ABSTRACT

SUN, YANG. Machine Learning Methods And Their Application In EHR Data. (Under the direction of Dr. Yi-Hui Zhou).

Electronic health records (EHR) have been widely adopted, and have the potential to greatly improve clinical services, diagnostics and to inform clinical decision making process. Though machine learning algorithms have been widely recognized and have shown promising performance in the domain of EHR data, researchers face multiple challenges to progress and achieve better performance. Methods for pre-processing EHR data, missing data imputation and conducting individual causal effect estimation remain underdeveloped. In this dissertation, we develop data pipelines and machine learning algorithms to address the above analytical challenges in EHR data. First, we propose methods that build a pipeline to extracts EHR data from public available electronic ICU databases, which made complex database accessible to the machine learning community. We improve the in-hospital mortality prediction accuracy using a combination of neural network missing data imputation approach and decision tree based outcome prediction algorithm. Second, we improve the predictive accuracy of representation learning and adversarial networks in the estimation of individual treatment effects via introducing a structure keeper which maintains the correlation between the baseline covariates and their corresponding representations in the high dimensional space. We train a discriminator at the end of representation layers to trade off representation balance and information loss and show that the proposed discriminator minimizes an upper bound of the treatment estimation error. By also considering the correlations between the learned representation space and the original covariate feature space, we are able to address the tradeoff between distribution balance and information loss. We conduct experiments with simulated data and real-word observational data. Our proposed Structure Maintained Representation Learning (SMRL)

algorithm outperforms state-of-the-art methods. We demonstrate the above algorithms on the real EHR data from MIMIC-III database. Last, we propose a generative conditional approach to impute the missing values of a given dataset. The generative conditional missing imputation networks (GCMI) presents theoretical properties under the MCAR and MAR mechanism. In addition, we incorporate the multiple imputation via chained equation-based approach to increase the model stability and imputation performance. We conduct extensive simulation and real data experiments on benchmark datasets to show the superior performance of the proposed methods against other state-of-the-art missing imputation methods.

Machine Learning Methods And Their Application In EHR Data

by
Yang Sun

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Statistics

Raleigh, North Carolina
2024

APPROVED BY:

Dr. Wenbin Lu	Dr. Sujit Ghosh
Dr. Marie Davidian	Dr. Yi-Hui Zhou Chair of Advisory Committee

BIOGRAPHY

Yang Sun was born in Taiyuan, Shanxi in China. He was admitted to Zhejiang University in 2013 and obtained his Bachelor's degree in Statistics. In the fall of 2017, he entered the Department of Statistics at North Carolina State University. His research topics include missing data imputation, machine learning and deep learning. He successfully defended his dissertation in 2023.

ACKNOWLEDGEMENTS

Foremost, I extend my heartfelt appreciation to Dr. Yi-hui Zhou, my esteemed academic advisor, for her patient guidance, continual encouragement, and generous support throughout the entirety of my Ph.D. journey. I am thankful for the invaluable feedback and insightful suggestions she provided regarding my research project, both in our face-to-face interactions and during remote meetings.

I extend my gratitude to the other distinguished members of my dissertation committee, namely Dr. Wenbin Lu, Dr. Sujit Ghosh, and Dr. Marie Davidian, for dedicating their time to serve on my committee and for imparting valuable feedback.

I acknowledge with gratitude the contributions of many other wonderful people at NCSU who significantly enriched my academic experience. Dr. Jacqueline Hughes-Oliver, Dr. Ryan Martin, and Dr. Soumendra Lahiri deserve special mention for delivering outstanding courses that laid the groundwork for my Ph.D. research. The faculty members in the Department of Statistics are appreciated for providing a comprehensive array of courses, and I extend my thanks to all the staff members for their exceptional service and timely assistance. I am also grateful to my peers in the Ph.D. program at NCSU, whose presence made my academic and personal life vibrant and diverse.

I extend my appreciation to my summer intern mentor and peers at Meta, Dr. Xiangfei Meng, Mohan Yang, and Siyun Yang. Their mentorship not only enhanced my practical skills in data science and machine learning but also offered valuable insights into my career development.

Lastly, my deepest love and gratitude go to my parents. I am and will always be proud to be their son. I express thanks to my relatives for their unwavering love and support throughout this journey.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER

1

A MACHINE LEARNING PIPELINE FOR MORTALITY PREDICTION IN THE ICU

1.1 Summary

Mortality risk prediction for patients admitted into the intensive care unit (ICU) is a crucial and challenging task, so that clinicians are able to respond with timely and appropriate clinical intervention. This becomes more urgent under the background of COVID-19 as a global pandemic. In recent years, electronic health records (EHR) have been widely adopted, and have the potential to greatly improve clinical services and diagnostics. However, the large proportion of missing data in EHR poses challenges that may reduce the accuracy of

prediction methods. We propose a cohort study that builds a pipeline that extracts ICD-9 codes and laboratory tests from public available electronic ICU databases, and improve the in-hospital mortality prediction accuracy using a combination of neural network missing data imputation approach and decision tree based outcome prediction algorithm. We show the proposed approach achieves a higher area under the ROC curve, ranging from 0.88-0.98, compared with other well-known machine learning methods applied to similar target population. It also offers clinical interpretations through variable selection. Our analysis also shows that mortality prediction for neonates was more challenging than for adults, and that prediction accuracy decreases as patients stayed longer in the ICU.

1.2 Introduction

The intensive care unit (ICU) admits patients with life-threatening injuries and the most severe illnesses (1). Patients in the ICU receive much more clinical care than other patients, requiring consistent monitoring and are at high risk of death. For example, under the background of COVID-19 as a global pandemic, the mortality rates of ICU admitted COVID-19 patients are between 30%-50%. Among the various aspects of clinical research, ICU mortality prediction plays an important role as it not only helps researchers to identify the patients in danger, but in principle can save ICU beds for patients most in need (2; 3). Therefore, building accurate mortality models and identifying important risk factors are needed more than ever to inform clinical decisions and answer urgent research questions.

Existing literature developed score systems to pre-process data using baseline patient characteristics. For example, the Acute Physiology And Chronic Health Evaluation system (APACHE)(4) and the Simplified Acute Physiology Score (SAPS) (5; 6) mostly reply on clinical expertise for variable selection and importance weights. However, none of these models are consistently suitable for all patients in ICUs due to insufficient accuracy or lack of generality

(7). In order to increase the accuracy of mortality prediction, researchers have designed various models and scoring systems with specific requirements. For example, Moridani et al. (8) proposed a mortality scoring system to predict heart disease. Although the model resulted in acceptable results when risks are predicted at an early stage, the sample size in this study was too small to represent the target population (only 90 selected patients), and only one disease factor was considered.

Electronic health records (EHR) contain valuable information such as demographics, lab test, vital signs and disease diagnosis code. In the United States, more than 30 million hospital patient visits happen each year, for which 83% generate electric health records. For researchers, data mining using such a rich data source could potentially provide deeper understanding and achieve satisfactory prediction results. One common challenge in utilizing electric health records is that data are not always available in a clean and tidy format, and much of the primary work is to prepare the data into a standard form amenable to analysis. This issue is imperative for ICU prediction because the clinical decisions have to be made in a timely matter.

Analysis on EHR data could be significantly challenging since there is considerable missing data. The characteristics of the patient records vary greatly among heterogeneous patients, which results in inconsistent and incomplete data. Improper dealing with missing data can lead to significant biased results, especially for data with complex structure. This issue is growing as EHR data become more widespread and datasets increase in scope and size. In EHR data, missing values usually frequently outnumber non-missing values, with missing rates ranging from 20% to 80% (9). In most EHR data, non-missing features may account for part of the missingness patterns among different patients. Therefore, in principle much of the missingness might be “recoverable" in an approximately unbiased manner, given a sufficiently sophisticated approach to imputation. Numerous methods have been developed to impute missing values. These include spectral analysis (10), kernel

methods (11), the EM algorithm (12) for certain models, matrix completion (13) and matrix factorization (14), to name a few. Multiple imputation (15; 16) can be further applied with above missing imputation methods to reduce the variance, by repeating the imputation steps multiple times and taking the average of the results, which also provides estimates of uncertainty. However, existing generative imputation methods have limitations. The EM methods require assumptions about the distributions and are cumbersome to apply to mixed features including both categorical and continuous data, which are very common in EHR data (e.g. ICD-9 codes and laboratory test values). Recent developments in deep learning have shown promising alternatives. For example, the generative adversarial imputation nets (GAIN) (17) train a generator and discriminator aversarially to impute missing data according to the true underlying distribution.

In this paper, we build an accurate ICU prediction pipeline that extracts ICD-9 codes and laboratory tests from the Multiparameter Intelligent Monitoring in Intensive Care (MIMIC) III database. Our model handles large numbers of patients with various diseases and significant missing data. We show that the proposed prediction model based on LightGBM outperforms other ICU mortality prediction models reported in literature under similar target population.

1.3 Materials and Methods

1.3.1 MIMIC Electronic Health Records data

The Multiparameter Intelligent Monitoring in Intensive Care (MIMIC) (18) is a public critical care database which includes all patients admitted to the ICUs of Beth Israel Deaconess Medical Center in Boston, MA since 2001. Three major versions of MIMIC database are available for research analysis. The MIMIC-II includes 24,508 adult patients (age> 15,

as defined by the MIMIC database) from 2001 to 2008, and MIMIC-III augments it with the data collected between 2008 - 2012, resulting in a total of 38,597 unique adult patients and 7870 neonates. In addition, the data in MIMIC-III is more reliable and complete to facilitate further research than in MIMIC-II (19). The MIMIC database provides information about patients' demographics, diagnosis codes, laboratory tests, and clinical events, for over 350 million values across various sources of data. Among the collected values, the laboratory test is one of the most important data sources, constituting more than 90% of total events. Table 1.1 summarizes the patient characteristic in different versions of MIMIC database. Noted that the identifier has been changed across versions. Although SUBJECT_ ID (unique to a patient) has been kept consistent between MIMIC-II and MIMIC-III, HADM_ ID (unique to a patient hospital stay) and ICUSTAY_ ID (unique to a patient ICU stay) have been regenerated and will not match between the two databases. Also, some item-ID mapping and Schema have changed, and many new tables have been added into MIMIC-III.

We use MIMIC-III database to build a machine learning pipeline for ICU patients' in-hospital mortality prediction, which includes 38,597 unique adults and 7,870 neonates. Separate models are built for adults and neonates due to their distinct event rates, ICD-9 diagnoses and covariate distributions. To protect patients' privacy, all patients' data are de-identified, and the reported times of the events are randomly drifted for each patient so that only the relevant time intervals within clinical events are known. There are 6,984 features among ICD-9 codes, 726 features among laboratory tests and tens of other demographics features. After feature engineering, a total of 601 features including demographics (age and gender), diagnosis codes (ICD-9 Code) and laboratory tests are selected in the machine learning models (Fig 1.1). ICD-9 Code documents diagnoses and procedures associated with hospital utilization. For neonates' ICD-9 records, we removed the V codes because they are designed for occasions when circumstances other than a disease or injury result in an encounter or are recorded by providers as problems or factors that influence care.

Table 1.1: Adult patient characteristics in different versions of MIMIC database.

Adult patient statistics	MIMIC-II	MIMIC-III
Distinct adult patients	24,508	38,597
Hospital admissions	26,870	49,785
Distinct ICU stays	31,782	55,423
Age (yrs)	65.5 (51.9, 77.7)	65.8 (52.8, 77.8)
Gender (male)	17,857 (56.2%)	27,983 (55.9%)
ICU length of stay (days)	2.1 (1.1, 4.3)	2.1 (1.2, 4.6)
Hospital length of stay (days)	7 (4, 13)	6.9 (4.1, 11.9)
Hospital mortality	3,092 (11.5%)	5,748 (11.5%)

1.3.2 Data preprocessing and Feature Engineering

ICU patients cohort

All patients in the MIMIC-III database are included, resulting in 38,597 unique adult patients, and 7,870 unique neonates. For each patient, we include their first admission to the ICU.

ICD-9 code

Each patient's ICD-9 codes are documented at the end of ICU stay, including information during the whole period, however, they are not supposed to contain the information after discharge. We extract and calculate the frequency of distinct ICD-9 codes appeared in the database. The number of ICD-9 codes can be as high as tens of thousands (14,567 by definition and 6,984 in our data), which could cause not only low predictive performance but also memory issues. The observed distribution of ICD-9 codes shows that some high frequent codes dominant the whole diagnoses dataset. As shown in Table A.1, the top 10 common ICD-9 codes cover 76.9% of the dataset and the top 50 cover 93.6%. Thus we consider to use the adjusted ICD-9 categories to reduce the feature dimension. The first 3 digits contain the general condition of a patient so that have been commonly used to

represent disease categories (20). Therefore, We follow this tradition and convert features into 1070 dummy variables (one-hot encoding) as indicators for each small group of disease codes. The last two line of Table A.1 show after grouping the ICD-9 codes into categories, the top 10 and top 50 most common categories cover 84.2% and 96.8% separately (21).

Although we already reduced the dimension of diagnosis code by extracting the first 3 or 4 digits, some features remain to be too sparse, which may cause common problems such as increasing the space and time complexity of models, causing algorithms to behave in unknown ways due to overfitting. To further simplify the feature space, we drop the adjusted ICD-9 codes which were documented on fewer than 48 patients, following our observation that the median of the 3-digit frequency in the database was 48.5. In this manner we are able to reduce the diagnosis code space to 535 unique values (as in Fig 1.1).

Laboratory tests

The Lab table in MIMIC-IV spans more than just the patient's ICU stay, and in fact covers their whole hospital stay (and sometimes includes outpatient stays). As a result, the extracted laboratory test values allows the possibility to predict the patient's mortality rate before admission to the ICU.

Among the 726 unique laboratory test features, we select only the tests documented after the patient's first inpatient admission. We also aggregate the laboratory test into each day. If there are more than one same laboratory test for a patient within the same day, we take their average values. If a laboratory test does not exist for a patient in one day, we treat it as a missing value. After the initial aggregation, our lab data are organized into a 3-dimensional patient-lab array, representing patient ID, date and lab test features, separately.

For ICU patients, there are no universal standard rules for prescription of laboratory tests, resulting in a large and sparse lab feature space. Frassica (22) in 2005 compared 45,188 lab tests and profiles in three ICUs: surgical ICU, pediatric ICU, and medical ICU,

and discovered that 80% of the tests and profiles in the three ICUs are covered by < 25 tests. Sharafoddini (23) combined the previous information and modified the lab tests items based on the Medical Information Mart for Intensive Care III database, and then selected 36 lab items for predicting in-hospital mortality. Inspired by the above analyses, we hypothesized that using more laboratory test items might not improve the prediction accuracy, as the three-dimension patient-lab-day array is extremely sparse and have a high missing data rate. Fig A.1 presents the missing proportion of 36 common laboratory tests. Many lab features have missing rate greater than 50%, and the missing rates can be as high as 80%, potentially posing important downstream effects on accuracy for some prediction methods.

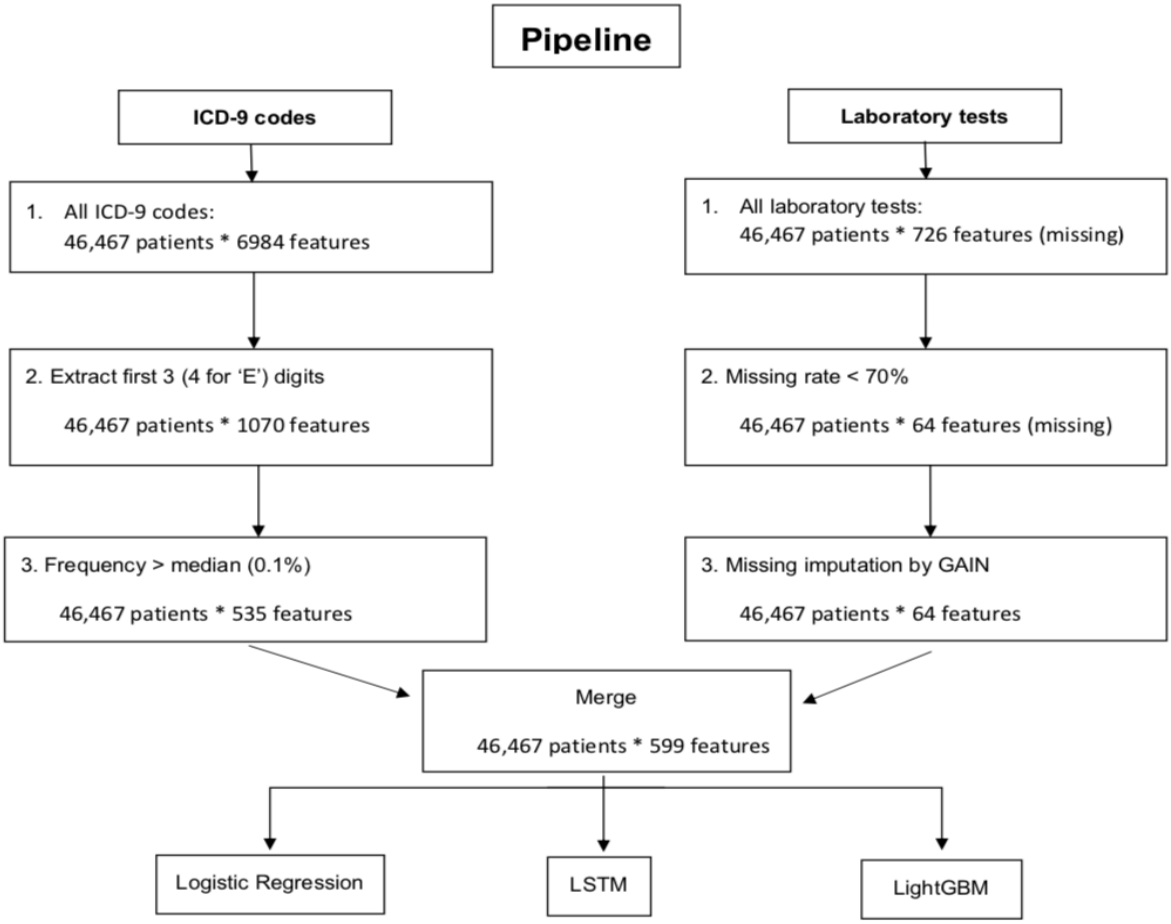
To reduce the dimension of laboratory test features, we first take the average of the patient's test value prior to the target (response) day as the patient's corresponding laboratory test value. In this manner, the sparse time series of laboratory tests are transformed into scalar values. As a result, the 3-dimensional patient-lab array becomes a patient-lab matrix. While we mitigate the sparsity problem of laboratory tests, the missing rates sill remain problematic.

Second, we filter out the laboratory tests with missing rate greater than 70% (as being effectively uninformative). This reduces the dimension of laboratory items to 64. The remaining missing data are imputed to improve the prediction accuracy. Details about missing data imputation are provided in Section 1.3.3.

Data Normalization

Some of the prediction methods are sensitive to the scale of values, and data normalization provides a more uniform effective weight. We re-scale all the features using the

Figure 1.1: Feature Engineering Pipeline of MIMIC-III database.



min-max normalization technique,

$$X_i = \frac{X_{i,raw} - min(X_{i,raw})}{max\left(X_{i,raw}\right) - min(X_{i,raw})} \tag{1}$$

where X_i is the scaled value for ith feature, and $X_{i,raw}$ is the original value. Other approaches are possible, but min-max scaling has the advantage of preserving interpretability for binary variables, as [0,1] maps to [0,1] after scaling.

The missing imputation step is for recovering training data, so that the machine learning models would be trained better to capture the structure of the features if proper imputation

methods are applied. In training steps, we applied GAIN on training data to obtain the complete dataset for model training. In prediction steps, we combine the original training data and the prediction data, and then apply GAIN on the whole dataset to obtain the complete prediction features as input of the models.

1.3.3 Missing Data Imputation via GAIN

The goal of imputation in the mortality prediction context is not to provide unbiased estimates of parameters governing the data structure, as is often the case in formal missing data methods, but to use missing data imputation as a device to increase prediction accuracy for ICU outcome. However, the general rationale behind missing data imputation is valid for prediction even if the predicted values are biased, as prediction does not use the final outcome measure. Here we operate under the standard assumption of missing-at-random as in the missing data literature.

Generative Adversarial Imputation Nets (GAIN) (17) is a missing data imputation algorithm based on the well-known Generative Adversarial Nets (GAN) framework. The generator and discriminator are trained adversarially to learn the desired missing data distribution. GAIN was defined for irregularly-sampled temporal data and has been shown to substantially outperform previous methods including Multiple Imputation by Chained Equations, Matrix Completion and Expectation Maximization on a variety of UCI datasets (17). A review paper for deep learning methods for medical time series data by Sun et al. also suggested superiority of GAN based missing data imputation methods for EHR data (24).

Let $X = (X_1, ..., X_D)^T$ be a vector of random variables (either continuous for laboratory tests or binary for re-coding of ICD-9 codes), whose distribution follows $P(X)$. In the presence of missing values, we use a mask vector $M = (M_1, ..., M_D)^T$ with $M_d \in \{0, 1\}$ for $d = 1, \ldots, D$ to inform the generator which values are missing or are present.

The generator in GAIN, G, takes realized values of X, M and a random noise vector Z as input, and outputs a vector of imputed values $\hat{X}$. The noise vector Z is introduced to add randomness to imputed values so they're not the same every time, and thus avoid model overfitting.

The discriminator of GAIN, D, is used as an adversary of G to determine whether a value is observed or imputed. In the discriminator, a hint vector H is further supplied to ensure that the discriminator forces the generator to generate the desired distribution.

The loss of GAIN is defined by

$$V(D,G)=\mathbb{E}_{\hat{X},M,H}\Big[M^T \log D(\hat{X},H)+(1-M)^T \log\big(1-D(\hat{X},H)\big)\Big],$$

where log is element-wise logarithm and $V(D,G)$ depends on G through the corresponding imputed clinical features matrix $\hat{X}$. The training objective of GAIN is a minimax problem given by $\min_G \max_D V(D,G)$.

We apply GAIN on the whole dataset to obtain the complete feature dataset as input of the subsequent outcome prediction models. Fig 3.2 summarizes the pipeline of our missing data imputation procedure.

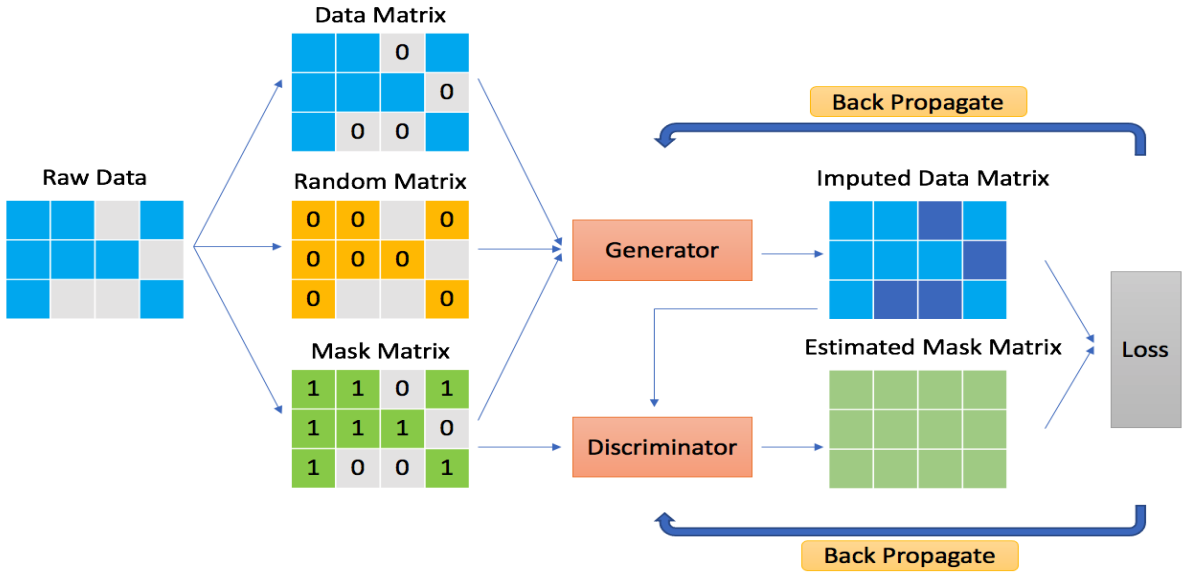
1.3.4 Outcome Prediction Models Under Comparison

We compare a number of binary outcome prediction models for the ICU mortality prediction task, including several popular machine learning methods. These models are briefly reviewed as follows.

1.3.5 Regularized logistic regression

We consider logistic regression as a baseline to predict the mortality probability. To avoid over-fitting, regularization is used to control the complexity of the model by penalizing

Figure 1.2: Pipeline of the missing data imputation procedure. The zeros in the Mask Matrix indicate the positions of missing data, while ones indicate the observed data. Data Matrix is generated from Raw Data with missing values imputed by 0. Random Matrix includes random noise at the missing positions and 0 otherwise.



large regression coefficients. The loss function of regularized logistic regression with L_K penalty is

$$V(\big(h_\theta(X), y\big) = -y\log(h_\theta(X)) - (1-y)\log(1-h_\theta(X)) + \sum_{k=1}^{K}\lambda_k\|\theta\|_k^k,$$

where $h_\theta(X)$ is the predicted probability of event y occurring given X, θ is a vector of the regression coefficients, and λ_k controls the amount of regularization applied to the model. When $K=1$, the loss function with L_1 penalty corresponds to Lasso logistic regression. When $K=2$ and $\lambda_1=0$, the loss function with L_2 penalty corresponds to Ridge logistic regression. Finally, when $K=2$ and $\lambda_1,\lambda_2\neq 0$, the loss function with weighted L_1 and L_2 penalty corresponds Elastic Net logistic regression. We compare the performance of these three different regularized logistic regression models for ICU mortality prediction. There is one tuning parameter for each of the Lasso and Ridge model (L_1 and L_2 penalty). For

Elastic Net, we set the same weight for L_1 and L_2 penalty term, so that it also has one tuning parameter. In the presence of large number of features, the L_1 penalty might be a better choice since it results in sparse coefficients and a simpler model with less features. For each tuning parameter of the penalty terms, a set of ten candidate values in a logarithm scale between $1e^{-4}$ and $1e^{4}$ is selected by cross-validation according to prediction accuracy.

LightGBM

Ke et al. (25) developed an efficient boosting decision tree using gradient-based one-side sampling (GOSS) and exclusive feature bundling (EFB), called LightGBM. LightGBM significantly improves the training time based on decision tree algorithms, while achieving the same or even better accuracy. To introduce the algorithm of LightGBM, we first review the Gradient Boosted Decision Trees (GBDT) method.

GBDT is a popular machine learning algorithm that iteratively constructs an ensemble of weak decision tree learners through boosting (26). Given a training set $\{(X_1, y_1), (X_2, y_2), \cdots, (X_n, y_n)\}$, where X are the feature samples and y are the mortality labels, the classification function $F(\boldsymbol{X}) = \sum_{m=1}^{T} f_m(\boldsymbol{X})$ is trained iteratively to minimizing the loss function $V(y, F(x))$:

$$\widehat{F} = \arg\min_{F} E_{X,y}[V(y, F(X))],$$

where T is the number of iterations, the newly added decision tree f_m is chosen to minimize the aggregated loss. However, if the data has a large sample size or high dimension, GBDT is not able to provide accurate and efficient results.

LightGBM combines exclusive features into one feature and reduce the time complexity of training LightGBM from $O(n * D)$ to $O(n * g)$ (27), where g is the group number of exclusive features. In addition, GOSS ignores the data with minor gradient so that we do not need to go through all data during the update steps. As a result, LightGBM has the ability

to achieve the same accuracy as GBDT, but with significantly reduced training time and memory usage with large datasets.

Recurrent neural network

As computational power increases, deep learning methods have become more powerful in recent years. Recurrent Neural Network (RNN) (28) arranges an input layer, several hidden layers, and an output layer sequentially to “recall" historical outcomes. This has been made possible through the structure design of hidden layers, in which the output of a previous layer is then used as a part of the input for the current layer. Schmidhuber and Hochreiter (29) proposed a special RNN model called Long short-term memory (LSTM) to capture subtle effects over long term memory. LSTM includes three gates besides the essential parts of RNN: the input gate quantifies the importance of the input information, the forget gate decides which information needs to be carried over, and the output gate generates the information of the next hidden layer. Together, these internal gates help to avoid the vanishing and exploding gradient problems that can occur with RNNs. The LSTM has been shown to work well for modeling sequential data. In this paper, we consider applying LSTM models to handle the correlations between features and longitudinal measurements from the laboratory tests. The last sigmoid layer outputs the predicted ICU mortality probability.

1.4 Experimental Settings

The data preprocessing and feature engineering steps described in Section 1.3.2 results in a total of 601 features (536 binary and 65 continuous). We apply logistic regression with L1 penalty, logistic regression with L2 penalty, logistic elastic net, LightGBM and LSTM separately for ICU mortality predictions. The performance is evaluated based on the area under the receiver-operator characteristic curve (AUROC), and the area under the precision-

recall curve for prediction accuracy (AUPRC). A high value of AUROC and AUPRC indicates the prediction model have adequate discrimination to distinguish patients that died in the hospital and those that did not. The AUPRC is more robust under highly skewed datasets (as the neonate cohort in this study), and give a more informative picture of an algorithm's performance (30). Various rules of thumb suggest that the AUC of a binary classifier should be greater than 0.8 or 0.9. We obtain estimated 95% confidence interval (CIs) of AUROC and AUPRC via 100 bootstrap samples. In addition to the adult and neonate cohort, we also want to examine the performance of methods on four adult sub-cohorts : patients total length of stay in ICU within 24 hours (adult within 1 day), total length of stay in ICU within 48 hours (adult within 2 day), total length of stay in ICU more than 24 hours (adult $>$ 1 day), and total length of stay in ICU more than 48 hours (adult $>$ 2 day). These subgroups are of clinical interest and have distinct underlying distributions. The prediction evaluation on these sub-cohorts would allow a more accurate classification and facilitate the understanding about potential differences of patients in different stages.

Comparison of the above 5 algorithms relies on 8-fold cross-validation on the patient level. For each patient, only the first ICU admission is used. That is, we randomly split the whole data set into 8 mutually exclusive blocks. The models are first trained on 7 blocks (training data set, including 87.5% of the cohort) and then used to predict on the rest of the 12.5% of the cohort (validation data set). This process is repeated 8 times to iterate through all 8 blocks. For each of the prediction algorithm, the hyper-parameters are selected via cross-validation to minimize the error, and performance measures are aggregated over all 8 iterations.

