Synthetic Data Sharing and Estimation of Viable Dynamic Treatment Regimes with Observational Data

by

Nina Zhou

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Biostatistics)
in The University of Michigan
2021

Doctoral Committee:

Professor Ivo D. Dinov, Co-Chair
Professor Lu Wang, Co-Chair
Research Associate Professor Daniel Almirall
Assistant Professor Zhenke Wu
Professor Chuanwu Xi

Nina Zhou

zhounina@umich.edu

ORCID iD: 0000-0002-1649-3458

ACKNOWLEDGEMENTS

Firstly, I would like to express my sincere gratitude to my advisors Dr. Lu Wang and Dr. Ivo Dinov for their continuous support of my Ph.D. study and related research, for their patience, motivation, and immense knowledge. Their guidance helped me in all the time of research and writing of this thesis. Besides my advisors, I am grateful to the the rest of my thesis committee: Dr. Zhenke Wu, Dr. Daniel Almirall and Dr.Chuanwu Xi for their insightful comments and encouragements, but also for the hard question which incented me to widen my research from various perspectives.

I would like to thank my fellow doctoral students for their feedback, cooperation and of course friendship. In addition, I would like to express my gratitude to Dr. Kristen Herold for proofreading all my writings.

Last but not the least, I would like to thank my family: my parents and grand parents for supporting me spiritually throughout writing this thesis and my life in general.

TABLE OF CONTENTS

LIST OF FIGURES

Figure

LIST OF TABLES

Table

LIST OF ALGORITHMS

Algorithm

ABSTRACT

Significant public demand arises for rapid data-driven scientific investigations using observational data, especially in personalized healthcare. This dissertation addresses three complementary challenges of analyzing complex observational data in biomedical research.

The ethical challenge reflects regulatory policies and social norms regarding data privacy, which tend to emphasize data security at the expense of effective data sharing. This results in fragmentation and scarcity of available research data. In Chapter II, we propose the DataSifter approach that mediates this challenge by facilitating the generation of realistic synthetic data from sensitive datasets containing static and time-varying variables. The DataSifter method relies on robust imputation methods, including missForest and an iterative imputation technique for time-varying variables using the Generalized Linear Mixed Model (GLMM) and the Random Effects-Expectation Maximization tree (RE-EM tree). Applications demonstrate that under a moderate level of obfuscation, the DataSifter guarantees sufficient per subject perturbations of time-invariant data and preserves the joint distribution and the energy of the entire data archive, which ensures high utility and analytical value of the time-varying information. This promotes accelerated innovation by enabling secure sharing among data governors and researchers.

Once sensitive data can be securely shared, effective analytical tools are needed to provide viable individualized data-driven solutions. Observational data is an important data source for estimating dynamic treatment regimes (DTR) that guide personalized treatment decisions. The second natural challenge regards the viabil-

ity of optimal DTR estimations, which may be affected by the observed treatment combinations that are not applicable for future patients due to clinical or economic reasons. In Chapter III, we develop restricted Tree-based Reinforcement Learning to accommodate restrictions on feasible treatment combinations in observational studies by truncating possible treatment options based on patient history in a multi-stage multi-treatment setting. The proposed new method provides optimal treatment recommendations for patients only regarding viable treatment options and utilizes all valid observations in the dataset to avoid selection bias and improve efficiency.

In addition to the structured data, unstructured data, such as free-text, or voice-note, have become an essential component in many biomedical studies based on clinical and health data rapidly, including electronic health records (EHR), providing extra patient information. The last two chapters in my dissertation (Chapter IV and Chapter V) expands the methods developed in the previous two projects by utilizing novel natural language processing (NLP) techniques to address the third challenge of handling unstructured data elements. In Chapter IV, we construct a text data anonymization tool, DataSifterText, which generates synthetic free-text data to protect sensitive unstructured data, such as personal health information. In Chapter V, we propose to enhance the precision of optimal DTR estimation by acquiring additional information contained in clinical notes with information extraction (IE) techniques. Simulation studies and application on blood pressure management in intensive care units demonstrated that the IE techniques can provide extra patient information and more accurate counterfactual outcome modeling, because of the potentially enhanced sample size and a wider pool of candidate tailoring variables for optimal DTR estimation.

The statistical methods presented in this thesis provides theoretical and practical solutions for privacy-aware utility-preserving large-scale data sharing and clinically meaningful optimal DTR estimation. The general theoretical formulation of the meth-

ods leads to the design tools and direct applications that are expected to go beyond the biomedical and health analytics domains.

CHAPTER I

Introduction

There is a significant public demand for rapid data-driven scientific investigations using observational data, especially in personalized healthcare that accounts for individual variability. Despite the fact that there are 3.3×10^{13} Gigabytes of available digital content in 2018, less than 5% of the data has been analyzed according to the International Data Corporation [87, 56]. While translating data into practical solutions, ethical and practical challenges arise. Specifically, in this dissertation, our work is motivated by three challenges of analyzing complex observational data in biomedical research.

The ethical challenge reflects regulatory policies and social norms regarding data privacy, which hinder efficient data sharing and result in the scarcity of available data in research. Existing strategies, including differentially private algorithms using graphical models [99, 12], are not scalable for high dimensional data and are incapable of handling time-varying data with correlation across time for each subject.

Once sensitive data can be securely shared, analysis tools are needed to provide viable individualized data-driven solutions. Personalized medicine tailors medical treatment to the individual characteristics of each patient. Rapid data growth in recent years has made observational data a major data source for personalized medicine. Dynamic treatment regime (DTR) is a pre-specified sequence of decision-rules that

guide personalized treatment decisions for patients. A common objective is to use data to estimate the "optimal DTR," which optimizes a desired clinical outcome. A natural challenge arises when some observed treatment combinations are not applicable for future patients due to clinical or economic reasons, which may lead to uninterpretable optimal DTR estimations. For example, the DTR may contain recalled drugs in a particular stage or has a weaker treatment following an aggressive treatment for nonresponders. In addition, methods do not exist to estimate the best DTR among predefined feasible treatment combinations. Finally, the traditional approach of deleting patient records involving inapplicable treatment combinations may lead to selection bias.

In addition to the structured data, unstructured data such as free-text, or voice note provides extra information and has become an essential component in many biomedical data sources, including electronic health records (EHR). The recent success in natural language processing techniques has generated much interest in applying unstructured data analysis methods to biomedical research. However, on the one hand, there is no existing method to enable safe and rapid sharing of the unstructured information. Although suppressing names, addresses, dates, and phone numbers is a common practice in anonymized clinical notes, personal identity information in EHR can be easily obtained from detailed descriptions in narrative reports. On the other hand, we notice a lack of research in estimating dynamic treatment regimes using additional free-text information extracted from clinical notes. Since EHR datasets are constructed for billing and patient care management, many clinically relevant covariates are evident in its narrative contents but are not available in the structured data.

To address these challenges, in Chapter II, we propose the DataSifter algorithm to create synthetic data from sensitive datasets containing static and time-varying variables, which accelerates innovation by enabling secure synthetic data sharing among

data governors, researchers, and trainees. The DataSifter method relies on robust imputation methods, including missForest and an iterative imputation technique for time-varying variables using the Generalized Linear Mixed Model (GLMM) and the Random Effects-Expectation Maximization tree (RE-EM tree). Applications based on simulated and real clinical data demonstrate the balance between the preservation of the data utility (analytical value) and the reduction of re-identification risk (privacy) associated with sharing obfuscated data. Our extensive simulation shows that under a moderate level of obfuscation, in addition to guaranteeing per subject obfuscation of time-invariant data, DataSifter also protects sensitive information in the time-varying records without a substantial impact on the analytical value of the sifted dataset.

To solve one of the technical challenges and provide viable treatment regimes, in Chapter III, we develop the Restricted Tree-based Reinforcement Learning to accommodate restrictions on feasible treatment combinations in observational studies by truncating possible treatment options based on patient history in a multi-stage multi-treatment setting. Such constrained optimization procedure is conducted backward from the last treatment stage so that patients in the restricted arms can still contribute to some of the stage-wise optimal regime estimations when their history up to that stage is considered feasible. Our algorithm provides optimal treatment recommendations for patients regarding viable treatment options and utilizes all valid observations in the dataset to avoid selection bias and improve efficiency.

With extra information in unstructured EHR data, we can refine the previous solutions. Chapter IV and Chapter V elevate the previous two chapters by utilizing novel techniques in natural language processing. In Chapter IV, we construct a free-text data anonymization tool – DataSifterText, which offers synthetic free-text data that protects patients' Protected Health Information (PHI). According to our clinical data applications, the proposed technique protects the distribution of the original

text corpus, offers individual level data obfuscation, and enables collaborative data analytics without compromising personally identifiable information. The DataSifter-Text algorithm provides sufficient privacy protection by disguising the location of true and obfuscated tokens. In Chapter V, we enhance the precision of optimal DTR estimation with extra information in clinical notes. Simulation studies and application on blood pressure management in intensive care units demonstrated that utilizing information extraction (IE) techniques in DTR estimations enables clinical decision support for larger study populations, provides more accurate counterfactual outcome modeling, and supports a wider pool of candidate tailoring variables.

The main contributions of this dissertation include (1) the creation and implementation of the partially synthetic data publishing tool DataSifter, which handles static and time-varying data (Chapter II). (2) The development of the Restricted Tree-based Reinforcement Learning for accommodating restrictions on feasible treatment combinations in observational studies when estimating optimal DTRs (Chapter III). (3) The extension of the DataSifter algorithm to unstructured data obfuscation (Chapter IV), and (4) the application of IE techniques in optimal DTR estimations (Chapter V). The proposed methods enable collaborations to expedite bench-to-bedside translational research and improve the precision of healthcare management. The overall structure of the dissertation is illustrated in **Figure** 1.1.

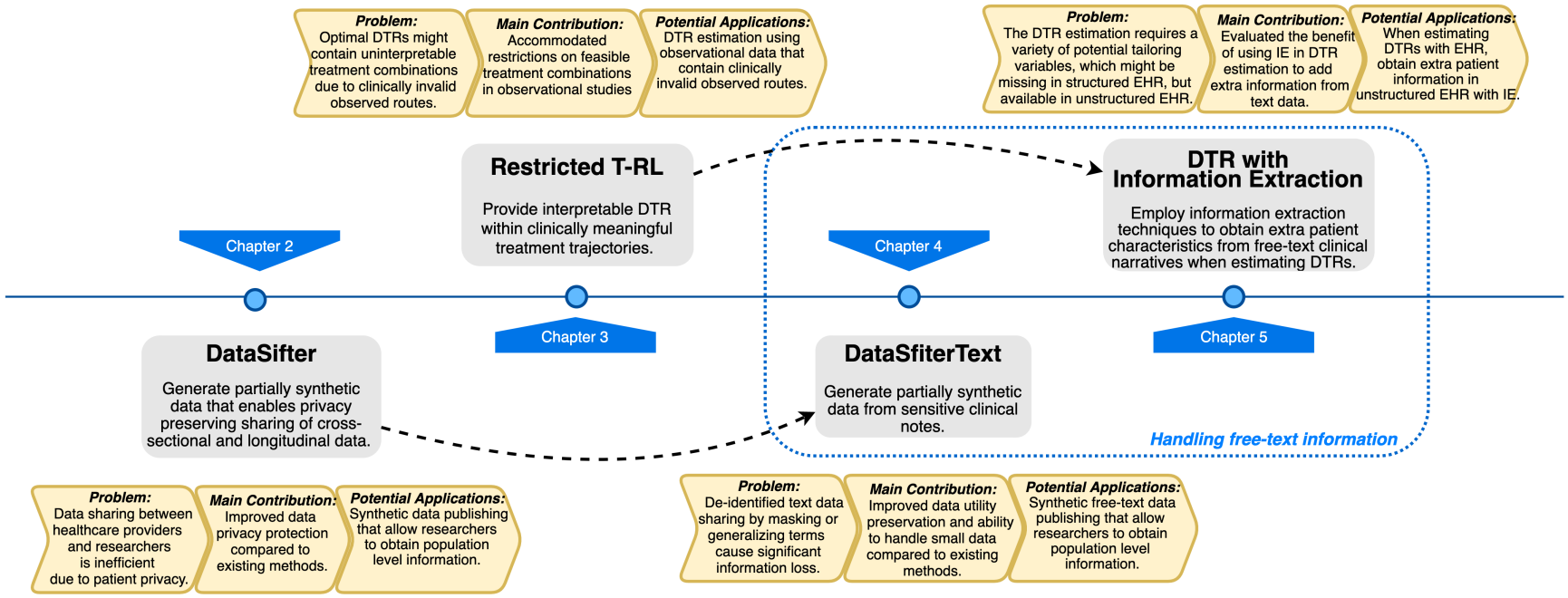

Figure 1.1: Dissertation Flowchart

CHAPTER II

DataSifter: Statistical Obfuscation of Sensitive Datasets

2.1 Introduction

Open science advocates the sharing of data, methods, source-code, end-to-end protocols, computational services, and peer-reviews. The benefits of open science on many aspects of common human experiences are well documented [54, 86, 48, 8, 14]. Along with the well-known exponential increase of the amount of newly acquired data, there is also an equally sticking exponential decay of the value of data that is stored but not processed, shared, or augmented [19, 79]. However, sharing data without loss of privacy is difficult, especially in the medical and healthcare settings. In fact, 66% of the participants in the 2017 Health Information National Trends Survey were concerned about data privacy when health information is electronically exchanged [52].

For data sharing involving Protected Health Information (PHI), organizations' Institutional Review Boards (IRBs) need to review the research before the required information can be retrieved from existing medical records and processed to extract valuable information. IRB's initial review process may take up to 4 months, however this process has significant variability depending on the type of review, e.g., expedited,

exempt, or full board reviews may take additional time, from 16 to 631 days [30]. In the United States, healthcare systems own the property rights for Electronic Health Records (EHR), and researchers have to bear the costs of data extraction and transfer under data use agreements [7].

Such regulation guarantees the protection of individual privacy rights, but delays researchers' abilities to gain access to appropriate information, build models and rapidly validate scientific discovery, which slows the knowledge transfer and basic science translation into clinical practice. The resulting slow and restricted data access may limit data utility for answering specific scientific questions. For example, in 2015, Keegan et al. [37] examined the relationship between ethnicity and short-term breast cancer survival using 2010 Kaiser Permanente Northern California EHR data. However, to obtain both the demographics and the cancer treatments for patients across facilities, they had to reduce the study cohort time frame from 8 years to 3 years in order to have both datasets available and link them together, which significantly impacted the statistical power of this scientific investigation. Thus, there are enormous benefits in developing new statistical methods to facilitate secure and quick information exchange between data stewards and data science experts.

Three existing strategies provide secure mechanisms for modeling, processing, and interrogating sensitive cross-sectional data. These include secure enclave access, data encryption (e.g., fully homomorphic encryption), and synthetic data publishing. Secure data enclave environments [1, 32] offer a platform for researchers to analyze sensitive data without compromising risks for misuse, fraud, and other violations. Many health information storage solutions, e.g., EHRs, rely on technology that provides managed data access for research in safe and controlled environments [68]. A number of possible unmodified database management systems (DBMS) can be utilized to provide secure data enclaves [2, 5]. Second, data encryption methods, including Fully Homomorphic Encryption (FHE), encode the data to allow computations di-

rectly on the resulting ciphertext [35, 26, 96]. FHE relies on homomorphic computing (result-preserving property) on the ciphertext without exposing the sensitive raw data to independent researchers, analysts, or data scientists. The above two mechanisms provide secure channels for data transfer and storage, but do not shorten the data sharing process.

In response, the third strategy, synthetic data generation emerged, which was first proposed by Rubin 1993 [70] with two classes of generating methods, as summarized by Reiter and Raghunathan [59]. The fully synthetic data sets are created by conditional distributions estimated from sensitive datasets. Popular methods for constructing these conditional distributions include Bayesian network [99], graphical models [12] and multiple imputation (MI) [70, 55]. However, these solutions are not scalable for sensitive datasets with higher dimensions (larger number of records and variables), especially given that variable selection can be burdensome for parametric models and greedy searches for causal nodes in graphical models are computationally expensive with too many candidates. Partially synthetic data refers to a set of multiple-imputed data replacing sensitive data value cells with imputations [42, 57, 58]. This class of methods treats data obfuscation as a missing data handling problem, where they generate artificial missingness for sensitive values in the dataset and impute the value with the remaining untouched data. As a result, partially synthetic data provides valid statistical inference. However, combined information from a set of multiple imputed datasets indicates the locations of true and obfuscated cells resulting no privacy protection for the true cells. In practice, covering all possible sensitive values is barely achievable and selecting the obfuscation location is a subjective and critical step for data privacy protection.

To promote effective data sharing, we propose the DataSifter frame work, which is designed to help data governors safely publish synthetic subsets of their sensitive dataset or share deidentified data (with certain level of obfuscation) that enable spe-

cific types of data analysts. Under the proposed framework, the trade-off between data privacy and information utility determines the level of obfuscation associated with partially synthetic data generation. We define *data privacy* as the disclosure risk of specific data values given an intruder's prior knowledge and published synthetic data. Given a predefined inferential model and some specific clinical or research questions, *data utility* is the analytical value of the data measured by the deviation of the model inference based on the original and the partially synthetic datasets. Multiple imputation (MI) methods are designed to minimize the loss of data utility while the DataSifter framework focuses on maximizing the data privacy protection under acceptable data utility. In this chapter, we will evaluate the quality of partially synthetic datasets based on the above two criteria.

We developed the first generation of the DataSifter technique (DataSifter I) [45] that handles high dimensional cross-sectional data. Perturbing individual level records, it allows researchers to securely acquire population level information that closely approximates the true signal. The core DataSifter technique relies on two processes supporting the critical statistical obfuscation of the data. First, it randomly and artificially generates missingness in the data, following the Missing Completely At Random (MCAR) mechanism [69], and uses robust iterative imputation methods, e.g., missForest [80], to approximate the original information. Second, DataSifter I classifies neighboring cases (similar observations) using Euclidean and Gower distances for continuous and categorical variables, respectively. Within each neighborhood cluster, DataSifter I randomly swaps a subset of feature values between similar records. This second operation guarantees partial change for each record while preserving the geometrical information on the data in feature space.

However, time-varying correlated data cannot be processed by DataSifter I, while such data, including longitudinal data, are ubiquitous and provide important information for many biomedical and health conditions. For example, in EHR databases, pa-

tient characteristics and disease progression variables are collected repeatedly across visits. Maintaining or preserving the within-subject covariance structure among time-varying measurements presents another layer of challenges. Currently, there are no automated procedures that enable secure sharing of time-varying correlated data with sensitive information. The proposed new algorithm, *DataSifter with Time-varying Correlated Data (DataSifter II)*, extends the DataSifter functionality in the case of dealing with large, cross-sectional, time-varying, and self-correlated data elements. DataSifter II introduces artificial missingness and embed robust longitudinal imputation methods to handle high dimensional sensitive data with time-varying measures. As illustrated in **Figure** 2.1, our proposed procedure operates on the time-varying data separately from the time-invariant (cross-sectional) data elements and then integrates the two parts to compile the obfuscated *sifted* dataset (output). DataSifter II preserves the data utility while introducing a proper level of privacy protection. The newly developed DataSifter II R package is available on GitHub `https://github.com/SOCR/DataSifterII`. With the proposed algorithm, data governors can create and validate *sifted* data objects with time-varying components prior to their release or sharing, allowing user-defined secure level and protecting the original within-subject covariance structure.

The rest of the manuscript is organized as follows: in Section 2.2, we introduce the DataSifter I procedure, evaluate its performance under different simulation settings using simple metrics and apply the algorithm to ABIDE dataset. In Section 2.3.1 we formally define the data privacy and utility measurement for partially synthetic data evaluation. Specifically, in section 2.3.1.2, we define the disclosure risk and show that partially synthetic datasets generated by the DataSifter framework provide better privacy protection than that of the MI method. Section 2.3.2 describes the DataSifter II protocol. Section 2.3.3 validates the data utility preservation and privacy protection of the proposed algorithm under different simulation settings and compares

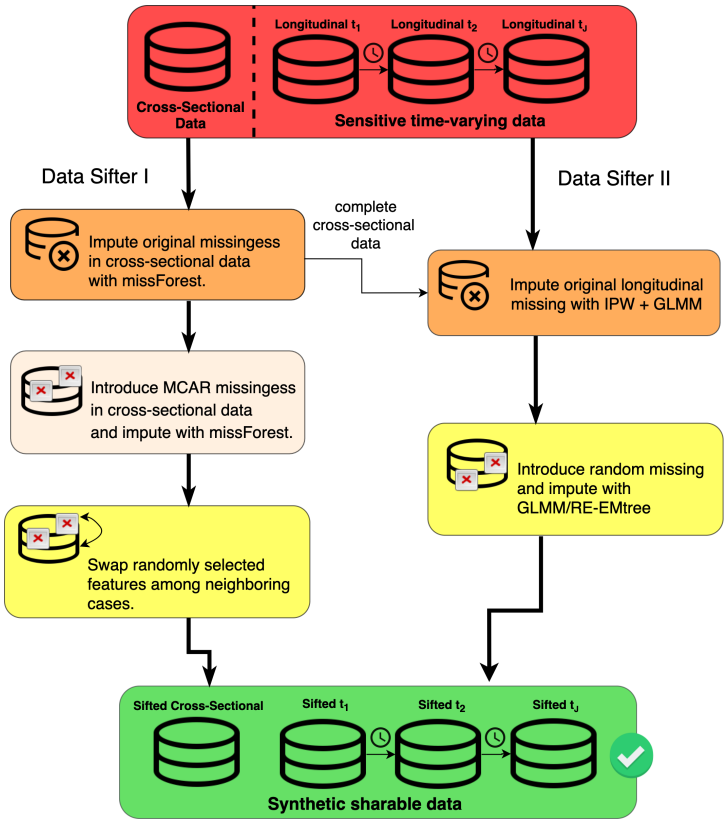

Figure 2.1: Graphical workflow depicting the organization of the DataSifter technique.

the performance against the MI method. In Section 2.3.4, we apply DataSifter II to the MIMIC-III clinical data and demonstrate its performance in maintaining a careful balance between protecting sensitive information and preservation of the data utility. We summarize the findings and discuss the expected impact and future developments in Section 2.4.

