by [32] proposes a deep Markov model (shorten for AttDMM) by integrating hidden Markov models (HMMs), neural networks, and attention mechanisms. The AttDMM pays attention to improving the prediction performance of an HMM by incorporating neural networks into the HMM structure, then using the HMM hidden state representation (i.e., the HMM output) directly to make ICU mortality prediction. Experimental results on the MIMIC-III dataset demonstrate the effectiveness and superiority of AttDMM in the mortality risk prediction compared to a standard implementation of HMMs and deep prediction methods such as and ICU-LSTM.

The study by [5] proposes a novel deep neural network with a modular structure (shorten for TAttNet) to carry out health risk prediction tasks using EHR data. The TAttNet mainly models long-term dependencies and short-term correlations of multivariate clinical time series data with deep neural network-based modules. It is worth noting that the previous ICD-9 diagnosis and procedure codes are modeled as auxiliary information. Experimental results on the MIMIC-III dataset demonstrate the effectiveness and superiority of TAttNet in the mortality risk prediction compared to existing deep prediction methods such as RETAIN [407] and T-LSTM [148].