Table 1.2: The mortality rate, accuracy (ACC), AUROC, and AUPRC returned from LightGBM in different patient cohorts.

Groups	Patients	ICU	Alive	Dead	Death%	ACC	AUROC	AUPRC
Adult Total	38557		28675	9882	25.63%	82.47%	90.42%	87.22%
Adult >1 day	35474	>24 hr	26895	8579	24.18%	81.62%	89.00%	74.52%
Adult >2 day	33029	>48 hr	24957	8072	24.44%	80.95%	88.62%	73.60%
Adult 1 day	3083	0 - 24 hr	1780	1303	42.26%	89.78%	97.03%	96.50%
Adult 2 day	2445	24 - 48 hr	1938	507	20.74%	81.75%	89.78%	76.80%
Neonate Total	7649		7588	61	0.80%	95.01%	98.92%	63.66%

1.5 Results

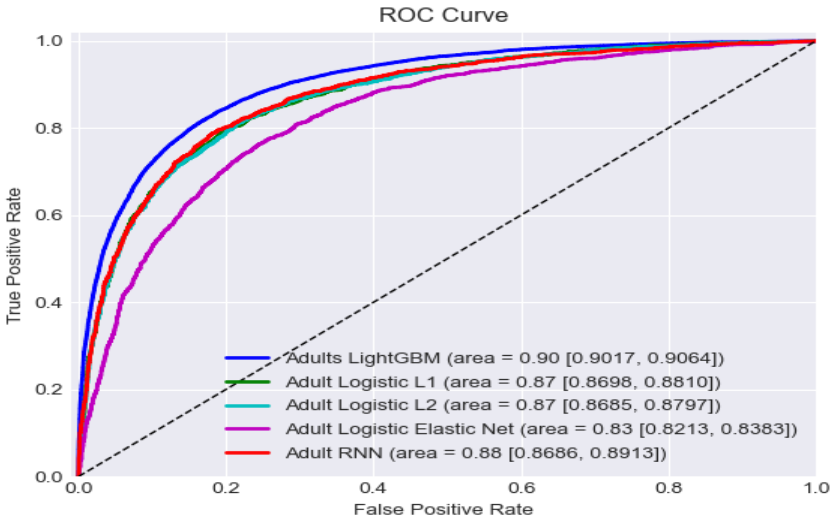
1.5.1 Comparison on the adult cohort

Table 1.2 summarizes the in-hospital mortality rates on different cohorts under study. The mortality rate of the overall adult cohort is 25.63%, with adults admitted to ICU within 1 day having a much higher mortality rate (42.26%) than those admitted after 1 day (24.18%). In contrast, the overall neonate cohort has a very low mortality rate (0.8%). Due to the skewness of this data, AUPRC is a more objective evaluation metric.

The performance of the 5 candidate algorithms on the overall adult cohort, as evaluated by AUROC and AUPRC, are presented in Fig 1.3 and Fig 1.4. The LightGBM achieves the highest prediction accuracy and discrimination result, with an average AUROC 90% (95%CIs : 0.902, 0.906) and AUPRC 87% (95%CIs : 0.8660, 0.8746). The RNN LSTM model performs slightly worse than LightGBM, with AUROC 88%(95%CIs : 0.869, 0.891) and AUPRC 83%(95%CIs : 0.822, 0.849). The logistic models have the lowest prediction accuracy, with L_1 and L_2 penalty achieve similar performance, with AUROC 87% (95%CIs : 0.870, 0.881) and AUPRC 82% (95%CIs : 0.816, 0.833). The logistic elastic net is the worst among all the methods, with AUROC 83% (95%CIs : 0.821, 0.838) and AUPRC 75% (95%CIs : 0.740, 0.768). These results are expected as LightGBM and RNN LSTM are non-parametric models involving

more hyper-parameters, and thus are more flexible to learn complex relationship between features and mortality status.

Figure 1.3: Cross-validation area under receiver operating characteristic curve (AUROC) for 5 candidate algorithms on the adult cohort.

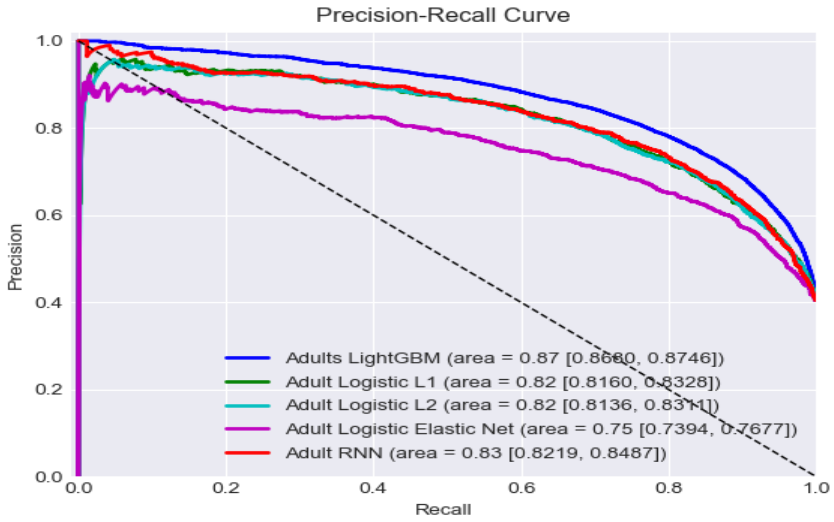


Among the models, LightGBM not only produces a more accurate prediction, but also provides a relatively interpretable model with feature selections. We examine the top 30 features identified by LightGBM (as in Fig A.2). The most important features are all laboratory tests including platelet count, red cell distribution width, alanine aminotransferase and blood urea nitrogen. None of the ICD-9 codes are selected by the model probably due to their relatively sparsity in the database.

1.5.2 Comparison among Sub-cohorts

Table 1.2 presents the AUROC and AUPRC for additional sub-cohorts using algorithm with the best performance (LightGBM). The AUROCs of adult > 1 day, adult > 2 day and

Figure 1.4: Cross-validation area under precision-recall curve (AUPRC) for 5 candidate algorithms on the adult cohort.



adult within 2 day sub-cohorts are similar to the adult overall cohort, achieving about 90% AUROC, but slightly lower AUPRC. The prediction accuracy in adult within 1 day sub-cohort is higher, with 97.0% AUROC and 96.5% AUPRC. The AUROC of neonates is 99.0%, much higher than 90.4% for adults, but the AUPRC is only 63.7%. This is due to the extremely low mortality rate in the neonates cohort.

1.5.3 Comparison with Other Literature

Several previous publications have also proposed mortality prediction models using the MIMIC database. Each study adopted different version of the data and unique inclusion criteria, resulting in sample sizes ranged from hundreds to tens of thousands. In addition, different evaluation metrics and specifically designed features have been chosen to suit their own models. Therefore, it is difficult to fairly compare the model performance across studies. To provide an approximate comparison to our results, we exclude studies with too specific cohort inclusion criteria, or without AUROC reported, or with sample size less

Table 1.3: Literature reported prediction performance on in-hospital mortality using MIMIC database.

Reference	# Patients	Model Type	Model	AUROC	MIMIC
(32)	14,739	Linear	LR	82%	II
Inclusion criteria: Have SAPS-I, LOS 24hr, first ICU stay only					
(33)	19,308	Linear	SVM	84%	II
Inclusion criteria: Age>18, >100 words across all notes					
(34)	9,683	Non-Linear	AutoTriage	88%	III
Inclusion criteria: Age>18, In MICU, >1 obs. for all features, 500hr $\geq LOS \geq 17hr$					
(31)	42,276	Non-Linear	LSTM	86%	III
Inclusion criteria: Age>18, only one ICU stay during the hospital admission					
(35)	24,508	Non-Linear	SuperLearner	88%	II
Inclusion criteria: Age>15					

than 9,000. Table 1.3 shows the literature reported prediction performance on in-hospital mortality using MIMIC database. Among these studies, Harutyunyan et al. (31) used the most similar data and criteria as our study. Our pipeline achieves a 4 percentage point increase compared with their results. Overall, while some of the reported AUROCs are close to 90%, our pipeline produces the highest AUROC for in-hospital mortality prediction compared with similar studies.

1.5.4 Validation of the pipeline using eICU database

The eICU Collaborative Research Database collects data from a combination of many critical care units throughout the continental United States. The data in the collaborative database covers 139,367 patients who were admitted to critical care units in 2014 and 2015. To validate the method of constructing the proposed pipeline and test the generalizability of the results, we rebuild the mortality prediction pipeline using adult patients (age > 15,

as defined in MIMIC-III) with their first admission records in the eICU database. Fig A.3 presents the corresponding data preprocessing and feature engineering pipeline. As a result, there are 117,456 unique patients and 436 features (389 ICD-9 categories, 45 laboratory tests and 2 demographic features), with in-hospital mortality rate 9.7%. Fig A.4 shows that our pipeline provides satisfactory prediction accuracy on this database. LightGBM still outperforms other methods, achieving the highest AUROC 0.92 (95%CI : 0.91, 0.92).

1.6 Conclusion and Discussion

We propose a pipeline that extracts ICD-9 codes and laboratory tests from public available electronic ICU databases, and improve the in-hospital mortality prediction accuracy using a combination of neural network missing data imputation and decision tree based algorithms. Among the 5 candidate algorithms, LightGBM results in the highest prediction accuracy. Our analysis also show that although the proposed model achieves a very high AUROC for neonates, the AUPRC is relatively low, indicating mortality prediction is more challenging for neonates than for adults due to the extreme low event probability. The predicted accuracy decreases as patients stay longer in the ICU. This is probably because the patients stayed longer are more heterogeneous and involve more complicated disease status. For our data, LightGBM has a higher AUROC compared with other well-known machine learning and deep learning methods, and it also refined the interpretable model by identifying the most important features.

Besides the improvement of AUROC, a precise mortality prediction before the patient enters ICU is much more meaningful when ICU resources are in shortage, especially during crises, as exemplified recently in the COVID-19 pandemic. Using our pipeline, we are able to predict severe ill patients' mortality probability as long as the needed laboratory test results are available, even if they are not admitted to the ICU. Thus extensions of this

work might therefore be used for allocation of scarce ICU resources. We also consider this manuscript to be useful for clinical support and decision-making. The top 30 features selected by LightGBM have been reported by clinical experts to be important indexes associated with mortality rate (23), which further validates the feature importance ranking returned from our LightGBM model. We note that resource allocation depends on a complex relationship between the likelihood of survival, and benefits to the patient or society at large, which will include considerations of remaining years of life. Thus our methods to improve prediction of neonatal mortality are of especial interest, as patient age serves as (effectively) a meaningful predictor of mortality, an important stratification variable, and an important indicator of patient benefit due to the remaining years of life. These aspects raise important ethical considerations that are beyond our scope. However, we believe our approaches serve as an essential input into these considerations. Further extensions to our approaches may also involve differing phenotype or predictors of outcomes, which may be useful in hypothesizing the effect of different treatments, as a step toward justifying experimental clinical trials.

Although we propose a mortality rate prediction pipeline that outperforms the literature reported performance, there are some limitations of this work. First, for MIMIC-III and eICU databases, the ICD-9 codes are documented at the end of patients' ICU stay. Therefore, the ICD-9 features include the information during the ICU stay, and there is no way to filter them based on time. We follow the approach used by El-Rashid et al.(36) and Huang et al.(21), which have used ICD-9 codes in MIMCI-III in a similar way to provide information about patients' disease information. However, if timestamp is accompanying the ICD-9 codes in other databases, the disease diagnosis could be used in a more flexible approach for dynamic prediction. Second, both MIMIC and eICU data are collected from US hospitals, which may cause generalization issues when trying to apply the models in non-US hospitals. However, the three types of features, demographics, diagnosis codes (ICD-9 Code:

International Classification of Diseases) and laboratory tests, are used worldwide. As a result, for non-US hospital settings, the integration of the proposed pipeline into local, national, and international healthcare systems will still be useful and save lives as long as the three types of data are collected, and the models are properly trained.

CHAPTER

2

METHODS

2.1 Summary

EHR plays an important role in improving patient-centered analysis. There is plenty of literature proposing suggestions from different aspects to help improve the quality of EHR data, however, completeness is still the fundamental one. EHR data cause a significant challenge for researchers as a result of massive missing data, which is not collected due to no intention or missing after the measurement. Not correctly dealing with missing data may lead to significant bias. Besides, many recently powerful analysis methods such as machine learning and deep learning methods require the complete data set, which would be prohibited by the incompleteness of EHR data. These problems are going to be more

relevant as EHR systems grow fast worldwide. Though the missing problem is noted by researchers, it is still substantial and unavoidable, and there is much literature working on this issue. In this work, we will compare the missing imputation methods for our EHR dataset.

In section 2 we conduct a literature review for 3 type of missing mechanisms: MCAR, MAR, and MNAR, and then summarize 7 methods of missing imputation methods: Data deletion, Mean/Median/Mode Imputation, Maximum Likelihood and Multiple imputation, Missing Forest, Generative Adversarial Imputation Nets (GAIN), Nuclear-norm Minimization ("SoftImpute"), Implicit Joint Modeling of Observed and missing data ("SoftImpute-concat"). In section 3, we conduct simulation for 4 missing patterns across with 4 data distributions under the above 7 missing imputation methods. Finally, in section 4, we compare the simulation results and try to summary the pattern.

2.2 Introduction

The electronic health record (EHR) system has been widely used in recent years. An EHR is an electronic system used by healthcare systems to collect and store patients' medical data. EHRs are widely used across clinical care and health care to capture a great amount of information about patients over time. EHRs contain a variety of patient level features, such as demographics, diagnoses, problem lists, medications, vital signs, and laboratory data. According to the National Academies of Medicine, an EHR has multiple core functionalities, including the capture of health information, orders and results management, clinical decision support, health information exchange, electronic communication, patient support, administrative processes, and population health reporting.

As the use of EHRs has made it easier to access clinical data, there has been growing interest in analysing data collected during the course of clinical care. EHRs outperforms

many existing registries and data repositories in volume, and the reuse of these data may decrease the costs associated with clinical research. The data from EHRs also provides a window into the medical care, status, and outcomes of a diverse population coming from actual patients. The secondary use of data collected in EHRs is a promising step towards decreasing research costs, and speeding the rate of new medical discoveries.

EHR plays an important role in improving patient-centered analysis. There are plenty of literature proposing suggestions from different aspects to help improve the quality of EHR data, however, completeness is still the fundamental one (37) (38). EHR data contains a lot of missing information that if left unsolved could reduce the validity of inferences concluded from the data. To deal with the missing data appropriately is tough work since sometimes it is hard to differentiate between missing data and a negative value. In other words, EHR data cause a significant challenge for researchers as a result of massive missing data that is not collected due to no intention or missing after the measurement. Not correctly dealing with missing data may lead to significant bias. For example, a patient without a documented value of heart failure may truly not have disease or the clinician may have simply not documented the condition.

Many recently powerful analysis methods such as machine learning and deep learning methods require the complete data set, which would be prohibited by the incompleteness of EHR data. These problems are going to be more significant as EHR systems grow fast worldwide. Though the problem of missing is noted by researchers, it is still substantial (39) (40) and unavoidable (41), and there is much literature working on this issue (42) (43) (44). Approaches for reducing missing data in EHR systems come from multiple angles, including: increasing structured data documentation, reducing data input errors, and utilization of text mining or natural language processing. In this work, we will compare the missing imputation methods for our EHR dataset.

2.3 Missing Data Analysis

Missing data exists almost everywhere when we deal with the real-world data, and it often causes some troubles for further analysis. Though missing data is a nuisance for our study, the expectation of collecting data without missingness is impractical. Even the most careful plan for collecting data cannot guarantee that each subject would provide a complete record, so missing data has been believed as one of the most important issues to scientists. Missing data decreases the ability of scientists to analyse the deep structure and the inner relationship behind the observations, and therefore, it causes a significant negative impact to scientific research. As the example in section 1, sometimes it is hard to differentiate between missing data and a negative value, so just ignoring the missing ones would result in losing the chance to utilize the informative data. (45) Many important scientific and business decisions are based on the serious analysis of the data, so dealing with the missing data in an appropriate way is recognized as a crucial step.

2.3.1 Missing Mechanism

MCAR, MAR, and MNAR

There are a lot of missing imputation methods for dealing with missing data. However, since the appropriate choice for missing imputation depends on finding out the right missing pattern behind the data, we should study the missing patterns before choosing the imputation methods. (46) analyse several missing reasons and summary them into four categories: missing completely at random (MCAR), missing at random (MAR), missing that depends on unobserved predictors, and missing that depends on the missing value itself. However, Missing values that depend on unobserved predictors and missing values that depend on the missing value itself can also be considered missing not at random (MNAR).

Let $Y = \left(Y^{\text{obs}}, Y^{\text{mis}}\right)$ be the complete data matrix consisting of the observed and unob-

served values. Further, let M be a missing data indicator matrix, taking the value 1 if y_{ij} is missing and 0 if y_{ij} is observed.

The data is MCAR if the missingness is independent of any observation, observed or missing, in the data set:

$$P\left(M \mid Y^{\text{obs}}, Y^{\text{mis}}\right) = P(M)$$

MCAR occurs when each value within the same feature has the same probability of being missing. That means, the values in the dataset will be missing totally randomly, and we cannot predict which specific value will have more chance to be missing. When the missing pattern is MCAR, there will be no reason why a specific response is missing. MCAR is convenient for analysis, but is often unrealistic for the real data. (47)

MAR, or missing at random, means the missing probability of a value is the same only within groups defined by the data:

$$P\left(M \mid Y^{obs}, Y^{mis}\right) = P\left(M \mid Y^{obs}\right)$$

Therefore, if the reason for a value being missing is not due to the missing variables themselves, then the reason may be due to other observed variables. (48) The word "random" here might cause some confusion, because the MAR is not actually random and it shows a systematic pattern there the bias of the probability for missing data has a relationship with other observed variables. It should be noted that MAR is a much broader set than MCAR. Mostly, missing data methods should be considered starting from the MAR assumption (47)

The last one, if a missing mechanism is not MCAR nor MAR, then we call it missing not at random, MNAR. The probability of missing is correlated with the missing value itself:

$$P\left(M \mid Y^{\text{obs}}, Y^{\text{mis}}\right) = P\left(M \mid Y^{\text{obs}}, Y^{\text{mis}}\right)$$

Therefore, it depends on the unobserved information which is not available to the analysis. Due to this reason, the missing values cannot be recovered without bias. Therefore, MNAR is considered the most complicated missing pattern.

Rubin's distinction could help us understand why there would be some gaps between the performance of the methods and our expectations. The theory indicates that the existence of missing data could cause loss of statistical power, however, under MCAR assumption, the estimation for parameters does not lead to bias by the missing data (49).

Missing Mechanism for EHR Data

EHR data present significant challenges as a result of not having been collected specifically for research purposes and are subject therefore to considerable missing data. Improper handling of missing data can result in significant bias. These issues will become more relevant as EHRs become more widespread, as the volume of discrete electronic data continues to grow, and as the access to this data becomes increasingly available.

To unbiasedly impute the missing values under MNAR, the algorithm must incorporate the model generating the missing data. (50) In most EHR data, other features may account for part of the missing variation among different patients. Therefore, it is reasonable to impute missing data under MAR assumption. (51) uses simulation by EHR data to show that imputed MNAR missing values without an MNAR model could lead to bias estimation, however, the bias is supposed to be small. To test whether the missing pattern is MAR or MNAR is impossible without contacting patients. However, (52) states that as the number of covariates used to predict the imputation increases, the MAR assumption is able to be more plausible. Followingthis idea, MAR missing imputation methods could be generally applied to real-world dataset to achieve a relatively good accuracy with the help of enough covariates.

2.3.2 Missing Proportions

While the missing data mechanisms are widely studied by scientists, the study of the proportion of missing data does not achieve an agreement in academia. (53) claims that the missing imputation should be adopted when the missing proportion is greater than 10 percent. However, (54)) states that 5 percent or less missing is negligible and when the missing rate is greater than 5 percent, the missing data should be imputed. It seems researchers prefer to impute missing values even when there is a small amount missing.

In the later simulation, we will also consider the effect of missing proportion.

2.3.3 Literature Review: Methods of Handling Missing Data

The core idea of imputation is to carefully replace the missing values with some predicted values, while avoiding importing the bias in the data (55). In recent years, there are plenty of missing imputation methods proposed to deal with the incomplete data. In this study, we will introduce some imputation methods we would like to apply to the EHR data.

Data deletion

One of the standard methods to deal with missing data is to delete the missing values, e.g. list-wise deletion and pair-wise deletion.

List-wise deletion is the easiest way to transform the missing dataset into a complete dataset, and it's the default method to handle incomplete data in many softwares. This most popular method can be useful when we have a large amount of complete lists and the missing pattern is MCAR. However, if the missing pattern is not MCAR, the list-wise deletion would cause bias for estimation. Besides, it will always cause loss of power due to the unused deleted data. (56) There are different opinions for this simple method, such as (57) claims it is difficult to design the best rule for general case, because the effect of

list-wise deletion depends on more than just missing proportion. However, (58) state that dropping a small proportion of the sample under MCAR, there would not be a significant problem.

Pair-wise deletion tries to solve the problem that the sample size might be too small after list-wise deletion. By this way, the calculation of the means and variance of variables are based on all available data, so we incorporate more information than list-wise deletion. However, the disadvantages is that the estimation of the mean and variance could be biased since they are not based on the same sample. Besides, the standard errors cannot be calculated for such estimators. (59) states that the pair-wise deletion should be used only when the following procedure take deletion into account.

Hot Deck Imputation

Hot deck imputation involves replacing missing values of one or more variables for a non-respondent (called the recipient) with observed values from a respondent (the donor) that is similar to the non-respondent with respect to characteristics observed by both cases. In some versions, the donor is selected randomly from a set of potential donors, which we call the donor pool; we call these methods random hot deck methods. In other versions a single donor is identified and values are imputed from that case, usually the "nearest neighbour" based on some metric; we call these methods deterministic hot deck methods, since there is no randomness involved in the selection of the donor. Other methods impute summaries of values for a set of donors, such as the mean, rather than individual values; we do not consider these as hot deck methods, although they share some common features. We note that our use of "deterministic" describes the way in which a donor is selected in the hot deck, and differs from the use of "deterministic" to describe imputation methods that impute the mean or other non-random value.

There are several reasons for the popularity of the hot deck method among survey

practitioners. As with all imputation methods, the result is a rectangular data set that can be used by secondary data analysts employing simple complete-data methods. It avoids the issue of cross-user inconsistency that can occur when analysts use their own missing-data adjustments. The hot deck method does not rely on model fitting for the variable to be imputed, and thus is potentially less sensitive to model misspecification than an imputation method based on a parametric model, such as regression imputation. Having said this, it is important to keep in mind that the hot deck makes implicit assumptions through the choice of metric to match donors to recipients, and the variables included in this metric, so it is far from assumption free. Another attractive feature of the hot deck is that only plausible values can be imputed, since values come from observed responses in the donor pool. There may be a gain in efficiency relative to complete-case analysis, since information in the incomplete cases is being retained. There is also a reduction in non-response bias, to the extent that there is an association between the variables defining imputation classes and both the propensity to respond and the variable to be imputed.

Mean/Median/Mode Imputation

Another simple method for dealing with missing imputation is Mean imputation, which is also called "unconditional mean" imputation (60) by replacing the missing value with the mean of all the observed values within the same variable. But the Mean imputation methods could be used only on numerical data and does not work for categorical data. So the Mode imputation method is proposed to deal with such cases. However, by Mean imputation, the distribution of the variables could be narrowed since all the imputed values take the mean value, which imports the bias into the dataset and then leads to the biased estimators. (61) (62) (63)

Actually, the disadvantage of any single imputation methods is that the distribution would be narrowed and as a result (as Figure 2.1 shows), the standard errors would be

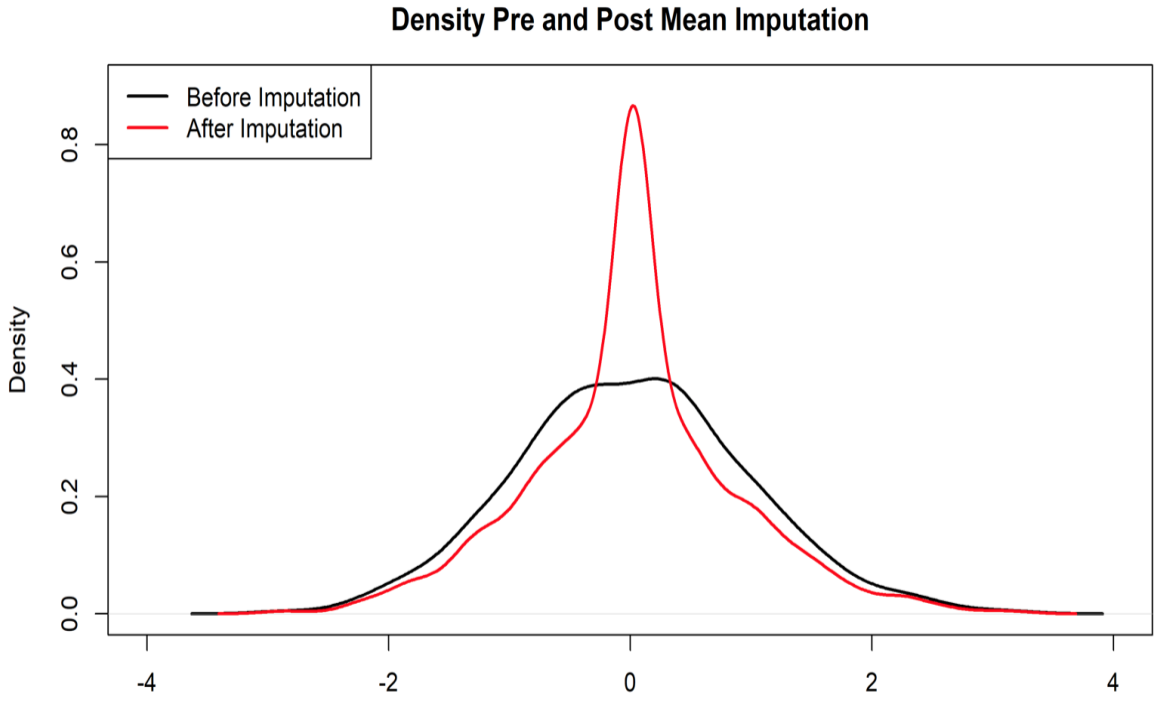


Figure 2.1: Density of Variable Pre and Post Mean Imputation

underestimated, confidence intervals would be narrowed, and p-values would be also underestimated (64).

The Median imputation works in a similar way as Mean imputation by replacing the missing values with medians instead of means of each variable, and it could be more robust when there are extreme values.

Maximum Likelihood

Now we're ready to consider maximum likelihood (ML), which is a close competitor to multiple imputation. Under identical assumptions, both methods produce estimates that are consistent, asymptotically efficient and asymptotically normal. With or without missing data, the first step in ML estimation is to construct the likelihood function. Suppose that

we have n independent observations ($i = 1, \ldots, n$) on k variables $\left(y_{i1}, y_R, \ldots, y_{ik}\right)$ and no missing data. The likelihood function is

$$L = \prod_{i=1}^{n} f_i\left(y_{i1}, y_{i2}, \ldots, y_{ik}; \theta\right)$$

where $f_{(}.\big)$ is the joint probability (or probability density) function for observation i, and θ is a set of parameters to be estimated. To get the ML estimates, we find the values of θ that make L as large as possible. Many methods can accomplish this, any one of which should produce the right result.

Now suppose that for a particular observation i, the first two variables, y_1 and y_2, have missing data that satisfy the MAR assumption. (More precisely, the missing data mechanism is assumed to be ignorable). The joint probability for that observation is just the probability of observing the remaining variables, y_β through y_{ik}. If y_1 and y_2 are discrete, this is the joint probability above summed over all possible values of the two variables with missing data:

$$f_i^*\left(y_{i3}, \ldots, y_{ik}; \theta\right) = \sum_{y_1} \sum_{y_2} f_i\left(y_{i1}, \ldots, y_{ik}; \theta\right)$$

If the missing variables are continuous, we use integrals in place of summations:

$$f_i^*\left(y_{i3}, \ldots, y_{ik}; \theta\right) = \int_{y_1} \int_{y_2} f_i\left(y_{i1}, y_{i2}, \ldots y_{ik}\right) d\, y_2 d\, y_1$$

Essentially, then, for each observation's contribution to the likelihood function, we sum or integrate over the variables that have missing data, obtaining the marginal probability of observing those variables that have actually been observed. As usual, the overall likelihood is just the product of the likelihoods for all the observations. For example, if there are m observations with complete data and $n - m$ observations with data missing on y_1 and y_2,

the likelihood function for the full data set becomes

$$L = \prod_{i=1}^{m} f_i\left(y_{i1}, y_{i2}, \ldots, y_{ik}; \theta\right) \prod_{m+1}^{n} f_i^*\left(y_{i3}, , \ldots, y_{ik}; \theta\right)$$

where the observations are ordered such that the first m have no missing data and the last $n-m$ have missing data. This likelihood can then be maximized to get ML estimates of θ.

In summary, Maximum Likelihood removes the missing data from the likelihood as if they do not exist. By this way, it incorporates all the information from available data to estimate the parameter that results in the most possible likelihood generating the sample data (47). However, there are also disadvantages. The most important one is that the required distribution assumption may not be available especially in complicated EHR data. Without a distribution family assumption, we cannot even build the model. Besides, the desired property of Maximum Likelihood estimates is based on 'large sample', and the estimation could be worse based on a small sample.

Multiple Imputation

A scientific estimand Q is a quantity of scientific interest that we can calculate if we would observe the entire population. For example, we could be interested in the mean income of the population. In general, Q can be expressed as a known function of the population data. If we are interested in more than one quantity, Q will be a vector. Note that Q is a property of the population, so it does not depend on any design characteristics. Examples of scientific estimands include the population mean, the population (co)variance or correlation, and population factor loadings and regression coefficients, as well as these quantities calculated within known strata of the population. Examples of quantities that are not scientific estimands are sample means, standard errors and test statistics. We can only calculate Q if the population data are fully known, but this is almost never the case.

The goal of multiple imputation is to find an estimate $\hat{Q}$ that is unbiased and confidence valid. We explain these concepts below.