2.2 DataSifter I: Partially Synthetic Time-invariant Data Generation

2.2.1 DataSifter I Overview

The core of the DataSifter I is an iterative statistical computing approach that provides the data-governors controlled manipulation of the trade-off between sensitive information obfuscation and preservation of the joint distribution. The DataSifter I is designed to satisfy data requests from pilot study investigators focused on specific target populations. The DataSifter I handles high dimensional data with various of data elements including multiple numerical or categorical features and one unstructured variable. However, the obfuscation of the unstructured variable is simply done by reallocating the free-text documents in the swapping step. The content of the text is not being altered to protect privacy. A more sophisticated method dealing with unstructured data can be found in Chapter IV.

At each step, the algorithm generates instances of complete datasets that in aggregate closely resemble the intrinsic characteristics of the original cohort; however, at an individual level, the rows of data are substantially obfuscated. This procedure drastically reduces the risk for subject re-identification by stratification, as meta-data for all subjects is randomly and repeatedly encoded. Probabilistic (re)sampling, distance metrics and imputation methods play essential roles in the proposed DataSifter I obfuscation approach. In regard to the designed data requests, the main assump-

tions of the DataSifter I technique include: (A1) Incomplete observations are driven by missing at random (MAR) or missing completely at random (MCAR) mechanisms [41]; (A2) The utility of each feature is equally important; (A3) Large random samples of the original data preserves the overall joint distribution. These assumptions are standard and allow us to manage the data or quantify data utility. (A1) allows accurate imputations (A2) is essential in calculating subject-pair distances, and (A3) promotes subject-wise parallelization. We use the following framework to form the DataSifter I algorithm. Three sources of obfuscation have been applied to the data during the DataSifter I technique: (1) initial data imputation (in the preprocessing step), (2) artificially create and impute missingness (in the imputation step), and (3) swapping data values in the neighborhood (in the obfuscation step). Here we define all the mappings that has been employed for obfuscation.

2.2.2 Notation

Define $\mathcal{X}$ as the counterfactual complete sensitive dataset for sifting consisting m features and n cases. Let's use $1 \leq j \leq m$ to denote features and $1 \leq i \leq n$ to denote cases:

$$\mathcal{X}_j = (X_1, ..., X_j, ..., X_m) \in R^{n \times m}, X_j = (X_{1,j}, ..., X_{n,j})^T, 1 \leq j \leq m.$$

In the above expression, $X_{i,j}$ denotes the i^{th} subject's j^{th} feature value. We define the utility information embedded in a dataset as the knowledge about the joint distribution of the holistic data including all variables. By preservation of utility, we mean the relative conservation of the signal energy that suggests small deviation of the sifted-data joint distribution from the original (raw) data joint distribution. Clearly, this does not hold true for large obfuscation levels.

Missing data is pervasive in almost all real-world datasets. We define the hypo-

thetical complete j-th feature as:

$$X_j = (X_{obs,j}, X_{mis,j}),$$

where $X_{mis,j}$ denotes a vector containing the actual values of the missing data portion. What we observe is denoted as $\tilde{X}_j = (\tilde{X}_{obs,j}, N_j)$,here N_j represents the missing cells. The length of $\tilde{X}_{obs,j}$ is n_j and the length of N_j is $n - n_j$.

2.2.3 User-controlled Parameters

Sifting different data archives requires customized data management. Five specific parameters mediate this management:

k_0: A binary parameter indicating whether or not to obfuscate the unstructured feature, if any.

k_1: The percent of artificial missing data values that should be artificially introduced prior to imputation. Missingness is stochastically introduced to all data elements except the unstructured variable. The range of this parameter can be between 0 and 0.4. We set an upper bound of 40% missingness in order to keep the remaining dataset informative.

k_2: The number of times to repeat the introduction-of-missing-and-imputation step. Five options are available from 0 to 4.

k_3: The fraction of structured features to be obfuscated in all the cases. Available options can vary between 0 and 1.

k_4: The fraction of closest subjects to be considered as neighbours of a given subject. This implies that the top k4×100% of the closest-distance subjects of a given subject can be considered as candidates for its neighbours. Then, the final neighbouring status of any subject is determined by an additional hard cut off.

Table 2.1 illustrates the combinations of k_i parameters implemented in the al-

gorithm to accommodate the user defined balance between privacy protection and obfuscation. The level of obfuscation spans the range from raw data (no obfuscation) to synthetically simulated data (complete obfuscation). Our highest level of obfuscation, i.e. 'indep', refers to the synthetic dataset sample from the marginal empirical distributions of all the features.

Table 2.1: DataSifter I k parameter vector mapping determining the level of obfuscation.

Obfuscation level	k_0	k_1	k_2	k_3	k_4
None	0	0	0	0	0
Small	0	0.05	1	0.1	0.01
Medium	1	0.25	2	0.6	0.05
Large	1	0.4	5	0.8	0.2
Indep	Output synthetic data with independent features				

2.2.4 Preprocessing

The preprocessing steps of the original data might vary for different datasets. **Figure** 2.2 illustrates the procedures included in the DataSifter I preprocessing step. The overall goal is to delete uninformative features with a constant value or have too much missingness and impute missing values in the original data. To impute the missing values, we use a non-parametric imputation method missForest [80], albeit many alternative strategies are also possible. As an iterative non-parametric imputation method of mixed data types, missForest fits a random forest model for each feature separately during one iteration using the observed data as training data and provide predictions for the missing cells. Hence, the random forest model for imputing a specific column uses all other variables in the dataset as predictors. In each iteration, the imputation of the entire dataset starts in the column with least missing values and ends in the column with most missing values. It stops to iterate when the difference between the latest and prior imputed data matrix is at least as great as the previous difference measured or the maximal iteration limit achieves. The difference

between matrices in sets of continuous ($\mathbf{N}$) and categorical ($\mathbf{F}$) variables are defined as

$$\Delta_{\mathbf{N}} = \frac{\sum_{k \in \mathbf{N}} \|\hat{X}_k^{(r)} - \hat{X}_k^{(r-1)}\|_2}{\sum_{k \in \mathbf{N}} \|\hat{X}_k^{(r)}\|_2}$$

and

$$\Delta_{\mathbf{F}} = \frac{\sum_{k \in \mathbf{F}} \sum_{i=1}^{n} I(\hat{X}_{ik}^{(r)} \neq \hat{X}_{ik}^{(r-1)})}{\text{Number of missing cells in categorical variables}},$$

where $\hat{X}_k^{(r)}$ is the imputed vector and $\hat{X}_{ik}^{(r)}$ is the imputed value for subject i of the k^{th} variable in the r^{th} iteration. We choose missForest algorithm, rather than other model-based multiple imputation methods, for the following reasons: (1) missForest employs random forest imputation that can cope with complex EHR data, which typically involves mixed-type data, complex interactions, and non-linear relations; (2) it relies on limited modeling assumptions; and (3) it is relatively efficient for large-scale and high-dimensional data.

DataSifter I produces a complete dataset after the preprocessing step. For simplicity, we denote n as the number of subjects in the dataset, k as the number of informative features filtered by the preprocessing step, and M as the number of subjects per batch during the parallel process. Alternative preprocessing methods are possible as long as the aims are met.

2.2.5 Imputation Step

Following the data preprocessing, the DataSifter I continues with the imputation and the obfuscation steps. During the imputation step, the DataSifter I algorithm first introduces random artificial missing values to the complete dataset, which synthetically provides privacy protection. The artificial missingness obeys missing completely at random (MCAR) mechanism as the missingness is introduces stochastically for case and feature indices [41]. Assume we have n entries of data and denote $\mathbf{Y} = (\mathbf{Y}_{obs}, \mathbf{Y}_{mis})$ as the full data, where $\mathbf{Y}_{obs}$ represents the observed part and $\mathbf{Y}_{mis}$

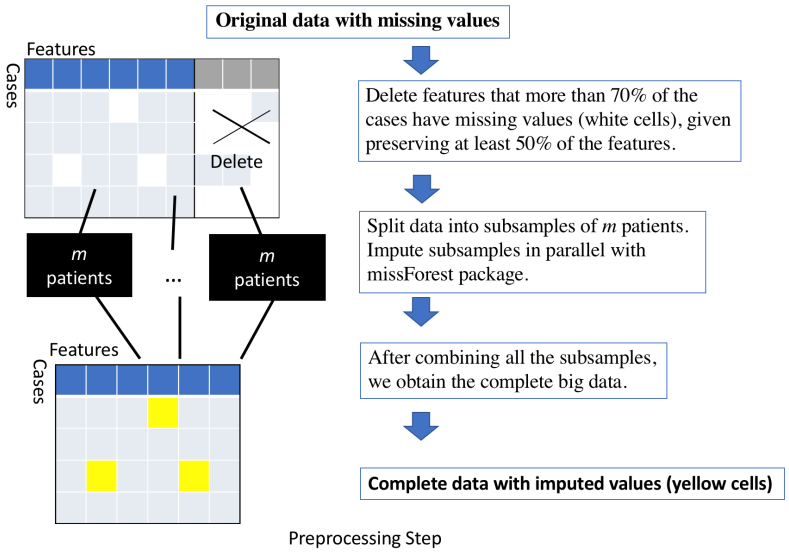

Figure 2.2: Flow Chart for Preprocessing Step.

the missing part. Let R denote the missing indicator with $R_i = I(Y_i \in \mathbf{Y}_{mis})$ for $i = 1, 2, .., n$. Because the MCAR assumption is satisfied, we have the following relationship: $P(\mathbf{R}|\mathbf{Y}) = P(\mathbf{R})$.

This relationship between observed and missing values guarantees that the fully observed data represents a random sample of the complete data. Accurate imputations of the missing values based on the observed values can be obtained with robust imputation methods. Thus, as described above, the introduction of missing data has limited effect in altering the joint distribution of the data during the imputation process. Similar to the preprocessing step, we use the missForest [80] to impute the artificial missingness with modified stopping criteria. Since the true missing value is known in this step, we define the stopping criterion for variable k under tolerance level ϵ as

$$\frac{\|X^*_{mis,k} - \hat{X}^{(r)}_{mis,k}\|_1}{\|X^*_{mis,k}\|_1} < \epsilon,$$

where $X^*_{mis,k}$ denotes the true values of the artificially missing cells in variable k and

$\hat{X}_{mis,k}^{(r)}$ denotes the imputed values at r^{th} iteration.

Following the imputation step, the outputted "sifted" dataset, X_{work}, has the following properties: (1) individual cases are manipulated, yet complete, protecting individual privacy, since hackers cannot distinguish "true" values from imputed values that are in the same format; (2) subjects with introduced missingness can still play an important role in the analysis after the imputation.

2.2.6 Obfuscation Step

During the obfuscation step, the DataSifter I repeatedly swaps the unstructured feature value and randomly selected structured feature values based on the closest neighbours to ensure a balance between data privacy and preservation of the feature distributions. The algorithm relies on distance metrics to determine neighbourhoods for all cases [24,25] and swaps feature values between closely adjacent neighbouring pairs. We compute pair-wise distances between all cases using a weighted distance measure: (1) Euclidean distances for normalized numerical features, and (2) Gower's distance for categorical features [28]. To obtain the distance matrix, we divide the current dataset outputted by the imputation step into three subsets and re-index the elements as numerical subset $X_{num} = (\vec{x}_1, \vec{x}_2, ..., \vec{x}_l) = (\hat{x}_1, \hat{x}_2, ..., \hat{x}_n)^T$, categorical dataset $X_{cat} = (\vec{y}_1, \vec{y}_2, ..., \vec{y}_{p-1-l}) = (\hat{y}_1, \hat{y}_2, ..., \hat{y}_n)^T$, and the unstructured feature X_{unstr}, where we have l numerical features, $p-1-l$ categorical features and one unstructured feature. For X_{num}, we apply a map algorithm f, which calculates the Euclidean distance for every pair of cases and maps the input data metric to the target distance metric. $f : R^{n\times l} \to R^{n\times n}$ is defined below. For $X = (x_1, x_2, ..., x_n)^T$, $f(X) = D_E = (e_{ij})$, where

$$e_{ij} = \begin{cases} \frac{\|x_i - x_j\|_2 - min_{i,j}\{\|x_i - x_j\|_2\}}{max_{i,j}\{\|x_i - x_j\|_2\} - min_{i,j}\{\|x_i - x_j\|_2\}} & , i < j \\ 0 & , otherwise \end{cases}$$

$\forall i, j$. We utilize f to obtain $f(X_{num}) = D_E = (e_{ij})_{n\times n}$. For the categorical subset, we define the mapping algorithm g which calculates the distance for categorical features via Gower's rule. For $X = (x_1, x_2, ..., x_n)^T$, $X \in R^{n\times p}, g(X) = D_G = (g_{ij})$. $\forall i, j$,

$$g_{i,j} = \sum_{s=1}^{p} \frac{g_{ijs}}{m}$$

Here, g_{ijs} is an indicator function related to the s^{th} feature, which is defined as,

$$g_{ijs} = \begin{cases} 0 & , x_{is} = x_{js} \\ 1 & , x_{is} \neq x_{js} \end{cases}.$$

Similarly, for $X_{cat} = (\vec{y}_1, \vec{y}_2, ..., \vec{y}_n)$, we attain $D_G = (g_{ij})_{n\times n} = g(X_{cat})$. Under assumption (A2), we weigh the distance and obtain the complete paired-distances metric,

$$D = (d_{ij})_{n\times n}, \forall i, j, d_{ij} = e_{ij} \times \frac{l}{p} + g_{ij} \times \frac{p-1-l}{p},$$

where l/p and $(p-1-l)/p$ represents the weights for the Euclidean and Gower distances, respectively. Two criteria are used to determine the neighboring status for subject pairs: (1) Closest $k_4 \times n$ neighbors regarding the pair distances; and (2) a hard cut off. In distance matrix D, for each i, we rank the paired distances d_{ij} as $\{d_{i1}, d_{i2}, ..., d_{in}\}$. Then, we find the maximum distance of the top $k_4 * 100\%$ $d_{i,floor(k_4\times n)}$, where $floor(k_4 \times n)$ rounds the number of cases to select to the lower integer. We use the cutoff to identify the potential neighbors of the i^{th} individual:

$$neighbor(i) = \{(i, j) : d_{ij} < d_{i,floor(k_4\times n)}\}, \forall i = 1, ..., n.$$

In addition, we set up a criterion to narrow the neighborhood. Let

$$c = inf\{d_{ij}\} + sd\{d_{ij}\}.$$

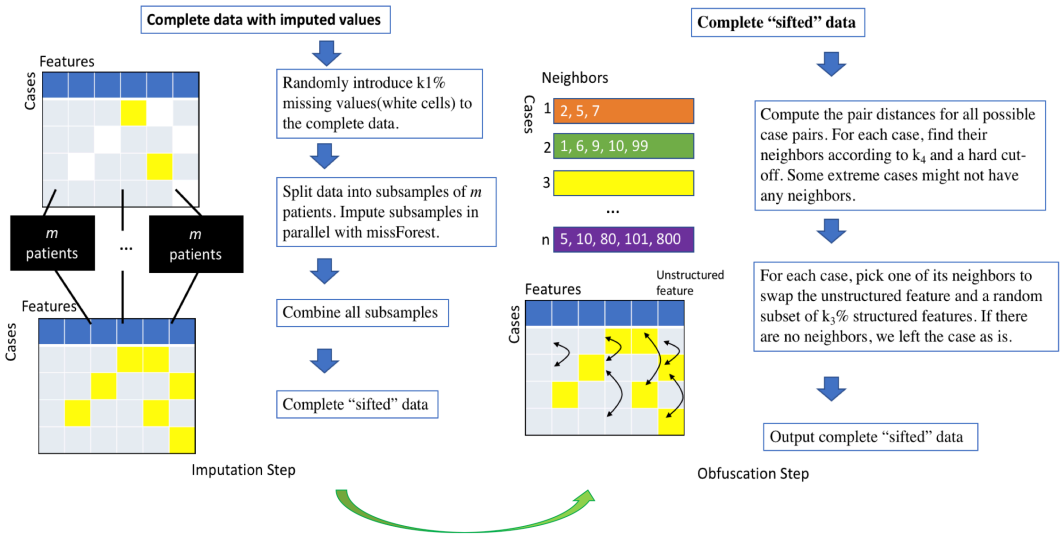

Figure 2.3: Flow Chart for Imputation and Obfuscation Steps.

Here, $sd\{d_{ij}\}$ refers to standard deviation of all the d_{ij}'s in D. We only preserve the neighbors that satisfy $d_{ij} \leq c$. The final set of neighbors, i.e. $neighbor_{final}$, is defined as follows:

$$neighbor_{final}(i) = \{(i,j)|(i,j) \in neighbor(i), d_{ij} \leq c\}, \forall i = 1, ..., n.$$

For extreme subjects that have no neighbors selected by the above process, we do not apply the obfuscation step. One subject could have multiple neighbors. For every subject, a neighboring subject is randomly selected as its swapping partner. We randomly swap a subset of randomly chosen features among each swapping pair. A detailed flow chart illustrating the imputation and obfuscation steps can be found in **Figure** 2.3.

2.2.7 Pseudo Code

In this section, we define X_{str} as the structured feature subset of current data, which consists of X_{num} and X_{cat}. Also, $Rand(X, r)$ is a function that randomly picks r elements without replacement in set X.

Input:

(1) The dataset after preprocessing $X_{work} = (\vec{x}_1, \vec{x}_2, ..., \vec{x}_p) \in R^{n \times p}$ with n cases and p features. There are one unstructured and $p-1$ structured features in the dataset. After the preprocessing step, p is less than or equal to the number of features in the original dataset. Each $\vec{x}_i$ is a column vector, $\vec{x}_i = (x_{1i}, x_{2i}, ..., x_{ni})^T$, $i = 1, ..., p$

(2) The categorical level of obfuscation L="none", "small", "medium", "large", "indep", or alternatively a specific parameter vector $(k_0, k_1, k_2, k_3, k_4)$.

Special cases:

If L ='none', the output is X_{work} and if L ='indep', the output is dented by X_{new}. Each feature in X_{new} is a synthetic sample from the empirical distribution of the corresponding feature in X_{work}.

Algorithm 1 DataSifter I

for i in $1 : k_2$ **do**
    Introduce $k_1 \times n \times (p-1)$ missing values to X_{str}.
    Impute missingness (e.g., via missForest) and update X_{work}.
end for
if $k_0 = 1$ **then**
    for $i = 1 : n$ **do**
        $(i, j) = Rand(neighbor^*(i), 1)$
        Swap the unstructured value for the pair (i, j) in X_{work}
    end for
end if
for $i = 1 : n$ **do**
    $j = Rand(neighbor^*(i), 1)$
    $Z_i = Rand(\{1, ..., p-1\}, k_3 \times (p-1)) = z_{i,1}, ..., z_{i,k_3\% \times (p-1)}$
    for $t \in Z_i$ **do**
        Swap $X_{str}[i, t]$ with $X_{str}[j, t]$
    end for
end for

2.2.8 Simulation

2.2.8.1 Simulation Setup

We present three different simulation studies to demonstrate the performance of the DataSifter I algorithm and assess its capability to (1) obfuscate and guard against stratification attempts for re-identification and (2) manage the overall data structure and preserve useful information in the resulting "sifted" data. In all experiments, we use $n = 1,000$ as number of subjects. In the first simulation, a binary outcome (Y) and five covariates ($X_i, i = 1, ..., 5$) were simulated; X_1 to X_4 were independently generated by normal distributions with the following distribution specifications:

$$X_1, X_2 \sim N(0, 1), X_3 \sim N(-1, 1), \text{ and } X_4 \sim N(0, 2).$$

The binary variable X_5 was directly dependent on X_1 and X_2: $logit(X_{5i}) = 0.5 - 4X_{1i} - X_{2i}$. The binary outcome variable was generated as follows:

$$logit[P(Y_i = 1|X)] = 10 + 10 \times X_{1i} + 10 \times X_{2i} - 5 \times X_{3i} - 20 \times X_{4i} - 15 \times X_{5i} + \epsilon_i,$$

where the residuals were independent and identically distributed (iid) namely $\epsilon_i \sim N(0, 1), i = 1, .., n$. Missingness for X_1 and X_2 was then introduced based on X_5 to meet the MAR criteria, which mimicked the real data situation. Denote $X_{i,1mis} = I(X_{i1} = NA)$ and $X_{i,2mis} = I(X_{i2} = NA)$, where i is the subject indicator. Missingness was introduced using the following probabilities:

$$P(X_{i,1mis} = 1) = P(X_{i,2mis} = 1) = \begin{cases} 0.193, & \text{if } X_5 = Y = 0 \\ 0.060, & \text{if } X_5 + Y = 1 \\ 0.003, & \text{if } X_5 + Y = 2 \end{cases}$$

As mentioned earlier, in the Imputation section, we can impute the original missing

values in the dataset prior to applying the subsequent DataSifter I algorithmic steps. The second simulation demonstrates an example of count outcomes. A Poisson model was used to generate the data.

$$P(Y_i = n) = (\lambda_i^n)/n! \times e^{-\lambda_i},$$

where

$$log(\lambda_i) = 0.2 + 0.5 * X_1 + 1 * X_2 - 0.5 * X_3 - 1 * X_4 - 1.5 * X_5 + \epsilon_i,$$

with iid residuals $\epsilon_i \sim N(0, 1)$. The covariates $X_i, i = 1, ..., 4$ were generated using uniform distributions. We constructed X_5 based on X_1 and X_2 and used a similar strategy as in the first binary simulation to introduce missingness. The third simulation involves continuous outcomes, where the response Y is generated by a similar linear model as in the first experiment; however, it uses an identity link yielding a continuous outcome:

$$Y = 10 + 10 * X_1 + 10 * X_2 - 5 * X_3 - 20 * X_4 - 15 * X_5 + \epsilon_i.$$

Again, the residuals were iid $\epsilon_i \sim N(0, 1)$. All covariates were generated from uniform distributions and the missing patterns were stochastically determined as in the first binary experiment. For all simulation studies, we focused on verifying whether the "sifted" datasets preserve a certain level of the energy that was present in the original true signals, relative to null signals. In addition, we examined the trade-offs between the level of obfuscation and the residual value (utility) of the resulting "sifted" data as a measure of the algorithm's performance. To make all three simulations more realistic, we augmented the original outcome and the (real) five covariates, with 20 additional null features that acted as decoy or "noisy" control features. All 20 null

features were uniformly distributed with various ranges and were independent of the outcome.

For each simulation, we derived 30 "sifted" datasets under a range of privacy levels, from "none" to "indep" level of obfuscation. To assess the privacy protection ability, we measured the Percent of Identical Feature Values (PIFV) between the "sifted" outcome and the original data for all the cases under each obfuscation level, i.e., we compared each subject's original and "sifted" records and measured the ratio between the number of identical values over the total number of features. For determine utility preservation, we used regularized linear models, with an elastic net regularization term, to identify the salient variables. Internal 10-fold statistical cross-validation was used to validate the results of the elastic net feature selection. Let $\mathbf{X}$ denote the covariate matrix (subjects $\times$ features =1,000 $\times$ 25), $\mathbf{y}$ denote the outcome, and β denote the elastic net parameter estimates obtained by optimizing the following objective function. We have

$$\hat{\beta}_{enet} = argmin_{\beta}(\mathbf{y} - \mathbf{X})^T(\mathbf{y} - \mathbf{X}) + \lambda\alpha\|\beta\|^2 + (1-\alpha)\|\beta\|^2,$$

where α is the parameter to determining the blend of the LASSO and Ridge contributions to the penalty, and λ is the regularization penalty parameter [100]. In our experiments, we used $\alpha = 0.8$ giving a slight dominance to the LASSO penalty. A regularization parameter tuning procedure was also performed, using misclassification error rate for binary simulation, deviance for count simulation, and mean squared error for continuous simulation. The largest λ value, which is within one standard error of the minimum cross-validated error, was selected as the optimal parameter [24]. When the estimated coefficient was different from zero, we considered this evidence that the corresponding feature represented a "true" predictor. On the other hand, zero coefficient estimates corresponded to "false" predictors. Recall that in all

simulations, there were five true predictors and 20 null variables. The true positives (number of true features identified) and the false positives (number of null features identifies as true predictors) were recorded for all experiments and each privacy level.

2.2.8.2 Simulation Results - Protection of sensitive information (privacy)

The privacy protection power relies heavily on the user-defined privacy level and the intrinsic information structure. Our results showed that for high privacy levels, PIFVs were close to 0% for all numerical features. For datasets including categorical features, the algorithm provided PIFVs similar to the lowest PIFV between any pair of different subjects in the original dataset. The overall privacy protection performance of the DataSifter I was excellent. Based on the overall simulation performance, a default recommended privacy level may be set at "medium." However, this is also subject to the sensitivity of the data, the specific characteristics of the data, and the trustworthiness of the data requestor. **Figure** 2.4 illustrates the relationship between PIFVs for the synthetic datasets and user-defined privacy levels. The outcome labels "binary", "count", and "continuous" refer to the first experiment, second experiment, and third experiments, respectively. As expected, the graph shows that preservation of sensitive information is better protected when the privacy level is higher. For all three simulations, the DataSifter I had similar performance in terms of PIFV. The outliers in the "none" level resulted from imputation of originally missing values. When the obfuscation was set at "medium"level, the variance of the PIFV was the largest as the levels of obfuscation might differ among individuals when using random sampling. "Small" level of obfuscation manipulated less of the data, with limited range around the neighborhood of each case. Hence, it generated smaller PIFV variances among individuals. On the other hand, "large" obfuscation level had small variance for PIVF as it changed most of the features for all cases. Under the "large" obfuscation setting, PIFV was around 25% for all three experiments, which provided

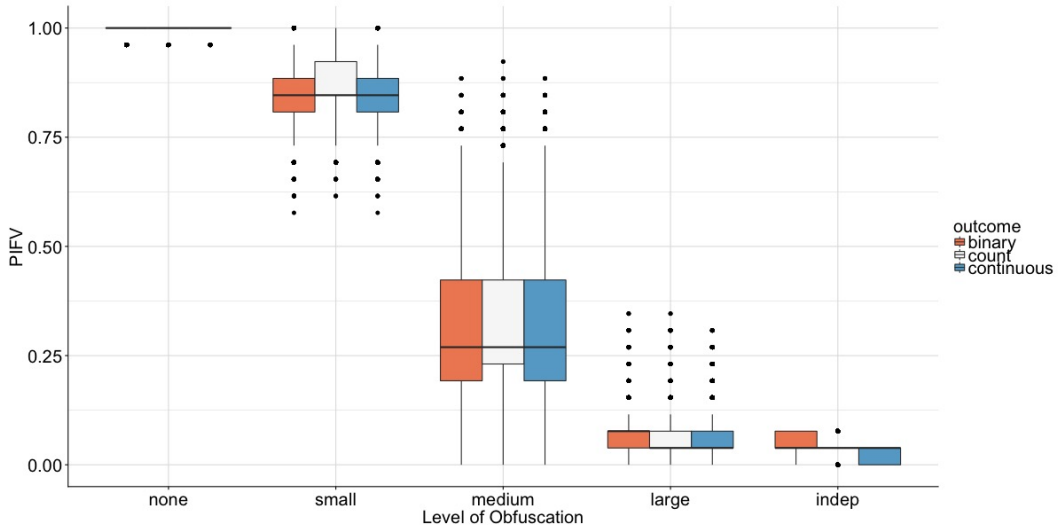

Figure 2.4: Boxplots of Percent of Identical Feature Values (PIFV) under Different Privacy Levels. Binary outcome refers to the first experiment; Count refers to the second experiment; Continuous refers to the third experiment. Each box represents 30 different "sifted" data or 30,000 "sifted" cases.

reliable protection for patient privacy. Under the "medium" level, around 75% of the cases had more than 50% of their data elements different from their original (true) counterparts. The synthetic data under "indep" changed almost all the feature values for every subject. Remember that these five original obfuscation levels represent simple examples of specifying the 5D Data-Sifter-control parameter vector k.

2.2.8.3 Simulation Results - Preserving utility information of the original dataset.

Next, we assessed the DataSifter I algorithm's integrity, in terms of its ability to maintain utility information, i.e., preserve the energy or information content of the original data. A detailed explanation can be found in section 2.2.2. Our results suggest that up to moderate obfuscation levels, the algorithm maintains a fair amount of information (data energy). However, as expected, this ability fades away for larger obfuscation levels. Also, different k parameter vectors have varying effects on the overall utility preservation. The results illustrating the DataSifter I ability to

conserve the data energy are illustrated in **Figure** 2.5. We report the true positive (TP) and false positive (FP) number of feature selections for the three simulation experiments. These results showed that the DataSifter I is able to preserve the signal energy in the original data. As expected, and in contrast to the privacy preservation ability, the performance of the technique to maintain data utility is better under low obfuscation levels. Different outcome types also affect the utility preservation. The simulations show that information energy preservation in the continuous outcome case is slightly better, compared to binary and count outcomes. In the continuous outcome simulation, for obfuscation levels below "large", regularization and variable selection via elastic net successfully identified all five important predictors in almost all "sifted" datasets, and the number of false positives was mostly zero. In addition, the variations of TPs and FPs among different privacy levels was the smallest among the three simulation experiments. The count outcome simulation performed similarly well; under "medium" obfuscation, elastic net was able to select 3 out of 5 features over 75% of the times. Count outcome simulation was not always stable. For instance, some datasets undergoing extreme "sifting" had zero true features selected; however, the algorithm also kept low the false negative rate. The binary outcome simulation demonstrated the least utility preservation as it had the highest false positive rates and the largest variability among all settings. Based on **Figure** 2.5, there is almost no true signal, or false signal, captured in the synthetic "indep" setting, which results from the elimination of the correlations among features. The extreme "indep" case aims to achieve maximum protection for patient privacy. As a consequence, the resulting "sifted" data provides little utility.

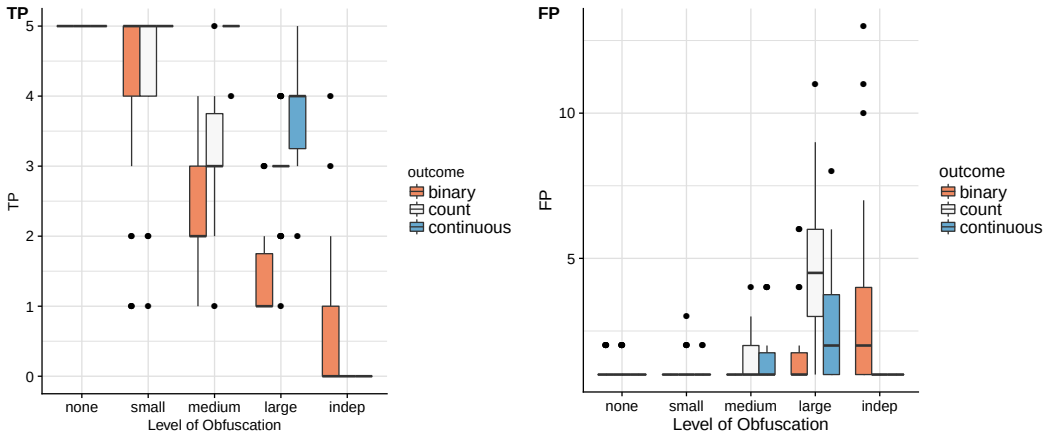

Figure 2.5: Logistic Model with Elastic Net Signal Capturing Ability. TP is the number of true signals (total true predictors = 5) captured by the model. FP is the number of null signals that the model has falsely selected (total null signals=20).

2.2.9 Clinical Data Application: Using DataSifter I to Obfuscate the ABIDE Data

We demonstrate the functionality of the DataSifter I on the Autism Brain Imaging Data Exchange (ABIDE) dataset [18]. The ABIDE dataset represents a multi-institutional effort for aggregating and sharing the imaging, clinical and phenotypic data of 1,112 volunteers [18]. The data includes resting-state functional magnetic resonance imaging (rs-fMRI) structural MRI, and phenotypic information of 539 patients (autism spectrum disorder) and 573 age-matched asymptomatic controls. In our study, we selected a subsample of 1,098 patients including 528 ASD and 570 controls. The dataset has 500 structural MRI biomarkers and phenotypical information such as age, sex and IQ. It is a very challenging case-study due to the heterogeneity of the data, format of the data elements, and the complexity of mental health phenotypes. We use the ABIDE data to showcase the performance of the DataSifter I technique on a convoluted multiplex study. The ABIDE dataset comprises 1,098 patients and 506 features. We included one unstructured feature-"image data file name" ("Data") in the dataset to show the DataSifter I ability to obfuscate unstructured text elements. Resembling the simulation experiments, we built a $dataSifter()$ function that has

five different levels of obfuscation to demonstrate the obfuscation utility trade-off. Obfuscation was assessed using PIFV as the simulation studies. We applied random forest [40] to predict the target binary outcome autism spectrum disorder (ASD) status (ASD vs. control) as a proxy of the algorithm's utility to maintain the energy of the original dataset into the "sifted" output. Predictions of the ASD status was conducted with the randomForest package. When specifying the parameters in the $dataSifter()$ function, level of obfuscation can be set by level. Here we used five different obfuscation levels. The level of obfuscation can be alternatively specified using a set of k combinations as function arguments. For example, to perform "small" obfuscation level, we can specify $k_0 = 0, k_1 = 0.05, k_2 = 1, k_3 = 0.1, k_4 = 0.01$ in the $dataSifter()$ function, which creates a flexible way to manage obfuscation levels. In this study, as mentioned above, the unstructured variable was named as "Data". If there are no rich text variables, the set of unstructured.names can be left to default (i.e., NULL). Explicit sensitive information like the subject ID, i.e., $subjID$ column, needs to be removed from the original dataset in advance. The batch size for the algorithm is defined by the parameter $batchsubj$. As mentioned in the Methods section, the DataSifter I algorithm operated on batches to provide scalability and alleviate the computational complexity. We recommend using a relatively small $batchsubj$ and large number of cores for datasets with a huge number of cases (e.g., hundreds of thousands). The maximum number of iterations for the missForest imputation algorithm is set to one to minimize the computational cost determined by imputing a large number of features. We obtained five "sifted" output datasets corresponding to different obfuscation levels: no ("none" obfuscation), s ("small" obfucation), m ("medium" obfuscation), l ("large" obfuscation), and i ("indep" synthetic data from empirical distributions of each feature). We then inspected the obfuscations made to the original dataset. Boxplots for PIFV was then plotted in **Figure** 2.6 **A** to illustrate the overall obfuscation effect. As expected, PIFV decreases with level of

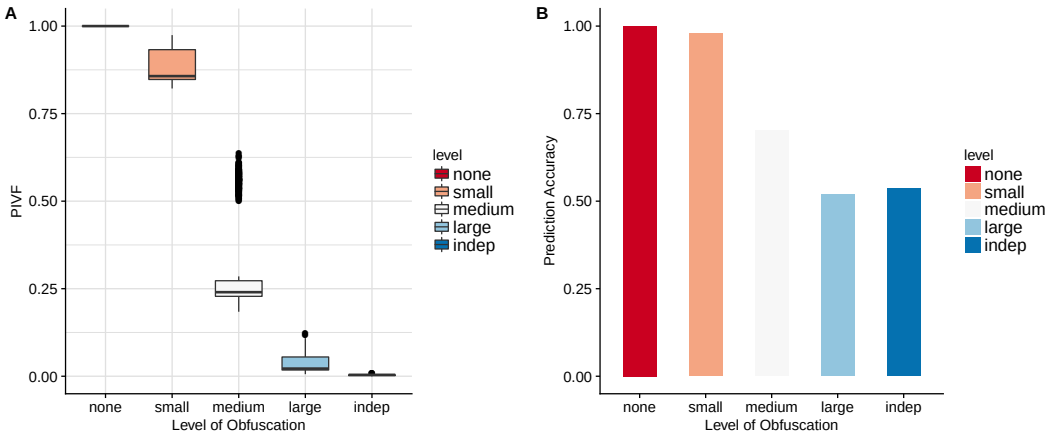

Figure 2.6: Boxplots of PIFVs for ABIDE under different levels of DataSifter I obfuscations. Each box represents 1,098 subjects among the ABIDE sub-cohort.

obfuscation. Comparing the application with the simulation experiments, the algorithm works better with a larger number of features. Under "medium" obfuscation level, the algorithm achieved 50% and 25% PIVF for the binary simulation data and ABIDE data, respectively.

To assess the utility information, we use the "sifted" datasets as training sets to fit random forest. These trained models provided predicted values for 632 complete cases in the original ABIDE data. The random forest built using no dataset predicted all outcomes correctly. s, m, l and i datasets were able to provide predictions with 98%, 70%, 52% and 54% accuracy, respectively. The prediction accuracy of all the datasets are illustrated in **Figure** 2.6 (B). Again, this result shows the trade-off between utility and the user-controlled privacy levels.

2.3 DataSifter II: Partially Synthetic Time-varying Data Generation

2.3.1 Privacy and Utility Measurement for Partially Synthetic Data

2.3.1.1 Data Structure and Notations

Time-varying correlated data are common in most biomedical and epidemiology studies. For example, in multi-center studies, we typically measure the target variables across all subjects at a single time point, but the subjects may be correlated within each center. In longitudinal data, the target variables are measured repeatedly at baseline and during follow-up, and thus we have intrinsic within-subject correlations. In this case, to reduce measurement errors, researchers take repeated measurements on the same subjects, which may also involve within-subject correlations. The DataSifter II framework can be applied to any correlated data. For illustration purposes in this study, we investigate the use of DataSifter II on longitudinal data.

Consider a longitudinal EHR dataset with n patients, each recorded until J_i^{th} visit, where the time intervals between visits are similar across patients. We collect m_l longitudinal variables at each visit and m_s static variables for patient characteristics. For simplicity, we denote time-varying variables as Y's and time-invariant variables as X's. In the following sections, we use $i = 1, ..., n$ to denote patients; $j = 1, ..., J_i$ to denote the visit time, which allow different visit times among patients; and k to index the variables (columns) in the dataset such that for static variable $k = 1, ..., m_s$, and for longitudinal variable $k = 1, ..., m_l$. Dummy variables are created for all categorical

longitudinal variables. Then the longitudinal measurements for subject i are

$$\mathbf{Y}_i = \begin{bmatrix} Y_{i,1,1} & ... & Y_{i,1,m_l} \\ Y_{i,2,1} & ... & Y_{i,2,m_l} \\ ... & ... & ... \\ Y_{i,J_i,1} & ... & Y_{i,J_i,m_l} \end{bmatrix}$$

with patient i's, time j's record of longitudinal variable k denoted as $Y_{i,j,k}$. The time-invariant variables of subject i are denoted as $\mathbf{X}_i = (X_{i1}, ..., X_{im_s})$, which can also be represented as $\mathbf{X}_i = (\mathbf{X}_{i1}, ..., \mathbf{X}_{im_s})$ to match the dimension of $\mathbf{Y}_i$ where $\mathbf{X}_{ik} = (X_{i,1,k}, ..., X_{i,J_i,k})^T$, and $k = 1, ..., m_s$ repeats the static variable for J_i times.

Missing data occurs often in longitudinal observations. Missingness can come from a completely missing record or partially missing record, where patient i does not have all data available for some visits. In this case, we denote the missing indices for k^{th} outcome as $mis_k = \{(i,j)|Y_{i,j,k} = \text{NA }\}$, and observed indices as $obs_k = \{(i,j)|Y_{i,j,k} \neq \text{NA}\}$. Fully observed long format data $\mathbf{Y}_{(\sum_i J_i)\times(m_l)} = (\mathbf{Y}_1, ..., \mathbf{Y}_n)^T$ has $\sum_i J_i$ rows and m_l columns. Similarly, the static variables are denoted as $\mathbf{X}_{(\sum_i J_i)\times m_s} = (\mathbf{X}_1, ..., \mathbf{X}_n)^T$. We use $\mathbf{D} = [\mathbf{X}, \mathbf{Y}]$ to denote the observed dataset.

While generating partially synthetic data, we view data obfuscation as an artificial missing creation and imputation procedure. Here we focus on obfuscating time-varying data and further denote each row in partially synthetic time-varying data as $(\mathbf{Y}_{i,j,\text{nrep}_{ij}}, \hat{\mathbf{Y}}_{i,j,\text{rep}_{ij}})$, where $\text{nrep}_{i,j}$ is a set of indexes for unreplaced variables, $\text{rep}_{i,j}$ is a set of indexes for replaced variables compared with the original row such that $\text{nrep}_{i,j} \cup \text{rep}_{i,j} = \{1, ..., m_l\}$, and $\hat{\mathbf{Y}}_{i,j,\text{rep}_{ij}}$ is a vector of the synthetic values created for patient i's record at time j. Similarly we denote one row in the synthetic static data as $(\mathbf{X}_{i,\text{nrep}_i}, \hat{\mathbf{X}}_{i,\text{rep}_i})$, where nrep_i and rep_i are indexes for unreplaced and replaced variables, and $\hat{\mathbf{X}}_{i,\text{rep}_i}$ denotes a vector of synthetic values for patient i's static characteristics. Finally, we use $\mathbf{Z}$ to denote a synthetic dataset that is composed of

the time-varying and static synthetic data components.

2.3.1.2 Data Privacy Measurement

In this section, we formally define data privacy in the form of disclosure risk and compare the disclosure risk between MI and DataSifter methods. Assume we have a m_l-dimensional partially synthetic time-varying data vector $\mathbf{y}_{i,j} = (\mathbf{y}_{i,j,\text{nrep}_{i,j}}, \mathbf{y}_{i,j,\text{rep}_{i,j}})$ corresponding to each individual i at time j in the original dataset $\mathbf{D}$ and the static portion of data is complete. We have a partially synthetic dataset $\mathbf{Z}$ that follows a similar joint distribution as the original data with unchanged static variables (unobfuscated). Specifically, we denote the partially synthetic dataset generated by DataSifter as $\mathbf{Z}_{sift}$ and U ($U \geq 2$) multiply imputed (MI) datasets as $\mathbf{Z}_{MI} = (\mathbf{Z}_{MI}^{(1)}, ..., \mathbf{Z}_{MI}^{(U)})$. To compare the *disclosure risk* between the *sifted* and multiply imputed partially synthetic datasets, we closely follow the Bayesian risk approach described in [60].