Unbiasedness means that the average $\hat{Q}$ over all possible samples Y from the population is equal to Q. The formula is

$$E(\hat{Q} \mid Y) = Q$$

The explanation of confidence validity requires some additional symbols. Let U be the estimated variance-covariance matrix of $\hat{Q}$. This estimate is confidence valid if the average of U over all possible samples is equal or larger than the variance of $\hat{Q}$. The formula is

$$E(U \mid Y) \geq V(\hat{Q} \mid Y)$$

where the function $V(\hat{Q} \mid Y)$ denotes the variance caused by the sampling process.

In summary, the goal of multiple imputation is to obtain estimates of the scientific estimand in the population. This estimate should on average be equal to the value of the population parameter. Moreover, the associated confidence intervals and hypothesis tests should achieve at least the stated nominal value.

The Maximum Likelihood and Multiple Imputation share a lot of similarities. However, the mechanism of Multiple Imputation is different as it includes two steps (65). During the first step, each missing value is filled with several candidates sampled from an imputation model, which generates the complete dataset. The benefit of using several imputations rather than one is to measure the uncertainty of the imputation, which is also the shortcoming of traditional single imputation methods. During the second step, the desired analysis is directly based on each complete dataset, and then the result is aggregated into a single value. In this way, the standard error of Multiple Imputation is more accurate than traditional methods, which tends to underestimate the standard error or variance. (47) Another advantage of Multiple Imputation is that it can be applied to almost any data, while

the Maximum Likelihood requires the assumption of the joint distribution, which is often complicated or even not practical in real data. (66) proposed a likelihood base imputation method, however, for EHR data with enormous and complicated types of variables, the likelihood based methods is difficult to apply. Multiple imputation by chained equations, also called MICE, is a common way to implement the Multiple Imputation, and variables are modelled individually. However, the disadvantage of Multiple Imputation is that it requires large computing resources and the model selection is limited (67).

Missing Forest

Decision Tree is also a widely used popular method.The core idea of Decision Tree is a classifier that divides the paths in a node point and each path leads to a different outcome. While the node has no child node, it is a leaf node; otherwise it is called an internal node. Each leaf node has a class label and each internal label has a criteria for dividing the paths. Following these criteria, each data will go into a single leaf node and then labeled with that class. There are plenty of criteria used in the Decision Tree to help divide the paths, such as CART, ID3, C4.5, OC1 and J48. Among such criteria, CART is the only one which could be used to deal with categorical data, which constructs the binary trees with up to two child nodes. But CART could result in unstable results. A complex tree might need a lot of computation resources to achieve low bias. (68) find the relationship between time complexity and the total levels of the categorical dataset. Rokach, 2016 claims that the bias will decrease as the complexity increases. Decision Tree imputation use observed data to construct the criteria trees for each variable and then impute the missing values. (69).

Random Forest is a group of the above classification trees, which decides the classification by the most votes among the single trees. (70). The Random Forest is widely used and seen as an ideal candidate because it could deal with high-dimension data, and it could achieve good accuracy without assumptions about the distribution and does not require

much computing time. (71). These properties make Random Forest have the ability to deal with almost any kind of data with fewer assumptions (72). Random Forest for missing imputation, called MissForest, focuses on making accurate predictions for missing data rather than sampling from distribution, so the missing imputation values may cause biased estimations.

We assume $X = \left(X_1, X_2, \ldots, X_p\right)$ to be a $n \times p$ -dimensional data matrix. We propose using an RF to impute the missing values due to its earlier mentioned advantages as a regression method. The RF algorithm has a built-in routine to handle missing values by weighting the frequency of the observed values in a variable with the RF proximities after being trained on the initially mean imputed dataset. However, this approach requires a complete response variable for training the forest.

Instead, we directly predict the missing values using an RF trained on the observed parts of the dataset. For an arbitrary variable X_s including missing values at entries $i^{(s)}$ mis $\subseteq \{1, \ldots, n\}$ we can separate the dataset into four parts: Note that $x^{(s)}$ obs is typically not completely observed since the index $i^{(s)}$ obs corresponds to the observed values of the variable X_s. Likewise, $x^{(s)}$ mis is typically not completely missing.

1. The observed values of variable X_s, denoted by $y^{(s)}$ obs;

2. the missing values of variable X_s, denoted by $y^{(s)}$ mis;

3. the variables other than X_s with observations $i^{(s)}$ obs $= \{1, \ldots, n\} \backslash i^{(s)}$ mis denoted by $x^{(s)}$ obs ;

4. the variables other than $\mathrm{X_s}$ with observations $i^{(s)}$ mis denoted by $x^{(s)}$ mis.

The stopping criterion γ is met as soon as the difference between the newly imputed data matrix and the previous one increases for the first time with respect to both variable

types, if present. Here, the difference for the set of continuous variables N is defined as

$$\Delta_N = \frac{\sum_{j \in N} \left(X_{\text{new}}^{\text{imp}} - X_{\text{old}}^{\text{imp}} \right)^2}{\sum_{j \in N} \left(X_{\text{new}}^{\text{imp}} \right)^2}$$

and for the set of categorical variables F as

$$\Delta_F = \frac{\sum_{j \in F} \sum_{i=1}^{n} I_{X_{\text{new}}^{\text{imp}} \neq X_{\text{old}}^{\text{imp}}}}{\#\,\text{NA}}$$

where NA is the number of missing values in the categorical variables.

However, (73) conducted a comparison study and found missForest consistently achieved the lowest error compared with other imputation methods, including k-nearest neighbors (k-NN) imputation and MICE, when data were missing completely at random (MCAR). (74) also recommended missForest when variables have high inter-correlations. (75) reported that missForest produced substantially biased estimates for variables missing at random (MAR) and poor coverage of confidence intervals compared with CALIBERrf impute. (76) demonstrated that the relative performance of missForest varied with the MCAR data patterns and did not show a clear advantage. Overall, the imputation accuracy and applicability of missForest is still unclear.

Generative Adversarial Imputation Nets (GAIN)

We consider a deep learning model, which we call Generative Adversarial Imputation Nets (GAIN) (17), which generalizes the well-known GAN (77) and has the ability to operate successfully even when there is no complete data. For traditional neural network, the known complete dataset is required for training the network. But for GAIN, there is no request for the complete training data, which is quite often lacked in real data.

The adversarial modeling framework (GAN) is most straightforward to apply when the

models are both multilayer perceptrons. To learn the generator's distribution p_g over data x, we define a prior on input noise variables $p_z(z)$, then represent a mapping to data space as $G\left(z;\theta_g\right)$, where G is a differentiable function represented by a multilayer perceptron with parameters θ_g. We also define a second multilayer perceptron $D\left(x;\theta_d\right)$ that outputs a single scalar. $D(x)$ represents the probability that x came from the data rather than p_g. We train D to maximize the probability of assigning the correct label to both training examples and samples from G. We simultaneously train G to minimize $\log(1-D(G(z)))$

In other words, D and G play the following two-player minimax game with value function $V(G,D)$:

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1-D(G(\boldsymbol{z})))]$$

Next we introduce the GAIN, which is built based on GAN. The generator in GAIN aims to impute missing data, while the discriminator aims to distinguish the "real data" (observed data) and the "fake data" (imputed data). The discriminator is designed to minimize the loss function, while the generator's goal is to maximize the chance to "cheat" the discriminator. Thus, the two components of the neural network are against each other. GAIN also follows the same architecture of GAN, while adding "hint" information for discriminator to make sure the adversarial process is as our aim. The hint matrix helps the generator following the true underlying data distribution.

Denote $X = (X_1, ..., X_d)$ is the random variable (either continuous for laboratory tests or binary for ICD-9 codes), whose distribution we will denote as $P(X)$. There is a Mask matrix which tells the Generator which values are missing or are present: $M = (M_1, ..., M_d)$ is a random variable taking values in $\{0,1\}^d$, i.e. M is the missing indicator matrix for X

For each $i \in \{1, ..., d\}$, we define a new random variable $\tilde{X} = (\tilde{X}_1, ..., \tilde{X}_d) \in \tilde{X}$ in the

following way:

$$\tilde{X}_i = \begin{cases} X_i, & \text{if } M_i = 1 \\ *, & \text{otherwise} \end{cases}$$

so that M indicates which components of X are observed. Note that we can recover M from $\tilde{X}$.

The generator, G, takes (realizations of) $\tilde{X}$, M and a noise variable, Z, as input and outputs $\bar{X}$, a vector of imputations. The Random matrix adds randomness to imputed values (so they're not the same every time). There is also Hint vector which is supplied to the Discriminator to ensure that the Discriminator forces the Generator to learn.

Then we define the random variables $\bar{X}, \hat{X} \in X$ by

$$\bar{X} = G(\tilde{X}, M, (1-M) \odot Z)$$
$$\hat{X} = M \odot \tilde{X} + (1-M) \odot \bar{X}$$

There is also Hint vector H which is supplied to the Discriminator to ensure that the Discriminator forces the Generator to learn. Then we define the quantity $V(D,G)$ to be

$$V(D,G) = \mathbb{E}_{\hat{X},M,H}\Big[M^T \log D(\hat{X},H) + (1-M)^T \log\big(1-D(\hat{X},H)\big)\Big],$$

where log is element-wise logarithm and dependence on G is through $\hat{X}$.

Then, as the standard GAN, we define the objective of GAIN to be the minimax problem given by $\min_G \max_D V(D,G)$, and the corresponding $\hat{X}$ is our imputed clinical features matrix.

Nuclear-norm Minimization ("SoftImpute")

Matrix completion is a method to recover a low rank matrix, which is a hot topic in recent literature. It has been applied to many scenarios, such as collaborative filtering (78), multi-task learning (79), system identification (80), and sensor localiza- tion (81)).

The direct search for the lowest rank solutions of equality constraints is NP-hard problem (82), so there is plenty of literature using trace-norm as the key to implement the matrix completion. Trace-norm is the sum of the singular values of a matrix, which is viewed as a convex relaxation for the rank of the matrix. It also could be seen as an analogue to relaxing the sparsity of a vector to its l1-norm, which performs well enough for both empirically and theoretically in compressed sensing.

In several problems of contemporary interest, arising, for instance, in recommender system applications, for example, the Netflix Prize competition (SIGKDD and Netflix), the observed data are in the form of a large sparse matrix, $Y_{ij}, (i,j) \in \Omega$, where $\Omega \subset \{1,\ldots,m\} \times \{1,\ldots,n\}$, with $|\Omega| \ll mn$. Popularly dubbed as the matrix completion problem, the task is to predict the unobserved entries, under the assumption that the underlying matrix is of low rank. This leads to the natural rank regularized optimization problem:

$$\min_X \frac{1}{2} \|P_\Omega(X-Y)\|_F^2 + \lambda \,\mathrm{rank}(X)$$

where $P_\Omega(X)$ denotes the projection of $X_{m\times n}$ onto the observed indices Ω and is zero otherwise; and $\|\cdot\|_F$ denotes the usual Frobenius norm of a matrix. Above problem, however, is computationally difficult due to the presence of the combinatorial rank constraint.

(81) claim that under MCAR or MAR scenarios, Nuclear-norm Minimization is able to achieve accurate matrix completion with high probability. (13) propose a method called SoftImpute to iteratively estimate the matrix by thresholding the singular values, which solves the following minimization problem:

$$\min_M \frac{1}{2}\left\|P_\Omega(X-\hat{X})\right\|_F^2 + \lambda\|\hat{X}\|_*$$

where Y is the original missing matrix, $P_\Omega(X)$ is the projection of X with holding the observed data, and substitute the missing values with 0. The algorithm follows two stages iterated until convergence:

1) Replace the missing values in X with the corresponding entries in current estimated $\hat{X}$

$$\hat{Y} \leftarrow P_\Omega(X) + P_\Omega^\perp(\hat{X})$$

2) Update $\hat{X}$ by computing the soft-threshold SVD of $\hat{Y}$

$$\hat{Y} = USV^T$$

$$\hat{X} \leftarrow US_\lambda V^T$$

where the S_λ sets all singular values less than λ to 0 and replace all other singular values with $(D_i - \lambda)_+$

(81) shows that under some assumptions on the underlying "population" matrix, a solution to second optimization approximates a solution to first optimization reasonably well. The estimator obtained from second optimization works quite well: The nuclear norm shrinks the singular values and simultaneously sets many of the singular values to zero, thereby encouraging low-rank solutions. It is thus not surprising that second optimization has enjoyed a significant amount of attention in the wider statistical community over the last decade.

Typical assumptions under which the nuclear norm works as a good proxy for the low-rank problem require the entries of the singular vectors of the "true" low-rank matrix to be sufficiently spread and the missing pattern to be roughly uniform. The proportion

of observed entries needs to be sufficiently larger than the number of parameters of the matrix $O((m+n)r)$, where r denotes the rank of the true underlying matrix. Since the key assumption for applying SoftImpute is the low rank property, SoftImpute requires much less computation resources and time than Multiple Imputation, so it is ideal for a high-dimension sparse matrix (83).

Implicit Joint Modeling of Observed and missing data ("SoftImpute-concat")

Inspired by the above SoftImpute algorithm, (84) proposed a new algorithm to incorporate the joint information between observed data and the missing pattern, which can recover matrix under MNAR scenario. The proposed to concatenate the original observed matrix with the mask matrix – which is the binary matrix with 1 indicating observed and 0 indicating missing, and then apply SoftImpute or other algorithm to recover the original matrix. As above, the most important assumption is low rank. They applied SoftImpute-concat in synthetic data and showed it outperformed the SoftImpute method. However, there is no theoretical analysis for this improvement and no real dataset used for testing the difference.

2.4 Experiment Setup

2.4.1 Real-world Datasets

We used the Medical Information Mart for Intensive Care III (MIMIC-III) (18) database, which includes 46520 different patients (adults and neonates) admitted to the Beth Israel Deaconess Medical Center in Boston, Massachusetts, between 2001 and 2012. The extracted adult (aged 15 years or older) and neonate patients have admission to 1 of the 6 ICUs – medical ICU, surgical ICU, cardiac care unit, cardiac surgery recovery unit, neonatal ICU, and trauma surgical ICU.

Data Collection

There are plenty of features in MIMIC database, including categorical, continuous, and text data. The ICD-9 code is the most common categorical data representing patients' diseases. The International Classification of Diseases Clinical Modification, 9th Revision (ICD-9 CM) is a list of codes intended for the classification of diseases and a wide variety of signs, symptoms, abnormal findings, complaints, social circumstances, and external causes of injury or disease. The numerical format of the diagnosis codes usually ranges from three to five digits that are assigned to a unique category. However, in the MIMIC database, we have no information about whether it is missing or it is just negative when a patient does not have a specific ICD9-code. It might be another interesting topic for missing imputation, but in this paper, it is not an ideal data for testing different imputation methods since we cannot even acquire a complete dataset for evaluation.

For continuous data, we choose the laboratory test data, which includes 726 unique items among our patient cohort, and we only select the tests which are documented after the patient's first admission. The missing proportions of laboratory tests are up to 90%, which cause a significant impact for further analysis based on such data. Therefore, to study the appropriate missing imputation methods for laboratory tests is an important work.

Data Processing

In this work, we calculate the mean of 726 laboratory tests within first 24 hours after each patient's first admission to ICU. After dropping out the laboratory tests which have missing proportions greater than 70%, we have 64 items left. Then we select all the patients with the complete records for the 64 laboratory tests, which results in 714 patients.

Therefore, our complete dataset for following missing imputation simulation is a matrix with dimension 714 patients * 64 laboratory tests.

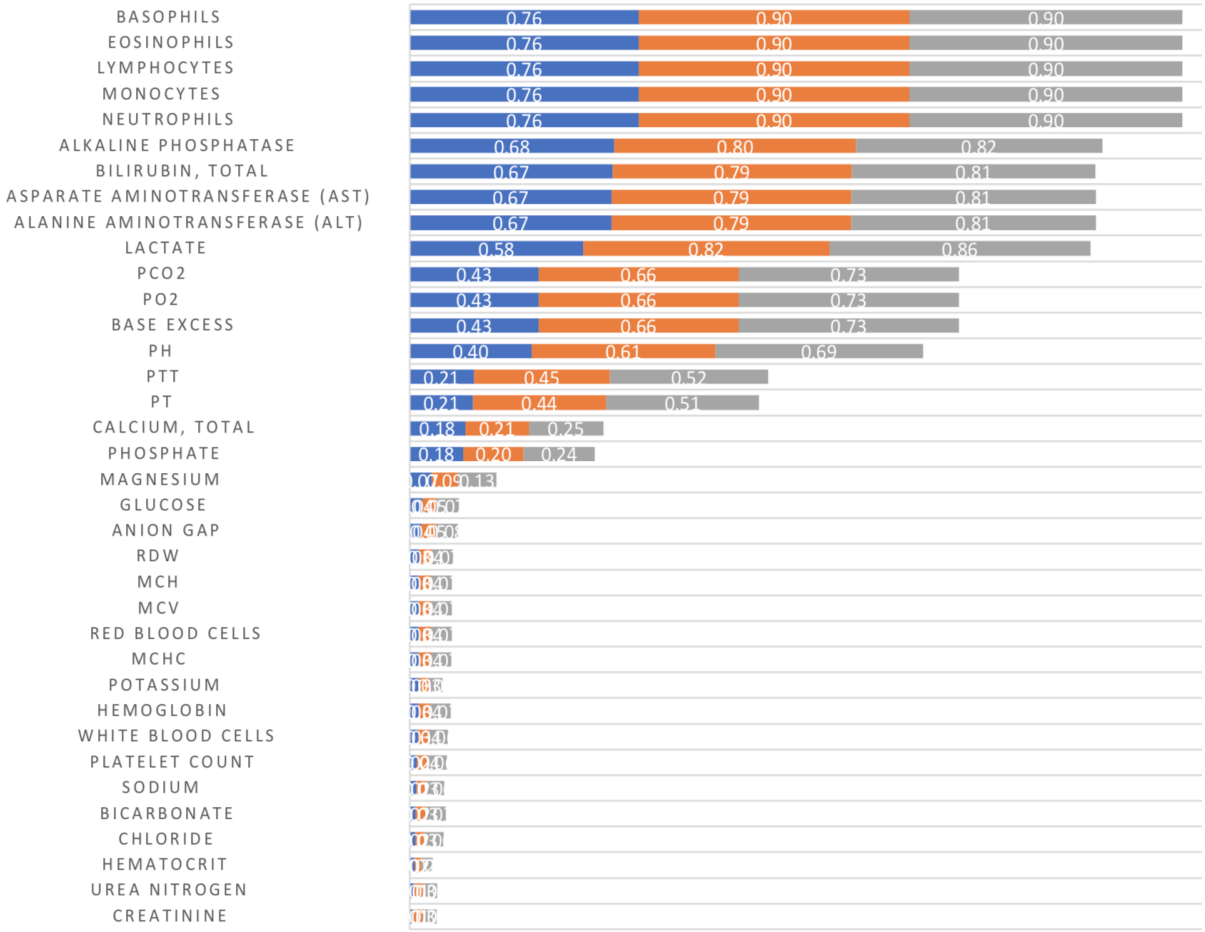


Figure 2.2: Missing Proportions of 36 Major Laboratory Tests

2.4.2 Synthetic Datasets

To match the above real data, our synthetic matrix is designed to also have 714 * 64 dimensions. We consider 3 scenarios: independent case, group correlated case, and low rank case.

For the independent scenario, we sample 714 rows from a 64-dimension multivariate normal distribution with identity covariance matrix, which means all 64 variables are independent.

For the grouped case, we sample 714 rows from a 64-dimension multivariate normal distribution with 8-blocked covariance matrix. This setting means the 64 variables can be divided into 8 groups, and each group has 8 variables. Within the same group, the variable has strong mutual correlation (we set it as 0.7).

For the last one, low rank case, we generate a 714 * 64 matrix with rank 10, which also means there are significant linear correlations within the 64 variables.

Besides, the means of each entry are also tuned to 0.5.

Till now, we have four sets of data matrix (all with complete data and dimension 714 * 64), one is from EHR real data, and the other three are synthetic dataset. Next, we will consider four kinds missing mechanisms for each of four sets of data matrix.

2.4.3 Setting Missing Mechanism

Following the above literature review, here we adopt 4 missing mechanisms, 2 MCAR and 2 MAR.

1 MCAR:

We set a missing proportion p, and for each element in the matrix, it is set to missing with probability p.

2 MCAR:

We set the parameters α and β to determine a Beta distribution. For each feature i, draw a $p_i \sim Beta(\alpha, \beta)$, and set each element within feature i to missing with probability p_i.

3 MAR: We set the parameters β_0 and β_1, and for each element x in the matrix, it is set to missing with probability $p_x = \frac{e^{\beta_0+\beta_1 * x}}{1+e^{\beta_0+\beta_1 * x}}$.

4 MAR: We set the parameters μ_0 and μ_1, for each feature i, we draw $\beta_{0i} \sim N(\mu_0, 1)$ and $\beta_{1i} \sim N(\mu_1, 1)$, and then set each element within feature i to missing with probability $p_x = \frac{e^{\beta_{0i}+\beta_{1i} * x}}{1+e^{\beta_{0i}+\beta_{1i} * x}}$. However, it is reasonable to always control $p_x < 0.9$

2.4.4 Setting Missing Proportions

To evaluate the performance of imputation methods within different scenarios, we should also control the missing proportions. Considering the missing proportion in laboratory tests can be significantly large, it is reasonable to set the missing proportion from 10% to 70% for our simulations.

Each method takes a 714 * 64 matrix as input, and then calculate the accuracy based on the out put. Repeat above data generation and accuracy estimation 1000 times.

Therefore, we tune the parameters in the above settings accordingly:

1 MCAR: $p = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7]$

2 MCAR: $a = 5, b = [45, 20, 12, 7, 5, 3, 2]$

3 MAR: $b_0 = -2, b_1 = [0, 2, 4, 5, 7, 8, 10]$

4 MAR: $\mu_0 = -2, \mu_1 = [0, 2, 4, 5, 7, 8, 10]$

By this setting, the expectation of missing proportions are [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7], which covers the low missing case to the high missing case.

Now we have four sets of dataset with missing values, each sets includes four missing mechanism, and each missing mechanism has missing rate from 0.1 to 0.7.

2.4.5 Evaluation Criteria

We evaluate these methods based on their imputation accuracy on the ground truth entries in synthetic data and MIMIC ICU fully observed subset. The accuracy is measured by normalized test error (13):

$$\frac{\left\|P_{\Omega}^{\perp}(Y-M)\right\|_{F}^{2}}{\left\|P_{\Omega}^{\perp}(Y)\right\|^{2}}$$

(The notation is the same as 2.3.6)

Therefore, for the synthetic rating data and ICU subset, we can explicitly compare the performance of 7 methods under 4 missing patterns for 7 missing proportions.

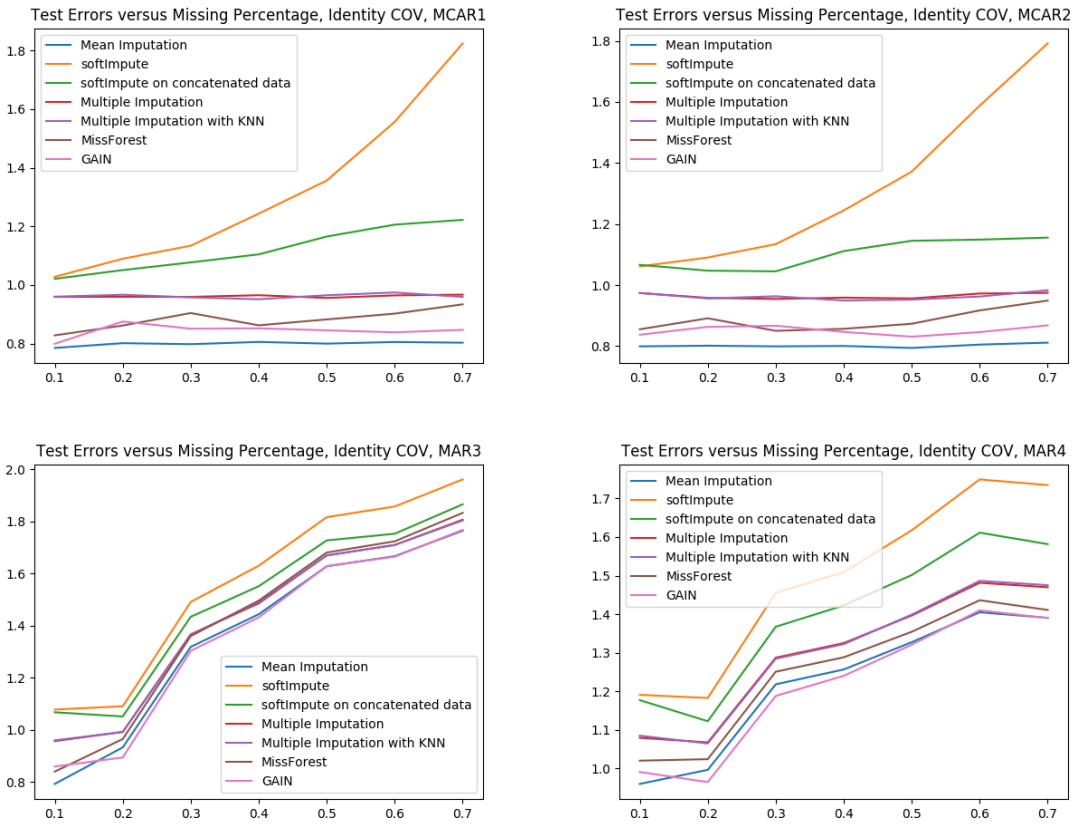


Figure 2.3: Identity covariance

2.5 Conclusion and Discussion

For the Multiple Imputation method, we iterate 5 times for each single imputation, and use the average of 10 imputation multiples as the imputation value. For SoftImpute and SoftImpute-concat, the input matrix is double-normalized, and the maximum rank is set based on the SVD result if the input matrix is completed.

It can be seen that under the identity covariance case, the simple baseline Mean Imputation method performs best, which is not surprising because all other imputation methods assume some inner correlation within the features. However, it is not true at this time since all variables are independent. Therefore, the assumptions for all other methods do not

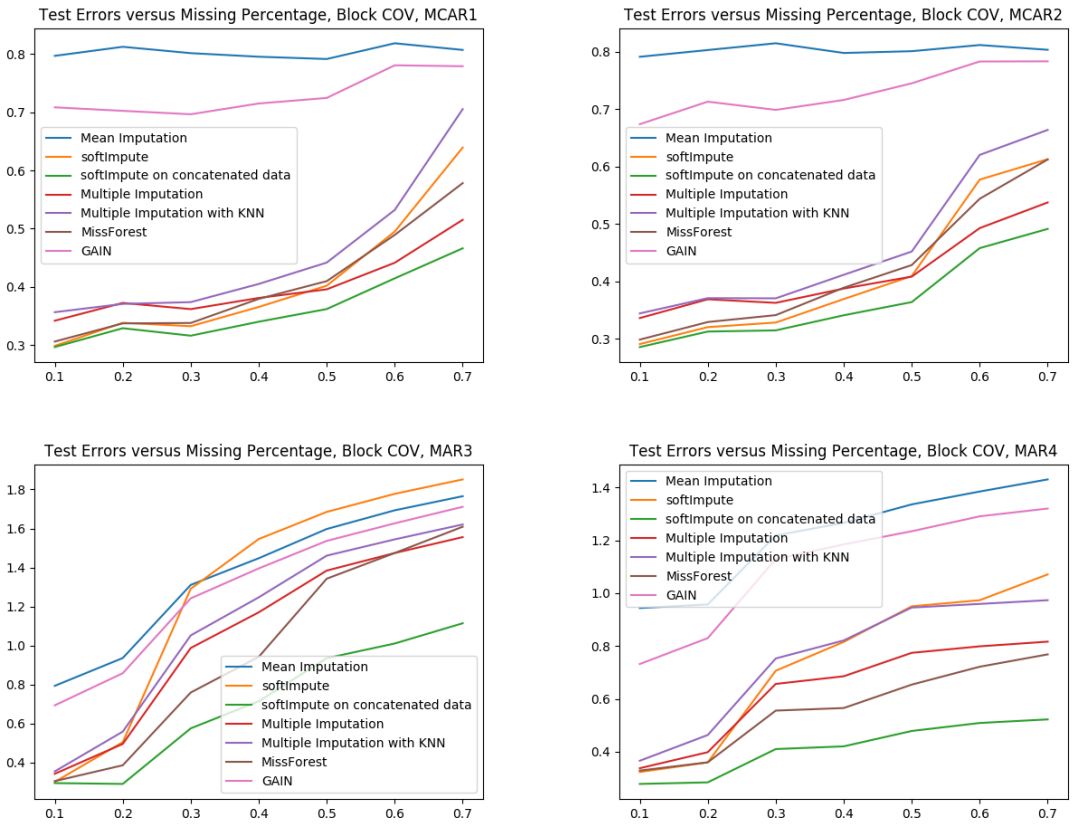


Figure 2.4: Block covariance (block size 8, correlation 0.7)

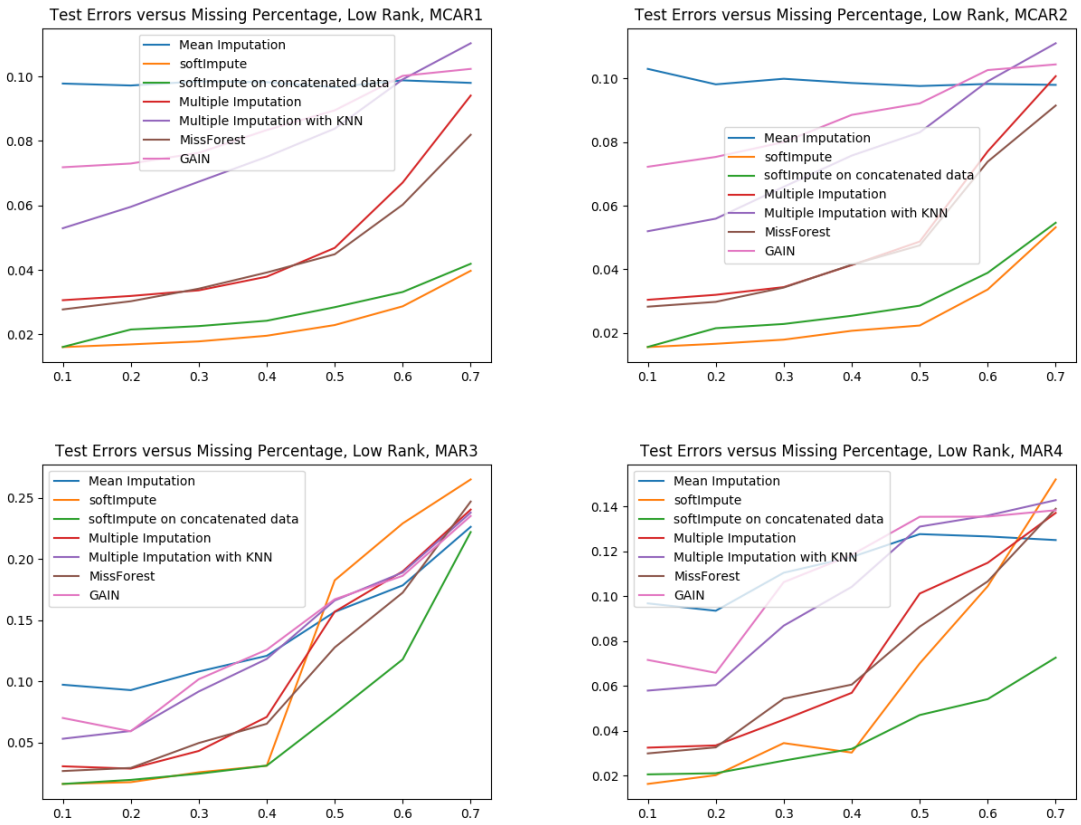


Figure 2.5: Low rank (rank = 10)

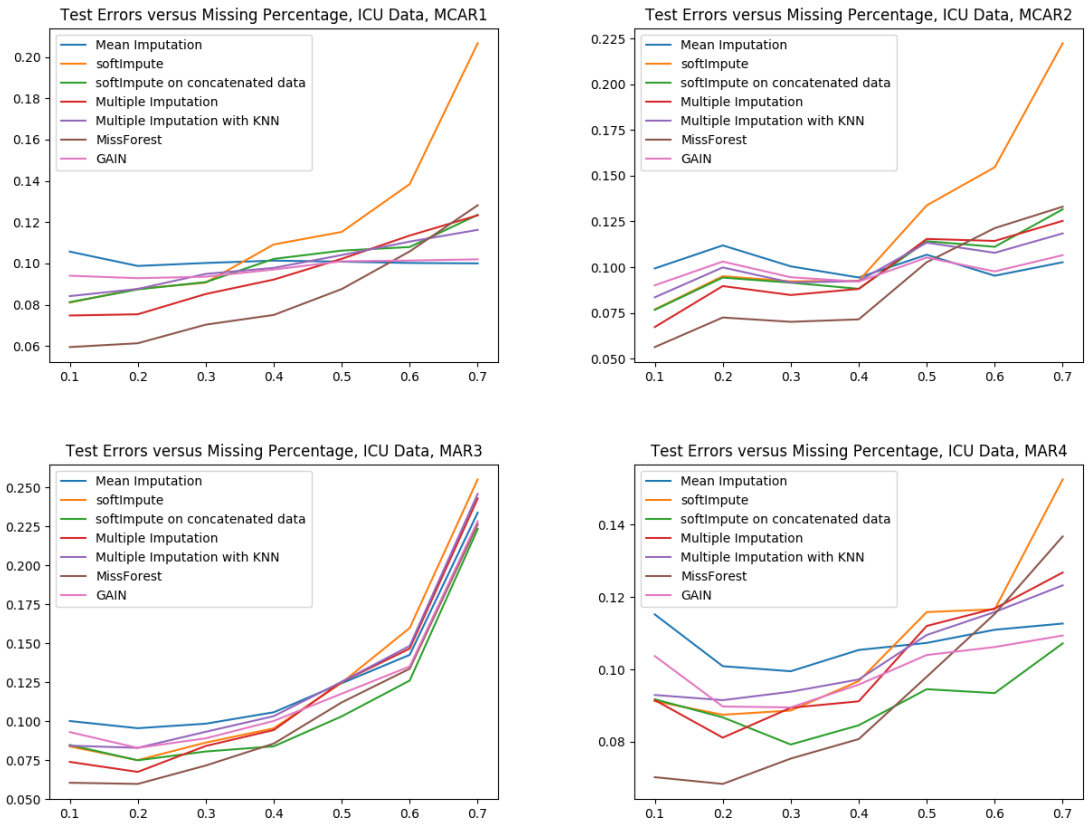


Figure 2.6: Laboratory test data

hold, and they actually incorporate wrong information into the imputation values. Mean Imputation has the simplest assumption which does not consider the interactions with other variables, so it is not mislead. Therefore, Mean Imputation performing better than others under independent case is reasonable. Also, because of the same reason, Mean Imputation is almost the worst imputation method in the other 3 cases.