Suppose an intruder is interested in learning some the true values in the $\mathbf{y}_{i,j}$ vector. Let A represent the intruder's prior knowledge about the original dataset $\mathbf{D}$, which is often referred to a subset of records in $\mathbf{D}_{-(i,j)} = \mathbf{D} \setminus \{\mathbf{y}_{i,j}\}$. Let S denote any information known by the intruder about the synthetic data generation procedure. Then, define the disclosure risk for $y_{i,j,k}$ as the conditional distribution

$$\underbrace{p(Y_{i,j,k} = y_{i,j,k}|\mathbf{Z}, A, S)}_{disclosure\ risk},$$

where $j \in \text{nrep}_{i,j} \cup \text{rep}_{i,j}$. The intruder cannot infer the location (indices) of unchanged ($\text{nrep}_{i,j}$) and changed ($\text{rep}_{i,j}$) cells in $\mathbf{Z}_{sift}$, whereas the unchanged cells in a set of U multiple-imputed datasets would imply the index locations for $\text{nrep}_{i,j}$. Hence, for the output *sifted* dataset, we have the disclosure risk:

$$p(Y_{i,j,k} = y_{i,j,k}|\mathbf{Z}_{sift}, A, S).$$

For $\mathbf{Z}_{MI}$ we have

$$p(Y_{i,j,k} = y_{i,j,k}|\mathbf{Z}_{MI}, A, S) = \{1 - I(k \in \text{nrep}_{i,j})\} p(Y_{i,j,k} = y_{i,j,k}|\mathbf{Z}_{MI}, \mathbf{Y}_{i,j,\text{nrep}_{i,j}} = \mathbf{y}_{i,j,nrep_{i,j}}, A, S) + I(k \in \text{nrep}_{i,j}),$$

where $I(k \in \text{nrep}_{i,j})$ is an indicator that takes the value of 1 when the data cell (i, j, k) of the multiple imputed datasets are not replaced with obfuscated value. When $k \in \text{nrep}_{i,j}$, the intruder knows that $\mathbf{Z}_{MI}$ contains true value at cell (i, j, k) so that the discourse risk is 1. On the other hand, when $k \notin \text{nrep}_{i,j}$, it is appropriate to assume that knowing which columns in the record for patient i's j^{th} visit contains true values (knowing $\mathbf{Y}_{i,j,\text{nrep}_{i,j}} = \mathbf{y}_{i,j,\text{nrep}_{i,j}}$) yields similar or higher disclosure risk compared to not knowing the locations as it provides more information about the true covariates for $Y_{i,j}$. When the information contained in $\mathbf{Z}_{sift}$ and $\mathbf{Z}_{MI}$ is similar regarding to inferring the distribution of $Y_{i,j,k}$, we have

$$p(Y_{i,j,k} = y_{i,j,k}|\mathbf{Z}_{MI}, A, S) \geq p(Y_{i,j,k} = y_{i,j,k}|\mathbf{Z}_{sift}, A, S).$$

Therefore, when both synthetic datasets contain comparable information, the DataSifter output has smaller, or rarely similar, disclosure risk compared to the multiple imputation method.

Data governors can quantify the privacy protection level of the synthetic data using our disclosure risk defined above. Specifically, when calculating the maximal privacy loss for each record, we assume the intruder knows all other records in the raw dataset, i.e. $A = \mathbf{D}_{-(i,j)}$, and the imputation model for the synthetic data is known. The proposed data privacy measurement (PM) for cell (i, j, k) is defined as

the difference between the expected and observed value:

$$\begin{aligned} PM_{i,j,k} =& E(Y_{i,j,k}|\mathbf{Z}, \mathbf{D}_{-(i,j)}, S) - y_{i,j,k} \\ =& \left\{ \int y \cdot p(Y_{i,j,k} = y|\mathbf{Z}, \mathbf{D}_{-(i,j)}, S) dy \right\} - y_{i,j,k}. \end{aligned}$$

In practice, the conditional model for $Y_{i,j,k}$ is constructed using $\mathbf{D}_{-(i,j)}$ with the identical model specification as the missing imputation model. For $Y_{i,j,k}$ in $\mathbf{Z}_{sift}$ or replaced cells in $\mathbf{Z}_{MI}$, we calculate the expected difference between the model prediction given other covariates in the synthetic data and the true value. For $\mathbf{Z}_{MI}$, the privacy measurement of $Y_{i,j,k}$ for unchanged cells is 0. Assume we introduce a percent of artificial missingness to the original data and $\mathbf{D}_{-(i,j)}$ provides sufficient information for accurate $Y_{i,j,k}$ predictions. For MI, there are $(1-a)$ of $Y_{i,j,k}$ with $PM = 0$ and the remaining cells have equal or slightly smaller PM. Thus, DataSifter is expected to improve PM by at least $1/a$ times compared to MI. We examine the average privacy measurement for every time-varying variable in the simulation and application sections below.

2.3.1.3 Data Utility Measurement

Given a pre-specified model, we can obtain the desired utility of the partially synthetic data by comparing the model fitted with the original and synthetic data in terms of model inference and/or prediction accuracy. For model inference, the data governor can consider a feasible parametric model regressing a summary variable on other covariates. For example, in EHR data, we can predict patients' comorbidity score over time, which represents a summary score for the patient's medical conditions. To test the data utility based on a regression coefficient β, we first fit the desired model with the original dataset and obtain its confidence interval. Then, we generate L partially synthetic datasets under the same target parameter setting, where L is a

large positive integer. By fitting the desired model on each synthetic data, we obtain a set of $\hat{\beta}_l$, $l = 1, ..., L$ and corresponding confidence intervals $(\text{LB}_{\hat{\beta}_l}, \text{UB}_{\hat{\beta}_l})$. In the ideal case, where the true coefficient β^* is known, we can directly use the confidence interval coverage, i.e. $\frac{\sum_{l=1}^{L} I\{\beta^* \in (\text{LB}_{\hat{\beta}_l}, \text{UB}_{\hat{\beta}_l})\}}{L}$ as our utility measurement. In practice, we obtain the empirical confidence interval for $\hat{\beta}_l$ from the synthetic datasets and measure if it overlaps with the confidence interval provided by the original dataset. In terms of prediction accuracy, we can set aside a randomly selected test set and compare the prediction error between models constructed with the remaining original and synthetic data records.

2.3.2 DataSifter II Technique

The proposed DataSifter II procedure operates on static variables $\mathbf{X}$ and time-varying variables $\mathbf{Y}$ separately and merges the two components back together to form the final partially synthetic data. The static variables are obfuscated with DataSifter I algorithm. The DataSifter II requires complete static variables as candidate predictors while obfuscating $\mathbf{Y}$. We handle possible missingness for time-invariant variables by missForest technique [80]. When obfuscating $\mathbf{Y}$, we first impute the original missingness in $\mathbf{Y}$ with inverse probability weighted imputation models. Then, we randomly introduce missingness to the working time-varying data and impute back with a proposed robust imputation method.

The main assumptions of the DataSifter II include: (A1) the possible missingness in the original data follows missing at random (MAR) or missing completely at random (MCAR) missing mechanism; (A2) The utility of each variable is equally important. We consider above assumptions because (A1) guarantees the imputation accuracy and (A2) allows indistinctive obfuscation for each variable.

2.3.2.1 Sifting Static Variables with DataSifter I

We apply DataSifter I to obfuscate the static portion $\mathbf{X}$. DataSifter I use missForest to impute potential missingness in the original data, introduce artificial missingness to the working data and impute the missing cells back, and swap partial information for similar records. The resulting *sifted* data has complete records sharing the same format with the original data.

The imputation procedures in DataSifter aim to create a single complete dataset disguising the original or artificial missing positions. We use the missForest technique that outputs a single imputed dataset and is proven to have smaller imputation errors than common methods including multiple imputation [80, 89]. This non-parametric imputation technique sequentially imputes and updates the data by variable. In the first iteration, when imputing for the first target variable, it fills in the missing cells among other predictor variables with mean imputation. Then, it constructs random forest models (target variable versus all other variables) to provide imputations. In subsequent iterations, while imputing and updating by variable, the imputation accuracy for each target variable improves as the missing cells in all other variables are replaced with better predictions.

When imputing the original missing data, we employ the original stopping criterion in missForest. It stops to iterate when the difference between the latest and prior imputed data matrix is at least as great as the previous difference measured or the maximal iteration limit has achieved. The difference between matrices in sets of continuous ($\mathbf{N}$) and categorical ($\mathbf{F}$) variables are defined as

$$\Delta_{\mathbf{N}} = \frac{\sum_{k\in\mathbf{N}}(\hat{\mathbf{X}}_k^{(r)} - \hat{\mathbf{X}}_k^{(r-1)})^2}{\sum_{k\in\mathbf{N}}(\hat{\mathbf{X}}_k^{(r)})^2}$$

and

$$\Delta_{\mathbf{F}} = \frac{\sum_{k\in\mathbf{F}}\sum_{i=1}^{n} I(\hat{\mathbf{X}}_{ik}^{(r)} \neq \hat{\mathbf{X}}_{ik}^{(r-1)})}{\text{Number of missing cells in categorical variables}},$$

where $\hat{\mathbf{X}}_k^{(r)}$ is the imputed vector and $\hat{X}_{ik}^{(r)}$ is the imputed value for subject i of the k^{th} variable in the r^{th} iteration.

When imputing artificial missingness, the true missing value is known. We define the stopping criterion under tolerance level ϵ as

$$\frac{\|X^*_{mis_k,k} - \hat{X}^{(r)}_{mis_k,k}\|_1}{\|X^*_{mis_k,k}\|_1} < \epsilon,$$

where $X^*_{mis_k,k}$ is the true values in the working data after imputing original missing data and $\hat{X}^{(r)}_{mis_k,k}$ is the imputed values.

2.3.2.2 Iterative Imputation Algorithm for Time-varying Data

Although DataSifter I applies robust nonparametric imputation methods like missForest to impute static missing variables, effective missing imputation for time-varying data can be challenging. In this chapter, we propose an iterative imputation algorithm for longitudinal data similar to the missForest algorithm. The proposed algorithm considers two types of missing mechanisms (MAR and MCAR) and two modeling options (linear mixed model and RE-EMtree). It handles missingness in time-varying variables $\mathbf{Y}$ with complete static variables $\mathbf{X}$ as potential predictors.

Before the imputation, we initiate all missing cells with the closest value from the same subject (last value carry forward or next value carry backward). If the subject has no observations of certain variables, we initialize such missing cells with mean imputations. Then, we sort the variables ascendingly based on missing percentage so that $Y_{\cdot,\cdot,1}$ has the smallest missing percentage and $Y_{\cdot,\cdot,m_l}$ has the most missing. Next, we start our iterative imputation procedure. Within an iteration, we impute from the first to the last variable with missing. While imputing a target variable $Y_{\cdot,\cdot,k}$, we separate the working data into four groups: the observed values of the target variable $Y_{obs_k,k}$, variables other than the target among the observed rows $[\mathbf{Y}_{obs_k,-k}, \mathbf{X}_{obs_k,\cdot}]$, the missing cells of the target variable with current imputation values $Y_{mis_k,k}$, and

variables other than the target among the missing rows $[\mathbf{Y}_{mis_k,-k}, \mathbf{X}_{mis_k,\cdot}]$, where obs_k and mis_k are the patient and visit index sets (i, j) with observed and missing variable k, respectively. Imputation models for the target variable is constructed by regressing $Y_{obs_k,k}$ on $[\mathbf{Y}_{obs_k,-k}, \mathbf{X}_{obs_k,\cdot}]$ and we update $Y_{mis_k,k}$ based on the imputation model. The imputation of a following missing variable k' $(k < k' \leq m_l)$ will benefit from this update because we have better estimates of the missing values in $Y_{\cdot,\cdot,k}$ for constructing the imputation model or providing covariates when predicting $Y_{mis'_k,k'}$. The algorithm finalizes the imputation result of a target variable when the imputation model predictions for the observed values are close to the true values after multiple iterations. For artificial missing, we directly compare the true missing values with its predictions. The algorithm stops when less than two variables are not finalized or maximal iteration has achieved.

Imputation Model Under Missing at Random Under different missing patterns, the algorithm utilizes different imputation models. When we have MAR, the missingness depends on observed data and the complete observations might be biased. We utilize inverse probability weighting to obtain an unbiased pseudo sample for imputation model fitting. By pseudo sample, we mean the weighted sample that creates balance by up-weighting the underrepresented population and down-weighting the over-represented population in the complete observations, where the weights can be calculated at the subject level, or subject and time level, to allow better imputation under different situations. For subject level, the probability of missing for each subject denoted as $P\{I(\mathbf{Y}_i \text{ contains NA})\}$ is modeled with the corresponding logistic regression using all working complete static variables and a LASSO penalty is applied for variable selection. Weights are calculated by the estimated inverse probability of being observed

$$w_i = \frac{1}{1 - \hat{P}(\mathbf{Y}_i \text{ contains NA})}.$$

In observational data like EHR, missingness at the subject and time level happens sporadically under usual circumstances; i.e., missingness can happen at any time point for a patient. Similarly, subject i at time j will be weighted by $w_{i,j} = \frac{1}{1-\hat{P}(Y_{i,j}\text{is missing})}$, where $\hat{P}(Y_{i,j}$ is missing) is estimated by a Generalized Linear Mixed Model (GLMM) and LASSO penalty is applied for variables selection. After estimating the weights for the observed records, we construct the imputation model for each target longitudinal variable with a weighted linear mixed model. The linear mixed model with random intercept follows:

$$Y_{\cdot,\cdot,k} = \mathbf{X}^{*T}\beta + \mathbf{Z}^T\mathbf{b} + \epsilon,$$

where $Y_{\cdot,\cdot,k}$, $k \in \{1, ..., m_l\}$ is the target longitudinal outcome, $\mathbf{X}^*$ are the selected significant predictors, $\mathbf{Z}$ is the design matrix for random effects, $b_i \sim N(0, \sigma_b^2)$, and $\epsilon_i \sim N(0, \sigma^2)$.

Accordingly, $\text{Var}(Y_{\cdot,\cdot,k}) = \sigma^2(\mathbf{Z}\mathcal{T}\mathbf{Z}^T + I) = \sigma^2\mathbf{H}$, with $\mathcal{T} = \frac{\sigma_b^2}{\sigma^2}I_{n\times n}$. We estimate the imputation model by optimizing the weighted log likelihood for complete cases:

$$L^w = C - \frac{1}{2}log(|\mathbf{H}|) - \frac{1}{2}nlog(\sigma^2) - \frac{1}{2\sigma^2}(Y_{\cdot,\cdot,k} - \mathbf{X}^*\beta)^T\mathbf{W}^{*T}\mathbf{H}^{-1}\mathbf{W}^*(Y_{\cdot,\cdot,k} - \mathbf{X}^*\beta),$$

where C is a constant and

$$W^* = \begin{bmatrix} \sqrt{w_{1,1}} & 0 & ... & 0 \\ 0 & \sqrt{w_{1,2}} & ... & 0 \\ ... & ... & ... & ... \\ 0 & ... & 0 & \sqrt{w_{n,J_n}} \end{bmatrix}$$

is a $(\sum_{i=1}^n J_i) \times (\sum_{i=1}^n J_i)$ diagonal matrix. We obtain each stochastic imputation with $X_i^*\hat{\beta} + \hat{b}_i$, where $\hat{b}_i$ is randomly sampled from $N(0, \hat{\sigma_b}^2)$.

Imputation Model under Missing Completely at Random Under MCAR, we propose to employ two modeling options Generalized Linear Mixed Model (GLMM) [47] or Random Effects-Expectation Maximization tree (RE-EM tree) [77] as imputation models within the iterative procedure. The two procedures are referred to as DataSifter II GLMM and DataSifter II RE-EM. Note that GLMM can handle various data types, including continuous, binary, and count data, whereas RE-EM tree is an effective algorithm for continuous measurements. For the DataSifter II GLMM, variable selection is conducted separately with GLMM LASSO. Here we perform a grid search for the regularization parameter and use Bayesian Information Criterion (BIC) to select the best model. Since an appropriate starting point is crucial for model convergence, DataSifter II incorporates two methods to initiate the parameters when fitting GLMM LASSO. The first method estimates all the parameters by glmmPQL, which is using pseudo-likelihood. GlmmPQL estimates the mean and variance parameters iteratively with maximum likelihood assuming normality [95]. It approximates the true likelihood with a strong normality assumption, but provides a computationally efficient way of estimating initial regression parameters. When the signal is sparse, and the GLMM algorithm does not converge with the glmmPQL initial values, we may consider initialization using zeros or another user-specified initialization.

Selected variables are denoted as $\mathbf{X}^* = (\mathbf{X}_1^*, .., \mathbf{X}_s^*)$, which may come from longitudinal variables other than the target and static variables. We fit the following prediction model for every missing longitudinal variable,

$$\eta_{i,j} = g\{E(Y_{i,j})\} = \mathbf{X}_{i,j}^{*T}\beta + \mathbf{Z}_{i,j}^{T}\gamma_i,$$

where $g(\cdot)$ is a known link function, $\mathbf{Z}_{i,j}$ is the designed matrix for random effects γ_i, $\gamma_i \sim N(0, D)$, $i = 1, ..., n$ represents subjects, and $j = 1, ..., J_i$ are different time

points.

After estimating β and D using observed data, we impute the missing values by randomly sampling $\hat{\gamma}_i \sim N(0, \hat{D})$ and then obtain a *Best Linear Unbiased Prediction (BLUP)* imputation prediction $g^{-1}(\mathbf{X}_{i,j}^{*T}\hat{\beta} + \mathbf{Z}_{i,j}^{T}\hat{\gamma}_i)$ for $Y_{i,j,k}$ with missing values.

DataSifter II RE-EM provides an alternative robust obfuscation for continuous time-varying measurements. RE-EM tree model combines the tree-based estimation for fixed effects and parametric estimation for random effects [77]. RE-EM tree is a semi-parametric generalization of the linear mixed effect model:

$$Y_{i,j} = f(X_{i,j,1}, ..., X_{i,j,m_s}) + \mathbf{Z_{i,j}}^{T}\gamma_i + \epsilon_{i,j},$$

where $(\epsilon_{i,1}, ..., \epsilon_{i,J_i})^T \sim N(0, R_i)$, $\gamma_i \sim N(0, D)$, $f(\cdot)$ function denotes a regression tree in the previous model, and R_i records the variance-covariance structure for i^{th} error term. RE-EM tree enjoys the capability of modeling the non-linear trend for fixed effects so that variable selection can be avoided. Parameter estimation for the RE-EM tree follows a two-step procedure: First, when estimating $f(\cdot)$, RE-EM tree adapts the CART tree algorithm. Assuming that $\gamma_i's$ are known and equal to the current estimate $\gamma_i^{(r)}$, the outcome of $f(\cdot)$ is $Y_{i,j} - \mathbf{Z_{i,j}}^{T}\gamma_i^{(r)}$. Fitting the tree is a binary recursive procedure that splits the whole population into similar subgroups. The default minimum number of subjects in the terminal node is set to 20. Also, a new split will be made when the reduction in sum of squares between the individual outcome and group average is less than 1%. In other words, we set the *complexity parameter (cp)* to be 0.01. To avoid overfitting, pruning is done by 10-fold cross-validation after the initial fitting. The final tree is selected with the largest *cp* value within one standard error above the minimized 10-fold cross-validated error. Second, extract the random effect estimates $\hat{\gamma}_i$ from a linear mixed model that regress Y_{ij} on $\hat{f}(X_{i,j,1}, ..., X_{i,j,m_s})$.

2.3.2.3 Create Partially Synthetic Time-varying Data

Using the proposed iterative imputation tool, we can create partially synthetic time-varying data by handling potential missing in $\mathbf{Y}$, generate artificial missingness and impute back.

Similar to the preprocessing step for static variables, we intend to initiate the process with a complete dataset containing both static and time-varying data. The working complete static data $\mathbf{X}$ can be obtained from missForest imputation. If missing values exist in the time-varying variables, we pre-process the data using the proposed imputation algorithm under MAR assumption. In practice, we can choose the subject level or subject and time level propensity model for missing based on different missing patterns, i.e., we model $P\{I(\mathbf{Y}_i \text{ contains NA})\}$ if missingness usually happens at subject level and $P\{I(\mathbf{Y}_{ij} = \text{NA})\}$ if missingness happens sporadically. Since the true missing values are unknown, we finalize the imputation result for a target variable k when $\frac{\|\mathbf{Y}_{obs_k,k} - \hat{\mathbf{Y}}^{(r)}_{obs_k,k}\|_1}{\|\mathbf{Y}_{obs_k,k}\|_1} < \epsilon$ at current iteration r for a pre-specificed tolerance level ϵ, where $k \in \{1, ..., m_l\}$.

Following the initial imputation, we start Sifting by introducing artificial random missing values to the longitudinal variables in the working complete dataset. Such randomly generated missingness follows a MCAR missing mechanism, which guarantees that the unweighted complete-case analysis is bias-free. We then impute the missing variables one by one with the proposed imputation procedure under MCAR with a data driven choice of either the parametric or semi-parametric imputation model.

2.3.2.4 Implementation of DataSifter II

We use Algorithm 2 to summarize the proposed imputation method for time-varying variables. The algorithm finalize the imputation for variable $Y_{\cdot,\cdot,k}$ at r^{th}

iteration when

$$\frac{\|\mathbf{Y}_{obs_k,k} - \hat{\mathbf{Y}}^{(r)}_{obs_k,k}\|_1}{\|\mathbf{Y}_{obs_k,k}\|_1} < \epsilon$$

at tolerance level ϵ. When imputing artificial missing data, the original missing values are given. Hence, we have an alternative criteria for determining if imputation can be finalized

$$\frac{\|\mathbf{Y}_{mis_k,k} - \hat{\mathbf{Y}}^{(r)}_{mis_k,k}\|_1}{\|\mathbf{Y}_{mis_k,k}\|_1} < \epsilon.$$

Algorithm 2 Time-varying data missing imputation algorithm

Input: complete static variables $\mathbf{X} \in \mathbb{R}^{(\sum_i J_i) \times m_s}$, time-varying variables $\mathbf{Y} \in \mathbb{R}^{(\sum_i J_i) \times (m_l)}$, missing mechanism (MAR or MCAR), imputation model (GLMM or RE-EMtree), and tolerance level ϵ.
Initially impute the missing cells in Y with a combination of last value carry forward and next value carry backward.
Sort the m_l variables in $\mathbf{Y}$ based on missing rate so that the first variable in $\mathbf{Y}$ has the least missing and the last variable in $\mathbf{Y}$ has the most missing.
Create a list of missing variable indexes vlist $= \{k_m, ..., m_l\}$ where missingness appear from the k_m^{th} variable.
Sort the m_l variables in $\mathbf{Y}$ based on missing rate so that the first variable in $\mathbf{Y}$ has the least missing and the last variable in $\mathbf{Y}$ has the most missing.
Create a list of missing variable indexes vlist $= \{k_m, ..., m_l\}$ where missingness appear from the k_m^{th} variable.
repeat
    for $k \in$ vlist **do**
        Separate data in four groups with

$$\left[\begin{array}{c|cc} Y_{obs_k,k} & \mathbf{Y}_{obs_k,-k} & \mathbf{X}_{obs_k,\cdot} \\ \hline Y_{mis_k,k} & \mathbf{Y}_{mis_k,-k} & \mathbf{X}_{mis_k,\cdot} \end{array}\right]$$

        if missing mechanism = MAR **then**
            Construct propensity score model for missingness and calculate the inverse probability of missing for records with row indexes obs_k.
            Perform variable selection with LMM with LASSO using $Y_{obs_k,k}$ as outcome and $[\mathbf{Y}_{obs_k,-k}, \mathbf{X}_{obs_k,\cdot}]$ as potential predictors.
            Fit inverse probability weighted LMM with $Y_{obs_k,k}$ as outcome and selected predictors as covariates.
            Impute missing values $Y_{mis_k,k}$ using the weighted LMM.
        else
            if imputation model = GLMM **then**
                Fit GLMM with LASSO regularization regressing $Y_{obs_k,k}$ on $[\mathbf{Y}_{obs_k,-k}, \mathbf{X}_{obs_k,\cdot}]$.
            else
                Fit RE-EMtree regressing $Y_{obs_k,k}$ on $[\mathbf{Y}_{obs_k,-k}, \mathbf{X}_{obs_k,\cdot}]$.
            end if
            Impute missing values $Y_{mis_k,k}$ using the fitted imputation model.
        end if
    end for
    iteration = iteration +1
    if Imputation finalizing criteria at tolerance level ϵ has met **then**
        Exclude k from vlist.
    end if
until iteration $>$ maxit or the length of vlist ≤ 1.
Output *sifted* time-varying variables $\mathbf{Y}^s$.

We summarize the implementation of DataSifter II with Algorithm 3.

Algorithm 3 DataSifter II

1: **Input:** static variables $\mathbf{X} \in \mathbb{R}^{n \times m_s}$, time-varying variables $\mathbf{Y} \in \mathbb{R}^{(\sum_i J_i) \times (m_l)}$, imputation model I, DataSifter I obfuscation level L, percent of artificial missing to introduce a, and tolerance level ϵ.
2: Operate DataSifter I on the static variables under obfuscation level L and obtain complete static variables. For patient i create J_i replicates of the working complete static record to create $\mathbf{X}^s \in \mathbb{R}^{\sum_i J_i \times m_s}$.
3: Operate **Algorithm 1**($\mathbf{X}^s$,$\mathbf{Y}$,MAR,GLMM,ϵ) to impute possible original missingness in $\mathbf{Y}$.
4: Introduce random missingness to a% of data values for data obfuscation purpose among the m_l time-varying variables in the working data and obtain data with artificial missingness denoted as $\mathbf{Y}^*$. Record real values of the missing cells.
5: Operate **Algorithm 1**($\mathbf{X}^s$,$\mathbf{Y}^*$,MCAR,I,ϵ) to obtain *sifted* time-varying variables $\mathbf{Y}^s$.
6: **Output**: A single complete and *sifted* dataset $[\mathbf{X}^s, \mathbf{Y}^s] \in \mathbb{R}^{\sum_i J_i \times (m_s + m_l)}$.

2.3.2.5 Residual Diagnostics

The residuals or errors introduced by DataSifter II obfuscated values follow a mixture distribution. When the final imputation model for variable $Y_{\cdot,\cdot,k}$ at its last iteration r_k satisfies $\frac{\|Y_{mis_k,k} - \hat{Y}^{(r_k)}_{mis_k,k}\|_1}{\|Y_{mis_k,k}\|_1} < \epsilon$, the summation of absolute residuals is controlled by ϵ and the original observed values. On the other hand, the residual is $Y_{mis_k,k} - \hat{Y}^{(\text{maxit})}_{mis_k,k}$. Thus, the model fitting can be assessed with the observed versus predicted values diagnostic plot. First, we subset the obfuscated cells. Then, we plot the observed values in the vertical axis and the predicted values on the horizontal axis. When the two values are only different by a small error term, the diagnostic plot shapes like a diagonal line. If the presence of significant outliers is detected, we may consider alternative strategies to remedy these atypical cases, e.g., removing outliers from the final shareable dataset to better protect the data utility.

2.3.3 Simulation Studies

2.3.3.1 Simulation Setup

In this section, we conduct controlled simulation studies to evaluate the data privacy and utility protection of DataSifter and multiple imputation methods. The original simulation data is generated with $n = 500,$ or $n = 1,000$ subjects, each with J_i time points, where J_i varies from 1 to 10 with equal probability, two longitudinal variables (Y_1 and Y_2), five static independent true predictors ($X_1, X_2..., X_5$), and $w = 5$ or $w = 20$ white noise variables. The static true predictors are generated by normal distributions with different means and unit variance. The white noise variables are also generated by normal distributions, but with a different set of means and larger variances. The longitudinal variable Y_1 is associated with static variables only (X_1, X_2, X_3) and Y_2 is associated with both static (X_4, X_5) and longitudinal (Y_1) variables. We consider linear and non-linear associations when generating Y_1 and Y_2.

Under linear association, Y_1 is generated by the following Linear Mixed Model:

$$Y_{i,j,1} = 1 - X_{1,i} - 0.5X_{2,i} - 0.3X_{3,i} + 0.8Visit_{i,j} + b_{0i} + \epsilon_{i,j},$$

where $i = 1, ..., n$ is the indicator for patients, and $j = 1, .., J_i$ is the indicator for time. Here J_i is the total visit number for patient i and $J_i \in \{1, 2, ..., 10\}$, b_{0i} is a subject specific random intercept that follows a $N(0, 1)$ distribution, and $\epsilon_{i,j}$ are independent for different time points and follows $N(0, 4)$. Variable Y_2 depends on two static variables and Y_1.

$$Y_{i,j,2} = -15 + 0.2Y_{i,j,1} - X_{4,i} + 0.2X_{5,i} + 2Visit_{i,j} + b_{1i} + \epsilon_{i,j}.$$

Similarly $b_{1i} \sim N(0, 1)$ and $\epsilon_{i,j} \sim N(0, 4)$. We know that under random intercept $V(Y_{i,\cdot,1}) = Z_i D Z_i^T + \sigma^2 I_{J_i}$ where $Z_i = 1_{J_i \times 1}$, $D = Var(b_{0i})$ and $\sigma^2 = Var(\epsilon_{i.j})$.

Thus, $Cov(Y_{i,j,1}, Y_{i,j',1}) = Var(b_{0i})$ and $Corr(Y_{i,j,1}, Y_{i,j',1}) = \frac{Cov(Y_{i,j,1}, Y_{i,j',1})}{\sqrt{Var(Y_{i,j,1})Var(Y_{i,j',1})}} = \frac{Var(b_{0i})}{Var(\epsilon_{i,j})+Var(b_{0i})}$. After some calculations, $Corr(Y_{i,j,1}, Y_{i,j',1}) = 0.2$ for all i and $j \neq j'$. Similarly, $Corr(Y_{i,j,2}, Y_{i,j',2}) = 0.2$.

We also consider cases with non-linear relationships. Similar to the linear setting, we construct models with compound symmetry correlation structure. Our two longitudinal variables are derived by:

$$Y_{i,j,1} = 10 + 3\sin(X_{1,i}) - 0.2X_{2,i}^2 - 0.1X_{1,i} \cdot |X_{3,i}| + Visit_{i,j} + b'_{0i} + \epsilon'_{i,j},$$

and

$$Y_{i,j,2} = 2 + 0.05\sin(Y_{i,j,1}) + 0.4\exp\{\cos(X_{4,i})\} - 0.02Y_{i,j,1} \cdot |X_{5,i}| + 2Visit_{i,j} + b'_{1,i} + \epsilon'_{i,j},$$

where $b'_{0i} \sim N(0, 9)$, $b'_{1i} \sim N(0, 16)$, and $\epsilon'_{i,j} \sim N(0, 64)$. We have $Var(Y_{i,j,1}) = 73$, $Corr(Y_{i,j,1}, Y_{i,j',1}) = 0.12$, $Var(Y_{i,j,2}) = 80$, and $Corr(Y_{i,j,2}, Y_{i,j',2}) = 0.2$, where $j \neq j'$.

The complete data generated by the above procedure will be used to examine different data obfuscation methods. To mimic real-world data, we also consider a scenario where some observations in Y_1 and Y_2 contains missing values, which follow the MAR missing data mechanism. First we define the missing indicator for variable Y_1 to be $M(Y_{i,j,1}) = I(Y_{i,j,1} = NA)$, $\forall i \in \{1, ..., n\}$ and $j \in \{1, ..., J_i\}$ and similarly for Y_2. The original missingness $M(Y_{i,j,1})$ and $M(Y_{i,j,2})$ is generated from the two sets of logistic regressions. The first set considers different probabilities of missingness at individual and time level. Under this model, we allow partially missing subjects.

$$logit[P\{M(Y_{i,j,1}) = 1\}] = -2 + 10X_{1,i} - 10Visit_{i,j} + b_i.$$
$$logit[P\{M(Y_{i,j,2}) = 1\}] = 3 - 4X_{5,i} - 12Visit_{i,j} + b'_i.$$

Here $b_i, b_i' \sim N(0,1)$. The two models will provide around 20-30% missingness for each longitudinal variable.

We compare the performance of four types of synthetic datasets: DataSifter with GLMM generated on complete original data, DataSifter with RE-EM tree generated on complete original data, multiple imputed synthetic data generated on complete original data, and DataSifter with RE-EM tree generated on original data that contains missing. All the *sifted* data are generated without static data obfuscation (L =no obfuscation). We applied the multiple imputation method using two-level normal models with homogeneous within group variances as the imputation model to create multiply imputed partially synthetic datasets, which is implemented in *mice* R package [74, 6]. We compare the first three types of synthetic data to assess the obfuscation performance. To demonstrate that DataSifter can successfully handle missingness in the original data, we further show that the decrease in data utility preservation is small after bring in original missingness under the RE-EM tree imputation method. All synthetic data are generated by randomly introducing 20% artificial missingness in Y_1 and Y_2 and impute the missing cells back. Static variables including the white noise variables are not obfuscated. One hundred replications are constructed for each type of synthetic datasets under every simulation setting.

2.3.3.2 Simulation Results

Using the proposed data utility and data privacy measurement, we evaluate synthetic datasets generated by DataSifter and MI. Data utility is measured in terms of prediction accuracy and inference based on models trained on synthetic datasets. For prediction accuracy, we construct test sets with identical sample size as the training dataset ($n = 500$ or $n = 1,000$ and $J_i \in \{1, ..., 10\}$). We then use the predictive models constructed on the synthetic datasets to predict the target longitudinal variables in the test set. Absolute deviance in predicted and observed values of Y_1 and

Y_2 are calculated as the prediction error. Model inference is measured by the 95% confidence interval coverage of the true parameter value among the 100 replications under linear association scenario. Data utility is examined using the average privacy measurement (PM) for the first 100 records of Y_1 and Y_2. As defined in section 2.3.1.2, for an obfuscated (replaced) target cell $Y_{i,j,k}$, we use the conditional model fitted by $\mathbf{D}_{-(i,j)}$ to represent intruder's prior knowledge and $PM_{i,j,k}$ is the difference between the conditional mean and the observed $y_{i,j,k}$.

The average prediction errors on test data are summarized in **Table** 2.2. Based on 100 replications, under most simulation settings, the average test error on Y_1 and Y_2 are similar across different synthetic data generation methods and these results are indistinguishable from the original data. For Y_2 under linear association, the MI method provides a slightly better prediction results of less than 15% improvement. This indicates the parameter estimations with synthetic data generated by DataSifter are relatively accurate. Note that whether or not we have missingness in the original datasets, the DataSifter RE-EM tree provides similarly accurate coefficient estimates. Moreover, stable results are observed under both linear and nonlinear associations, training sample sizes and noise levels, which suggests that our proposed imputation method is robust.

Since the proposed imputation method is aimed at minimizing imputation error rather than accounting for uncertainty of the missing values, the 95% confidence interval constructed on *sifted* datasets are narrower than the original data. As shown in **Table** 2.3, for variable Y_1, the MI method achieves desired 95% coverage while the DataSifter GLMM achieves 89-98% accuracy. Due to the slower convergence rate of non-parametric estimations and non-linear model specification, the synthetic datasets generated by DataSifter RE-EM have smaller CI coverage ranging from 76-94%. For Y_2 , the CI coverage for X_4 (one of the static predictors) is relatively small under the DataSifter methods (43- 68%). Nevertheless, we observe that the DataSifter GLMM

Training sample	n = 500							
Variable	Y_1				Y_2			
Association, Noise level	Linear, w = 5	Linear, w = 20	Nonlinear, w = 5	Nonlinear, w = 20	Linear, w = 5	Linear, w = 20	Nonlinear, w = 5	Nonlinear, w = 20
Original	1.858	1.858	20.471	20.471	1.903	1.903	7.649	7.649
Multiple Imputation	1.851	1.867	20.295	20.289	1.903	1.922	7.693	7.818
DataSifter GLMM	1.903	1.901	20.538	20.525	2.151	2.144	7.949	7.949
DataSifter RE-EM	1.896	1.896	20.503	20.516	2.153	2.149	7.705	7.710
DataSifter RE-EM with original missing	1.871	1.873	20.121	20.101	2.184	2.205	7.728	7.730
Training sample	n = 1,000							
Variable	Y_1				Y_2			
Association, Noise level	Linear, w = 5	Linear, w = 20	Nonlinear, w = 5	Nonlinear, w = 20	Linear, w = 5	Linear, w = 20	Nonlinear, w = 5	Nonlinear, w = 20
Original	1.870	1.870	20.746	20.746	1.861	1.861	7.410	7.410
Multiple Imputation	1.863	1.883	20.563	20.596	1.863	1.891	7.420	7.520
DataSifter GLMM	1.911	1.911	20.817	20.801	2.100	2.111	7.738	7.741
DataSifter RE-EM	1.911	1.914	20.806	20.797	2.109	2.122	7.456	7.459
DataSifter RE-EM with original missing	1.864	1.864	19.985	19.998	2.245	2.248	7.729	7.732

Table 2.2: Mean absolute deviation (prediction error) for test dataset based on model fitted on original and synthetic datasets. The test datasets are generated separately with the same sample size as the training sets.

method (87% and 84%) provide a much better CI coverage for predictor Y_1 compared to the MI method (39% and 21%). This is because the proposed iterative imputation method updates the missing predictor information during each iteration while MI is solely based on complete data. The CI coverage based on the DataSifter RE-EM tree method is smaller when the original data contains missing values. However, the reduction in CI coverage is alleviated for a larger sample size.

The average privacy measurement for different synthetic datasets is illustrated in **Figure** 2.7. Each boxplot records the distribution of average privacy measurement for Y_1 and Y_2 over 100 replications among the first 100 records. Under all scenarios, the two DataSifter methods have similar privacy measurement values and distribu-

Training sample	n = 500											
Variable	Y_1						Y_2					
Association, Noise level	Linear, w = 5			Linear, w = 20			Linear, w = 5			Linear, w = 20		
Covariate	X_1	X_2	X_3	X_1	X_2	X_3	X_4	X_5	Y_1	X_4	X_5	Y_1
Original	0.940	0.950	0.960	0.940	0.950	0.960	0.970	1.000	0.960	0.970	1.000	0.960
Multiple Imputation	0.940	0.970	0.950	1.000	1.000	1.000	0.960	1.000	0.860	0.980	1.000	0.390
DataSifter GLMM	0.900	0.920	0.930	0.940	0.950	0.930	0.680	0.930	0.890	0.590	0.940	0.870
DataSifter RE-EM	0.870	0.860	0.870	0.900	0.940	0.860	0.670	0.850	0.740	0.600	0.850	0.720
DataSifter RE-EM with original missing	0.690	0.850	0.690	0.660	0.840	0.700	0.490	0.840	0.760	0.440	0.880	0.730
Training sample	n = 1,000											
Variable	Y_1						Y_2					
Association, Noise level	Linear, w = 5			Linear, w = 20			Linear, w = 5			Linear, w = 20		
Covariate	X_1	X_2	X_3	X_1	X_2	X_3	X_4	X_5	Y_1	X_4	X_5	Y_1
Original	0.950	0.960	0.960	0.950	0.960	0.960	0.960	0.970	0.880	0.960	0.970	0.880
Multiple Imputation	0.960	0.980	0.960	0.990	1.000	1.000	0.960	0.940	0.660	0.990	0.990	0.210
DataSifter GLMM	0.930	0.960	0.960	0.890	0.980	0.960	0.430	0.860	0.790	0.440	0.910	0.840
DataSifter RE-EM	0.930	0.890	0.760	0.880	0.940	0.770	0.440	0.730	0.510	0.430	0.740	0.550
DataSifter RE-EM with original missing	0.730	0.850	0.760	0.730	0.840	0.770	0.260	0.870	0.520	0.270	0.840	0.530

Table 2.3: Confidence interval (95%) coverage based on 100 replicates under different models. The coverage records the percent of times that CIs from models trained on original and synthetic datasets cover the true parameter estimate.

tions. The MI method offers significantly lower privacy measurement, see **Figure** 2.7. In fact, when introducing 20% artificial missingness to the longitudinal variables, the average mean privacy measurement is around 5.25 times higher in *sifted* datasets compared to multiply imputed datasets. This result provides empirical evidence for the privacy measurement (PM) improvement derivation, see Section 2.3.1.2. Compared to multiple imputed datasets, *sifted* datasets have at least $1/a$ times higher average PM, where a is the percent of the introduced artificial missing values in the data.

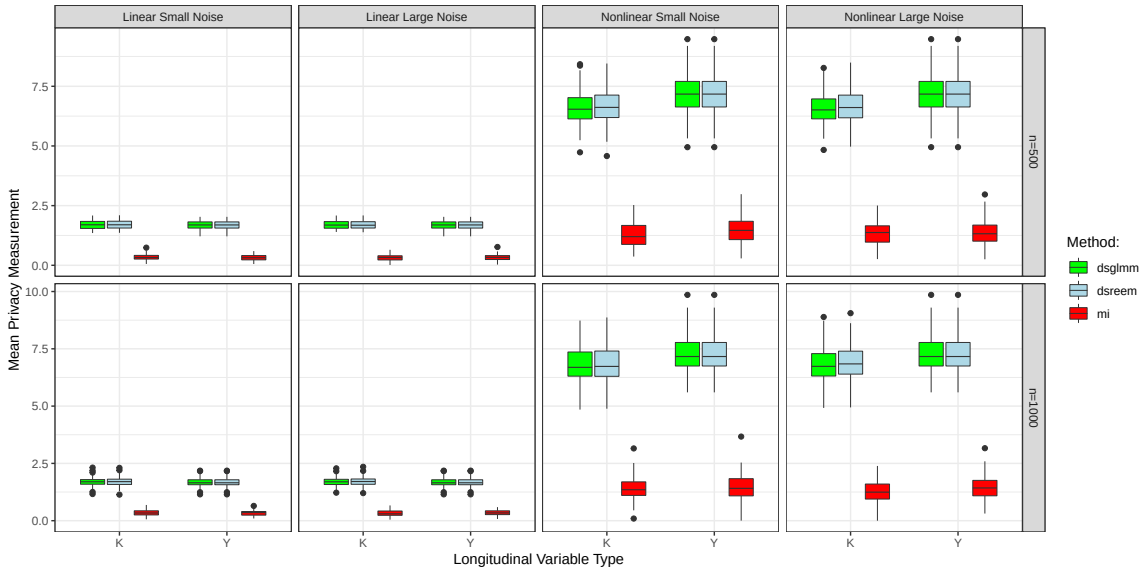

Figure 2.7: Average privacy measurement among first 100 rows in the synthetic datasets. The scenario with small noise level contains $w = 5$ and large noise level contains $w = 20$ white noise variables.