SoftImpute and SoftImpute-concat are the worst imputation methods under the identity covariance case, it is natural since the matrix is obviously full rank here, which significantly violates their assumptions. When the correlation parameter of the block covariance matrix is large, the block matrix setting encourages the low rank of the input matrix. From the results, we can see that for block covariance and low rank cases, SoftImpute and SoftImpute-concat are indeed the best. For ICU data, they perform also well. By exploratory analysis of the ICU dataset, we find the group pattern of variables exists, but not that much. It helps us understand their performance.

For comparison between the two methods, it is clear that almost under all scenarios, the SoftImpute-concat outperforms the SoftImpute. Especially when the missing proportion is large, the SoftImpute often performs terribly. It suggests that adding the missing mask in SoftImpute-concat is able to increase the imputation accuracy.

Multiple imputation methods perform quite stable in terms of accuracy scores compared with other imputation methods. They do not achieve the best nor the worst in the four scenarios, and show a robust performance as the missing proportion increases. The Multiple imputation with mean and Multiple imputation with KNN can achieve very different accuracy in different scenarios, which suggests the choice of imputation methods in each step within Multiple imputation should be carefully considered according to the dataset.

MissForest always performs pretty well in our 4 scenarios. This tree method does not need much tuning parameters nor requiring assumptions about distributional aspects

of the data, which makes it a very handy and powerful imputation method. In real data analysis, if we do not have much prior information about the dataset, maybe applying MissForest as a baseline is an acceptable choice.

From the simulation results, it can be concluded that the GAIN imputation method performs acceptable in our ICU dataset, but it does worse under all three synthetic scenarios. However, one main reason is that GAIN, as a neural network method, requires finely tuning parameters. We try to tune the parameters suitable for ICU dataset, and use the same parameters for 4 datasets * 4 missing mechanisms * 7 missing proportions = 112 cases. Therefore, its not performing well under 3 synthetic dataset does not mean the method is weak. After finely tuning for each case, the deep neural network has the potential to achieve a better accuracy.

In conclusion, for our MIMIC ICU laboratory test data, if we have confidence about the low rank setting, SoftImpute and SoftImpute-concat could achieve ideal accuracy while do not require much computation resources. (A further idea is that if we have categorical data such as ICD-9 codes but with explicit missing or not index, then due to its sparse and low rank property, SoftImpute and SoftImpute-concat imputation might perform surprisingly well.) Multiple imputation and MissForest are ideal to consider as they do not require tuning parameters finely and show a good performance, especially the MissForest. However, the GAIN, as a neural network method, has the best potencial if we tune the parameters appropriately.

CHAPTER

3

STRUCTURE MAINTAINED REPRESENTATION LEARNING NEURAL NETWORK FOR CAUSAL INFERENCE

3.1 Introduction

Estimating heterogenerous causal effects of a treatment has drawn increasing interests in many fields such as personalized medicine, policy making, and economics. While traditional methods focus on estimating the average causal effect on a target population, this approach is insufficient to draw inferences about differential causal effects due to the differential responses across different characteristics to a treatment. In this study, we

focus on answering the question "which treatment works best for whom" by estimating the conditional average treatment effects (CATE) or individualized treatment effect (ITE) based on observational data.

The fundamental challenge of causal inference is that for each individual, we only observe the outcome corresponding to the assigned treatment group (factual outcome), and the other potential outcome (counterfactual outcome) under the opposite treatment option is missing (85; 86). Therefore, from the prediction perspective, the standard supervised learning approach does not apply because the counterfactual is never observed and the true individual causal effects remain unknown. To infer the missing potential outcomes from observational data, one biggest challenge is that the treatment assignment mechanism is unknown and observational data usually suffers from selection bias, so the covariate distributions across treatment arms may be fundamentally different. For machine learning, this causes *distributional shift* problem when one tries to predict, and for statistical inference, this is known as *confounding*, where the confounders are variables associated with both treatment assignment and outcome, leading to biased estimation of causal effects when not properly accounted for (87). Classical methods address covariate imbalance via propensity score methods such as matching or weighting (88; 89; 87). However, these methods mainly focus on estimating the average causal effect and rely on correct estimation of the propensity scores. Moreover, the popular inverse probability weighting (90) may suffer from large variance when the overlap of covariate distributions is poor.

Recent developments in machine learning solve this problem via *representation learning* through deep neural networks such that the covariate distributions between treatment arms are balanced in the learned high dimensional representation space (91; 92). However, the covariates associated with treatment assignment usually offer valuable information about final estimate of the causal effect (93), and over emphasizing balance may lose such information of outcomes and harm the predictive accuracy (94). Therefore, representation

learning faces the trade-off between achieving good balance and maintaining predictive information. Distinct from the representation learning, another popular machine learning approach directly infers ITE based on the generative adversarial nets (GANs) (77), where the generators and discriminators are trained adversarially to learn the counterfactual outcomes and subsequently ITEs (95). These models also showed promising results to learn complex generative distributions and operate under limited model assumptions.

In this study, we capitalize on the success of representation learning and adversarial networks in the estimation of ITEs. First, we improve the predictive accuracy of representation learning via introducing a structure keeper which maintains the correlation between the baseline covariates and their corresponding representations in the high dimensional space. Second, we train a discriminator at the end of representation layers to trade off representation balance and information loss. We show that the proposed discriminator minimizes an upper bound of the treatment estimation error. Specially, we train the representation layers to fool a discriminator, which attempts to determine whether the given representations are from the treatment or the control arm. By also considering the correlations between the learned representation space and the original covariate feature space, we are able to address the tradeoff between distribution balance and information loss. We conduct experiments with simulated data and real-word observational data. The code of experiments can be found at https://github.com/SMRLNN/SMRLNN. Our proposed Structure Maintained Representation Learning (SMRL) algorithm outperforms state-of-the-art methods.

3.2 Problem setup and notations

Consider a sample of N individuals, where the treatment group ($Z = 1$) has N_1 subjects, and the control group ($Z = 0$) contains N_0 subjects. We operate under the Stable Unit Treatment Value Assumption (SUTVA) (96), each subject i has two potential outcomes

$Y_i(1)$ and $Y_i(0)$ under treatment and control. SUTVA implies the potential outcomes of each subject are not impacted by the treatments received by others, and there is only one version of each treatment. The fundamental challenge of causal inference is that we only observe the outcome corresponding to the assigned treatment group (factual outcome), $Y_i^F = Z_i Y_i(1) + (1 - Z_i) Y_i(0)$, and the other unobserved outcome (counterfactual outcome) Y_i^C is missing. Suppose we also observe a vector of P pre-treatment covariates, $X_i = (X_{i1}, ..., X_{iP})^T$. Denote the probability of receiving treatment giving covariates by $e(X_i) = \Pr(Z_i = 1|X_i)$, i.e. *propensity score*, and the conditional expectation of the potential outcome given the pre-treatment covariates with treatment z by $\mu_z(x) = E\{Y(z)|X = x\}$ for $z = 0, 1$. We are interested in estimating the conditional average treatment effect (CATE) or the individual treatment effect (ITE), defined as the expected difference of potential outcomes given the pre-treatment covariates $\tau(x) = \mu_1(x) - \mu_0(x) = E(Y(1) - Y(0) \mid X = x)$. In addition, another causal estimand commonly of interest is the average treatment effect (ATE), $\tau_{\text{ATE}} = E_{x \sim p(x)}\{\tau(x)\}$, where the expectation is taken on a pre-specified population of interest with covariate distribution $p(x)$. Estimating the causal estimands necessitates inferring the missing counterfactual outcome of each subject, and the following two standard assumptions are employed for identifiability (88), (1) Strong ignorabililty (or unconfoundness): $Z \perp\!\!\!\perp \{Y(1), Y(0)\}|X$; (2) Positively (or overlap): $0 < e(X) < 1$. The strong ignorabililty assumption states the treatment assignment of subjects is independent of the potential outcomes given covariates, mimicing a conditional randomization. The overlap assumption states each subject has non-zero probability being in the treatment or control group.

3.3 Relevant work

Previous machine learning methods for the estimation of ITE fall into three categories. The first category directly models the outcome response surface. For example, Causal

Forest (CF) (97; 98); Bayesian Additive Regression Trees (BART) (99); GAMLSS (100). The second category separately models the representation and the outcome surface such that the neural networks are encourages to learn balanced representations. For example, Treatment Agnostic Regression Network (TARNET) and Counterfactual Regression Network (CFRNET) (92; 91; 101). These methods proposed two possible statistical distances to measure the distribution discrepancies. Specifically, let p_1, p_2 be two distributions over a probability space S, the Integral Probability Metrics (IPM) is defined as $IPM_{\mathrm{G}}\left(p_1, p_2\right) = \sup_{g\in \mathrm{G}} \left| \int_S g(s)\left(p_1(s) - p_2(s)\right) ds \right|$, where $g : S \to \mathbb{R}$ belongs to a function family G. When G is the family of 1-Lipschitz functions, IPM becomes the Wasserstein distributional distances, and when G is the family of norm-1 reproducing kernel Hilbert space (RKHS) functions, IPM becomes the Maximum Mean Discrepancy (MMD) distances. Penalizing IPM loss forces the treated and control covariate distributions to be similar. The third category such as GANITE (95) extends GAN based method by attempting to learn the counterfactual distributions and the ITE distributions.

Our work is most similar to CFRNET, but as representation learning trades off between reducing bias and maintaining predictive information, Zhang et al. (102) argued that choice of the IPMs may critically impact the model performance and the overlap in representation space may be substantially biased. To tackle these challenges, We propose a structure keeper that emphasizes the correlation between the learned representations and the original covariates. In addition, instead of choosing an arbitrary IPM such as Wasserstein distances or MMD, we alternately optimize a discriminator to distinguish whether the representations are transformed from the treatment or control group.

3.4 Structure Maintained Representation Learning Neural Network for Causal Inference (SMRLNN)

Adversarial representation learning capitalizes on the recent development in utilizing representation learning to achieve covariate balance in the high-dimensional space. Instead of defining specific metrics, such as Wasserstein distance or MMD distance, to measure distances between two distributions, we propose to introduce an adversarial approach where we train a discriminator to differentiate whether the learned representations $\Phi(x)$ are from the treated or control arm. Hence, the discriminator forces the representation layers to map the covariate probability space to an overlapped probability space.

3.4.1 Representation Balancing

Traditional balancing methods such as propensity score weighting focus on balancing the first moment condition, i.e. absolute mean difference, between two treatment groups. However, when the treatment effect is heterogeneous across patients' baseline covariates, higher moment balance is required to achieve unbiasedness. Therefore, propensity score methods suffer from bias even when the true propensity scores are provided. In addition, in practice, the propensity scores have to be estimated from real data, and mis-specification of the propensity score could result in high bias and low precision.

In contrast, the balancing property of representation learning is guaranteed by the discriminator, which forces the distribution similarity between two treatment groups. Therefore, representation learning usually performs better under complex propensity score models and to estimate heterogeneous treatment effects.

Let $\Phi : X \rightarrow R^d$ be a representation function that maps from the covariate probability space X to a representation space R^d, such that the covariate distributions of different treatment arms are balanced in R^d. The representation functions are constructed by a deep

neural network, and we accomplish the goal of achieving covariate balance by adding a discriminator after the representation layers. The representation balancing discriminator $D : R^d \rightarrow R$ belongs to a class of classifiers which differentiates whether the learned representations $\Phi(x)$ are from the treated or control arm. The representation layers and discriminator are trained iteratively such that the learned representations are balanced between treatment arms to be able to fool the discriminator. When we update the representation layers, the parameters of the discriminator are fixed. As a result, the penalization will make the representation layers to map samples toward the decision boundary. Therefore, traditional GANs may suffer from no loss when samples lie in a long way on the correct side of the decision boundary. To stabilize the training, we follow the the work of LSGANs (103) by defining the objective of the representation balancing component as minimizing the following losses from a two-player game

$$L_D(\Phi(x|z=0),\Phi(x|z=1)) = \frac{1}{2}\mathbb{E}_{(x|z=0)}(D(\Phi(x))-1)^2 + \frac{1}{2}\mathbb{E}_{(x|z=1)}(D(\Phi(x))+1)^2 \tag{3.1}$$

$$L_\Phi(\Phi(x|z=0),\Phi(x|z=1)) = \frac{1}{2}\mathbb{E}_{(x|z=0)}(D(\Phi(x)))^2, \tag{3.2}$$

where Equation (3.1) is minimized with respect to the discriminator D, and Equation (3.2) is minimized with respect to the representation layers. These modified losses generate more gradients by penalizing the samples lying close to the decision boundary, thus resulting in more stabilized training performance. In addition, Mao et al. (103) proves that minimizing Equation (3.1) and (3.2) yields minimizing the Pearson χ^2 divergence between $p(x|z=0)+p(x|z=1)$ and $2p(x|z=1)$.

3.4.2 Representation Structure Keeper

The aim of representation layers are to balance the covariate distributions in the learned represented space, but to keep the prognostic information contained by covariates. In this section, we introduce a structure keeper on top of the representation layers based on the Representation Structure Keeper (RSK). The RSK allows for calculating correlation between two sets of variables in high dimensional space. For the given pairs of sample of covariates and their representations $((X_1, \Phi(X)_1), \ldots, (X_n, \Phi(X)_n))$, denote the projection of X and $\Phi(X)$ in a chosen direction by

$$P_X = W_X X, \quad P_{\Phi(X)} = W_{\Phi(X)} \Phi(X),$$

where W_X and $W_{\Phi(X)}$ are the $K \times P$ projection matrices of X and $\Phi(X)$. The RSK solves the projection matrices such that the correlation defined by the top K projection directions between the covariates and the representations are maximized. For example, denote the correlation matrix of P_X and $P_{\Phi(X)}$ by

$$corr(P_X, P_{\Phi(X)}) = \text{corr}\left(W_X X,\ W_{\Phi(X)} \Phi(X)\right) \tag{3.3}$$

$$= \frac{W_X' \hat{\mathbb{E}}\left[X \Phi(X)'\right] W_{\Phi(X)}}{\sqrt{W_X' (\hat{\mathbb{E}}[XX'] + \lambda_1 I) W_X W_{\Phi(X)}' (\hat{\mathbb{E}}[\Phi(X)\Phi(X)'] + \lambda_2 I) W_{\Phi(X)}}}, \tag{3.4}$$

then W_X and $W_{\Phi(X)}$ are optimized such that

$$L_{RSK}(x, \Phi(X)) = \max_{W_X, W_{\Phi(X)}} \sum_K \text{diag}(W_X' C(X, \Phi(X)) W_{\phi(X)}'), \tag{3.5}$$

with

$$W_X' (C_{XX} + \lambda_1 I) W_X = 1$$

$$W_{\Phi(X)}' (C_{\Phi(X)} + \lambda_2 I) W_{\Phi(X)} = 1.$$

where $\text{diag}(C(X, \Phi(X)))$ represents the diagonal elements of the correlation matrix. More-

over, the correlation matrix $C(X,\Phi(X))$ in (3.5) can be decomposed to

$$C(X,\Phi(X)) = \hat{\mathbb{E}}\left[\begin{pmatrix} X \\ \Phi(X) \end{pmatrix}\begin{pmatrix} X \\ \Phi(X) \end{pmatrix}'\right] = \begin{bmatrix} C_{XX} & C_{X\Phi(X)} \\ C_{\Phi(X)X} & C_{\Phi(X)\Phi(X)} \end{bmatrix},$$

and the corresponding Lagrangian of RSK optimization problem is

$$\begin{aligned} L\left(W_X, W_{\Phi(X)}\right) = \min_{\lambda_X,\lambda_{\Phi(X)}} \Big[& -W_X' C_{X\Phi(X)} W_{\Phi(X)} + \frac{\lambda_X}{2}\left(W_X'(C_{XX}+\lambda_1 I)W_X - 1\right) \\ & + \frac{\lambda_{\Phi(X)}}{2}\left(W_{\Phi(X)}'(C_{\Phi(X)\Phi(X)}+\lambda_2 I)W_{\Phi(X)} - 1\right)\Big] \end{aligned}$$

Therefore, our representation structure keeper is designed to optimize the objective function $L\left(\lambda, W_X, W_{\Phi(X)}\right)$, and to achieve this, one can take derivatives with respect to x and $\Phi(X)$, giving

$$(C_{XX}+\lambda_1 I)^{-1} C_{X\Phi(X)}(C_{\Phi(X)\Phi(X)}+\lambda_2 I)^{-1} C_{\Phi(X)X}\hat{W}_X = \rho^2 \hat{W}_X \tag{3.6}$$

$$(C_{\Phi(X)\Phi(X)}+\lambda_2 I)^{-1} C_{\Phi(X)X}(C_{XX}+\lambda_1 I)^{-1} C_{X\Phi(X)}\hat{W}_{\Phi(X)} = \rho^2 \hat{W}_{\Phi(X)}, \tag{3.7}$$

where the eigenvalues ρ^2 are the squared canonical correlations and the eigenvectors $\hat{W}_X$ and $\hat{W}_{\Phi(X)}$ are the normalized canonical correlation basis vectors. Therefore $\hat{W}_X$ and $\hat{W}_{\Phi(X)}$ are the solutions of a symmetric eigenvalue problem of the form $Ax = \lambda x$.

Then our loss of representation structure keeper is:

$$L_{RSK}(X,\Phi(X)) = \sum_K \operatorname{diag}\left(\hat{W}_X'\hat{\mathbb{E}}\left[X\Phi(X)'\right]\hat{W}_{\Phi(X)}\right) \tag{3.8}$$

3.4.3 Outcome Prediction Network

Let $H : R^d \times \{0,1\} \rightarrow Y$ be the class of outcome prediction functions defined over the representation space R^d. We implement the standard feed-forward deep neural networks that takes the last layer of representation component and the observed treatment assignment as inputs and output an outcome prediction, $\hat{y}_i = H(\Phi(x_i), z_i)$. Then, the empirical mean squared error (MSE) loss function for outcome prediction is $L_{out}(H,\Phi) = \frac{1}{N}\sum_{i=1}^{N}\left(\hat{y}_i - y_i^F\right)^2$, and the total training loss function can be expressed as

$$L_{FL} = L_{out}(H,\Phi) + \lambda R(\Phi),$$

where $R : R^d \rightarrow R$ is a regularization function and λ is a regularization coefficient that penalizes the complex of the representation architecture.

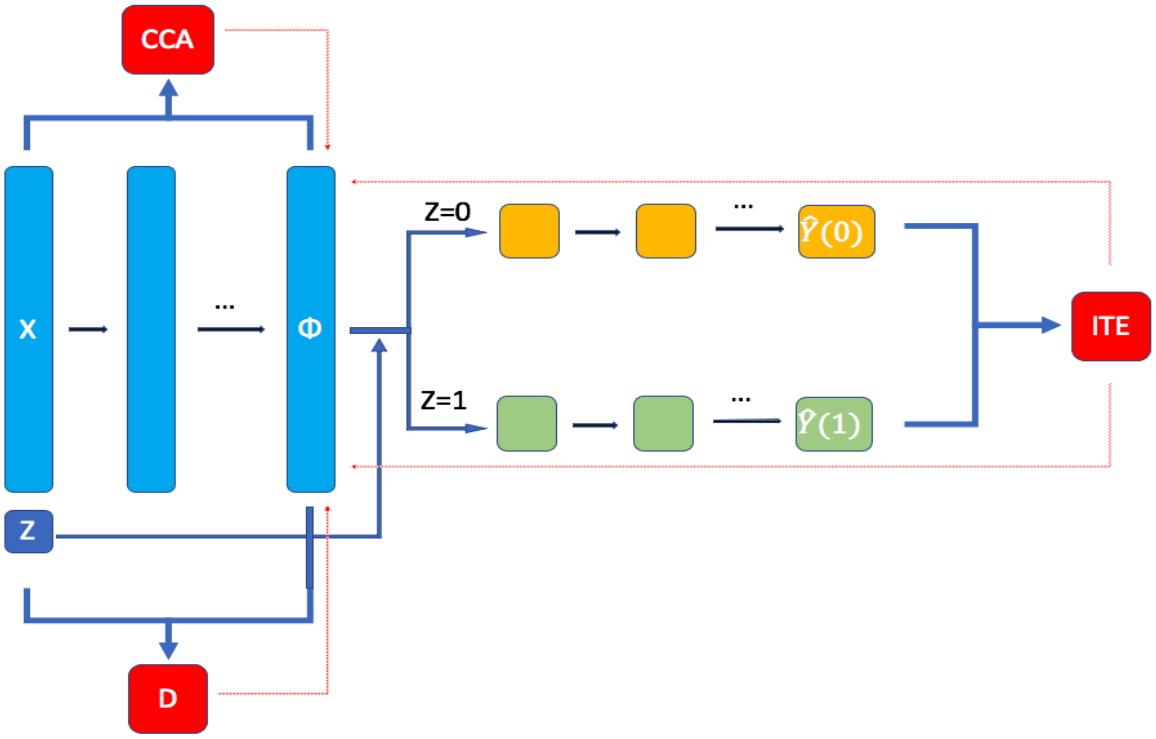
3.4.4 Algorithm

The architecture of our proposed neural networks is summarized in Figure 1, and the optimization steps are summarized in Algorithm 1.

Data: Sample pairs $\left(x_1, z_1, y_1\right), \ldots, \left(x_n, z_n, y_n\right)$, representation structure network Φ_W with standard normal initial weights W, representation balancing network D_U with standard normal initial weights U, outcome network H_V with standard normal initial weights V

Result: $\hat{\tau}_i = \hat{Y}_i(1) - \hat{Y}_i(0)$

Figure 3.1: SMRLNN Structure: X represents the covariates; z represents the treatment assignment; $\Phi : X \rightarrow R^d$ is a representation function; $\hat{Y(0)}$ and $\hat{Y(1)}$ are the predicted potential outcomes; D is the Representation Balancing; CCA is the Representation Structure Keeper



while *not converged* **do**

- Sample mini-batch $\{i_1, i_2, \ldots, i_m\} \subset \{1, 2, \ldots, n\}$
- Calculate the gradients of the representation structure Keeper: $g_1 = \nabla_W L_{RSK}(\Phi)$
- Calculate the gradients of the representation balancing parts: $g_2 = \nabla_U L(D)$, $g_3 = \nabla_W L(\Phi)$
- Calculate the gradients of the outcome model: $g_4 = \nabla_V L_{FL}(H, \Phi)$, $g_5 = \nabla_W L_{FL}(H, \Phi)$
- Update weights parameters $[W, U, V] \leftarrow \left[W - \eta\left(\alpha g_1 + \beta g_3 + g_5\right), U - \eta\left(g_2\right), V - \eta\left(g_4\right)\right]$
- Check convergence criterion

end

$$\hat{\tau}_{H,\Phi}(x) = H(\Phi(x),1) - H(\Phi(x),0)$$

$$L_{PEHE}(H,\Phi) = \sum_{x \in X} \left(\hat{\tau}_{H,\Phi}(x) - \tau(x)\right)^2 p(x)$$

3.5 Theorem

When focusing on the Integral Probability Metric (IPM) and Precision in Estimation of Heterogeneous Effect (PEHE), defined as $L_{PEHE}(H,\Phi) = \sum_{x \in X} \left(\hat{\tau}_{H,\Phi}(x) - \tau(x)\right)^2 p(x)$, where $\hat{\tau}_{H,\Phi}(x) = H(\Phi(x),1) - H(\Phi(x),0)$ is the treatment effect estimate for unit x, Shalit et al.(91) have shown that the error of PEHE is upper bounded by the sum of the expected factual loss and the IPM. We introduce H divergence to quantify the discriminator assessed balance condition, and show that the prediction error can be upper bounded by the sum of the expected factual loss and the H divergence criteria.

Let D denote the family of binary discriminators $D : \Phi(X) \rightarrow [0,1]$, then we define the H divergence (104) between two probability density distributions as:

$$d_D(\Phi) = \max_{D \in D} \left| \frac{1}{N_0} \sum_{x_i \in X_0} D\left(\Phi(x_i)\right) - \frac{1}{N_1} \sum_{x_j \in X_1} D\left(\Phi(x_j)\right) \right|$$

where X_1 and X_0 are covariate distributions over treatment and control groups.

To facilitate the mathematical derivations, we first introduce the following definitions. Define the expected loss for the unit and treatment pair (x, t) as:

$$\ell_{H,\Phi|z}(x) = \int_{Y_z} L_Y\left(Y(z), H(\Phi(x), z)\right) p\left(Y(z) \mid x\right) d\, Y(z),$$

and the maximum loss among the two treatment groups is $\ell_{H,\Phi}^{max}(x) = max\left(\ell_{H,\Phi|z=0}(x),\ \ell_{H,\Phi|z=1}(x)\right)$.

The expected factual loss and counterfactual losses of H and Φ are, respectively:

$$L_F(H,\Phi) = \frac{1}{N}\sum_{i=1}^{N} \ell_{H,\Phi|z=z_i}(x_i)\, p\,(x_i, z = z_i)$$
$$L_C(H,\Phi) = \frac{1}{N}\sum_{i=1}^{N} \ell_{H,\Phi|z=z_i}(x_i)\, p\,(x_i, z = 1-z_i),$$

and by the law of iterated expectations,

$$L_F(H,\Phi) = p_0 \cdot L_{F|z=0}(H,\Phi) + p_1 \cdot L_{F|z=1}(H,\Phi)$$
$$L_C(H,\Phi) = p_0 \cdot L_{CF|z=1}(H,\Phi) + p_1 \cdot L_{CF|z=0}(H,\Phi),$$

where $p_0 = p(z=0)$ and $p_1 = p(z=1)$, and $p\,(x,z) = p_0 \cdot p(x|z=0) + p_1 \cdot p(x|z=1)$.