2.3.4 Clinical Data Application using MIMIC-III

The Medical Information Mart for Intensive Care III (MIMIC-III) represents a sizable single-center database that provides patients' medical records in a large tertiary care hospital between 2001 and 2012. MIMIC-III data stores information related to patients' admission, including vital signs, medications, laboratory measurements, length of stay, survival data, and more [34]. We consider a subset of 7,080 patients who had at least two visits to the hospital who contributed 17,594 hospital admission records with demographic variables, including insurance type, gender, race, age, marital status, and death after admission. Admission information such as insurance type, admission type, and month of admission is also available. MIMIC-III contains de-identified or coded data that is considered not involving protected health information. However, the data request process including taking an online course and submitting an application with specific research topics and requested information can still take more than three weeks. Using the data for any rigorous scientific investigation requires the researcher to go through a time consuming data request procedure, while

at the end the investigation may find no significant results. DataSifter II allows a quicker turnaround for checking the potentials of research hypotheses. For example, we want to investigate the association between length of stay in tertiary care hospitals and Medicaid insurance type controlling other patient demographic variables using the MIMIC-III data. We illustrate how to use the *sifted* data to answer our initial research question, and evaluate both utility and privacy protection performance in the *sifted* MIMIC-III data. We also compare the synthetic MIMIC-III datasets generated by DataSifter II with that of the multiple imputation method. A linear mixed effect model is used to regress the length of hospital stay on patient characteristics. The privacy protection effort is measured by the *privacy measurement* (PM) for age and length of hospital stay among the first 100 records.

First, we obfuscate the following longitudinal variables: (1) length of stay, (2) month of admission, (3) death after visit, (4) age at visit. We consider generating two types of Siftered data: with (L =medium) or without (L =no obfuscation) static data obfuscation using DataSifter I. By using the DataSifter II protocol, we introduce 20% missingness in the longitudinal variables specified above to obtain the first type of *sifted* data without static obfuscation. The RE-EM tree model is used as the imputation model because of its more flexible mean structure. Then, we generate the second type of *sifted* data with further obfuscation on the static variables using DataSifter I under the medium level of obfuscation, which entails two rounds of artificial missing introduction and imputation, each one randomly obfuscating 25% of the cells. The other setting for the medium obfuscation level defines neighbors as the cases with the closest top 5% distance and swap 60% of the features with a neighboring case. As a comparison, we also create partially synthetic data with multiple imputation (MI) based on 20% random artificial missingness on the four longitudinal variables. In each replication, the MI dataset and the Siftered dataset without static obfuscation have the same amount of data cells being replaced. The

Sifterd dataset with static obfuscation has the highest level of privacy protection among the three by altering an extra 25% of the cells in the static data. Fifty replications are generated for each type of partially synthetic data.

Next we compare the model parameter estimates between models fitted on the original data and on the three different types of synthetic data, assuming the following linear mixed effect model:

$$\begin{aligned}\text{Length of stay}_{i,j} =& \beta_0 + \beta_1 \text{Age} + \beta_2 \text{Medicaid} + \beta_3 \text{Private insurance} + \beta_4 \text{White} + \beta_5 \text{Black} \\ &+ \beta_6 \text{Male} + \beta_7 \text{Emergency admission} + \beta_8 \text{Urgent admission} \\ &+ \beta_9 \text{Single} + \beta_{10} \text{English language} \\ &+ \beta_{11} \text{Visit}_{i,j} + b_{1i} \text{Visit}_{i,j} + \epsilon_{i,j},\end{aligned}$$

where $b_{1i} \sim N(0, \sigma^2_{b1})$, and $\epsilon_{i,j} \sim N(0, \sigma^2)$.

Results in **Figure** 2.8 show that the DataSifter II provides much better privacy protection than the MI method with a small loss in data utility. Medicaid is not associated with the length of stay in any synthetic and original data fitted models. According to **Figure** 2.8 **A**, most of the mean PM by row (record) are below five among the 50 multiply imputed synthetic datasets. The average PM is 0.33 for age at admission and 1.53 for length of hospital stay. The DataSifter method provides a significant improvement in terms of PM with 14.87 and 8.17 as the average PMs for age and hospital stay, respectively. We can also infer from **Figure** 2.8 **A** that the mean PMs can vary considerably from row to row in *sifted* datasets without static obfuscation.

Figure 2.8 (B) illustrates the deviance of parameter estimates between the model fitted with three types of synthetic dataset and the original linear mixed model. Only significant parameter estimates are shown in the plot. The box plots represent the empirical confidence intervals (CIs) or the distribution of $\hat{\beta}$'s among 50 replicates. The black dots and purple intervals illustrate the coefficient estimates and CIs from

the original model. According to **Figure** 2.8 **B**, all the empirical CIs from MI and DataSifter without static obfuscation overlap with the CIs acquired from original data. The MI created synthetic datasets provide the most accurate $\hat{\beta}$'s that align closely with the original estimates and a small estimation bias is observed for the *sifted* data without static obfuscation. Five out of seven empirical CIs from the model constructed from the *sifted* data with static obfuscation have overlapped with the original CIs. The results suggest that the data utility is well preserved in *sifted* datasets after intensive obfuscation.

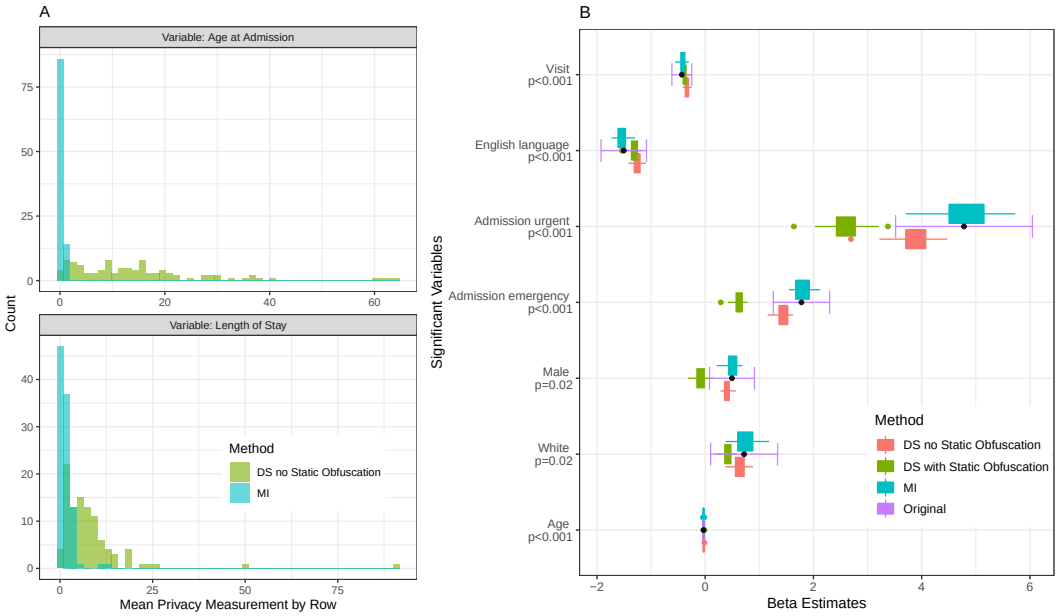

Figure 2.8: MIMIC III synthetic data privacy (**A**) and utility (**B**) evaluation. Plot **A** summaries the distribution of mean privacy measurement for age and length of hospital stay for first 100 rows across 50 synthetic datasets generated by DataSifter (without static obfuscation using DataSifter I) and multiple imputation. Plot **B** compares the significant coefficient estimates (p-value < 0.05) among the models fitted with original data, and synthetic datasets generated by DataSifter II (with or without static obfuscation) and multiple imputation. The boxes illustrate the distribution of coefficients estimated on 50 synthetic datasets. The black dots and purple intervals are the parameter estimates and confidence intervals from the linear mixed model fitted by the original dataset.

Since none of the fitted models obtained from the *sifted* datasets shows significance of the effect β_2, researchers who are interested in the relationship between Medicaid

and length of hospital stay may conclude "no statistical association" from any one of the *sifted* datasets presented in the simulation study.

2.4 Discussion

The results shown above illustrate that the DataSifter technique balances between maintaining the energy of the original data (preserves information utility) while simultaneously introducing a level of privacy protection safeguarding against re-identification of sensitive information contained in the archive.

According to the reported simulation studies, under a careful set-up for user-defined privacy levels, DataSifter I can successfully provide privacy protection while maintaining data utility. The clear negative relationship between the level of obfuscation and the proportion of PIFVs indicates that a high user-specified privacy level does provide increased privacy protection for sensitive information. Using DataSifter under "large" or "indep" settings, patient privacy was highly protected. Data re-identification was almost impossible by stratification filtering of the targeted patients via known feature values. This is due to the method's inability to distinguish between real, imputed, or obfuscated values within each real feature, and the relatively small proportion of untouched data elements. Of course, caution needs to be exercised, as multiple queries resulting in repeated "sifted" data instances may expose the overlapping "true" values especially for low levels of obfuscation. However, the large proportion of "sifted" elements protects sensitive information and may allow data users to request a small number of data queries. The application of DataSifter I on ABIDE provided a realistic demonstration of how to employ the proposed algorithm on EHR. Also, the application confirmed DataSifter I's ability to handle high dimensional data. The excellent prediction performances on the "medium" obfuscation level suggested similar data utility between original and "sifted" data.

For DataSifter II, The simulation results based on introducing 20% artificial

missingeness suggest that data utility is better preserved for longitudinal variables that depend only on static variables (Y_1) compared to variables that depend on both static and longitudinal variables (Y_2). The two imputation method options, GLMM and RE-EM tree, provide accurate and computationally efficient imputations for Y_1 and Y_2 under the both linear and nonlinear generative models. The RE-EM tree method is efficient computationally when the number of longitudinal variables is large, and the number of subjects is small compared to the number of variables in the data.

Compared to the multiple imputation method, DataSifter II provides extended privacy protection with moderate utility loss in terms of CI coverage. We showed that the synthetic datasets generated by DataSifter are more powerful for predictive model constructions rather than inferential tasks by design to seek for the smallest imputation error in each obfuscated cell.

We examined the privacy protections of the longitudinal synthetic data proportion using a proposed privacy measurement. The original DataSifter I method can further ensure that partial information in the time-invariant cross-sectional data is obfuscated for any non-isolated subject and our application shows that this extra level of protection only introduces a small bias in model-based statistical inference. When complete obfuscation for all data elements is required, we recommend the generation of fully synthetic longitudinally *sifted* datasets. This new option is implemented in the DataSifter II R package. It is specified by the setting of level-of-obfuscation parameter, which is also available in the DataSifter I technique. We define a local value range $(a_{i,j}, b_{i,j})$ for each longitudinal variable, individual i, time j, and as the closest-neighborhood N of nearby observations. Then, we randomly sample with a uniform distribution bounded by $a_{i,j}$ and $b_{i,j}$ to generate each data point. The fully synthetic version of DataSifter II introduces errors that are locally bounded. However, it does not guarantee preservation of within-subject correlations.

The experimental results show that the sifted datasets preserve the information content of the original data (utility preservation) at the same time they provide powerful data privacy protections. Depending on the specific data characteristics, the DataSifter II performance may be impacted in terms of its efficiency and balance between privacy-protection and utility-reservation. We employed linear (GLMM) model and tree (RE-EM tree) structure to approximate the distribution for each longitudinal variable conditional on other variables in the data. The utility preservation for each longitudinal variable may be affected by (1) the complexity of the relationship, (2) empirical variance of the target time-varying variable, (3) the data type of the predictors, and (4) alternative within-subject covariance structures.

The DataSifter algorithm provides data governors and researchers with a semi-automated and reliable framework for sensible information exchange. Despite perturbing individual level records, the overall *sifted* time-varying data shares similar population level information with the original process. DataSifting allows owners of data, business managers, biomedical scientists, and clinical researchers to create and study pseudo populations by sharing obfuscated data objects, instead of the original sensitive information, and conducting studies on these *sifted* datasets. This information exchange process promotes rapid and effective testing of *a priori* research hypotheses (confirmatory analytics), as well as, data-driven discovery science and formulation of novel translational science questions (exploratory analytics). Many biomedical areas, clinical settings, and research and development partnerships may benefit from the DataSifter technology to conduct advanced trans-disciplinary research and translate basic science advances into clinical practice.

CHAPTER III

Robust Estimation for Viable Optimal DTRs with Restricted Arms Using Observational Data

3.1 Introduction

Personalized medicine (PM) or precision medicine is an increasingly popular area of study including a broad range of approaches, which provide individualized solutions for drug therapy or preventive care. PM is an important research topic since significantly different treatment effects of certain medical procedures can be observed from patient to patient. To make single-stage treatment recommendations, traditional PM approaches cluster patients into groups based on their cross-sectional demographics or genetic information. Dynamic treatment regime (DTR) is an effective vehicle for personalized medicine under the time-varying treatment settings of chronic conditions like drug abuse, cancer, and diabetes, where adaptive treatments are necessary for patients at multiple stages [10]. To make effective treatment decisions at each stage, DTR uses patients' past medical history and current disease status to offer the most informed individualized treatment recommendation for chronic diseases. Nevertheless, making sequential decisions can be challenging due to the intricate causal relationships between a sequence of treatments and the final clinical outcome.

Various methods have been developed for estimating optimal DTRs. Marginal

structural models with inverse probability weighting (IPW) [51, 90], Q- and A-learning methods [93, 82, 50, 76], and dynamical system models [61] stand out as the most commonly used parametric methods for estimating optimal DTR. These methods are highly interpretable but rely heavily on parametric assumptions, which however could be violated in some scenarios. For example, we may have too many covariates or limited information to specify reasonable parametric models. Over the past few years, nonparametric approaches have been proposed to allow more flexibilities and relax assumption restrictions. Among such non-parametric approaches, tree-based methods not only relax the linear model assumption but also facilitate the easy interpretability for physicians to understand and use in practice. Laber and Zhao (2015) [39] employed an inverse probability weighted purity measure to build decision trees for optimizing the potential health outcomes for a single stage, where the patient individual medical history are used to tailor the treatment as potential tree nodes. Tao and Wang (2018) [85] generalized the method to a doubly robust approach, which provides more robustness using observational data and can also handle multi-stage treatment situations. Cut-off based verdicts in the tree structure can provide clear guidance to future clinical practices.

When answering the inferential questions in DTR researches, pre-designed Sequential Multiple Assignment Randomized Trials (SMART) are ideal to meet the desirable casual assumptions. Although SMART were a growing data source for comparing or constructing DTR, it is still not easy to perform and implement SMART in practice since such trials can be expensive and time-consuming. As we enter the big data era, massive amounts of health records are available for scientific discoveries and observational studies have their own merits and values for evaluating the optimal DTRs. There has been a large literature to develop statistical methods for DTR in observational studies [39, 84, 85]. However, a natural challenge arises since some observed treatment routes with good-targeted outcomes can be inapplicable or not

recommended for future patients. Possible sources of the inapplicable routes include drug recalls and inconsistent treatment applications from different physicians. For example, it is not meaningful to construct a DTR that contains recalled drugs in a certain stage or has a weaker medicine following a more aggressive treatment. Although there is a vast literature for estimating optimal DTR, directly applying the existing DTR methods to data with inapplicable treatment routes can lead to unrealistic recommendations for some patients. On the other hand, simply deleting the patient records with restricted arms is not a solution either since that may induce selection bias and interpretable DTR is not guaranteed. Therefore, a growing number of researchers and clinicians are calling for new methods that solve such constrained optimization problem in observational studies.

In this chapter, we develop the Restricted Tree-based Reinforcement Learning (RT-RL) method to accommodate restrictions in observational studies by truncating possible treatment options based on patient history in a multi-stage multi-treatment setting. The proposed method provides robust and interpretable estimations by employing doubly robust augmented inverse probability weighted estimators in a tree-based algorithm. Our algorithm provides optimal treatment recommendations for patients regarding only applicable treatment options and utilizes all valid observations in the dataset to avoid selection bias and improve efficiency.

The rest of the paper is organized as follows: In section 3.2 we formulize the restricted optimization problem and our proposed solution by specifying the truncation process for the patient history and treatment options and describing the Restricted T-RL estimating procedure. In section 3.3, using simulation studies, we demonstrate the interpretability, effectiveness, and robust performances of our method by comparing RT-RL with the naive method that deletes all the information in the restricted arms. In section 3.4, we apply the algorithm to Global Appraisal for Individual Needs (GAIN) to estimate interpretable two-stage DTR to guide the level of care placement

for adolescents with substance use disorder. The performance of our proposed method is compared with the native T-RL and non-dynamic regimes. Finally, in section 3.5 we summarize and discuss the findings.

3.2 Method

3.2.1 Notations and Setup

We consider an observational study with n patients and T treatment stages where there are $K_j\,(K_j \geq 2)$ potential treatment options at the j^{th} treatment stage, $j = 1, ..., T$. Patients are observed to follow one of the treatments available at each stage. For brevity, we suppress the patient index $i\,(i = 1, .., n)$ when no confusion exists. Let A_j denote the treatment at the j^{th} treatment stage, where A_j may take a value a_j, a specific value for the j^{th} treatment assignment that belongs to the stage specific treatment space $\mathcal{A}_j = \{1, ..., K_j\}$. Let $\overline{\mathbf{A}}_T \equiv (A_1, \ldots, A_T)$ with a bar "-" denote the T-stage sequence of treatment indicators. Similarly, we denote the observed treatment routes with $\overline{a}_T \equiv (a_1, \ldots, a_T)$. As a general notation in the following, we use a bar "-" and a subscript j on a variable to denote the history of this variable up to the j^{th} stage. For example, $\overline{\mathbf{A}}_{j-1} \equiv (A_1, \ldots, A_{j-1})$ to denote the past treatments prior to stage j. Let R_j denote the clinical outcome observed following A_j, also known as rewards, which depends on the precedent patient characteristics $\mathbf{X}_j$ and previous treatments $\overline{\mathbf{A}}_{j-1}$. We consider the overall outcome of interest as some functional of the reward history such that $Y \equiv f(R_1, ..., R_T)$, where $f(.)$ is a pre-specified function (e.g., sum), assuming Y is bounded and preferable with larger values. To recommend personalized optimal treatment for future patients at stage j, we infer from the observed Y, the current candidate treatments $a_j \in \mathcal{A}_j$ and the patient history $\mathbf{H}_j = (\overline{\mathbf{A}}_{j-1}, \mathbf{X}_j^T)^T \in \mathcal{H}_j$.

However, some of the observed treatment sequences are not applicable for future

patients due to clinical or practical reasons. We define such treatment routes as restricted treatment arms, which should be excluded from the domain of the DTRs of interest during estimation. Suppose there are M_j restricted arms at stage j, and denote them as $\overline{\mathbf{b}}_{m,j} = (b_{1,m,j}, ..., b_{j,m,j})$ where $m = 1, ..., M_j$ and $b_{j',m,j} \in \mathcal{A}_{j'}$ for $j' = 1, ..., j$. Excluding restricted treatment arms from the observed treatment stages, we obtain a set of viable treatment sequences from stage 1 to stage j denoted as $\overline{\mathcal{A}}_j^{res} \equiv \overline{\mathcal{A}}_j / \{\overline{\mathbf{b}}_{m,j}, m = 1, ..., M_j\}$. Correspondingly, the domain of viable patient history is $\mathcal{H}_j^{res} = \{\mathbf{h}_j : \overline{\mathbf{a}}_{j-1} \in \overline{\mathcal{A}}_{j-1}^{res}\}$. In the rest of the paper, we denote the random variable of viable treatment history as $\mathbf{H}_j^{res}$, and the value of $\mathbf{H}_j^{res}$ is in $\mathcal{H}_j^{res}$.

We use $g^{res} = (g_1^{res}, ..., g_T^{res})$ to denote a sequence of restricted viable rules for personalized treatment decisions across the different treatment stages (total T stages), i.e., a viable restricted dynamic treatment regime (DTR), where g_j^{res} is a meaningful mapping function from the domain of restricted patient history $\mathcal{H}_j^{res}$ to the range of viable treatment options at j given the past ($\mathcal{A}_j^{res}$). Denote the collection of such restricted meaningful mappings as $\mathcal{G}^{\text{res}} = (\mathcal{G}_1^{\text{res}}, ..., \mathcal{G}_T^{\text{res}})$. Specifically, $g_j^{res} \in \mathcal{G}_j^{\text{res}}$ maps from viable treatment histories to applicable stage j treatments conditional on $\overline{A}_{j-1}$ such that $(\overline{\mathbf{A}}_{j-1}, g_j^{res}(\mathbf{H}_j^{res})) \in \overline{\mathcal{A}}_j^{res}$.

To identify the optimal restricted DTR among $\mathcal{G}^{res}$, we consider the counterfactual framework for causal inference defined in Robins, 1986. At stage T, let $Y^*(A_1, ..., A_{T-1}, a^T)$ or $Y^*(a^T)$ denote the counterfactual outcome for patients received treatment a_T conditional on previous treatments. Our goal is to find the optimal one among the restricted DTRs in $\mathcal{G}_T^{\text{res}}$ that maximizes the expected counterfactual outcome, i.e.,

$$g_T^{\text{res,opt}} = \underset{g_T^{res}(\mathbf{H}_T^{res}) \in \mathcal{G}_T^{res}}{argmax} E\left[\sum_{a_T=1}^{K_T} Y^*(a_T) I\{g_T^{res}(\mathbf{H}_T^{res}) = a_T\}\right].$$

For any stage j before T, we seek the best regime by maximizing the expected coun-

terfactual outcome with all future treatments optimized to avoid confounding. We denote the counterfactual outcome at stage j given $\overline{\mathbf{A}}_{j-1}$ and future optimal treatments $g_{j+1}^{\text{res, opt}}, \ldots, g_T^{\text{res, opt}}$ as $Y^*\left(\overline{\mathbf{A}}_{j-1}, a_j, g_{j+1}^{\text{res, opt}}, \ldots, g_T^{\text{res, opt}}\right)$. Since such counterfactual outcomes cannot be observed, we estimate the stage-wise pseudo-outcome with observed data considering viable constricted treatment space by

$$PO_j^{res} = \hat{E}^{res}\left\{Y^*\left(A_1, \ldots, A_j, g_{j+1}^{\text{res, opt}}, \ldots, g_T^{\text{res, opt}}\right)\right\},$$

at stage j for $j = 1, ..., T-1$, where E^{res} denotes the expectation considering only the viable patient history and treatment routes space, $g_j^{res,opt} \in \mathcal{G}_j^{res}$ and $(A_1, ..., A_j) \in \overline{\mathcal{A}}_j^{res}$, which is equivalent to the recursive form

$$PO_j^{res} = \hat{E}_{\mathbf{H}^{res}}\left\{PO_{j+1}^{res} \middle| A_{j+1} = g_{j+1}^{\text{res,opt}}\left(\mathbf{H}_{j+1}^{res}\right), \mathbf{H}_{j+1}^{res}\right\}.$$

At the final stage, $PO_T^{res} = Y$. Let $\text{PO}_j^{res,*}(\overline{A}_{j-1}, a_j)$, or $\text{PO}_j^{res,*}(a_j)$ for brevity, denote the counterfactual pseudo-outcome for a patient treated with $a_j \in \mathcal{A}_j$ and viable past treatments $(A_1, ..., A_{j-1}) \in \overline{\mathcal{A}}_{j-1}^{res}$. We have

$$\text{PO}_j^{res,*}(g_j^{res}) = \sum_{a_j=1}^{K_j} \text{PO}_j^{res,*}(a_j) I\{g_j^{res}(\mathbf{H}_j^{res}) = a_j\}.$$

Our optimization problem at stage $j\,(j < T)$ among the meaningful mappings becomes:

$$g_j^{\text{res,opt}} = \underset{g_j(\mathbf{H}_j^{res}) \in \mathcal{G}_j^{res}}{argmax}\ E[\text{PO}_j^{res,*}(g_j^{res})].$$

3.2.2 Constrained Optimization Procedure

To connect the counterfactual outcomes and counterfactual pseudo-outcomes with observed data, we make the following three assumptions considering only the applicable treatment routes [51, 66, 53]:

(1) **Consistency**. If the patient were given the treatments accordingly, the observed outcome would be the same as the counterfactual outcome, which indicates $Y = \sum_{a_T=1}^{K_T} Y^*(a_T) I(A_T = a_T)$ for stage T, where $A_T \in \mathcal{A}_T^{res}$. Similarly, the estimated pseudo outcome agrees with the counterfactual pseudo-outcome $PO_j^{res} = \sum_{a_j=1}^{K_j} \mathrm{PO}_j^{res,*}(a_j) I\{A_j = a_j\}$ for stage $j < T$ and $A_j \in \mathcal{A}_j^{res}$.

(2) **No unmeasured confounding**. For any treatment sequence $\overline{a}_T = (a_1, .., a_T)$, treatment A_j is independent of future outcomes (rewards), given $\mathbf{H}_j^{res}$ a random variable that takes value in $\mathcal{H}_j^{res}$, i.e.,

$$A_j \perp (R_j(\overline{a}_j), ..., R_T(\overline{a}_T)) | \mathbf{H}_j^{res}, \forall j = 1, ..., T.$$

(3) **Positivity**. There exists constants $0 < c_0 < c_1 < 1$ such that with probability 1 the propensity score

$$\pi_{a_j}(\mathbf{H}_j^{res}) = Pr(A_j = a_j | \mathbf{H}_j^{res}) \in (c_0, c_1).$$

With the three assumptions, we can bridge the pseudo outcome estimated from the observational data with the expected counterfactual pseudo outcome for a specific regime $g_j^{res} \in \mathcal{G}_j^{res}$ at any stage $j < T$. Conditional on $\mathbf{H}_j^{res}$, we have

$$E\{\mathrm{PO}_j^{res,*}(g_j^{res})\} = E_{\mathbf{H}_j^{res}} \left[\sum_{a_j=1}^{K_j} E\{\mathrm{PO}_j^{res,*}(a_j) | \mathbf{H}_j^{res}\} I\{g_j^{res}(\mathbf{H}_j^{res}) = a_j\} \right].$$

With no unmeasured confounder assumption, it is easy to show that

$$E\{\mathrm{PO}_j^{res,*}(g_j^{res})\} = E_{\mathbf{H}_j^{res}} \left[\sum_{a_j=1}^{K_j} E\{\mathrm{PO}_j^{res,*}(a_j) | A_j = a_j, \mathbf{H}_j^{res}\} I\{g_j^{res}(\mathbf{H}_j^{res}) = a_j\} \right].$$

Then, using consistency assumption and positivity assumption, we can link the coun-

terfactual with the estimated pseudo outcome

$$E\{\mathrm{PO}_j^{res,*}(g_j^{res})\} = E_{\mathbf{H}_j^{res}}\left[\sum_{a_j=1}^{K_j} E\{\mathrm{PO}_j^{res}|A_j = a_j, \mathbf{H}_j^{res}\} I\{g_j^{res}(\mathbf{H}_j^{res}) = a_j\}\right].$$

Let $\mu_{j,a_j}^{res}(\mathbf{H}_j^{res}) = E(PO_j^{res}|A_j = a_j, \mathbf{H}_j^{res})$, then our goal is to find

$$g_j^{\mathrm{res,opt}} = \underset{g_j^{res}\in\mathcal{G}_j^{\mathrm{res}}}{argmax} E_{\mathbf{H}_j^{res}}\left[\sum_{a_j=1}^{K_j} \mu_{j,a_j}^{res}(\mathbf{H}_j^{res}) I\{g_j^{res}(\mathbf{H}_j^{res}) = a_j\}\right],$$

at stage j in the space of applicable treatment options. Likewise, we have

$$g_T^{\mathrm{res,opt}} = \underset{g_T^{res}\in\mathcal{G}_T^{\mathrm{res}}}{argmax} E_{\mathbf{H}_T^{res}}\left[\sum_{a_T=1}^{K_j} \mu_{T,a_T}^{res}(\mathbf{H}_T^{res}) I\{g_T^{res}(\mathbf{H}_T^{res}) = a_T\}\right],$$

with $\mu_{T,a_T}^{res}(\mathbf{H}_T^{res}) = E(Y|A_T = a_T, \mathbf{H}_T^{res})$.

The proposed method RT-RL utilizes the backward induction technique [4] to estimate the restricted DTR. The counterfactual mean outcome is estimated by AIPW estimator similar to [84]. We have modified the estimation of $E\{Y^*(a_T)\}$ under the viable patient history space $\mathcal{H}_T^{res}$ at final stage T. Assume we have observed n patients with $n^{res,T}$ patients received applicable treatment combinations until stage T. We propose to estimate $E\{Y^*(a_T)\}$ with $\mathbb{P}_{n^{res,T}}\{\hat{\mu}_{T,a_T}^{\mathrm{res,AIPW}}(\mathbf{H}_T^{res})\}$, where

$$\hat{\mu}_{T,a_T}^{\mathrm{res,AIPW}}(\mathbf{H}_T^{res}) = \frac{I(A_T = a_T)}{\hat{\pi}_{T,a_T}(\mathbf{H}_T^{res})} Y + \left\{1 - \frac{I(A_T = a_T)}{\hat{\pi}_{T,a_T}(\mathbf{H}_T^{res})}\right\} \hat{\mu}_{T,a_T}^{res}(\mathbf{H}_T^{res}).$$

Proposition 1 (Double Robustness). Assume patient observations $\{\mathbf{X}_i, \overline{\mathbf{A}}_{i,(T-1)}, A_{i,T}, Y_i\}_{i=1}^n$ are independent and identically distributed that follows certain multivariate distribution $\mathbf{p}$. A subset of $n^{res,T}$ patients has viable past treatment routes until stage T. We define viable patient observations as $\{\mathbf{H}_{iT}^{res}, A_{iT}, Y_i\}_{i=1}^{n^{res,T}} \equiv \{\mathbf{X}_i, \overline{\mathbf{A}}_{i,(T-1)}, A_{iT}, Y_i\}_{i=1}^{n^{res,T}}$ such that $\overline{\mathbf{A}}_{i,(T-1)} \in \overline{\mathcal{A}}_{T-1}^{res}$. $\mathbb{P}_{n^{res,T}}\{\hat{\mu}_{T,a_T}^{\mathrm{res,AIPW}}(\mathbf{H}_T^{res})\}$ is a consistent estimator of

$E\{Y^*(a_T)\}$ if either the propensity score model $\hat{\pi}_{T,a_T}(\mathbf{H}_T^{res})$ or the conditional mean model $\hat{\mu}_{T,a_T}^{res}(\mathbf{H}_T^{res})$ is correctly specified.

For the final stage T, we propose to estimate the optimal regime with

$$\hat{g}_T^{\text{res,opt}} = \underset{g_T^{res}\in\mathcal{G}_T^{\text{res}}}{argmax}\mathbb{P}_{n^{res,T}}\left[\sum_{a_T=1}^{K_T}\mu_{T,a_T}^{res,AIPW}(\mathbf{H}_T^{res})I\{g_T^{res}(\mathbf{H}_T^{res})=a_T\}\right]$$

$$= \underset{g_T^{res}\in\mathcal{G}_T^{\text{res}}}{argmax}\frac{1}{n^{res,T}}\sum_{i=1}^{n^{res,T}}\left[\frac{I(A_{iT}=g_T^{res}(\mathbf{H}_{iT}^{res}))}{\hat{\pi}_{T,A_T}(\mathbf{H}_{iT}^{res})}Y_i+\left\{1-\frac{I(A_{iT}=g_T^{res}(\mathbf{H}_{iT}^{res}))}{\hat{\pi}_{T,A_T}(\mathbf{H}_{iT}^{res})}\right\}\hat{\mu}_{T,A_T}^{res}(\mathbf{H}_{iT}^{res})\right],$$

where $\hat{\pi}_{T,A_T}(\mathbf{H}_T^{res})$ is the estimated propensity score model and $\hat{\mu}_{T,A_T}^{res}(\mathbf{H}_T^{res})$ denotes the conditional mean model. Similarly, for stage j $(1 \le j < T)$, our proposed estimator for $g_j^{\text{res,opt}}$ is

$$\hat{g}_j^{\text{res,opt}} =$$

$$\underset{g_j^{res}\in\mathcal{G}_j^{\text{res}}}{argmax}\frac{1}{n^{res,j}}\sum_{i=1}^{n^{res,j}}\left[\frac{I(A_{ij}=g_j^{res}(\mathbf{H}_{ij}^{res}))}{\hat{\pi}_{j,A_j}(\mathbf{H}_{ij}^{res})}PO_i^{res}+\left\{1-\frac{I(A_{ij}=g_j^{res}(\mathbf{H}_{ij}^{res}))}{\hat{\pi}_{j,A_j}(\mathbf{H}_{ij}^{res})}\right\}\hat{\mu}_{j,A_j}^{res}(\mathbf{H}_{ij}^{res})\right],$$

given the propensity score model $\hat{\pi}_{j,A_j}(\mathbf{H}_j^{res})$ and the conditional mean model $\hat{\mu}_{j,A_j}^{res}(\mathbf{H}_j^{res})$.

We utilize tree-based reinforcement learning (T-RL) to search for optimal treatment regime that closely follows the procedure proposed by [85], but only considering the viable restricted space. We propose to utilize the AIPW estimator $\mathbb{P}_{n^{res,j}}\{\hat{\mu}_{j,a_j}^{\text{res,AIPW}}(\mathbf{H}_j^{res})\}$ to estimate $E\{Y^*(a_T)\}$ or $E\{\text{PO}_j^{res,*}(a_j)\}$ that serves as the purity measure for tree model construction at stage T or $j < T$, respectively. The tree model divides patients into subgroups with alike histories and recommend the corresponding optimal treatment to each subgroup that maximizes the estimated group average stage-wise pseudo outcome. The proposed procedure provides viable individualized treatment solutions tailored by patient characteristics recorded in $\mathbf{H}_j^{res}$. While considering a partition over node Ω that split patients into two groups ω and

ω^c, we define the purity measure before the partition as:

$$\mathcal{P}_j(\Omega,\phi) = \max_{a_j\in\mathcal{A}_j^{res}} \mathbb{P}_{n^{res,j}}\Bigg[\sum_{a_j=1}^{K_j}\Bigg\{\frac{I(A_j=a_j)}{\hat{\pi}_{j,a_j}(\mathbf{H}_j^{res})}\mathrm{PO}_j^{res} + \{1-\frac{I(A_j=a_j)}{\hat{\pi}_{j,a_j}(\mathbf{H}_j^{res})}\}\hat{\mu}_{j,a_j}^{res}(\mathbf{H}_j^{res})\}\Bigg\} \times I(\mathbf{H}_j^{res}\in\Omega)\Bigg],$$

for $j = 1, ..., T$, where $PO_T^{res} = Y$ for final stage. We compare $\mathcal{P}_j(\Omega,\phi)$ with the purity measure after the partition

$$\mathcal{P}_j(\Omega,\omega) = \max_{a_1,a_2\in\mathcal{A}_j^{res}} \mathbb{P}_{n^{res,j}}\Bigg[\sum_{a_j=1}^{K_j}\Bigg\{\frac{I(A_j=a_j)}{\hat{\pi}_{j,a_j}(\mathbf{H}_j^{res})}\mathrm{PO}_j^{res} + \{1-\frac{I(A_j=a_j)}{\hat{\pi}_{j,a_j}(\mathbf{H}_j^{res})}\}\hat{\mu}_{j,a_j}^{res}(\mathbf{H}_j^{res})\}\Bigg\} \times I\{g_{j,\omega,a_1,a_2}^{res}(\mathbf{H}_j^{res})=a_j\}I(\mathbf{H}_j^{res}\in\Omega)\Bigg],$$

where $g_{j,\omega,a_1,a_2}^{res}$ is the treatment rule that assigns a_1 to patient subgroup ω and assigns a_2 to patient subgroup ω^c . We chose the best split ω based on largest improvement in purity measure $\mathcal{P}_j(\Omega,\omega) - \mathcal{P}_j(\Omega,\phi)$.

3.2.3 Implementing Restricted T-RL

We implement the Restricted T-RL by restricting and allocating patients based on past treatment sequences during the optimization procedure. This implies that only patients with viable treatment combination up to stage j, i.e. $\overline{\mathbf{A}}_j \in \overline{\mathcal{A}}_j^{res}$, can contribute to the treatment regime estimation at current stage j for $j = 1, ..., T$. Among patients with viable treatment sequences up to stage j, we fit separate T-RL models based on past treatment sequence for patients with $(\overline{\mathbf{A}}_{j-1}, a_j) \notin \overline{\mathcal{A}}_j^{res}, \exists a_j \in \mathcal{A}_j^{res}$ who are at risk of receiving inapplicable arms. In each model, we search the best treatment among $\{a_j \in \mathcal{A}_j^{res} : (\overline{\mathbf{A}}_j, a_j) \in \overline{\mathcal{A}}_j^{res}\}$. An additional T-RL is fitted using the remaining patient records with $\overline{\mathbf{A}}_{j-1} \in \overline{\mathcal{A}}_{j-1}^{res}$, and $(\overline{\mathbf{A}}_{j-1}, a_j) \in \overline{\mathcal{A}}_j^{res}, \forall a_j \in \mathcal{A}_j^{res}$ without restrictions on candidate optimal treatments.

Moreover, when the observed treatments of a patient satisfy $\overline{\mathbf{A}}_j \in \overline{\mathcal{A}}_j^{res}$ and $\overline{\mathbf{A}}_{j^*} \notin \overline{\mathcal{A}}_{j^*}^{res}, \forall j^* > j$ with $j < T$, this patient can only contribute to the optimal regime estimation for the first j stages. For this patient, since the observed outcome Y is not informative, PO_j^{res} is estimated using data from similar patients with applicable treatments until the $j+1$ stage. Prior to stage j, we can use backward induction to estimate the stage-wise pseudo outcomes.

When fitting tree models, we adapt the stopping criteria proposed by [85] that considers classic tree-based method pruning parameters to avoid overfitting. Node size (n_0) controls the minimal number of subjects in each final node, and the depth of the tree (d) avoids complicated regimes. Moreover, a positive constant λ for minimal purity improvement is used for examining if the current best split $\widehat{\omega}^{\text{opt}} = argmax_{\omega}[\mathcal{P}_j(\Omega, \omega)]$ satisfies $\mathcal{P}_j(\Omega, \omega)$-$\mathcal{P}_j(\Omega, \phi) > \lambda$, which guarantees user-specified meaningful splits. We set values for n_0 , d and λ for pruning based on specific applications.

For demonstration purposes, from now on, we consider a two stage problem ($T = 2$) where we have M inapplicable treatment routes $\overline{\mathbf{b}}_m = (b_{1,m}, b_{2,m}), m = 1, ..., M$. Under this setting, we assume all observed first stage treatments are viable and $a_2 = b_{2,m}$ given $a_1 = b_{1,m}$ are not viable for $m = 1, ..., M$.

In **stage 2**, we estimate the restricted treatment rule by allocating patients based on if A_1 agrees with any restricted arms. First, we construct the overall conditional mean model $\hat{\mu}_{2,a_2}^{res}$ with patients who have received viable treatment routes. Then, we separate the observations into $M+1$ groups so that patients with $A_1 = b_{1,m}$ are allocated to the m^{th} group and the rest of the patients are in the 0^{th} group. We define the truncated patient history, treatment set and treatment rules as follows. For the

0^{th} group:

$$\mathcal{H}_{02}^{\text{res}} = \mathcal{H}_2 \setminus \{(b_{1,m}, \mathbf{X}_2^T) : m = 1, ..., M\};$$

$$\mathcal{A}_{02}^{\text{res}} = \mathcal{A}_2;$$

$$\mathcal{G}_{02}^{\text{res}} = \{g_2^{res} : \mathcal{H}_{02}^{\text{res}} \mapsto \mathcal{A}_{02}^{\text{res}}\}.$$

For the m^{th} group:

$$\mathcal{H}_{\text{m2}}^{\text{res}} = \{(b_{1,m}, X_2^T)\};$$

$$\mathcal{A}_{\text{m2}}^{\text{res}} = \mathcal{A}_2 \setminus \{b_{2,m}\};$$

$$\mathcal{G}_{\text{m2}}^{\text{res}} = \{g_2^{res} : \mathcal{H}_{\text{m2}}^{\text{res}} \mapsto \mathcal{A}_{\text{m2}}^{\text{res}}\},$$

$m = 1, ..., M$.

Group specific propensity scores are estimated using data from patients who received applicable treatment routes

$$\hat{\pi}_{2,A_2}\left(\mathbf{H}_2^{res}\right) = \begin{cases} \hat{\pi}_{2,A_2}^{0}\left(\mathbf{H}_2^{res}\right), & \mathbf{H}_2^{res} \in \mathcal{H}_{02}^{res} \\ \hat{\pi}_{2,A_2}^{m}\left(\mathbf{H}_2^{res}\right), & A_2 \in \mathcal{A}_{m2}^{res} \text{ and } \mathbf{H}_2^{res} \in \mathcal{H}_{\text{m2}}^{\text{res}},\ m = 1, \ldots, M \end{cases}.$$

Then, separate T-RL models are fitted given corresponding estimated conditional mean model and propensity score model using patient observations in each restricted groups (from 1^{st} to M^{th} group) that have $\overline{\mathbf{a}}_2 = (b_{1,m}, c_{2,m})$, where $c_{2,m} \neq b_{2,m}$, to maximize the pseudo outcome at stage 2. We use all patients in the 0^{th} group to fit the remaining T-RL model, since all observed treatment sequences in this group are compatible to those of our interest.

We provide the following restricted estimation for the treatment rule for stage 2:

$$\hat{g}_2^{res,opt} = \begin{cases} \underset{g_2^{res} \in \mathcal{G}_{m,2}^{\text{res}}}{argmax} \mathbb{P}_{n^{res,m,2}} \left[\sum_{a_2=1}^{K_2} \mu_{2,a_2}^{res,AIPW}(\mathbf{H}_2^{res}) I\{g_2^{res}(\mathbf{H}_2^{res}) = a_2\} \right] & \mathbf{H}_2^{res} \in \mathcal{H}_{\text{m2}}^{\text{res}} \text{ and } a_2 \in \mathcal{A}_{m2}^{res} \\ \underset{g_2^{res} \in \mathcal{G}_{0,2}^{\text{res}}}{argmax} \mathbb{P}_{n^{res,0,2}} \left[\sum_{a_2=1}^{K_2} \mu_{2,a_2}^{res,AIPW}(\mathbf{H}_2^{res}) I\{g_2^{res}(\mathbf{H}_2^{res}) = a_2\} \right] & \mathbf{H}_2^{res} \in \mathcal{H}_{0,2}^{\text{res}} \end{cases},$$

where $m = 1, ..., M$, $n^{res,m,2}$ is the number of patients in the m^{th} group with $A_2 \neq b_{2,m}$ while $n^{res,0,2}$ is the number of patients in the 0^{th} group. We define $\mathcal{H}_2^{\text{res}} = \mathcal{H}_{0,2}^{\text{res}} \cup \mathcal{H}_{1,2}^{\text{res}} \cup ... \cup \mathcal{H}_{M,2}^{\text{res}}$. Since the recommendations are given conditional on A_1, all patients are guaranteed to receive applicable guidance from $\hat{g}_2^{res,opt}$.