Last, the expected variance of $Y(z)$ with respect to a distribution $p(x,z)$:

$$\sigma^2_{Y(0)}(p(x,z)) = \int_{X\times Y} \big(Y(0) - \mu_0(x)\big)^2\, p\,(Y(0)|x)\, p(x,z)\, d\,Y(0)\, d\,x$$
$$\sigma^2_{Y(1)}(p(x,z)) = \int_{X\times Y} \big(Y(1) - \mu_1(x)\big)^2\, p\,(Y(1)|x)\, p(x,z)\, d\,Y(1)\, d\,x$$

$$\sigma^2_{Y(z)} = \min\left\{\sigma^2_{Y(z)}(p(x,z)), \sigma^2_{Y(z)}(p(x,1-z))\right\}, z = 0,1$$
$$\sigma^2_{Y} = \min\left\{\sigma^2_{Y(0)}, \sigma^2_{Y(1)}\right\}$$

Theorem 1. Let $\Phi : X \to R$ be a one-to-one invertible representation function, and let p_Φ be the distribution induced by Φ over R, i.e. $p_\Phi(r|t=1)$ and $p_\Phi(r|t=0)$ are the covariate distributions under treatment and control induced over R. We then have for any outcome prediction function $H : R \times \{0,1\} \to Y$:

$$L_{PEHE}(H,\Phi) \le 2\left(L_{F|z=0}(H,\Phi) + L_{F|z=1}(H,\Phi) + d_D(\Phi)\cdot\sum_{x\in X} \ell^{max}_{H,\Phi}(x) - 2\sigma^2_Y\right)$$

See proof in appendix.

Remark. Theorem 1 establishes the lower bound of PEHE for any outcome prediction function using representation learning, when the distance of representation space and the original covariate space is measured by the H divergence. The first two terms in (A.2) relate to the outcome prediction error, and are optimized by the typical supervised learning process using neural networks. The third term involves the product of the treatment distribution distance quantified by H divergence and the maximum expected loss among the two treatment groups. While the maximum expected loss is fixed given the optimal outcome prediction function H and the representation function Φ, our proposed algorithm minimizes the H divergence via optimization of the discriminator introduced in Section 3.4.1. Theorem 1 lays the theoretical foundation to ensure the proposed algorithm to provide low prediction error of the ITE measured by PEHE, and we further validate the performance via synthetic and real data simulations in Section 3.6 and 3.7.

3.6 Simulation Study

We design simulations studies to compare a number of state-of-art machine learning methods that are popular for estimating the potential outcomes. The methods under comparison are Causal Forest (CF), Bayesian Additive Regression Trees (BART), Treatment Agnostic Regression Network (TARNET) and Counterfactual Regression Network (CFRNET), and Generative Adversarial Nets for inference of Individualized Treatment Effects (GANITE). CF is a nonparametric random forests based algorithm that provides desirable asymptotic properties, and serves as a popular benchmark method (97); BART applies a Bayesian modeling approach by building a sum-of-trees model (99); TARNET applies representation learning without penalizing the representation balance; CFRNET incorporates the IPM loss into representation leaning (101); and GANITE is a GAN based method to learn the counterfactual distributions (95).

3.6.1 Data Generation Process

We consider various combinations of sample sizes and outcome surfaces to examine the performance of the afore mentioned methods. In total, there are 4 (sample size) × 3 (outcome model) = 12 simulation scenarios.

We generate $N \in \{200, 300, 500, 1000\}$ patients, with $P = 15$ covariates that are multivariate normal distributed as $X_i = (X_{i1}, \ldots, X_{iP}) \sim N\left(0, \sigma^2\left[(1-\rho)I_P + \rho 1_P 1_P^T\right]\right)$, where $\sigma^2 = 1$ is the marginal variance and $\rho = 0.3$ controls the correlation between the covariates for $i = 1, \ldots, N$. For each subject, the observed treatment assignment Z_i is simulated from a Bernoulli distribution $Z_i \sim Bernoulli(e(X_i))$, where $e(X_i)$ is the propensity score. We assume the baseline covariates serve as confounders and the propensity score model is

$$\text{logit}[e(X_i)] = X_i^T \alpha, \quad \alpha \sim \text{Unif}\left([-1, 1]^P\right).$$

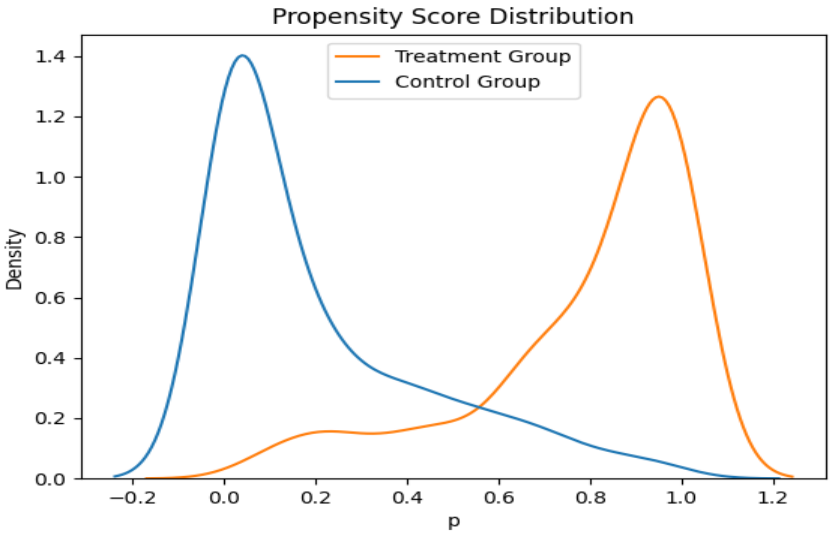
The realized values of the regression coefficients are α = (0.8, -0.8, -1, -0.8, 0.2, -0.4, 1, 0.6, 0.2, 0.6, -0.2, -0.4, -1, 0.6, 0.4), resulting in approximately 50% of the subjects being in the treatment group.

Overall, the observed outcome can be expressed as

$$Y_i = \mu_0(X_i) + Z_i \cdot c(X_i) + \epsilon_i, \quad \epsilon_i \overset{iid}{\sim} \mathbb{N}(0, 1), \tag{3.9}$$

where $\mu_0(X_i)$ is the conditional expectation of the potential outcome under control, $c(X_i)$ is the individual treatment effect that we are interested to estimate, and ϵ_i represents the random noise. This model implies that the conditional expectation of the potential outcome under treatment is $\mu_1(X_i) = \mu_0(X_i) + \text{c}(\text{X}_i)$. To assess the robustness of different methods, we consider three outcome generation processes that satisfy linear, piece-wise linear and non-linear surfaces, separately.

Figure 3.2: Propensity Score distribution



In *outcome model 1*, we assume a complex linear relationship motivated by Susan Athey at el. (105). Specifically,

$$\mu_0(X_i) = X_i^T \beta_0, \text{ with } \beta_0 \sim \text{Unif}\left([1,2]^P\right),$$
$$c(X_i) \sim X_i^T \beta_1 + 2,$$

The realized values of the outcome regression coefficients are $\beta_0 =$ (1.2, 1.1, 1.0, 1.8, 1.6, 2.0, 1.2, 1.3, 1.4, 1.1, 1.5, 1.1, 1.1, 1.0, 1.7), $\beta_1 =$ (1.5, 1.0, 1.9, 2.0, 1.5, 2.0, 2.0, 1.7, 2.0, 1.5, 1.4, 1.6, 1.9, 1.2, 1.2).

In *outcome model 2*, we assume a piece-wise linear relationship motivated by Kunzel et al. (106):

$$\mu_0(X_i) = X_i^T \beta_0, \text{ with } \beta_0 \sim \text{Unif}\left([-5,5]^P\right),$$
$$c(X_i) = 0.5\mathbb{I}(X_{i1} > 0.5) + \mathbb{I}(X_{i2} > 0.3) + 2\mathbb{I}(X_{i3} > 0, X_{i4} > 0.2)$$

where $\mathbb{I}(\cdot)$ stands for the indicator function, and the realized values of the outcome regression coefficients are $\beta_0 =$ (-5, 4, 3, -2, -2, -5, -2, 2, -2, 1, -3, -5, 4, 5, -4).

In *outcome model 3*, we assume a complex non-linear relationship motivated by Kang and Schafer (107):

$$\mu_0(X_i) = X_i^T \beta_0,$$
$$\mu_1(X_i) = \exp\big((X_i + W)\beta_0\big)$$
$$c(X_i) = \mu_1(X_i) - \mu_0(X_i)$$

where W is an offset matrix of the same dimension as X_i with every value equal to 0.5, β_0 is a vector of regression coefficients (0, 0.1, 0.2, 0.3, 0.4) randomly sampled with probabilities (0.6, 0.1, 0.1, 0.1, 0.1). The realized values of the outcome regression coefficients are $\beta_0 =$ (0.1, 0.2, 0.3, 0.1, 0, 0.3, 0, 0, 0, 0, 0, 0.1, 0, 0, 0.3).

Under each setting, we simulate 100 repeated data sets and evaluate the performance of different methods by the expected precision in estimation of heterogeneous effect (PEHE) (99),

$$\epsilon_{PEHE} = \frac{1}{n}\sum_{i=1}^{n}\Big(\widehat{\mu}_1(X_i) - \widehat{\mu}_0(X_i) - \big(\mu_1(X_i) - \mu_0(X_i)\big)\Big)^2,$$

where $\widehat{\mu}_0(X_i), \widehat{\mu}_1(X_i)$ are the estimated means from model, and $\mu_0(X_i), \mu_1(X_i)$ are the underlying true conditional means under control and treatment group. In addition of the estimation of individual treatment effect, we also evaluate the empirical absolute bias of ATE on the overall sample,

$$\epsilon_{ATE} = |\frac{1}{n}\sum_{i=1}^{n}\widehat{\mu}_1(X_i) - \widehat{\mu}_0(X_i) - ATE|,$$

where the true ATE is obtained from calculating the average treatment effect of a super population with 100,000 simulated subjects.

Table 3.1: Performance comparison between SMRLNN and state-of-the-art methods as the outcome model and sample sizes are varied with respect to ϵ_{PEHE}. The Monte Carlo SD is shown after ± .

Model	N	SMRLNN	TARNET	CFRNET	CF	BART	GANITE
1	200	**1.43 ± 0.07**	2.77 ± 0.09	2.17 ± 0.06	7.71 ± 0.77	5.96 ± 0.63	9.82 ± 0.76
1	300	**1.32 ± 0.11**	1.70 ± 0.05	1.46 ± 0.05	7.23 ± 0.70	4.88 ± 0.45	9.18 ± 0.57
1	500	**0.93 ± 0.07**	1.04 ± 0.02	0.99 ± 0.02	6.81 ± 0.52	3.64 ± 0.26	7.31 ± 0.51
1	1000	**0.70 ± 0.04**	0.82 ± 0.02	0.75 ± 0.02	6.15 ± 0.39	2.57 ± 0.15	6.46 ± 0.36
2	200	**1.56 ± 0.04**	2.22 ± 0.04	1.99 ± 0.03	2.70 ± 0.84	1.87 ± 0.25	2.57 ± 0.18
2	300	**1.51 ± 0.02**	1.68 ± 0.02	1.63 ± 0.02	2.43 ± 0.56	1.68 ± 0.22	2.55 ± 0.16
2	500	**1.35 ± 0.03**	1.43 ± 0.01	1.42 ± 0.01	2.22 ± 0.41	1.37 ± 0.17	2.43 ± 0.18
2	1000	1.15 ± 0.02	1.29 ± 0.01	1.27 ± 0.01	1.97 ± 0.28	**1.02 ± 0.13**	2.09 ± 0.16
3	200	**1.16 ± 0.13**	2.36 ± 0.07	2.08 ± 0.07	2.05 ± 0.67	2.06 ± 0.66	2.87 ± 0.72
3	300	**1.31 ± 0.12**	1.95 ± 0.06	1.66 ± 0.05	1.97 ± 0.58	1.91 ± 0.58	2.61 ± 0.60
3	500	**1.25 ± 0.22**	1.49 ± 0.06	1.28 ± 0.06	1.88 ± 0.47	1.77 ± 0.44	2.27 ± 0.47
3	1000	**1.01 ± 0.10**	1.17 ± 0.07	1.04 ± 0.06	1.83 ± 0.45	1.61 ± 0.42	2.08 ± 0.45

Table 3.2: Performance comparison between SMRLNN and state-of-the-art methods as the outcome model and sample sizes are varied with respect to ϵ_{ATE}. The Monte Carlo SD is shown after ± . The underlying true ATE of the three models are 2, 1.766, 3.306 respectively.

M	N	SMRLNN	TARNET	CFRNET	CF	BART	GANITE	DR
1	200	0.20 ± 0.07	0.53 ± 0.06	0.40 ± 0.05	0.56 ± 0.98	0.21 ± 1.04	0.67 ± 0.26	**0.14 ± 1.09**
1	300	0.18 ± 0.06	0.28 ± 0.03	0.29 ± 0.03	0.26 ± 0.80	0.08 ± 0.80	0.55 ± 0.21	**0.01 ± 0.91**
1	500	0.09 ± 0.04	0.25 ± 0.03	0.26 ± 0.03	0.20 ± 0.68	0.05 ± 0.62	0.43 ± 0.20	**0.02 ± 0.72**
1	1000	**0.06 ± 0.02**	0.20 ± 0.03	0.22 ± 0.03	0.21 ± 0.55	0.07 ± 0.46	0.39 ± 0.20	0.07 ± 0.45
2	200	0.29 ± 0.12	0.45 ± 0.06	0.38 ± 0.05	1.59 ± 1.33	0.30 ± 0.62	1.69 ± 0.62	**0.23 ± 0.66**
2	300	**0.18 ± 0.07**	0.32 ± 0.04	0.31 ± 0.04	1.50 ± 0.88	0.27 ± 0.40	1.63 ± 0.64	0.19 ± 0.44
2	500	0.15 ± 0.05	0.24 ± 0.03	0.25 ± 0.03	1.30 ± 0.63	0.18 ± 0.29	1.33 ± 0.55	**0.13 ± 0.25**
2	1000	**0.08 ± 0.02**	0.20 ± 0.03	0.20 ± 0.03	1.10 ± 0.46	0.13 ± 0.16	1.24 ± 0.51	0.12 ± 0.14
3	200	0.14 ± 0.06	0.39 ± 0.04	0.33 ± 0.04	**0.04 ± 0.28**	0.20 ± 0.28	0.45 ± 0.27	0.06 ± 0.31
3	300	0.13 ± 0.05	0.27 ± 0.03	0.25 ± 0.03	**0.05 ± 0.23**	0.15 ± 0.23	0.43 ± 0.29	0.13 ± 0.31
3	500	0.11 ± 0.05	0.22 ± 0.03	0.21 ± 0.03	**0.04 ± 0.20**	0.08 ± 0.17	0.28 ± 0.22	0.06 ± 0.19
3	1000	0.07 ± 0.03	0.17 ± 0.02	0.18 ± 0.02	**0.03 ± 0.12**	0.05 ± 0.12	0.24 ± 0.19	0.05 ± 0.14

Table 3.3: Performance comparison between SMRLNN and state-of-the-art methods as the numbers of covariates is increased with respect to ϵ_{PEHE}. The Monte Carlo SD is shown after ± .

P	N	SMRLNN	TARNET	CFRNET	CF	BART
50	200	**1.56 ± 0.13**	2.19 ± 0.12	2.68 ± 0.15	2.12 ± 0.20	1.66 ± 0.17
100	200	**1.75 ± 0.14**	2.33 ± 0.14	2.90 ± 0.16	2.23 ± 0.21	2.55 ± 0.20
200	200	**2.45 ± 0.16**	3.31 ± 0.35	3.49 ± 0.18	2.95 ± 0.37	3.69 ± 0.38
400	200	**3.44 ± 0.22**	7.37 ± 0.54	7.17 ± 0.61	4.59 ± 0.44	5.85 ± 0.40
800	200	**4.58 ± 0.31**	8.20 ± 0.44	5.37 ± 0.36	4.90 ± 0.60	6.89 ± 0.54

3.6.2 Simulation Results

Table 1 shows the performance of ITE estimation from different methods evaluated by PEHE. As the sample size increases from 200 to 1000, the PEHE of all methods monotonically decreases in all three outcome models. Overall, SMRLNN results in the smallest PEHE and Monte Carlo standard deviation across all the methods under comparison, substantially outperforming CF and GANITE. The difference is most pronounced when the outcome model is linear. Specifically, while the PEHE of GANITE ranges from 6.46 to 9.82, the PEHE of SMRLNN is only 0.70 to 1.43. Under piese-wise liner and nonlinear outcome surfaces, the PEHE of SMRLNN is about half of GANITE. The PEHE of CFRNET is slightly better than TARNET, showing penalizing the representation imbalance improves model performance when baseline covariates are imbalanced. Under Outcome model 1, CFRNET and TARNET outperform BART, while their performances are comparable under Outcome model 2 and 3. Last but not least, the Monte Carlo SD of CF, BART and GANITE are significantly larger than the representation learning based methods such as SMRLNN, TARNET and CFRNET.

Table 2 shows the performance of ATE estimation corresponding to Table 1. Similar to the trends observed in Table 1, the absolute bias of ATE estimation decreases as sample size increases. Our proposed method SMRLNN achieves the smallest bias and Monte Carlo standard deviation in comparison with other methods. The improvement of SMRLNN on

the ATE estimation is not as significant as the improvement of the ITE since our method is not designed for ATE estimation. While CF and GANITE remains having the largest bias, BART achieves comparable ATE bias with SMRLNN. The bias of TARNET and CFRNET lies between SMRLNN and CF under Outcome model 1 and 2, but CF results in the smallest ATE bias when outcome is nonlinear. Again, the variability of CF, BART and GANITE are significantly larger than the representation learning based methods.

Table 3 shows the performance of ITE estimation in high-dimension from different methods evaluated by PEHE. The sample size is fixed as 200, as the number of covariates increase from 50 to 800, the PEHE of all methods monotonically decreases in all three outcome models. Overall, SMRLNN results in the smallest PEHE and Monte Carlo standard deviation across all the methods under comparison.

The simulation results demonstrate that SMRLNN robustly outperforms state-of-the-art methods in terms of individual treatment effect estimations under a range of the examined scenarios. It also shows superiority in the estimation of ATE under linear and piece-wise linear outcome surfaces, as well as maintains small variance.

3.7 Real Data Experiments

In this section, we demonstrate the performance of SMRLNN architecture on real data experiments. The performance of causal inference methods is usually evaluated by a hybrid of real data variables and synthesized outcomes.

3.7.1 Infant Health and Development Program Dataset

The Infant Health and Development Program (IHDP) dataset (99) is a popular benchmark for evaluation of causal inference methods. IHDP is a randomized trial aiming to evaluate the efficacy of high-quality child care on premature infants. An observational study

was recreated from IHDP by removing a non-random portion from the subjects, resulting in 139 children in the treatment group, and 608 in the control. Following Shalit et al. (91), we use 25 pre-treatment covariates and the simulated response surface B of Hill et al. (99) For the IHDP data, since the outcome surface is known and both factual and counterfactual outcomes are simulated, we are able to compute the true ITE, and then evaluate using the empirical PEHE and ATE. We report the in-sample and out-of-sample performance on 100 replications of the data.

3.7.2 Jobs Dataset

Next, we evaluate various methods on another widely used benchmark based on a real-world dataset, Jobs (108; 91), which combines a randomized trial with observational data such that training can be conducted on both, but only the randomized data is used for evaluation. The Jobs data includes a binary outcome, 8 covariates, with the randomized trial having 297 treated and 425 controls, and the observational data having 2490 controls. For the Jobs data, since only factual outcomes are available but the testing set comes from a randomized controlled trial (RCT), empirical policy risk is used to evaluate the average loss on the randomized subset of Jobs. Policy Risk $\left(R_{pol}(\pi)\right)$ can be defined as the average loss in value when treating according to the policy implied by an ITE estimator:

$$R_{pol}(\pi) = \frac{1}{N}\sum_{i=1}^{N}\left[1-\left(\frac{1}{|\Pi_1 \cap T|}\sum_{i\in\Pi_1\cap T}\hat{y}_i(1)\times\frac{|\Pi_1\cap E|}{|E|}+\frac{1}{|\Pi_0\cap C\cap E|}\sum_{i\in\Pi_0\cap C\cap E}\hat{y}_i(0)\times\frac{|\Pi_0\cap E|}{|E|}\right)\right],$$

where $\hat{y}_i(z)$ is the predicted probability of employment under treatment z, $\Pi_z = \{i : z = \arg\max \hat{y}_i(z)\}$ is the set of randomized subjects whose predicted potential probability is larger under treatment z, E represents the set of subjects in the RCT, C is the set of control subjects, T is the set of treated, and $|\cdot|$ represents the sample size of a set. In addition, we

evaluate the empirical absolute bias of ATT on the randomized set E:

$$\epsilon_{ATT} = \left| \frac{1}{|T|} \left(\sum_{i \in T} \widehat{\mu}_1(X_i) - \widehat{\mu}_0(X_i) \right) - ATT \right|,$$

where $ATT = |T|^{-1} \sum_{i \in T} y_i - |C \cap E|^{-1} \sum_{i \in C \cap E} y_i$ is the average treatment effect for the treated calculated from the RCT set.

3.7.3 MIMIC-III Sepsis Cohort Dataset

The Medical Information Mart for Intensive Care-III (MIMIC-III) (18) is a public critical care database which includes all patients admitted to the ICUs of Beth Israel Deaconess Medical Center in Boston, MA from 2008 - 2012. The database contains information about patients' demographics, diagnosis codes, laboratory tests, vital signs, and clinical events, for over 350 million values across various sources of data. We evaluate the treatment effect of mechanical ventilation on in-hospital mortality in adult patients fulfilling the international consensus sepsis-3 criteria. Of the 20,225 eligible admissions, 4,210 (20.8%) received mechanical ventilation, and 1,208 (28.7%) experienced in-hospital deaths. We pre-specify 47 baseline covariates based on clinical knowledge, including demographics, Elixhauser premorbid status, vital signs, laboratory values, fluids and vasopressors received and fluid balance (109). Data variables with multiple measurements are recorded at the time of sepsis diagnosis. Table A.2 presents the baseline characteristics of these covariates. Significant imbalance is observed in many covariates, with mechanical ventilation patients being on average having more severer symptoms as evidenced by larger initial SOFA score, elixhauser score, SGOT, SGPT, IV fluid intake, and Urine output over 4 hours.

Propensity score matching (PSM) is a broadly used method for causal inference on real data. In the literature, the individual treatment effect $\tau(X_i)$ is usually approximated by a matched pair approach, i.e., find a nearest neighbor of unit i and take the difference in

Table 3.4: Performance of ITE estimation with IHDP real-world dataset. Bold indicates the method with the best performance for each dataset. The Monte Carlo SD is shown after ± .

Metric	SMRLNN	CFRNET	TARNET	CF	BART	GANITE
ϵ_{PEHE}	**0.74 ± .01**	0.76 ± .02	0.95 ± .02	3.8 ± 0.2	2.3 ± 0.1	2.4 ± 0.4
ϵ_{ATE}	**0.19 ± .01**	0.27 ± .01	0.35 ± .02	0.40 ± .03	0.34 ± .02	0.38 ± .03

Table 3.5: Performance of ITE estimation with Jobs real-world dataset. Bold indicates the method with the best performance for each dataset. The Monte Carlo SD is shown after ± .

Metric	SMRLNN	CFRNET	TARNET	CF	BART	GANITE
R_{pol}	**0.18 ±.01**	0.21 ± .01	0.21 ± .01	0.20 ± .02	0.25 ± .02	0.20 ± .02
ϵ_{ATT}	**0.05 ± .01**	0.08 ± .03	0.10 ± .03	0.07 ± .03	0.08 ± .03	0.08 ± .03

outcomes of the pair as the approximated "true" ITE (91). In this paper, we fitted a logistic regression propensity score mode with the 25 covariates to estimate the propensity score. For each patient receiving mechanical ventilation (MV), we find a matched pair using the k-nearst neighbor method without replacement, and then take the difference in outcomes of the pair. We evaluate different methods using PEHE based on this approximated ground truth ITE.

The performance of ITE estimation with three real-world dataset are shown in Table 4-7. On both the IHDP and MIMIC-III dataset, SMRLNN achieves the best accuracy with respect to ϵ_{PEHE} and ϵ_{ATE}. On the Jobs dataset, SMRLNN is also the best method with respect to R_{pol} and ϵ_{ATE}.

Table 3.6: Performance of ITE estimation with sepsis cohort from MIMIC-III dataset. Bold indicates the method with the best performance for each dataset. The Monte Carlo SD is shown after ± .

Metric	SMRLNN	CFRNET	TARNET	CF	BART	GANITE
ϵ_{PEHE}	**0.56 ± 0.07**	0.71 ± 0.10	0.63 ± 0.08	0.72 ± 0.11	0.64 ± 0.06	0.94 ± 0.12
ϵ_{ATE}	**0.04 ± 0.01**	0.08 ± 0.02	0.07 ± 0.01	0.09 ± 0.03	0.05 ± 0.01	0.11 ± 0.05

3.8 Conclusion and Discussion

In this paper, we proposed a novel representation learning algorithm to estimate the individual treatment effect. We then presented the generalized bounds for any representation learning function using the H divergence. As our proposed algorithm minimizes the H divergence via optimization of the discriminator, we also use a structure keeper to capture the valuable information from the original covariates to avoid information loss in the representation learning process. We showed that the proposed algorithm outperforms start-of-art methods in extensive synthetic settings under various sample sizes, covariates, outcome models, and real data benchmarks in randomized trials, social studies, and electronic health records applications. While the assumptions of strong ignorability is fundamental in the identification of causal estimands, they may not always hold in real-world scenarios. In cases where hidden or unobserved confounding variables are suspected to influence both the treatment assignment and the outcomes, it is imperative to consider strategies to account for and mitigate the impact of such hidden confounding. Further work can explore the development of algorithms capable of automatically detecting and controlling for hidden confounders for modeling complex relationships in high-dimensional data. In addition, preserving certain structures that may not be relevant could potentially introduce biases. To prevent an excessive penalty on the structure keeper component, one can opt to exclude this component, allowing for a baseline performance assessment of representation learning. Last, future studies are required to extend the methodology to accommodate multiple treatments.

CHAPTER

4

GENERATIVE CONDITIONAL MISSING IMPUTATION NETWORKS

4.1 Introduction

Missing data is an ubiquitous problem when dealing with the real-world data, and it often hinders the application and validation of statistical methods for downstream analysis. Missing mechanism is the first crucial factor to consider when dealing with missing data since since the appropriate choice for missing imputation depends on finding out the right missing pattern behind the data (46). Missing mechanisms can be summarized into three categories: missing completely at random (MCAR), missing at random (MAR), and missing not at random (MNAR). MCAR pertains to situations where the likelihood of data

being missing is unrelated to the data itself, whether observed or unobserved. Formally, MCAR is upheld when: $\Pr(R \mid X) = \Pr(R)$. Here, "R" and "X" denote missingness and data (both observed and missing data), respectively. Under MCAR, there is no discernible link between the data and the occurrence of missing values.(47). A more flexible and realistic assumption for missing data is MAR, where the likelihood of missingness is dependent on observed data but is independent of unobserved data after accounting for the observed information. Mathematically, MAR holds when: $\Pr(R \mid X) = \Pr(R \mid X_{obs})$. The final category of missingness is MNAR, in which the likelihood of missingness is linked to both observed and unobserved data. Mathematically, MNAR is described by the following equation: $\Pr(R \mid X) = \Pr(R \mid X_{obs}, X_{mis})$ This indicates that the relationship between missingness and the data persists even after accounting for observed information.

Traditional methods inappropriately handling missing data usually cause decreased sample size and power, loss of data representation and biased results (110). For example, complete case analysis assumes MCAR and would introduce bias under other missing mechanisms, if the missingness is independent of the outcome of interest given the variables in the model. It will always cause loss of power due to the unused deleted data. Last observation carried forward method for longitudinal studies are generally biased (111). Single imputation method fills in the missing values commonly imposing a regression prediction model, but it fails to take the uncertainty in the imputations into account, leading to inflated false positive discoveries and bias in the downstream analysis (61). On the contrast, multiple imputation (MI) methods such as Multiple Imputation by Chained Equations (MICE) (112; 113) are more robust and informative in that they capture the underling distribution of each variable more accurately, and provide uncertainty estimation.

Recent development of deep generative methods have shown promising performance to capture the latent structure, inter-variable correlations, and representations of complex

high-dimensional data (114). For example, Yoon et al. (17) proposed a generative adversarial nets (GAIN) to learn the desired distribution and assisted discriminator with information about the original missingness. Nazabal et al. (115) designed variational autoencoders for fitting incomplete heterogenerous data (HI-VAE). MisGAN proposed by Li, Jiang, and Marlin in 2019 (116) focuses on improving imputation accuracy by leveraging generative models within the GAN framework. Ghalebikesabi et al. presented "Deep Generative Missingness Pattern-Set Mixture Models" (117) which use deep generative techniques to address the complexity of missing data patterns, offering a new dimension of imputation quality. Deep Generative Modelling and Imputation of Incomplete Data Sets (MIWAE) (118) strives to improve the quality of imputed data through innovative generative techniques. The extension of the MIWAE model, Deep Generative Modelling with Missing Not at Random Data (not-MIWAE) (119), further demonstrates the application of deep generative modeling in complex, real-world datasets by addressesing the challenging issue of missing data that is not missing at random. Ma and Zhang contributed to the field in 2021 with their work on "Identifiable Generative Models" (120) that are designed for imputing missing data not at random, addressing even more intricate missing data scenarios. The studies referenced above underscore the increasing interest and innovation in the use of GANs for missing data imputation. While GAIN only handles homogenerous data type, and HI-VAE assumes different likelihoods models for heterogenerous data, in this paper, we propose a generative conditional approach to impute the missing values of a given dataset. Our approach is designed to address critical gaps in the existing literature, with a primary focus on flexibility in handling both continuous and categorical data without imposing distribution assumptions. Specially, we first show the generative conditional missing imputation networks (GCMI) presents theoretical properties under the MCAR mechanism, and then incorporate the multiple imputation via chained equation-based approach to increase the model stability and imputation performance. Last, We conduct extensive simulation

and real data experiments on benchmark datasets to show the superior performance of the proposed methods against other state-of-the-art missing imputation methods. Our approach is geared towards improving the quality and accuracy of imputed data, making them valuable tools for handling incomplete datasets in various domains.

4.2 Problem Formulation

Consider a data matrix with N samples/objects and P features/variables. Denote the data matrix as $X = (x_1, x_2, \ldots, x_P) \in \mathbb{R}^{N \times P}$, where $x_j = \left(x_{1j}, \ldots, x_{Nj}\right)^T$ is a feature vector containing N attributes for the jth feature, and $i = 1, \ldots, N$ is the row index. Each of the N attributes could be either missing or observed. When x_j is not completely missing or observed, the data matrix can be grouped into four parts based on the missing values of each feature vector x_j,: 1. The observed attributes of variable x_j, denoted by x_j^{obs}; 2. the missing attributes of variable x_j, denoted by x_j^{mis}; 3. the rest of feature matrix correspond to observed x_j, denoted by $X_{(-j)}^{obs}$; 4. the rest of feature matrix correspond to missing values of x_j denoted by $X_{(-j)}^{mis}$. In addition, we use the index set $miss(j)$, $obs(j)$ to denote the missing and observed rows from the jth feature, and n_j to denote the number of elements in $obs(j)$. The goal is to impute the missing values for each feature j without overfitting on the observed data. Moreover, missing imputation is commonly accompanied with downstream tasks such as outcome predictions. For example, researchers may be interested in filing a variety of patient level features in the electronic health records, such as demographics, diagnoses, medications, vital signs, and laboratory data, in order to predict an adverse or disease outcome. When the features are correctly imputed, it has the potential to enhance the subsequent performance.

4.2.1 Generative Conditional Networks for Missing Imputation (GCMI)

For each variable j, the goal is to generate the missing attributes x_j^{mis} given the observations from $X_{(-j)}^{mis}$, which can be sampled from the conditional distribution $p(x_j|X_{(-j)})$ of feature j. As a result, the GCMI is an ensemble of generative conditional networks consisting of P pairs of conditional generators and discriminators.

Generator G_j : The j-th generator G_j is designed to impute the missing values in the j-th column. Let z denotes an independent k-dimensional noise from $N(0,I)$, then the generator $G_j : \mathbb{R}^{P-1} \times \mathbb{R}^k \to \mathbb{R}^1$ is a function which maps $X_{(-j)}$ and noise z to a vector of imputations $\hat{x}_j = G_j(X_{(-j)}, z)$. In the training stage, the generator uses $X_{(-j)}^{obs}$ and x_j^{obs} to learn the conditional distribution of $p_{(x_{(j)}|X_{(-j)})}$, then we apply G_j on $X_{(-j)}^{mis}$ to generate imputation of x_j^{mis}.

Discriminator D_j : The j-th discriminator $D_j : \mathbb{R}^{P-1} \times \mathbb{R}^1 \to (0,1)$ is a function which given the information about $X_{(-j)}$, tries to distinguish whether the attributes in x_j are observed or imputed values.

We define the loss function of G_j, D_j as:

$$\min_{D_j} L_D(D_j) = \frac{1}{2}\mathbb{E}_{\boldsymbol{x}\sim p_{x_j}(\boldsymbol{x})}\left[(D_j(X_{(-j)}, \boldsymbol{x}) - 2)^2\right] + \frac{1}{2}\mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}\left[(D_j(X_{(-j)}, G_j(X_{(-j)}, \boldsymbol{z})))^2\right]$$
$$\min_{G_j} L_G(G_j) = \frac{1}{2}\mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}\left[(D_j(X_{(-j)}, G_j(X_{(-j)}, \boldsymbol{z})) - 1)^2\right],$$

Theoretical Properties of GCMI

To ensure the reliability of imputed values, we establish the theoretical properties of GCMI under the Missing Completely at Random (MCAR) mechanisms. Our goal is to find a generator $G_j(x_{-j}, z)$ that can be as close to the conditional distribution $P_{x_j|X_{(-j)}=x_{-j}}$ as possible. Matching the conditional distribution of $G_j(x_{-j}, z)$ with $P_{x_j|X_{(-j)}=x_{-j}}$ for a given $x_{-j} \in X_{(-j)}$ is equivalent to matching the joint distribution of $(X_{(-j)}, G_j(X_{(-j)}, z))$ and the joint

distribution of $(X_{(-j)}, x_j)$, if the same marginal distribution of $X_{(-j)}$ is involved (121). For this purpose, we first introduce the concept of f-divergence which measures the distribution difference between two probability functions. Suppose P and Q are two probability density functions on $\mathbb{R}^d$ with density p, q respectively, and Q is absolutely continuous with respect to P. The f-divergence (122) of Q with respect to P is defined by

$$\mathbb{D}_f(q\|p) = \int f\left(\frac{q(z)}{p(z)}\right) p(z) dz,$$

where f is a non-negative and convex function taking the minimum at $f(1) = 0$. By Jensen's inequality, $\mathbb{D}_f(q\|p) \geq 0$ for every q, p and $\mathbb{D}_f(q\|p) = 0$ if and only if $q = p$.

We will show that by defining the above loss functions, we actually use the χ^2 divergence, a specific form of the f-divergence.

$$\begin{aligned}
L_D(D_j) &= \frac{1}{2}\mathbb{E}_{\boldsymbol{x}\sim p_{x_j}(\boldsymbol{x})}\left[(D_j(X_{(-j)}, \boldsymbol{x}) - 2)^2\right] + \frac{1}{2}\mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}\left[(D_j(X_{(-j)}, G_j(X_{(-j)}, \boldsymbol{z})))^2\right] \\
&= \frac{1}{2}\int_{X_j} p_{x_j}(\boldsymbol{x})\left[(D_j(X_{(-j)}, \boldsymbol{x}) - 2)^2\right] \mathrm{d}\boldsymbol{x} + \frac{1}{2}\int_{Z} p_{\boldsymbol{z}}(\boldsymbol{z})\left[(D_j(X_{(-j)}, G_j(X_{(-j)}, \boldsymbol{z})))^2\right] \mathrm{d}\boldsymbol{z} \\
&= \frac{1}{2}\int_{X_j} p_{x_j}(\boldsymbol{x})\left[(D_j(X_{(-j)}, \boldsymbol{x}) - 2)^2\right] \mathrm{d}\boldsymbol{x} + \frac{1}{2}\int_{X_j} p_{G_j}(\boldsymbol{x})\left[(D_j(X_{(-j)}, \boldsymbol{x}))^2\right] \mathrm{d}\boldsymbol{x} \\
&= \frac{1}{2}\int_{X_j} \{p_{x_j}(\boldsymbol{x})\left[(D_j(X_{(-j)}, \boldsymbol{x}) - 2)^2\right] + p_{G_j}(\boldsymbol{x})\left[(D_j(X_{(-j)}, \boldsymbol{x}))^2\right]\} \mathrm{d}\boldsymbol{x}
\end{aligned}$$

To find the D_j which minimizes $L_D(D_j)$, let

$$\frac{\partial \{p_{x_j}(\boldsymbol{x})\left[(D_j(X_{(-j)}, \boldsymbol{x}) - 2)^2\right] + p_{G_j}(\boldsymbol{x})\left[(D_j(X_{(-j)}, \boldsymbol{x}))^2\right]\}}{\partial D_j(X_{(-j)}, \boldsymbol{x})} = 0,$$

then we have the optimal discriminator D_j for a fixed G_j:

$$D_j^*(X_{(-j)}, \boldsymbol{x}) = \frac{2p_{x_j}(\boldsymbol{x})}{p_{x_j}(\boldsymbol{x}) + p_{G_j}(\boldsymbol{x})}$$

Then we can reformulate $L_G(G_j)$ as:

$$
\begin{aligned}
L_G(G_j) &= \frac{1}{2}\mathbb{E}_{\boldsymbol{x}\sim p_{x_j}(\boldsymbol{x})}\Big[(D_j^*(X_{(-j)},\boldsymbol{x})-1)^2\Big]+\frac{1}{2}\mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}\Big[(D_j^*(X_{(-j)},G_j(X_{(-j)},\boldsymbol{z}))-1)^2\Big] \\
&= \frac{1}{2}\int_{X_j} p_{x_j}(\boldsymbol{x})\Big[(D_j^*(X_{(-j)},\boldsymbol{x})-1)^2\Big]\mathrm{d}\boldsymbol{x}+\frac{1}{2}\int_{Z} p_{\boldsymbol{z}}(\boldsymbol{z})\Big[(D_j^*(X_{(-j)},G_j(X_{(-j)},\boldsymbol{z}))-1)^2\Big]\mathrm{d}\boldsymbol{z} \\
&= \frac{1}{2}\int_{X_j} p_{x_j}(\boldsymbol{x})\Big[(D_j^*(X_{(-j)},\boldsymbol{x})-1)^2\Big]\mathrm{d}\boldsymbol{x}+\frac{1}{2}\int_{X_j} p_{G_j}(\boldsymbol{x})\Big[(D_j^*(X_{(-j)},\boldsymbol{x})-1)^2\Big]\mathrm{d}\boldsymbol{x} \\
&= \frac{1}{2}\int_{X_j} p_{x_j}(\boldsymbol{x})\left[(\frac{2p_{x_j}(\boldsymbol{x})}{p_{x_j}(\boldsymbol{x})+p_{G_j}(\boldsymbol{x})}-1)^2\right]\mathrm{d}\boldsymbol{x}+\frac{1}{2}\int_{X_j} p_{G_j}(\boldsymbol{x})\left[(\frac{2p_{x_j}(\boldsymbol{x})}{p_{x_j}(\boldsymbol{x})+p_{G_j}(\boldsymbol{x})}-1)^2\right]\mathrm{d}\boldsymbol{x} \\
&= \frac{1}{2}\int_{X_j} \frac{\big((p_{x_j}(\boldsymbol{x})+p_{G_j}(\boldsymbol{x}))-2p_{G_j}(\boldsymbol{x})\big)^2}{p_{x_j}(\boldsymbol{x})+p_{G_j}(\boldsymbol{x})}\mathrm{d}x \\
&= \frac{1}{2}\mathbb{D}_{\chi^2}(p_{x_j}+p_{G_j}\|2p_{G_j})
\end{aligned}
$$

where χ^2_{Pearson} is the Pearson χ^2 divergence. Thus minimizing above equation yields minimizing the Pearson χ^2 divergence between $p_{x_j}+p_{G_j}$ and $2p_{G_j}$.

Therefore, we proved that loss functions $L_D(D_j)$ and $L_G(G_j)$ achieves global minimum if and only if $D_j^*(X_{(-j)},\boldsymbol{x})=\frac{2p_{x_j}(\boldsymbol{x})}{p_{x_j}(\boldsymbol{x})+p_{G_j}(\boldsymbol{x})}$ and $p_{x_j}=p_{G_j}$

In training process, we consider the empirical version of $L_D(D_j)$ and $L_G(G_j)$:

$$
\begin{aligned}
\widehat{L}_D(D_j) &= \frac{1}{2n}\sum_{i=1}^{n}(D_j(X_{i,(-j)},x_{i,j})-2)^2+\frac{1}{2n}\sum_{i=1}^{n}(D_j(X_{i,(-j)},G_j(X_{i,(-j)},z_i)))^2 \\
\widehat{L}_G(G_j) &= \frac{1}{2n}\sum_{i=1}^{n}(D_j(X_{i,(-j)},G_j(X_{i,(-j)},z_i))-1)^2
\end{aligned}
$$

To further improve the generator accuracy, we add the following penalty to the generator loss:

$$
L_{acc}(x,\hat{x})=\begin{cases}(\hat{x}-x)^2, & \text{if } x \text{ is continuous} \\ -x\log(\hat{x})-(1-x)\log(1-\hat{x}), & \text{if } x \text{ is binary}\end{cases}
$$

4.3 Algorithm

Minimizing the loss functions can be achieved by an iterative process. First, we fill the missing values in X using mean imputation or other methods. Second, sort the columns of variables in an ascending order based on the missing proportion in x_j, $j = 1, \ldots, P$. Third, for each variable x_j, the missing values are imputed by first training an GCIN with response x_j^{obs} and predictors $X_{(-j)}^{obs}$; then, predicting the missing values x_j^{mis} by applying the trained GCMI to $X_{(-j)}^{mis}$.

The algorithm GCIN can be visualized in Figure 4.1, comprising two key components: a conditional generator (G) and a conditional discriminator (D). Both G and D utilize the rectified linear unit (ReLU) activation function.

The role of G is to generate imputed values for missing positions. It takes a random noise vector z as input, based on the given condition $X_{(-j)}$, and produces the imputed value for $X_{(j)}$ as output. The ReLU activation function is incorporated within the generator to introduce non-linearity and enhance the model's expressive capacity.

D is responsible for assessing the authenticity of the data samples it receives as input, aiming to distinguish between real and imputed values, also considering the given condition $X_{(-j)}$. The discriminator also employs the ReLU activation function to introduce non-linearity in its computations.

During training, the parameters of G and D are updated alternatively. In each iteration, the generator generates synthetic samples, which are then passed to the discriminator along with real samples from the training set. The discriminator evaluates the samples and provides feedback to both G and itself. By iteratively updating the parameters of G and D, GCIN aims to achieve a balance where the generator produces high-quality imputed values that are indistinguishable from real samples, and the discriminator becomes increasingly proficient at distinguishing between real and imputation values.

We then integrate a multiple imputation strategy based on chained equations to enhance the stability and imputation performance of our model. That is, the GCIN imputation procedure is repeated until a stopping criterion is met. The maximum iterations for chained equation imputation in GCMI is 20. Multiple imputation offers several significant benefits in the context of handling missing data. First, multiple imputation mitigates bias introduced by missing data by generating multiple complete datasets, each with a different set of imputed values. This reduces the impact of randomness in imputation and helps to recover the true underlying data distribution more effectively. Second, it has the ability to obtain valid variance quantification. Valid variance estimation is essential for accurately assessing the uncertainty associated with parameter estimates and hypothesis testing.The process involves combining the variability within each imputed dataset (within-imputation variance) with the variability between the imputed datasets (between-imputation variance) to obtain an overall estimate of variance. The aggregated variance is often calculated as a combination of within-imputation and between-imputation variances following Rubin's Rules (123). These rules combine the variances in a way that properly accounts for the uncertainty due to missing data, yielding a valid and robust estimate of the parameter's variance. GCMI, shown in Figure 4.2, is based on GCIN and multiple imputation to impute the missing values. The implementation details for them are provided in Algorithm 1.

4.4 Experiments

Methods for missing data imputation are usually evaluated by synthetic and real-data based simulations. In this section, we validate the performance of GCMI under various missing mechanisms, missing proportions and sample sizes regarding the accuracy of

covariates imputation. When the primary goal is to impute missing covariates, we compare the performance of GCMI with several traditional and state-of-the-art missing data imputation methods:

- **Mean Imputation**: Missing values are imputed using the mean of each continuous variable and mode of each categorical variable.
- **MICE**: An iterative multiple imputation method that models the missing data conditioning on other observed variables. We use the ***mice*** Python package with default hyperparameter setting.
- **Matrix imputation**: Fit a low-rank matrix approximation to a matrix with missing values via nuclear-norm regularization.
- **MissForest**: An iterative random forest imputation algorithm trained on the observed data to predict the missing data. We use the ***MissForest*** Python package with default hyperparameter setting.
- **GAIN**: An generative adversarial imputation method that imputes missing data by adversarially training the generator and discriminator with an additional hint matrix to reveal the missingness. We implement GAIN using author provided code at `http://github.com/jsyoon0823/GAIN` with default hyperparameter setting.

4.4.1 Synthetic data

We generate $N = 2000$ complete observations with $P = 15$ independent continuous variables from the multivariate normal distribution

$$\boldsymbol{X}_i = (X_{i1}, \ldots, X_{iP})^T \sim N\left(0, \sigma^2\left[(1-\rho)I_P + \rho\, 1_P 1_P^T\right]\right)$$

where $\sigma^2 = 1$ is the marginal variance and $\rho = 0.3$ controls the correlation between the covariates. A continuous response variable Y_i is generated from a linear regression model

$$Y_i = X_i^T \alpha + \epsilon_i, \quad \alpha \sim \text{Unif}\left([-1,1]^P\right), \epsilon_i \sim N(0,1), \tag{4.1}$$

where ϵ_i represents the individual level random noise and the regression coefficients α are simulated randomly from a uniform distribution with realized values (0.542, -0.769, 0.298, -0.156, 0.778, -0.391, -0.629, 0.311, 0.913, -0.025, -0.676, 0.512, 0.840, -0.265, -0.678).

4.4.2 Real-data

Besides the synthetic data, we compare the performance of GCMI with several state-of-the-art missing data imputation methods using three popular real benchmark datasets from University of California, Irvine (UCI) machine learning repository, and the Medical Information Mart for Intensive Care (MIMIC) database. (124).

Researchers and healthcare professionals have harnessed the ICU medical database to explore a wide range of critical care topics, from predictive modeling to treatment efficacy assessments. However, like many real-world clinical databases, the ICU dataset is not immune to the challenge of missing data, a ubiquitous issue in clinical research. Missing data can significantly impact the validity and generalizability of research findings, making it imperative to develop robust strategies for imputing missing values.

In this paper, we delve into the realm of missing data imputation within the context of the ICU Database. We aim to address the critical need for accurate and efficient methods to handle missing data in this rich clinical dataset. By doing so, we hope to enhance the reliability and applicability of research conducted using the ICU database, ultimately contributing to the advancement of critical care and patient outcomes. In the following sections, we will explore state-of-art missing imputation methods on two ICU dataset,

shedding light on their strengths and limitations in the pursuit of rigorous and insightful clinical research.

MIMIC-III Laboratory Data

The Medical Information Mart for Intensive Care-III (MIMIC-III) (18) is a public critical care database which includes all patients admitted to the ICUs of Beth Israel Deaconess Medical Center in Boston, MA from 2008 - 2012. The database contains information about patients' demographics, diagnosis codes, laboratory tests, vital signs, and clinical events, for over 350 million values across various sources of data.

The Lab table in the MIMIC-III dataset extends beyond the confines of the patient's ICU stay, encompassing their entire hospitalization, which may also include outpatient records. This comprehensive dataset offers an opportunity to predict a patient's mortality rate even before their admission to the ICU. However, numerous lab features exhibit missing rates exceeding 50%, and in some instances, these rates soar as high as 80%. Fig A.1 presents the missing proportion of 36 common laboratory tests. This high incidence of missing data may potentially have substantial repercussions on the accuracy of certain prediction methods. For ICU patients, the prescription of laboratory tests lacks universal standardization, leading to a vast and sparsely populated lab feature space. Previous research conducted by Frassica in 2005 (22) examined a staggering 45,188 lab tests and profiles across three distinct ICUs, revealing that 80% of the tests and profiles in these ICUs could be adequately represented by fewer than 25 tests. Building on this, Sharafoddini (23) modified lab test items based on the Medical Information Mart for Intensive Care III database and selected a set of 36 lab items for predicting in-hospital mortality. Drawing inspiration from these prior analyses, we posit the hypothesis that an increase in the number of laboratory test items may not necessarily result in improved prediction accuracy. This stems from the inherent characteristics of the three-dimensional patient-lab-day array, characterized by extreme sparsity and a high rate

of missing data. Therefore, out of the 726 distinct laboratory test features available, our analysis focuses on those documented after the patient's initial inpatient admission. These values are further aggregated on a daily basis to reduce sparsity. In cases where a patient has multiple instances of the same laboratory test on a given day, we compute their average values. Averaging these values helps to reduce the impact of noise and provides a more stable and representative estimate of the underlying variable. If a laboratory test is absent for a patient on a specific day, we treat it as a missing value. The informativeness of the testing procedure is evident, as it encompasses both the presence or absence of the test and the frequency of testing. Specifically, the decision for certain individuals to undergo testing carries valuable information. Conversely, patients who do not undergo testing likely exhibit variations in the distribution of covariates, possibly indicating differences in health status—ranging from being sicker to less sick, depending on the specific test in question. Subsequent to this initial aggregation, our laboratory data is structured into a 3-dimensional patient-lab array, with distinct dimensions representing patient ID, date, and laboratory test. Although we address the issue of laboratory test sparsity, the challenge of high missing rates still persists. We choose to perform imputation for the top features reported in literature for predicting mortality (125). Among these features, the most critical ones are platelet count, red cell distribution width, alanine aminotransferase, and blood urea nitrogen.

Since we do not have the underlying truth for the unobserved laboratory values, to evaluate the missing imputation performance, we randomly drop the observed data with extra missing rates from 10% to 50%, and then calculate the imputation error based on the known values of such dropped data and their imputed values.

eICU Collaborative Research Database

The eICU Collaborative Research Database is a vast and valuable resource for conducting in-depth research in the field of critical care medicine. Created through a collaboration between Philips Healthcare and the MIT Laboratory for Computational Physiology, this database was established to provide researchers with a comprehensive collection of clinical data from a diverse and extensive patient population.

The eICU Collaborative Research Database contains records from over 200,000 critical care unit admissions across more than 200 hospitals in the United States. With data on more than 139,000 individual patients and thousands of variables covering vital signs, laboratory results, medications, diagnoses, and interventions, this dataset offers a comprehensive view of patient care. Spanning several years and collected at high temporal granularity, it allows researchers to explore critical care practices at a national scale and over time. These quantities highlight the substantial breadth and depth of the eICU database, making it a valuable asset for research endeavors, including the investigation of missing data imputation techniques in critical care settings.

With the primary objective of improving patient care and outcomes, the eICU database contains a wealth of information spanning various medical domains, including patient demographics, vital signs, laboratory results, medications, and treatment interventions. By offering insights into the care and outcomes of those patients admitted to ICUs in the United States, the eICU database facilitates research in areas such as clinical decision support, predictive modeling, epidemiology, and quality improvement.

In our approach to handling the eICU Collaborative Research Database, we employ a data preprocessing pipeline akin to the one applied in the MIMIC-III dataset analysis. This process yields a dataset containing 40 crucial features for our imputation. Given the absence of ground truth for the unobserved laboratory values, we adopt a consistent evaluation

strategy to assess the performance of missing data imputation. To achieve this, we introduce additional missingness into the observed data, randomly discarding data points at rates ranging from 10% to 50%. Subsequently, we assess the quality of imputation techniques by computing the imputation error, comparing the imputed values to the known values of the dropped data.

4.4.3 Experiment Setup

Missing datasets are generated under MCAR, MAR, and MNAR mechanisms separately.

- MCAR: Every entry of the original complete covariates matrix $\boldsymbol{X}$ is randomly deleted with a constant probability $p \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6\}$.
- MAR: We generate missing values in $\boldsymbol{X_m} = (X_5, \ldots, X_p)^T$ according to the missing propensity score model $\text{logit}[M_j = 1] = \boldsymbol{X_c}^T \beta_j$, for $\boldsymbol{X_c} = (X_1, \ldots, X_4)^T, \beta_j \sim \text{Unif}\big([-1,1]^P\big)$, $j = 1, 2, \ldots, p$. M_j is the missing indicator vector for column j.
- MNAR: Every entry of the original complete covariates matrix $\boldsymbol{X}$ is randomly deleted with a constant probability $p = b_0 + b_1 * X_{ij}$, where $b_0 \in \{-4, -3.1, -2.5, -2, -1.5, -1\}$, $b_1 = 3$.

The neural networks used in the simulations are specified as follows. The batch size for training data is set as 256 (when $n_{train} = 5000$). The maximum training epochs is 10000. The maximum iterations for chained equation imputation in GCMI is 20.

4.5 Results

All the methods are evaluated based on the rooted mean squared error (RMSE) between the original complete covariates matrix X and the imputed covariates matrix $\hat{X}$. Under

each simulation scenario, we repeat the experiments on 100 Monte Carlo (MC) datasets, and report the average RMSE across 100 repetitions.

4.5.1 Synthetic data imputation

Figure 4.4, 4.5 and 4.6 show the RMSE for missing data imputation under various missing mechanisms and missing rates in the covariate matrix imputation only. When missing mechanisms is MAR, mean imputation results in the highest RMSE. This is expected as mean imputation doesn't assume any prediction model and therefore is unable to account for the correlations between covariates or use it for prediction. GAIN and Matrix imputation method generate the second least favorable result, followed by MissForest and MICE. Our proposed GCMI consistently results in the highest prediction accuracy and the lowest RMSE across a range of missing rates. The results suggest that multiple imputation based methods (MissForest, MICE and GCMI) have higher precision and are more stable compared to the single imputation methods (Mean, Matrix and GAIN). Similar trends are observed under MCAR and MNAR.

4.6 Real-data imputation

In this section, we present and discuss the results of missing imputation methods on two ICU datasets (MIMIC-III and eICU Collaborative Research Database) using the proposed GCMI method. The results are reported in terms of Root Mean Squared Error (RMSE) as mean ± standard error (SE). The missing rates vary from 10% to 50% for different features in the datasets. Throughout all real data experiments, the missing data are generated from the mechanism of Missing At Random (MAR). The outcomes for imputing missing covariates from the MIMIC-III and eICU datasets are presented in Tables 4.1 and 4.2.

The evaluation of missing imputation methods on the MIMIC-III Laboratory Data, as

outlined in Table 4.1, provides insights into their performance when applied to this specific critical care dataset. In a simulated scenario involving blockwise missing data with $n = 5000$ patients and $p = 24$ laboratory features, the missing rates were systematically varied from 10% to 50%, reflecting real-world challenges in handling missing values in clinical data.

The evaluation involved an array of imputation methods, ranging from basic techniques like mean imputation to more sophisticated approaches such as multiple imputation, MI-KNN, MissForest, Soft Impute, Soft_MaskImpute, GAIN, and the proposed GCMI.

Mean imputation, serving as a benchmark, consistently produced relatively stable imputation errors around 0.073 across all missing rates. However, it became evident that for the precision and reliability of clinical analyses, advanced imputation methods were necessary. Multiple imputation, MI-KNN, and MissForest, as more advanced techniques, exhibited a clear improvement in imputation accuracy as the missing rates increased. This trend is particularly important for MIMIC-III data analysis, where missing values can be prevalent, emphasizing the significance of accurate imputation for clinical decision support and research. In contrast, the Soft Impute method, which relies on matrix completion, demonstrated competitive but somewhat variable results. Its imputation errors ranged from 0.080 to 0.098, indicating a degree of sensitivity to the extent of missingness. Soft_MaskImpute, characterized by relatively high imputation errors, displayed diminishing performance as missing rates increased. This underscores the need for alternative strategies in cases where data exhibit high levels of missingness. GAIN, a method that employs generative adversarial networks, delivered reasonable imputation performance, but its imputation errors increased from 0.061 to 0.070 as the missing rates escalated. This suggests the method's sensitivity to the extent of missingness.

Most notably, the proposed GCMI method consistently outperformed the other imputation methods for the MIMIC-III dataset. It maintained impressively low imputation errors, ranging from 0.058 to 0.068, even as missing rates increased. This consistency underscores

the potential of GCMI as a robust and reliable solution for imputing missing values in the MIMIC-III Laboratory Data, offering clinicians and researchers a promising tool for enhancing the accuracy and reliability of their clinical analyses, predictive modeling, and decision support systems in the critical care domain.

In the eICU Collaborative Research Database, various missing data imputation methods were evaluated for a dataset of 5000 patients with 40 features, under MAR with missing rates ranging from 10% to 50%. In the table 4.2, the proposed GCMI also consistently outperformed other imputation techniques with imputation errors of 0.060 to 0.081. These results indicate the method's robustness and reliability, offering enhanced accuracy in imputing missing values in the ICU dataset.

In conclusion, the findings from this evaluation reveal the vital importance of selecting an appropriate imputation method when dealing with missing data in the ICU medical data. GCMI, in particular, offers a promising solution for researchers in the critical care domain, as it consistently delivers accurate imputations, regardless of the missing data rates, and may significantly enhance the reliability of subsequent analyses and predictions in this crucial healthcare context.

4.7 Conclusion and Discussion

In conclusion, our Generative Conditional Missing Imputation (GCMI) method introduced in this paper marks a significant advancement in the realm of data imputation. Harnessing the resilience of conditional GAN architectures, GCMI excels in handling missing data across various mechanisms in both simulated and real data experiments. It establishes a new standard in imputation accuracy, surpassing existing methods across a range of contexts. Although the training of GAN-based algorithms like GCMI can be challenging, particularly with small datasets, the benefits are clear. GCMI's adaptability to incomplete

datasets during training and its maintenance of inter-variable correlations are substantial strengths that enhance its applicability. The empirical validation against real-life datasets confirms GCMI's potential as a powerful tool in the imputation method arsenal, offering improved prediction quality even in the face of missing information. In practical scenarios, the assumptions of Missing At Random (MAR) and Missing Not At Random (MNAR) are frequently encountered in Electronic Health Record (EHR) data. While our empirical findings demonstrate the superior performance of the proposed algorithm under these assumptions, there is a need for future work to augment theoretical development. This will contribute to a deeper understanding of how the algorithm can be refined for enhanced applicability in specific scenarios governed by these assumptions.

Algorithm 1: Make an initial guess for the missing values in X using a chosen imputation method. Then, sort the variables x_j, $j = 1, \ldots, P$ based on the missing proportion starting with the lowest amount.

while *imputation stopping criterion not converged* **do**

for $j \in \{1, \ldots, P\}$ **do**

- Split data into fours parts as $x_j^{obs}, x_j^{mis}, X_{(-j)}^{obs}, X_{(-j)}^{mis}$.
- Fit a generative conditional imputation neural network (GCIN) model $x_j^{obs} \sim G_j(X_{(-j)}^{obs}, z)$, which includes
  Generator G_j with standard normal initial weights W, and Discriminator D_j with standard normal initial weights U.

while *GCIN not converged* **do**

- Update D_j by descending its stochastic gradient:

$$g_U = \nabla_U \frac{1}{B}\left[\sum_{i=1}^{B/2}\left(D_j(X_{i,(-j)}^{obs}, X_{i,j}^{obs}) - 2\right)^2 + \sum_{i=B/2+1}^{B}\left(D_j(X_{i,(-j)}^{obs}, G_j(X_{i,(-j)}^{obs}, z_i)\right)^2 + \alpha_1 \|U\|^2\right.$$

- Randomly select $c_i \in obs(j)$ for $i = 1, 2, \ldots, B$,
- Update G_j by descending its stochastic gradient:

$$g_W = \nabla_{\boldsymbol{W}} \frac{1}{B}\sum_{i=1}^{B}\left(D_j(X_{c_i,(-j)}^{obs}, G_j(X_{c_i,(-j)}^{obs}, z_{c_i})) - 1\right)^2 + \alpha_{21}\|W\|^2 + \alpha_{22} L_{acc}\left(x_{c_i,j}^{obs}, G_j(X_{c_i,(-j)}^{obs}, z_{c_i})\right)$$

- Update weights parameters $[W, U] \leftarrow \left[W - \eta_1 g_W, U - \eta_2 g_U\right]$
- Check GCIN convergence criterion

end

Update X: $x_j^{mis} \leftarrow G_j(X_{(-j)}^{mis}, z)$

end

Check imputation convergence criterion

end

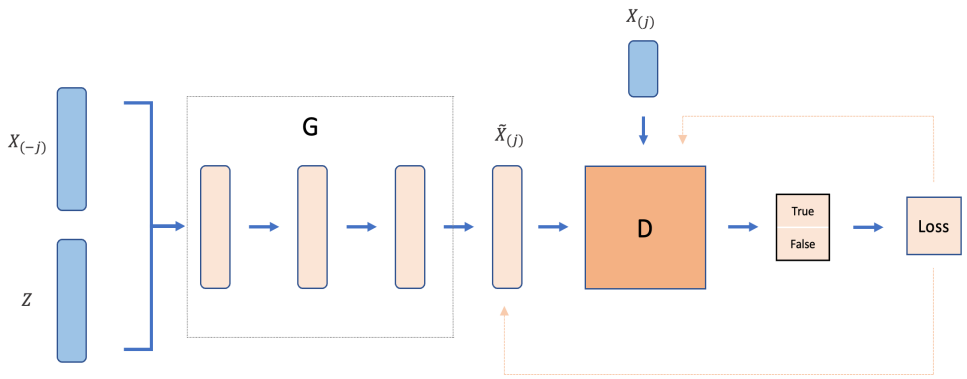


Figure 4.1: GCIN: G takes a random noise vector z as input, conditioned on the given condition $X_{(-j)}$, and produces the imputed value for $X_{(j)}$ as output. D aims to evaluate input data samples and differentiate between real and imputed values, considering the given condition $X_{(-j)}$. The ReLU activation function is incorporated within the generator to introduce non-linearity and enhance the model's expressive capacity.

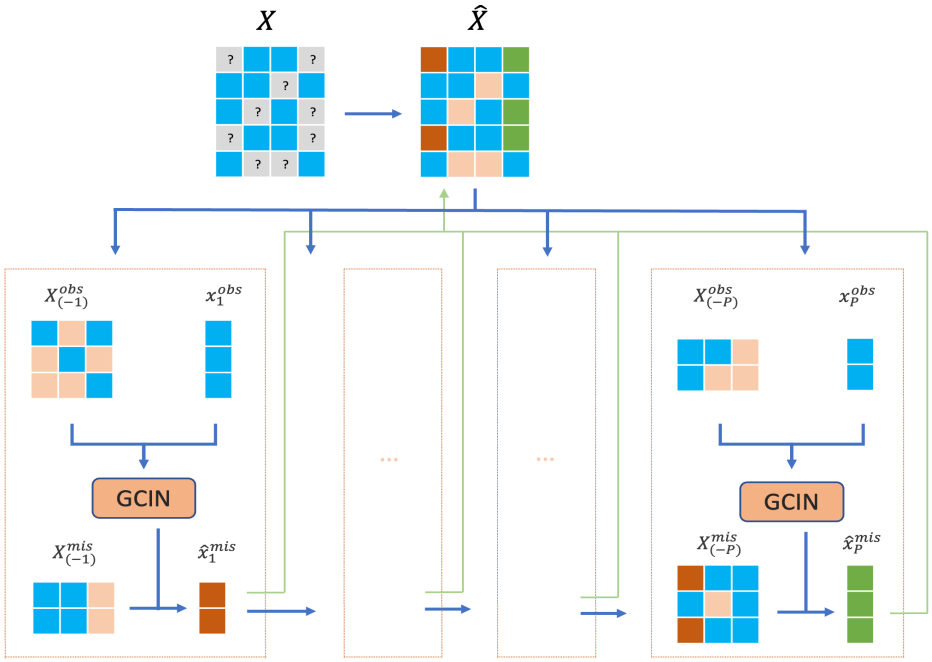


Figure 4.2: GCMI: Perform initial imputation for missing values in X using a chosen method and sort variables based on missing proportions. Iterate through each column j and fit a generative conditional imputation neural network (GCIN) on $x_j^{obs} \sim G_j(X_{(-j)}^{obs}, z)$. Once the GCIN algorithm converges, update X with $x_j^{mis} \leftarrow G_j(X_{(-j)}^{mis}, z)$ and check convergence for imputation.

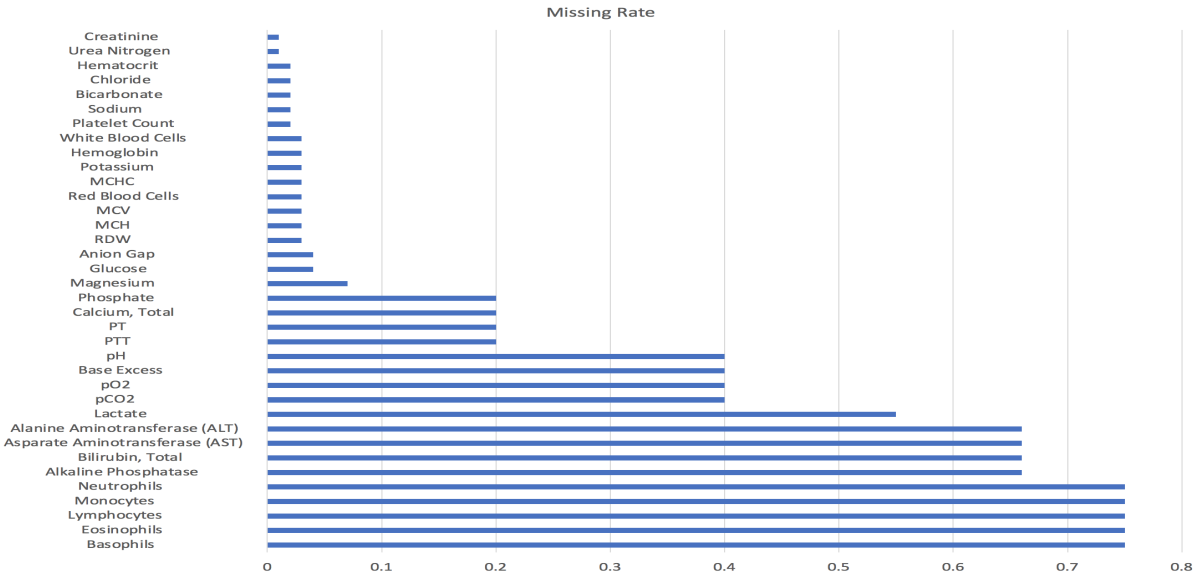


Figure 4.3: Missing Proportions of the 36 Common Laboratory Tests in MIMIC-III.

Figure 4.4: MCAR

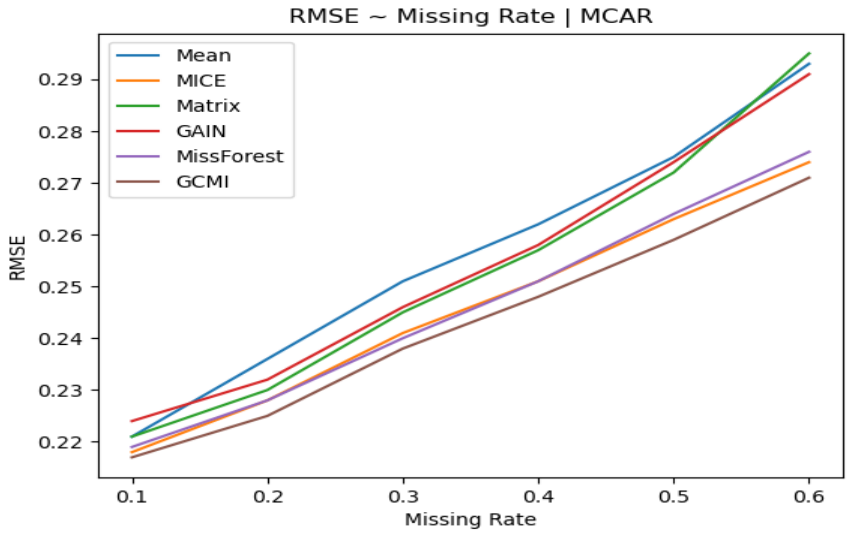


Figure 4.5: MAR

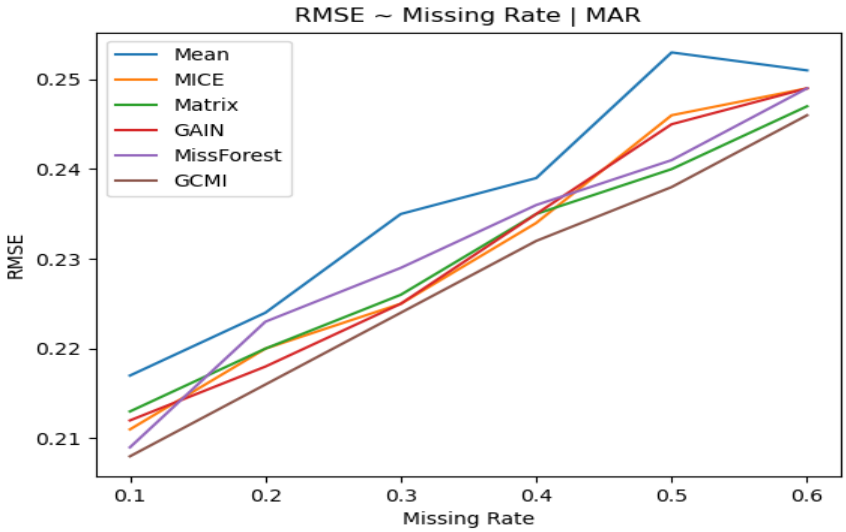


Figure 4.6: MNAR

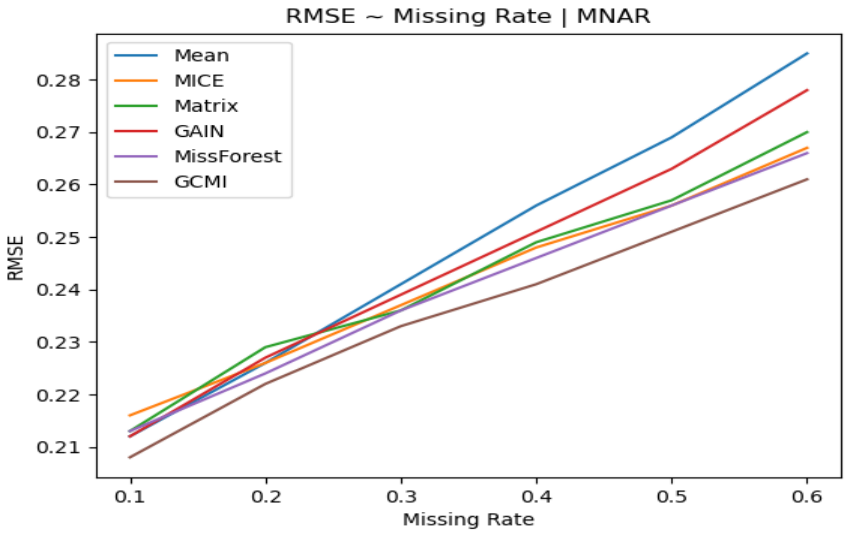


Table 4.1: MIMIC-III Laboratory Data - Missing Imputation: Simulation of Blockwise Missing Data with $n = 5000$ and $p = 24$ under MAR. The missing rates of the features range from 10% to 50%. Performance results are presented as mean ± standard deviation.

Missing Rate	0.1	0.2	0.3	0.4	0.5
Mean Imputation	0.073 ± 0.001	0.073 ± 0.001	0.073 ± 0.001	0.073 ± 0.001	0.073 ± 0.001
Multiple Imputation	0.061 ± 0.001	0.063 ± 0.001	0.065 ± 0.000	0.067 ± 0.001	0.069 ± 0.001
MI-KNN	0.064 ± 0.001	0.066 ± 0.001	0.068 ± 0.001	0.070 ± 0.001	0.071 ± 0.001
MissForest	0.059 ± 0.001	0.061 ± 0.001	0.064 ± 0.001	0.066 ± 0.001	0.069 ± 0.001
Soft	0.080 ± 0.002	0.094 ± 0.001	0.098 ± 0.006	0.095 ± 0.006	0.092 ± 0.004
Soft_Mask	0.097 ± 0.003	0.091 ± 0.003	0.097 ± 0.004	0.106 ± 0.005	0.125 ± 0.007
GAIN	0.061 ± 0.001	0.063 ± 0.001	0.065 ± 0.001	0.068 ± 0.001	0.070 ± 0.001
GCMI	$\mathbf{0.058 \pm 0.001}$	$\mathbf{0.061 \pm 0.001}$	$\mathbf{0.063 \pm 0.001}$	$\mathbf{0.066 \pm 0.001}$	$\mathbf{0.068 \pm 0.002}$

Table 4.2: eICU Collaborative Research Database - Missing Imputation: Simulation of Blockwise Missing Data with $n = 5000$ and $p = 40$ under MAR. The missing rates of the features range from 10% to 50%. Performance results are presented as mean ± standard deviation.

Missing Rate	0.1	0.2	0.3	0.4	0.5
Mean Imputation	0.097 ± 0.001	0.097 ± 0.001	0.097 ± 0.001	0.097 ± 0.001	0.097 ± 0.002
Multiple Imputation	0.065 ± 0.001	0.070 ± 0.001	0.073 ± 0.001	0.078 ± 0.001	0.083 ± 0.002
MI-KNN	0.073 ± 0.001	0.076 ± 0.001	0.080 ± 0.001	0.084 ± 0.001	0.088 ± 0.002
MissForest	0.062 ± 0.001	0.067 ± 0.001	0.073 ± 0.001	0.078 ± 0.001	0.083 ± 0.002
Soft	0.072 ± 0.001	0.076 ± 0.001	0.079 ± 0.001	0.084 ± 0.001	0.090 ± 0.002
Soft_Mask	0.100 ± 0.001	0.107 ± 0.002	0.112 ± 0.002	0.114 ± 0.002	0.113 ± 0.002
GAIN	0.076 ± 0.001	0.079 ± 0.001	0.082 ± 0.001	0.086 ± 0.001	0.112 ± 0.002
GCMI	$\mathbf{0.060 \pm 0.001}$	$\mathbf{0.065 \pm 0.002}$	$\mathbf{0.070 \pm 0.001}$	$\mathbf{0.076 \pm 0.001}$	$\mathbf{0.081 \pm 0.002}$

REFERENCES

[1] Marjut Varpula, Minna Tallgren, Katri Saukkonen, Liisa-Maria Voipio-Pulkki, and Ville Pettilä. Hemodynamic variables related to outcome in septic shock. *Intensive care medicine*, 31(8):1066–1071, 2005.

[2] Jean-Louis Vincent and Daniel De Backer. Circulatory shock. *New England Journal of Medicine*, 369(18):1726–1734, 2013.

[3] Daniel De Backer, Patrick Biston, Jacques Devriendt, Christian Madl, Didier Chochrad, Cesar Aldecoa, Alexandre Brasseur, Pierre Defrance, Philippe Gottignies, and Jean-Louis Vincent. Comparison of dopamine and norepinephrine in the treatment of shock. *New England Journal of Medicine*, 362(9):779–789, 2010.

[4] 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.

[5] 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.

[6] Jason Waechter, Anand Kumar, Stephen E Lapinsky, John Marshall, Peter Dodek, Yaseen Arabi, Joseph E Parrillo, R Phillip Dellinger, Allan Garland, Cooperative Antimicrobial Therapy of Septic Shock Database Research Group, et al. Interaction between fluids and vasoactive agents on mortality in septic shock: a multicenter, observational study. *Critical care medicine*, 42(10):2158–2168, 2014.

[7] Marc Leone, Pierre Asfar, Peter Radermacher, Jean-Louis Vincent, and Claude Martin. Optimizing mean arterial pressure in septic shock: a critical reappraisal of the literature. *Critical care*, 19(1):101, 2015.

[8] Maurizio Cecconi, Daniel De Backer, Massimo Antonelli, Richard Beale, Jan Bakker, Christoph Hofer, Roman Jaeschke, Alexandre Mebazaa, Michael R Pinsky, Jean Louis Teboul, et al. Consensus on circulatory shock and hemodynamic monitoring. task force of the european society of intensive care medicine. *Intensive care medicine*, 40(12):1795–1815, 2014.

[9] Kitty S Chan, Jinnet B Fowles, and Jonathan P Weiner. Electronic health records and the reliability and validity of quality measures: a review of the literature. *Medical Care Research and Review*, 67(5):503–527, 2010.

[10] Debashis Mondal and Donald B Percival. Wavelet variance analysis for gappy time series. *Annals of the Institute of Statistical Mathematics*, 62(5):943–966, 2010.

[11] Kira Rehfeld, Norbert Marwan, Jobst Heitzig, and Jürgen Kurths. Comparison of correlation analysis techniques for irregularly sampled time series. *Nonlinear Processes in Geophysics*, 18(3):389–404, 2011.

[12] Pedro J García-Laencina, José-Luis Sancho-Gómez, and Aníbal R Figueiras-Vidal. Pattern classification with missing data: a review. *Neural Computing and Applications*, 19(2):263–282, 2010.

[13] Rahul Mazumder, Trevor Hastie, and Robert Tibshirani. Spectral regularization algorithms for learning large incomplete matrices. *The Journal of Machine Learning Research*, 11:2287–2322, 2010.

[14] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

[15] Ian R White, Patrick Royston, and Angela M Wood. Multiple imputation using chained equations: issues and guidance for practice. *Statistics in medicine*, 30(4):377–399, 2011.

[16] 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.

[17] Jinsung Yoon, James Jordon, and Mihaela Van Der Schaar. Gain: Missing data imputation using generative adversarial nets. *arXiv preprint arXiv:1806.02920*, 2018.

[18] 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.

[19] Zheng Dai, Siru Liu, Jinfa Wu, Mengdie Li, Jialin Liu, and Ke Li. Analysis of adult disease characteristics and mortality on mimic-iii. *PloS one*, 15(4):e0232176, 2020.

[20] Wei-Qi Wei, Lisa A Bastarache, Robert J Carroll, Joy E Marlo, Travis J Osterman, Eric R Gamazon, Nancy J Cox, Dan M Roden, and Joshua C Denny. Evaluating phecodes, clinical classification software, and icd-9-cm codes for phenome-wide association studies in the electronic health record. *PloS one*, 12(7):e0175508, 2017.

[21] Jinmiao Huang, Cesar Osorio, and Luke Wicent Sy. An empirical evaluation of deep learning for icd-9 code assignment using mimic-iii clinical notes. *Computer methods and programs in biomedicine*, 177:141–153, 2019.

[22] Joseph J Frassica. Frequency of laboratory test utilization in the intensive care unit and its implications for large-scale data collection efforts. *Journal of the American Medical Informatics Association*, 12(2):229–233, 2005.

[23] Anis Sharafoddini, Joel A Dubin, David M Maslove, and Joon Lee. A new insight into missing data in intensive care unit patient profiles: observational study. *JMIR medical informatics*, 7(1):e11605, 2019.

[24] Chenxi Sun, Shenda Hong, Moxian Song, and Hongyan Li. A review of deep learning methods for irregularly sampled medical time series data. *arXiv preprint arXiv:2010.12493*, 2020.

[25] 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. In *Advances in neural information processing systems*, pages 3146–3154, 2017.

[26] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

[27] Cheng Chen, Qingmei Zhang, Qin Ma, and Bin Yu. Lightgbm-ppi: Predicting protein-protein interactions through lightgbm with multi-information fusion. *Chemometrics and Intelligent Laboratory Systems*, 191:54–64, 2019.

[28] Rafal Jozefowicz, Wojciech Zaremba, and Ilya Sutskever. An empirical exploration of recurrent network architectures. In *International conference on machine learning*, pages 2342–2350. PMLR, 2015.

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

[30] Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233–240, 2006.

[31] 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.

[32] Li-wei Lehman, Mohammed Saeed, William Long, Joon Lee, and Roger Mark. Risk stratification of icu patients using topic models inferred from unstructured progress notes. In *AMIA annual symposium proceedings*, volume 2012, page 505. American Medical Informatics Association, 2012.

[33] Marzyeh Ghassemi, Tristan Naumann, Finale Doshi-Velez, Nicole Brimmer, Rohit Joshi, Anna Rumshisky, and Peter Szolovits. Unfolding physiological state: Mortality modelling in intensive care units. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 75–84, 2014.

[34] Jacob Calvert, Qingqing Mao, Jana L Hoffman, Melissa Jay, Thomas Desautels, Hamid Mohamadlou, Uli Chettipally, and Ritankar Das. Using electronic health record

collected clinical variables to predict medical intensive care unit mortality. *Annals of Medicine and Surgery*, 11:52–57, 2016.

[35] 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.

[36] Nora El-Rashidy, Shaker El-Sappagh, Tamer Abuhmed, Samir Abdelrazek, and Hazem M El-Bakry. Intensive care unit mortality prediction: an improved patient-specific stacking ensemble model. *IEEE Access*, 8:133541–133564, 2020.

[37] Nicole Gray Weiskopf and Chunhua Weng. Methods and dimensions of electronic health record data quality assessment: enabling reuse for clinical research. *Journal of the American Medical Informatics Association*, 20(1):144–151, 2013.

[38] Kitty S Chan, Jinnet B Fowles, and Jonathan P Weiner. Electronic health records and the reliability and validity of quality measures: a review of the literature. *Medical Care Research and Review*, 67(5):503–527, 2010.

[39] Nicole G Weiskopf, George Hripcsak, Sushmita Swaminathan, and Chunhua Weng. Defining and measuring completeness of electronic health records for secondary use. *Journal of biomedical informatics*, 46(5):830–836, 2013.

[40] 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.

[41] Roderick J Little, Ralph D'Agostino, Michael L Cohen, Kay Dickersin, Scott S Emerson, John T Farrar, Constantine Frangakis, Joseph W Hogan, Geert Molenberghs, Susan A Murphy, et al. The prevention and treatment of missing data in clinical trials. *New England Journal of Medicine*, 367(14):1355–1360, 2012.

[42] Jonathan AC Sterne, Ian R White, John B Carlin, Michael Spratt, Patrick Royston, Michael G Kenward, Angela M Wood, and James R Carpenter. Multiple imputation for missing data in epidemiological and clinical research: potential and pitfalls. *Bmj*, 338, 2009.

[43] Craig D Newgard and Jason S Haukoos. Advanced statistics: missing data in clinical research—part 2: multiple imputation. *Academic Emergency Medicine*, 14(7):669–678, 2007.

[44] Jason S Haukoos and Craig D Newgard. Advanced statistics: missing data in clinical research—part 1: an introduction and conceptual framework. *Academic Emergency Medicine*, 14(7):662–668, 2007.

[45] Patrick E McKnight, Katherine M McKnight, Souraya Sidani, and Aurelio Jose Figueredo. *Missing data: A gentle introduction*. Guilford Press, 2007.

[46] Andrew Gelman and Jennifer Hill. *Data analysis using regression and multilevelhierarchical models*, volume 1. Cambridge University Press New York, NY, USA, 2007.

[47] Amanda N Baraldi and Craig K Enders. An introduction to modern missing data analyses. *Journal of school psychology*, 48(1):5–37, 2010.

[48] Tshilidzi Marwala. *Computational intelligence for missing data imputation, estimation and management: knowledge optimization techniques*. Information Science Reference Hershey, New York, 2009.

[49] Hyun Kang. The prevention and handling of the missing data. *Korean journal of anesthesiology*, 64(5):402, 2013.

[50] Thad Dunning, David A Freedman, Steven Turner, and William Outhwaite. Modeling selection effects. *The handbook of social science methodology, eds. Steven Turner and William Outhwaite*, pages 225–31, 2007.

[51] Jau-Huei Lin and Peter J Haug. Exploiting missing clinical data in bayesian network modeling for predicting medical problems. *Journal of biomedical informatics*, 41(1):1–14, 2008.

[52] Donald B Rubin, Hal S Stern, and Vasja Vehovar. Handling "don't know" survey responses: the case of the slovenian plebiscite. *Journal of the American Statistical Association*, 90(431):822–828, 1995.

[53] Derrick A Bennett. How can i deal with missing data in my study? *Australian and New Zealand journal of public health*, 25(5):464–469, 2001.

[54] Joseph L Schafer. Multiple imputation: a primer. *Statistical methods in medical research*, 8(1):3–15, 1999.

[55] Estevam R Hruschka, Eduardo R Hruschka, and Nelson FF Ebecken. Bayesian networks for imputation in classification problems. *Journal of Intelligent Information Systems*, 29(3):231–252, 2007.

[56] John W Graham. Missing data analysis: Making it work in the real world. *Annual review of psychology*, 60:549–576, 2009.

[57] Roderick JA Little and Donald B Rubin. Bayes and multiple imputation. *Statistical analysis with missing data*, pages 200–220, 2002.

[58] Joseph L Schafer and John W Graham. Missing data: our view of the state of the art. *Psychological methods*, 7(2):147, 2002.

[59] Lisa L Doove, Stef Van Buuren, and Elise Dusseldorp. Recursive partitioning for missing data imputation in the presence of interaction effects. *Computational Statistics & Data Analysis*, 72:92–104, 2014.

[60] Roderick JA Little. Regression with missing x's: a review. *Journal of the American statistical association*, 87(420):1227–1237, 1992.

[61] A Rogier T Donders, Geert JMG Van Der Heijden, Theo Stijnen, and Karel GM Moons. A gentle introduction to imputation of missing values. *Journal of clinical epidemiology*, 59(10):1087–1091, 2006.

[62] Liu Peng and Lei Lei. A review of missing data treatment methods. *Intelligent Information Management Systems and Technologies*, 1(3):412–419, 2005.

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

[64] Jaap Brand. *Development, implementation and evaluation of multiple imputation strategies for the statistical analysis of incomplete data sets*. 1999.

[65] Stef Van Buuren and KMICE Groothuis-Oudshoorn. Multivariate imputation by chained equations. *J Stat Softw*, 45(3):1–67, 2011.

[66] Michael W Robbins, Sujit K Ghosh, and Joshua D Habiger. Imputation in high-dimensional economic data as applied to the agricultural resource management survey. *Journal of the American Statistical Association*, 108(501):81–95, 2013.

[67] Elizabeth A Stuart, Melissa Azur, Constantine Frangakis, and Philip Leaf. Multiple imputation with large data sets: a case study of the children's mental health initiative. *American journal of epidemiology*, 169(9):1133–1139, 2009.

[68] Lane F Burgette and Jerome P Reiter. Multiple imputation for missing data via sequential regression trees. *American journal of epidemiology*, 172(9):1070–1076, 2010.

[69] Bhekisipho Twala. An empirical comparison of techniques for handling incomplete data using decision trees. *Applied Artificial Intelligence*, 23(5):373–405, 2009.

[70] GÃvSrard Biau, Luc Devroye, and GÃĄbor Lugosi. Consistency of random forests and other averaging classifiers. *Journal of Machine Learning Research*, 9(Sep):2015–2033, 2008.

[71] 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.

[72] Daniel J Stekhoven and Peter Bühlmann. Missforest—non-parametric missing value imputation for mixed-type data. *Bioinformatics*, 28(1):112–118, 2012.

[73] 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), 2013.

[74] Fei Tang and Hemant Ishwaran. Random forest missing data algorithms. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 10(6):363–377, 2017.

[75] 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.

[76] N Solaro, A Barbiero, G Manzi, and PA Ferrari. A simulation comparison of imputation methods for quantitative data in the presence of multiple data patterns. *Journal of Statistical Computation and Simulation*, 88(18):3588–3619, 2018.

[77] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[78] David Goldberg, David Nichols, Brian M Oki, and Douglas Terry. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12):61–70, 1992.

[79] Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. Convex multi-task feature learning. *Machine learning*, 73(3):243–272, 2008.

[80] Zhang Liu and Lieven Vandenberghe. Semidefinite programming methods for system realization and identification. In *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, pages 4676–4681. IEEE, 2009.

[81] Emmanuel J Candes and Yaniv Plan. Matrix completion with noise. *Proceedings of the IEEE*, 98(6):925–936, 2010.

[82] T Tony Cai, Wen-Xin Zhou, et al. Matrix completion via max-norm constrained optimization. *Electronic Journal of Statistics*, 10(1):1493–1525, 2016.

[83] Trevor Hastie, Rahul Mazumder, Jason D Lee, and Reza Zadeh. Matrix completion and low-rank svd via fast alternating least squares. *The Journal of Machine Learning Research*, 16(1):3367–3402, 2015.

[84] Aude Sportisse, Claire Boyer, and Julie Josse. Imputation and low-rank estimation with missing not at random data. *Statistics and Computing*, 30(6):1629–1643, 2020.

[85] Donald B Rubin. Causal inference using potential outcomes: Design, modeling, decisions. *Journal of the American Statistical Association*, 100(469):322–331, 2005.

[86] Peng Ding and Fan Li. Causal inference: A missing data perspective. *arXiv preprint arxiv.1712.06170*, 2017.

[87] José R Zubizarreta. Stable weights that balance covariates for estimation with incomplete outcome data. *Journal of the American Statistical Association*, 110(511):910–922, 2015.

[88] Paul R Rosenbaum and Donald B Rubin. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):41–55, 1983.

[89] Nathan Kallus. Generalized optimal matching methods for causal inference. *J. Mach. Learn. Res.*, 21:62–1, 2020.

[90] J M Robins, M A Hernan, and B Brumback. Marginal structural models and causal inference in epidemiology. *Epidemiology*, 11(5):550–560, 2000.

[91] Uri Shalit, Fredrik D Johansson, and David Sontag. Estimating individual treatment effect: generalization bounds and algorithms. In *International Conference on Machine Learning*, pages 3076–3085. PMLR, 2017.

[92] Fredrik Johansson, Uri Shalit, and David Sontag. Learning representations for counterfactual inference. In *International conference on machine learning*, pages 3020–3029. PMLR, 2016.

[93] Claudia Shi, David Blei, and Victor Veitch. Adapting neural networks for the estimation of treatment effects. *Advances in neural information processing systems*, 32, 2019.

[94] Ahmed Alaa and Mihaela Schaar. Limits of estimating heterogeneous treatment effects: Guidelines for practical algorithm design. In *International Conference on Machine Learning*, pages 129–138. PMLR, 2018.

[95] Jinsung Yoon, James Jordon, and Mihaela Van Der Schaar. Ganite: Estimation of individualized treatment effects using generative adversarial nets. In *International Conference on Learning Representations*, 2018.

[96] Donald B Rubin. Randomization analysis of experimental data: The fisher randomization test comment. *Journal of the American Statistical Association*, 75(371):591–593, 1980.

[97] Jonathan Davis and Sara B Heller. Using causal forests to predict treatment heterogeneity: An application to summer jobs. *American Economic Review*, 107(5):546–50, 2017.

[98] Stefan Wager and Susan Athey. Estimation and inference of heterogeneous treatment effects using random forests. *Journal of the American Statistical Association*, 113(523):1228–1242, 2018.

[99] Jennifer L Hill. Bayesian nonparametric modeling for causal inference. *Journal of Computational and Graphical Statistics*, 20(1):217–240, 2011.

[100] Maike Hohberg, Peter Pütz, and Thomas Kneib. Treatment effects beyond the mean using distributional regression: Methods and guidance. *PloS one*, 15(2):e0226514, 2020.

[101] Fredrik D Johansson, Nathan Kallus, Uri Shalit, and David Sontag. Learning weighted representations for generalization across designs. *arXiv preprint arXiv:1802.08598*, 2018.

[102] Yao Zhang, Alexis Bellot, and Mihaela Schaar. Learning overlapping representations for the estimation of individualized treatment effects. In *International Conference on Artificial Intelligence and Statistics*, pages 1005–1014. PMLR, 2020.

[103] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2794–2802, 2017.

[104] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1):151–175, 2010.

[105] Susan Athey, Guido Imbens, Thai Pham, and Stefan Wager. Estimating average treatment effects: Supplementary analyses and remaining challenges. *American Economic Review*, 107(5):278–81, 2017.

[106] Sören R Künzel, Jasjeet S Sekhon, Peter J Bickel, and Bin Yu. Metalearners for estimating heterogeneous treatment effects using machine learning. *Proceedings of the national academy of sciences*, 116(10):4156–4165, 2019.

[107] Joseph DY Kang and Joseph L Schafer. Demystifying double robustness: A comparison of alternative strategies for estimating a population mean from incomplete data. *Statistical science*, 22(4):523–539, 2007.

[108] Robert J LaLonde. Evaluating the econometric evaluations of training programs with experimental data. *The American economic review*, pages 604–620, 1986.

[109] Matthieu Komorowski, Leo A Celi, Omar Badawi, Anthony C Gordon, and A Aldo Faisal. The artificial intelligence clinician learns optimal treatment strategies for sepsis in intensive care. *Nature medicine*, 24(11):1716–1720, 2018.

[110] Roderick JA Little and Donald B Rubin. *Statistical analysis with missing data*, volume 793. John Wiley & Sons, 2019.

[111] Richard J Cook, Leilei Zeng, and Grace Y Yi. Marginal analysis of incomplete longitudinal binary data: a cautionary note on locf imputation. *Biometrics*, 60(3):820–828, 2004.

[112] Stef Van Buuren, Jaap PL Brand, Catharina GM Groothuis-Oudshoorn, and Donald B Rubin. Fully conditional specification in multivariate imputation. *Journal of statistical computation and simulation*, 76(12):1049–1064, 2006.

[113] Stef Van Buuren and Karin Groothuis-Oudshoorn. mice: Multivariate imputation by chained equations in r. *Journal of statistical software*, 45:1–67, 2011.

[114] Seongwook Yoon and Sanghoon Sull. Gamin: Generative adversarial multiple imputation network for highly missing data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8456–8464, 2020.

[115] Alfredo Nazabal, Pablo M Olmos, Zoubin Ghahramani, and Isabel Valera. Handling incomplete heterogeneous data using vaes. *Pattern Recognition*, 107:107501, 2020.

[116] Steven Cheng-Xian Li, Bo Jiang, and Benjamin Marlin. Misgan: Learning from incomplete data with generative adversarial networks. *arXiv preprint arXiv:1902.09599*, 2019.

[117] Sahra Ghalebikesabi, Rob Cornish, Chris Holmes, and Luke Kelly. Deep generative missingness pattern-set mixture models. In *International Conference on Artificial Intelligence and Statistics*, page 3727–3735. PMLR, 2021.

[118] Pierre-Alexandre Mattei and Jes Frellsen. Miwae: Deep generative modelling and imputation of incomplete data sets. In *Proceedings of the 36th International Conference on Machine Learning*, pages 4413–4423. PMLR 97, 2019.

[119] N. B. Ipsen, P-A. Mattei, and J. Frellsen. not-miwae: Deep generative modelling with missing not at random data. In *Proceedings of 9 International Conference on Learning Representations*, 2021.

[120] Chao Ma and Cheng Zhang. Identifiable generative models for missing not at random data imputation. In *Advances in Neural Information Processing Systems 34*, pages 27645–27658, 2021.

[121] Xingyu Zhou, Yuling Jiao, Jin Liu, and Jian Huang. A deep generative approach to conditional sampling. *Journal of the American Statistical Association*, pages 1–12, 2022.

[122] Syed Mumtaz Ali and Samuel D Silvey. A general class of coefficients of divergence of one distribution from another. *Journal of the Royal Statistical Society: Series B (Methodological)*, 28(1):131–142, 1966.

[123] Donald B. Rubin. Multiple imputation for nonresponse in surveys. *Wiley*, 81(394):543–603, 1987.

[124] Arthur Asuncion and David Newman. Uci machine learning repository, 2007.

[125] Yang Sun and Yi-Hui Zhou. A machine learning pipeline for mortality prediction in the icu. *International Journal of Digital Health*, 2(1), 2022.

[126] Kelwin Fernandes, Pedro Vinagre, and Paulo Cortez. A proactive intelligent decision support system for predicting the popularity of online news. In *Portuguese conference on artificial intelligence*, pages 535–546. Springer, 2015.

APPENDIX

APPENDIX

A

APPENDIX

A.1 Appendix for Chapter 1

A.1.1 Additional Figures

Figure A.1: Missing Proportions of the 36 Common Laboratory Tests

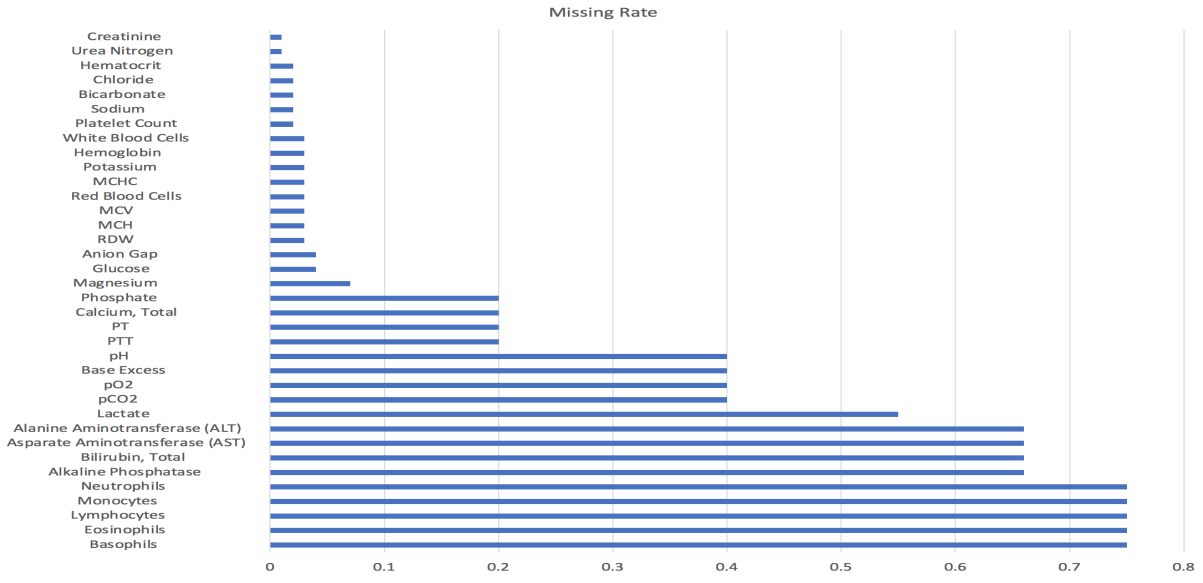


Figure A.2: Top 30 features selected by LightGBM to predict ICU mortality rate.

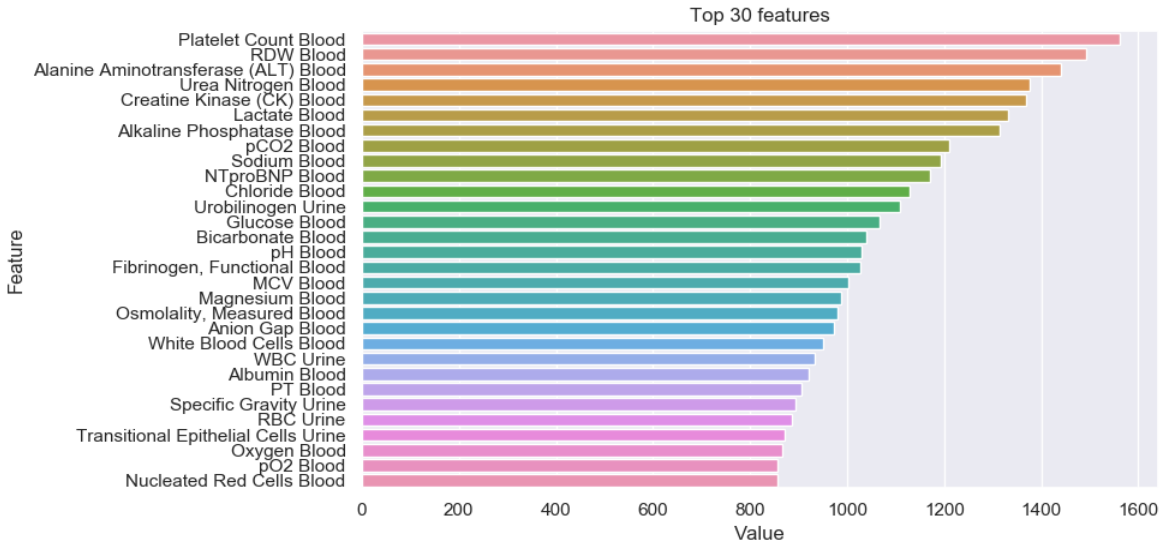


Figure A.3: Feature Engineering Pipeline using eICU database.

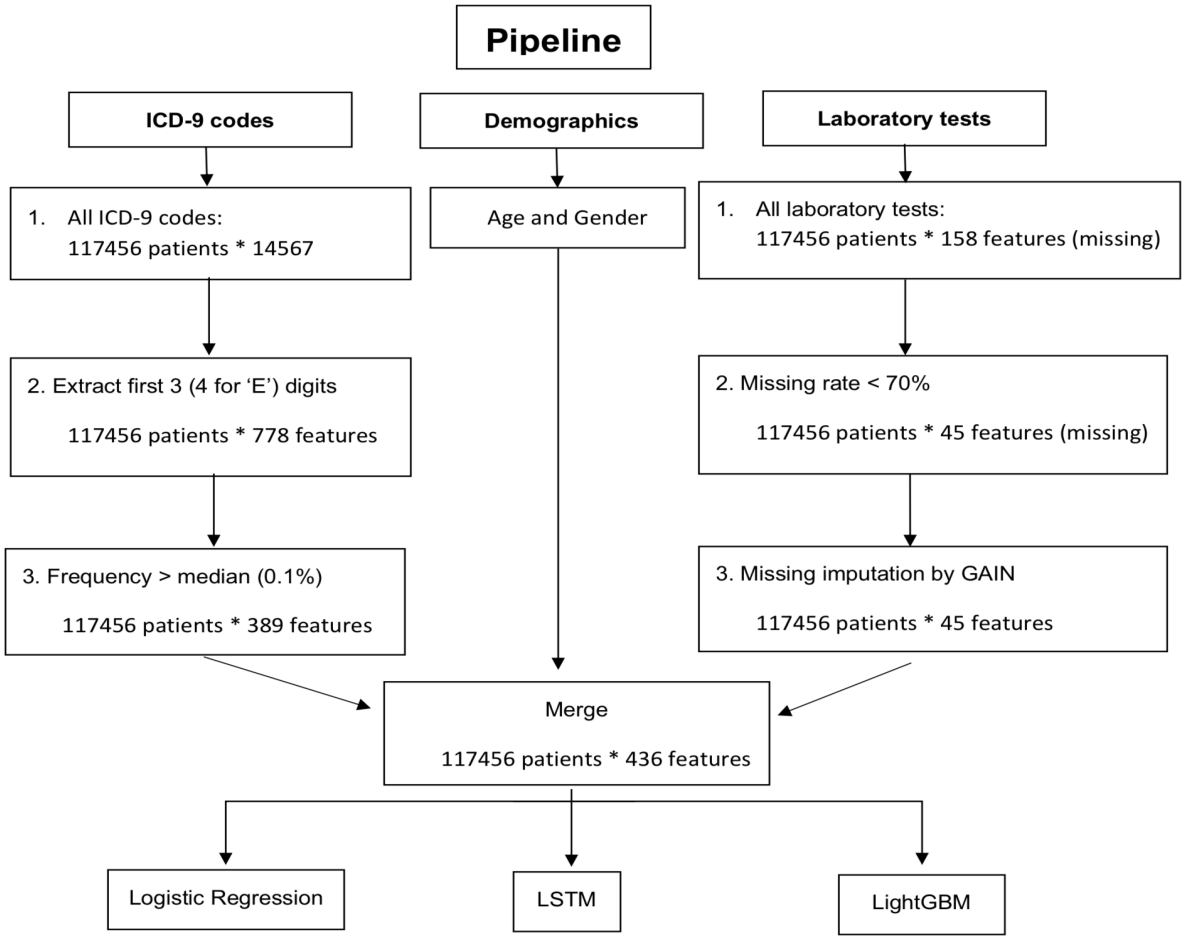
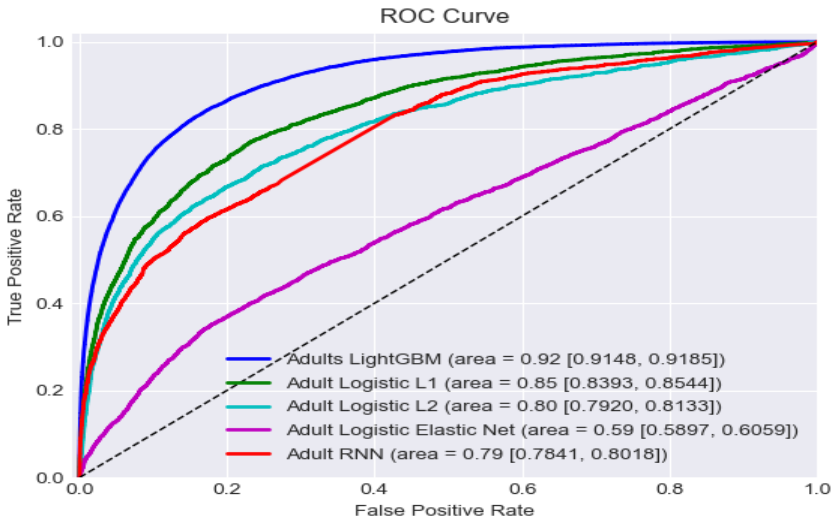


Figure A.4: Area under receiver operating characteristic curve (AUROC) for 5 candidate algorithms using eICU database.



A.1.2 Additional tables

Table A.1: Total Admission for Top 10 ICD-9 codes and top 10 ICD-9 categories

ICD-9 Code	Admissions (Coverage)
4019: Hypertension	20046 (38.02%)
4280: Congestive heart failure	12842 (24.36%)
42731: Atrial fibrillation	12589 (23.88%)
41401: Coronary atherosclerosis	12178 (23.10%)
5849: Acute kidney failure	8906 (16.89%)
25000: Diabetes Type II	8783 (16.66%)
2724: Hyperlipidemia	8503 (16.13%)
51881: Acute respiratory failure	7249 (13.75%)
5990: Urinary tract infection	6442 (12.22%)
53081: Esophageal reflux	6154 (11.67%)
top-10-code	40562 (76.93%)
top-50-code	49354 (93.60%)

ICD-9 Category	Admissions (Coverage)
401: Essential hypertension	20646 (39.16%)
427: Cardiac dysrhythmias	16774 (31.81%)
276: Disorders of fluid electrolyte	14712 (27.90%)
272: Disorders of lipoid metabolism	14212 (26.95%)
414: Other chronic ischemic heart disease	14081 (26.71%)
250: Diabetes mellitus	13818 (26.21%)
428: Heart failure	13330 (25.28%)
518: Other diseases of lung	12997 (24.65%)
285: Other and unspecified anemias	12404 (23.53%)
584: Acute kidney failure	11147 (21.14%)
top-10-cat	44419 (84.24%)
top-50-cat	51034 (96.79%)

A.2 Appendix for Chapter 3

A.2.1 Theorem

Theorem 1. Let $\Phi : X \to R$ be a one-to-one invertible representation function, and let p_Φ be the distribution induced by Φ over R, i.e. $p_\Phi(r|t=1)$ and $p_\Phi(r|t=0)$ are the covariate distributions under treatment and control induced over R. We then have for any outcome prediction function $H : R \times \{0,1\} \to Y$:

$$L_{PEHE}(H,\Phi) \leq 2\left(L_{F|z=0}(H,\Phi) + L_{F|z=1}(H,\Phi) + d_D(\Phi) \cdot \sum_{x \in X} \ell^{max}_{H,\Phi}(x) - 2\sigma^2_Y \right)$$

Proof:

By theorem 1 of (91), the following inequality holds:

$$L_{PEHE}(H,\Phi) \leq 2\big(L_C(H,\Phi) + L_F(H,\Phi) - 2\sigma^2_Y\big)$$

With above definitions and assumptions, we can prove:

$$L_C(H,\Phi) - \big[p_0 \cdot L_{F|z=1}(H,\Phi) + p_1 \cdot L_{F|z=0}(H,\Phi)\big] =$$

$$\big[p_0 \cdot L_{CF|z=1}(H,\Phi) + p_1 \cdot L_{CF|z=0}(H,\Phi)\big] - \big[p_0 \cdot L_{F|z=1}(H,\Phi) + p_1 \cdot L_{F|z=0}(H,\Phi)\big] =$$

$$p_0 \cdot \big[L_{CF|z=1}(H,\Phi) - L_{F|z=1}(H,\Phi)\big] + p_1 \cdot \big[L_{CF|z=0}(H,\Phi) - L_{F|z=0}(H,\Phi)\big] =$$

$$p_0 \sum_{x \in X} \big(\ell_{H,\Phi|z=1}(x)\big(p(x|z=0) - p(x|z=1)\big) + p_1 \sum_{x \in X} \ell_{H,\Phi|z=0}(x)\big(p(x|z=1) - p(x|z=0)\big) =$$

$$p_0 \sum_{r \in R} \big(\ell_{H,\Phi|z=1}(\Phi^{-1}(r))\big(p_\Phi(r|z=0) - p_\Phi(r|z=1)\big)\big)$$

$$+ p_1 \sum_{r \in R} \ell_{H,\Phi|z=0}(\Phi^{-1}(r))\big(p_\Phi(r|z=1) - p_\Phi(r|z=0)\big) \leq$$

$$p_0 \sum_{r \in R} \big(\ell_{H,\Phi|z=1}(\Phi^{-1}(r)) \cdot d_D(\Phi) + p_1 \sum_{r \in R} \ell_{H,\Phi|z=0}(\Phi^{-1}(r)) \cdot d_D(\Phi) \leq$$

$$p_0 \sum_{r \in R} \ell^{max}_{H,\Phi}(\Phi^{-1}(r)) \cdot d_D(\Phi) + p_1 \sum_{r \in R} \ell^{max}_{H,\Phi}(\Phi^{-1}(r)) \cdot d_D(\Phi) =$$

$$p_0 \sum_{x \in X} \ell^{max}_{H,\Phi}(x) \cdot d_D(\Phi) + p_1 \sum_{x \in X} \ell^{max}_{H,\Phi}(x) \cdot d_D(\Phi) =$$

$$d_D(\Phi) \cdot \sum_{x \in X} \ell^{max}_{H,\Phi}(x)$$

i.e. $L_C(H,\Phi) \leq p_0 \cdot L_{F|z=1}(H,\Phi) + p_1 \cdot L_{F|z=0}(H,\Phi) + d_D(\Phi) \cdot \sum_{x \in X} \ell^{max}_{H,\Phi}(x)$

Therefore,

$$
\begin{aligned}
&L_{PEHE}(H,\Phi) \leq 2\big(L_C(H,\Phi)+L_F(H,\Phi)-2\sigma_Y^2\big)\leq \\
&2\left[\left(p_0\cdot L_{F|z=1}(H,\Phi)+p_1\cdot L_{F|z=0}(H,\Phi)+d_D(\Phi)\cdot\sum_{x\in X}\ell_{H,\Phi}^{max}(x)\right)\right] \qquad \text{(A.1)}\\
&+2\big[\big(p_0\cdot L_{F|z=0}(H,\Phi)+p_1\cdot L_{F|z=1}(H,\Phi)\big)-2\sigma_Y^2\big]\leq \\
&2\left(L_{F|z=0}(H,\Phi)+L_{F|z=1}(H,\Phi)+d_D(\Phi)\cdot\sum_{x\in X}\ell_{H,\Phi}^{max}(x)-2\sigma_Y^2\right) \quad \square \qquad \text{(A.2)}
\end{aligned}
$$

A.2.2 Model Configuration

Our models were configured with varying numbers of representation layers (1, 2, or 3 layers) responsible for feature extraction, and hypothesis layers (1, 2, or 3 layers) involved in generating predictions. We explored different dimensions for both representation layers (20, 50, 100, or 200 units per layer) and hypothesis layers (20, 50, 100, or 200 units per layer), impacting model complexity and prediction expressiveness. In the training process, we employed diverse batch sizes (100, 200, 500, or 700 samples per batch) affecting training efficiency and algorithm stability. These parameter settings and architectural choices were essential components of our experimental framework.

Tables

A.3 Appendix for Chapter 4

A.3.1 UCI Machine Learning Datasets

The UCI Machine Learning Datasets, hosted by the University of California, Irvine, represent a vital resource for the machine learning community. The UCI Machine Learning Datasets have become a cornerstone for benchmarking and validating imputation algorithms, facilitating advancements in the field of missing data imputation. To demonstrate the effectiveness of our approach, we conduct real-data experiments on benchmark datasets. This empirical validation allows us to compare the performance of our proposed GCMI method with other state-of-the-art missing data imputation techniques.

Credit. The Credit Card Clients Dataset records customers demographics (e.g. gender,

Table A.2: Baseline characteristics table of the sepsis patients included in the MIMIC-III database. PT: Prothrombin Time; PTT: Partial Thromboplastin Time; SIRS: Systemic Inflammatory Response Syndrome; Shock index: systolic blood pressure/heart rate.

Mechvent	No(16015)	Yes (4210)	All (20225)
Gender	0.44	0.42	0.44
Age	65.17	63.04	64.73
Elixhauser score	3.89	4.12	3.94
Readmission to intensive care	0.34	0.29	0.33
Weight	75.40	79.16	76.19
SOFA	3.92	5.92	4.34
SIRS	0.95	1.14	0.99
Glasgow coma scale	11.26	7.96	10.57
Heart rate	77.91	79.21	78.18
Systolic	106.12	103.67	105.61
Mean blood pressure	69.02	68.92	69.00
Diastolic blood pressure	47.86	47.71	47.83
Shock Index	0.64	0.65	0.64
Respiratory rate	16.72	17.03	16.78
SpO2	94.69	95.11	94.78
Temperature	97.23	97.76	97.34
Potassium	3.86	3.80	3.84
Sodium	136.62	137.70	136.85
Chloride	102.21	103.26	102.43
Glucose	112.08	114.82	112.65
BUN	25.03	30.25	26.12
Creatinine	1.35	1.30	1.34
Magnesium	1.88	1.94	1.89
Calcium	8.04	7.95	8.02
Ionised calcium	1.07	1.07	1.07
CO2	23.22	24.38	23.46
SGOT	55.87	100.59	65.18
SGPT	49.75	79.65	55.97
Total bilirubin	1.20	1.56	1.28
Albumin	2.67	2.49	2.64
Hemoglobin	9.89	9.59	9.83
WBC count	10.40	11.96	10.72
Platelets count	212.25	215.37	212.90
PTT	31.62	32.57	31.82
PT	14.90	14.90	14.90
INR	1.36	1.35	1.36
Arterial_pH	7.33	7.34	7.33
paO2	83.52	87.07	84.26
paCO2	35.25	36.71	35.55
Arterial_BE	-3.17	-2.34	-3.00
Arterial_lactate	1.29	1.39	1.31
HCO3	22.21	21.81	22.13
PaO2_FiO2	201.55	183.22	197.74
Maximum dose of vasopressor over 4h	0.01	0.06	0.02
Current IV fluid intake over 4h	40.16	139.89	60.92
Urine output over 4h	73.95	152.92	90.39
Cumulated fluid balance since admission	573.02	1318.48	728.19

Table A.3: Credit Dataset - Missing Imputation: Simulation of Blockwise Missing Data with $n = 5000$ and $p = 23$ under MCAR. The missing rates of the features range from 10% to 60%. RMSE values are presented as mean ± standard deviation.

Missing Rate	0.1	0.2	0.3	0.4	0.5	0.6
Mean Imputation	0.243 ± 0.002	0.242 ± 0.002	0.243 ± 0.002	0.242 ± 0.002	0.243 ± 0.003	0.243 ± 0.005
Multiple Imputation	0.120 ± 0.002	0.141 ± 0.002	0.161 ± 0.002	0.177 ± 0.003	0.194 ± 0.003	0.210 ± 0.005
MI-KNN	0.136 ± 0.001	0.156 ± 0.003	0.175 ± 0.002	0.190 ± 0.002	0.206 ± 0.003	0.220 ± 0.004
MissForest	0.113 ± 0.002	0.157 ± 0.002	0.186 ± 0.002	0.213 ± 0.002	0.233 ± 0.003	0.255 ± 0.004
Soft	0.111 ± 0.003	0.137 ± 0.004	0.161 ± 0.006	0.186 ± 0.006	0.217 ± 0.003	0.246 ± 0.006
Soft_Mask	0.286 ± 0.015	0.478 ± 0.010	0.369 ± 0.003	0.303 ± 0.004	0.260 ± 0.004	0.249 ± 0.005
GAIN	0.136 ± 0.009	0.148 ± 0.007	0.170 ± 0.008	0.191 ± 0.006	0.217 ± 0.005	0.247 ± 0.006
GCMI	**0.105 ± 0.003**	**0.129 ± 0.003**	**0.153 ± 0.004**	**0.171 ± 0.003**	**0.188 ± 0.003**	**0.203 ± 0.006**

Table A.4: Letter Dataset - Missing Imputation: Simulation of Blockwise Missing Data with $n = 5000$ and $p = 16$ under MAR. The missing rates of the features range from 10% to 60%. RMSE values are presented as mean ± standard deviation.

Missing Rate	0.1	0.2	0.3	0.4	0.5	0.6
Mean Imputation	0.164 ± 0.000	0.163 ± 0.001	0.162 ± 0.001	0.164 ± 0.001	0.163 ± 0.001	0.163 ± 0.002
Multiple Imputation	0.127 ± 0.000	0.130 ± 0.001	0.131 ± 0.001	0.140 ± 0.001	0.144 ± 0.001	0.150 ± 0.002
MI-KNN	0.137 ± 0.000	0.137 ± 0.001	0.142 ± 0.001	0.145 ± 0.001	0.150 ± 0.001	0.153 ± 0.002
MissForest	**0.080 ± 0.001**	**0.085 ± 0.001**	0.106 ± 0.001	0.118 ± 0.002	0.127 ± 0.002	0.149 ± 0.004
Soft	0.131 ± 0.000	0.133 ± 0.001	0.137 ± 0.001	0.151 ± 0.001	0.155 ± 0.001	0.178 ± 0.003
Soft_Mask	0.643 ± 0.001	0.285 ± 0.002	0.296 ± 0.001	0.221 ± 0.001	0.180 ± 0.001	0.185 ± 0.002
GAIN	0.138 ± 0.000	0.135 ± 0.001	0.139 ± 0.001	0.140 ± 0.001	0.145 ± 0.001	0.215 ± 0.002
GCMI	0.086 ± 0.000	0.092 ± 0.001	**0.102 ± 0.001**	**0.115 ± 0.001**	**0.126 ± 0.001**	**0.146 ± 0.003**

education, age) and credit card payment history (e.g. payment status, payment delay, amount of bill statement) to predict whether a customer produces default-related user behavior in the next month. The dataset contains 30,000 samples with 23 features.

Letter. The Letter Recognition Database consists of 20000 total exemplars of the 26 English alphabet letters. The letter images are generated from 20 different fonts, and the objective of this dataset is to classify letter categories based on 16 integer variables (e.g. statistical moments and edge counts) extracted from images of the letters.

News. The online news popularity dataset consists of 39797 observations and 61 features derived from online news articles published from Mashable news service from January 2013 until January 2015 (126). The purpose of this dataset is to determine the popularity of the news based on number of shares. The dataset also collects articles' information such as average word length, the number of positive words, the number of images, number of keywords, etc.

Table A.5: News Dataset - Missing Imputation: Simulation of Blockwise Missing Data with $n = 5000$ and $p = 58$ under MAR. The missing rates of the features range from 10% to 60%. RMSE values are presented as mean ± standard deviation.

Missing Rate	0.1	0.2	0.3	0.4	0.5	0.6
Mean Imputation	0.227 ± 0.000	0.230 ± 0.001	0.230 ± 0.001	0.232 ± 0.001	0.230 ± 0.001	0.231 ± 0.001
Multiple Imputation	0.140 ± 0.000	0.157 ± 0.001	0.174 ± 0.001	0.194 ± 0.001	0.210 ± 0.001	0.224 ± 0.001
MI-KNN	0.204 ± 0.001	0.215 ± 0.001	0.216 ± 0.001	0.229 ± 0.001	0.234 ± 0.001	0.239 ± 0.001
MissForest	0.125 ± 0.002	0.156 ± 0.001	0.179 ± 0.001	0.199 ± 0.001	0.215 ± 0.001	0.223 ± 0.003
Soft	0.200 ± 0.000	0.209 ± 0.001	0.216 ± 0.001	0.229 ± 0.001	0.236 ± 0.001	0.245 ± 0.001
Soft_Mask	0.213 ± 0.010	0.230 ± 0.003	0.233 ± 0.003	0.236 ± 0.003	0.234 ± 0.004	0.235 ± 0.002
GAIN	0.213 ± 0.004	0.213 ± 0.004	0.219 ± 0.005	0.220 ± 0.005	0.238 ± 0.003	0.271 ± 0.003
GCMI	$\mathbf{0.121 \pm 0.001}$	$\mathbf{0.146 \pm 0.001}$	$\mathbf{0.171 \pm 0.001}$	$\mathbf{0.184 \pm 0.001}$	$\mathbf{0.202 \pm 0.001}$	$\mathbf{0.215 \pm 0.001}$