At **stage 1**, we perform backward induction to estimate the optimal decision rule. More specifically, we derive the pseudo-outcome for stage 1 with

$$PO_1^{\text{res}} = \hat{E}\left\{Y \middle| A_2 = \hat{g}_2^{res,opt}\left(\mathbf{H}_2^{res}\right), \mathbf{H}_2^{res}\right\} = \hat{\mu}_{2,\hat{g}_2^{res,opt}}\left(\mathbf{H}_2^{res}\right),$$

which is the estimated potential outcome one would have observed if the future treatments at stage 2 are already optimized by $\hat{g}_2^{res,opt}$. $\hat{g}_2^{res,opt}$ consider only applicable treatment routes and $\mathbf{H}_2^{res}$ covers all patients because there is no restriction on $\mathcal{A}_1$. Moving backward, at stage 1, we don't have any constrains on $\mathcal{H}_1$ and $\mathcal{G}_1$. Hence, we have $\mathbf{H}_1^{res} = \mathbf{H}_1$. Let $\hat{\mu}_{1,a_1}^{\text{res,AIPW}}(\mathbf{H}_1)$ denote $\hat{E}(PO_1^{res}|A_1 = a_1, \mathbf{H}_1)$. The estimated optimal treatment rule for a given patient at stage 1 is:

$$\hat{g}_1^{res,opt} = \underset{g_1^{res} \in \mathcal{G}_1}{argmax} \mathbb{P}_n \left[\sum_{a_1=1}^{K_1} \hat{\mu}_{1,a_1}^{res,AIPW}(\mathbf{H}_1) I\{g_1^{res}(\mathbf{H}_1) = a_1\} \right].$$

In practice, we use a modified version of the pseudo outcome [33] to reduce the bias due to possible model misspecification from the conditional model. $\widetilde{\text{PO}}_j^{\text{res}}$ is calculated with the pseudo outcome for previous stage minus the expected outcome gain from

receiving the optimal treatment:

$$\widetilde{\mathrm{PO}}_j^{\mathrm{res}} = \widetilde{\mathrm{PO}}_{j+1}^{\mathrm{res}} + \hat{\mu}^{\mathrm{res}}_{j,g_j^{res,opt}}\left(\mathbf{H}_j^{res}\right) - \hat{\mu}^{res}_{j,A_j}(\mathbf{H}_j)$$

for $j < T$. With backward induction and defining $\widetilde{\mathrm{PO}}_T = Y$, it is easy to show that:

$$\widetilde{\mathrm{PO}}_j^{\mathrm{res}} = Y + \sum_{t=j+1}^{T-1}\left\{\hat{\mu}^{\mathrm{res}}_{t,g_t^{res,opt}}\left(\mathbf{H}_t^{res}\right) - \hat{\mu}^{\mathrm{res}}_{t,A_t}\left(\mathbf{H}_t\right)\right\} + \hat{\mu}^{res}_{T,g_T^{res,opt}}\left(\mathbf{H}_j^{res}\right) - \hat{\mu}^{res}_{T,A_T}(\mathbf{H}_j)$$

for $j < T$. Thus, for stage 1, since $\mathbf{H}_1^{res} = \mathbf{H}_1$ we have:

$$\hat{\mu}^{\mathrm{res,AIPW}}_{1,a_1}(\mathbf{H}_1) = \frac{I(A_1 = a_1)}{\hat{\pi}_{1,A_1}(\mathbf{H}_1)}\widetilde{\mathrm{PO}}_1^{\mathrm{res}} + \{1 - \frac{I(A_1 = a_1)}{\hat{\pi}_{1,A_1}(\mathbf{H}_1)}\}\hat{\mu}_{1,a_1}(\mathbf{H}_1),$$

where $\hat{\pi}_{1,A_1}(\mathbf{H}_1)$ and $\hat{\mu}_{1,g_1}(\mathbf{H}_1)$ are estimated conditionally using all records.

3.2.3.1 Example to Illustrate the RT-RL Process

Figure 3.1 illustrates a 2 stage 3 treatments per stage RT-RL patient allocation procedure with a restricted arm $A_1 = 1, A_2 = 2$. At the second stage, patients with $A_1 = 1$ are allocated to the 1^{st} group for at risk of receiving the restricted arm. We use patient records with $A_1 = 1, A_2 \neq 2$ to fit the T-RL that seeks personalized optimal treatment among $A_2 = 0$ or $A_2 = 1$ for patients in group 1. Then, we correct the pseudo outcomes for patients in the restricted arm with $\widetilde{\mathrm{PO}}_1^{\mathrm{res}}$ so that they can contribute to the first stage optimal treatment estimation. The rest of patients are allocated to the 0^{th} group where a normal T-RL is fitted with no restrictions. The treatment regimes estimated by RT-RL provides viable treatment recommendations to future patients based on past treatment sequences.

Through this process, we utilized the observed data of the patients in the restricted route before their last stage to provide better estimates of $g_1^{res,opt}$ and avoid selection bias in the study population. In the meanwhile, the above process accommodates the

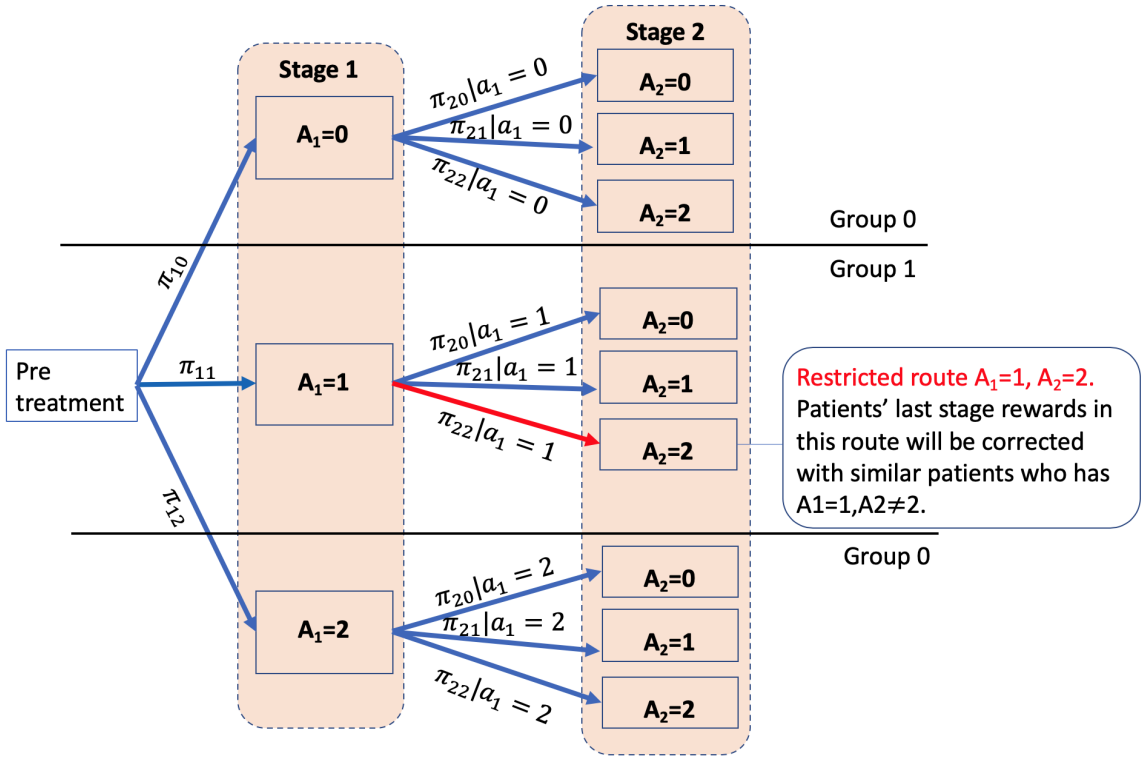

Figure 3.1: Patient allocation for a 2 stage 3 treatments per stage RT-RL estimation with inapplicable route $A_1 = 1, A_2 = 2$.

restrictions during the optimization procedure and guarantees the estimated optimal regime is interpretable and viable in practice.

3.2.3.2 RT-RL Alogorithm

We summarize the RT-TL process for $T \geq 2$ stages considering restricted routes $\overline{\mathbf{b}}_{m,j} = (b_{1,m,j}, ..., b_{j,m,j})$, $j = 1, .., T$ and $m = 1, ..., M$ with the following Algorithm.

[1]. **Input:** Outcome Y_i, patient character $\mathbf{X}_{ij}$ and treatment received $a_{i1}, ..., a_{iT}$, where $i = 1, ..., n$ and $j = 1, ..., T$; restricted routes $\overline{\mathbf{b}}_{m,j} = (b_{1,m,j}, ..., b_{j,m,j}), j = 1, .., T$ and $m = 1, ..., M$; n_0, d and λ for tree pruning.

[2]. Restrict the patient records considering $\overline{\mathbf{a}}_{i,(T-1)}$ and obtain $\mathcal{H}_T^{res}$. Estimate $\hat{\mu}_{T,a_T}^{res}(\mathbf{H}_T^{res})$ for and $a_T = 1, \ldots, K_T$ using all records.

[3]. Group subjects into the 0^{th} group when $\overline{A}_{T-1} \neq \overline{b}_{m,T-1} \forall m$. Classify subjects

into the m^{th} group when $\overline{A}_{T-1} = \overline{\mathbf{b}}_{m,T-1}$. Then we estimate group specific propensity models
$\hat{\pi}^0_{T,A_T}(\mathbf{H}^{res}_T), ..., \hat{\pi}^M_{T,A_T}(\mathbf{H}^{res}_T)$. In the m^{th} group, the propensity model is fitted with subjects with $a_T \neq b_{m,T}$ considering viable candidate treatments.

[4]. Fit separate T-RL models given (n_0 , d and λ) for each group with group specific propensity models and $\hat{\mu}^{res}_{T,a_T}(\mathbf{H}^{res}_T)$. Obtain $g^{res,opt}_T$ by combining the rules of all $M+1$ groups conditional on $\overline{\mathbf{a}}_{T-1}$.

[5]. For $j = T-1, T-2, ..., 1$ **do**

(a). With obtained $\hat{g}^{res,opt}_{j+1}$, we calculate $\widetilde{\mathrm{PO}}_j$ with $\widetilde{\mathrm{PO}}_{j+1}$ among patients with $\overline{\mathbf{a}}_{j+1} \in \mathcal{A}^{res}_{j+1}$. The calculation is done by estimating $E\left\{\widetilde{\mathrm{PO}}_{j+1}\middle| A_{j+1} = a_{j+1}, \mathbf{H}_{j+1}\right\}$ with random forest under default setting and obtain $\hat{\mu}^{res}_{j+1,g^{res,opt}_{j+1}}\left(\mathbf{H}^{res}_{j+1}\right)$ and $\hat{\mu}^{res}_{j+1,A_{j+1}}(\mathbf{H}^{res}_{j+1})$ with random forest model predictions. Then, for patients with $\overline{\mathbf{a}}_{j+1} \notin \mathcal{A}^{res}_{j+1}$ and $\overline{\mathbf{a}}_j \in \mathcal{A}^{res}_j$, we estimate the modified pseudo outcome at stage j with $\hat{E}(\widetilde{\mathrm{PO}}_j|\mathbf{H}^{res}_j)$ using the $\widetilde{\mathrm{PO}}_j$ we just calculated.

(b). Restrict the patient records considering $\overline{\mathbf{a}}_{i,(j-1)}$ and obtain $\mathcal{H}^{res}_j$. Estimate $\hat{\mu}^{res}_{j,a_j}(\mathbf{H}^{res}_j)$ for and $a_j = 1, \ldots, K_j$ using all records.

(c). Group subjects into the 0^{th} group when $\overline{A}_{j-1} \neq \overline{b}_{m,j-1} \forall m$. Classify subjects into the m^{th} group when $\overline{A}_{j-1} = \overline{\mathbf{b}}_{m,j-1}$. Then we estimate group specific propensity models $\hat{\pi}^0_{j,A_j}\left(\mathbf{H}^{res}_j\right), ..., \hat{\pi}^M_{j,A_j}\left(\mathbf{H}^{res}_j\right)$ considering viable candidate treatments.

(d). Fit separate T-RL models given (n_0 , d and λ) for each group with group specific propensity models and $\hat{\mu}^{res}_{j,a_j}(\mathbf{H}^{res}_j)$. Obtain $g^{res,opt}_j$ by combining the rules of all $M+1$ groups conditional on $\overline{\mathbf{a}}_{j-1}$.

[6]. End For

[7]. Output $g^{res,opt} = (g^{res,opt}_1, \ldots, g^{res,opt}_T)$.

3.3 Simulation Studies

We conduct simulate a studies to evaluate the performances of the proposed RT-RL compared to a naïve method with regular Reinforcement learning method that deletes all patients in the restricted arm. We consider 2 treatment stages ($T = 2$), and 3 treatments ($K = 3$) per stage ($A_j \in \{0, 1, 2\}$ for $j = 1, 2$) and one restricted treatment route ($A_1 = 1$ and $A_2 = 2$) Specifically, the simulation study is designed so that patients in the restricted treatment route have better outcomes and healthier baseline conditions. The results are examined by the agreement between DTR recommended treatments and optimal treatment routes excluding $A_1 = 1$ and $A_2 = 2$.

We generate 3 continuous covariates (X_1, X_2, X_3) independently from $N(0, 1)$. The clinical outcomes are generated from the sum of rewards at each stage ($Y = Y_1 + Y_2$) preferable with higher values. The treatment A_1 is generated from a multinomial distribution $P(A_1 = k) = \pi_{k1},\ k = 0, 1, 2$ with

$$\pi_{01} = \frac{p_{11}}{p_{11} + p_{21} + 1}, \pi_{11} = \frac{p_{21}}{p_{11} + p_{21} + 1}, \text{ and } \pi_{21} = 1 - \pi_{01} - \pi_{11},$$

where $p_{11} = \exp\{-0.2X_1 + 0.3X_2 - 0.2I(X_3 > -0.5) + 0.5\}$,$p_{21} = \exp\{0.3X_2 + 1.5I(X_3 > -0.5) + 0.5\}$. The underlying optimal treatments for stage 1 follow a rule:

$$g_1^{\text{opt}}(H) = \begin{cases} 0 & X_1 < 0, X2 < 0 \\ 1 & X_2 \geq 0 \\ 2 & X_1 \geq 0, X_2 < 0 \end{cases}$$

The rewards at stage 1 are defined generated as follows:

$$Y_1 = \exp(1 + 0.05 * X_2 - 2 * |A_1 - A_{\text{opt}}|) + \epsilon, \quad \epsilon \sim N(0,\ 0.5).$$

To mimic a typical setting where some treatments with worse side effects may per-

form better on the targeted clinical outcome, we allow, a large portion of patients to receive the restricted treatment sequence in this simulation, so that we observe a better clinical outcome for them. We allocate the treatment at stage 2 conditional on A_1 such that patient who has a greater possibility of receiving the restricted arm when $A_1 = 1$. A_2 follows a 3-level multinomial distribution with corresponding probabilities $(\pi_{02}, \pi_{12}, \pi_{22}) = \left(\frac{p_{12}}{p_{12}+p_{22}+1}, \frac{p_{22}}{p_{12}+p_{22}+1}, \frac{1}{p_{12}+p_{22}+1}\right)$, where $p_{12} = \exp\{0.05Y_1 - 0.05X_1 - 0.2I(X_3 > -0.5) - 1\}$ and $p_{22} = \exp\{0.08Y_1 - 0.2X_1 - 0.1I(X_3 > -0.5) - 1.6\}$. On the other hand, when $A_1 \neq 1$, $(\pi'_{02}, \pi'_{12}, \pi'_{22}) = \left(\frac{p'_{12}}{p'_{12}+p'_{22}+10}, \frac{p'_{22}}{p'_{12}+p'_{22}+10}, \frac{10}{p'_{12}+p'_{22}+10}\right)$, where $p'_{12} = \exp(-0.2Y_1 - 0.05X_1 + 4), p'_{22} = \exp(-0.08Y_1 + 0.6X_1 + 3)$. The optimal treatment regime for stage 2 permits higher rewards in the restricted route:

$$g_2^{\text{opt}}(H|A_1 = 1) = \begin{cases} 0 & Y_1 \leq 0.5 \\ 1 & 0.5 < Y_1 \leq 1.5 \\ 2 & Y_1 > 1.5 \end{cases}$$

$$g_2^{\text{opt}}(H|A_1 \neq 1) = \begin{cases} 0 & X_2 > -0.5 \\ 1 & X_2 \leq -0.5 \end{cases}$$

The rewards are then defined as:

$$Y_2 = \exp\{1 + I(X_3 > -0.5) * I(A_1 = 1) - 6 * \exp(|A_2 - g_2^{\text{opt}}(H)|)\} + \epsilon,$$

where $\epsilon \sim N(0,\ 0.5)$. When observed with a higher Y_1, at stage 2, the optimal rewards for patients in the restricted route are $E[Y_2|A_1 = 1, A_2 = g_2^{\text{opt}}(H) = 2] = e^2$, which is higher than that of the patients in the applicable routes $E[Y_2|A_1 \neq 1, A_2 = g_2^{\text{opt}}(H)] = e^1$.

We compare our method with a naive method where all patients on the restricted arm are deleted from the estimation and a regular reinforcement learning method is directly applied (naive T-RL). In both the naïve and Restricted T-RL, we utilize

AIPW version of the purity measure to allow robust estimation for pseudo outcomes. We set the minimum size of terminal nodes as 20, specify λ as 5% in the splitting criteria, and the maximal depth as three. One hundred replications are generated under each scenario considering different training sample size (n = 3,000, 5,000), and different propensity score model specifications (correct or incorrect). All estimated DTRs are evaluated on the agreement of the estimated and the true constrained optimal treatments of a randomly generated external test dataset with n = 1,000. The optimal regime for each patient is calculated empirically using the underlying rewards functions.

Table 3.1 summarizes the simulation results. Opt % shows the empirical mean of the percentage of subjects correctly classified to their constrained optimal treatments. Both methods perform well in the first stage, correctly specifying the true optimal treatment over 99% of patients. However, for the second stage and overall, the Restricted T-RL has significantly higher opt % than the naïve method. Both regimes improve patient outcomes, and the proposed algorithm has slightly lower improvements without recommending the restricted arm to future patients. Although deleting the patients in the restricted arms, the naïve method recommends the inappropriate treatment combination to 39-42% of patients in the test set. As demonstrated in the simulation results, both methods are doubly robust as similar results are observed under the correctly or incorrectly specified π model.

3.4 Application Adolescents Substance Use

We use a longitudinal observational dataset (n = 10,131), known as the Global Appraisal for Individual Needs (GAIN) [15], to estimate an optimal two-stage DTR to guide the level of care (LOC) placement for adolescent substance users. The DTR guides LOC placement over 0-3 months (stage 1) and 3-6 months (stage 2) to optimize substance use at month 12 after initial treatment. The observed possible

Table 3.1: Comparing simulation results between Restricted T-RL and the naive method. Standard errors are recorded in parenthesis. Opt % records the average percent of subjects in the test set that has being recommended the true optimal treatment route. $\widehat{E}\left(Y|\widehat{g}^{\text{opt}}\right) - \widehat{E}\left(Y\middle|g^{\text{obs}}\right)$ is the improvement of the estimated pseudo outcomes when following the estimated DTR versus the observed DTR. % of Recommendation with Restricted Arm shows the percent of subjects in the test set that has been recommended the restricted treatment arm.

	Correct π		Incorrect π	
Method	Restricted T-RL	Naïve T-RL	Restricted T-RL	Naïve T-RL
Training Sample		n = 3000		
Opt % Stage 1	99.3 (0.43)	99.3 (0.49)	99.1 (0.64)	98.6 (5.17)
Opt % Stage 2	81.2 (15.7)	29 (10.6)	78.9 (17.0)	29.2 (11.1)
Opt % Overall	80.9 (15.6)	28.6(10.7)	78.5 (16.9)	28.4 (11.4)
$\hat{E}(Y\|\hat{g}^{opt}) - \hat{E}(Y\|g^{obs})$	1.2 (1.6)	1.3 (1.6)	1.2 (1.6)	1.2 (1.6)
% of Recommendation with Restricted Arm	0	38.9	0	38.8
Training Sample		n = 5000		
Opt % Stage 1	99.4 (0.30)	99.4 (0.37)	99.3 (0.39)	99.1 (0.64)
Opt % Stage 2	88.1(13.4)	32.0 (9.2)	85.4 (15.3)	31.8 (9.6)
Opt % Overall	87.8 (13.3)	31.6 (9.2)	85.0 (15.3)	31.4 (9.7)
$\hat{E}(Y\|\hat{g}^{opt}) - \hat{E}(Y\|g^{obs})$	1.2 (1.6)	1.3 (1.5)	1.3 (1.6)	1.3 (1.5)
% of Recommendation with Restricted Arm	0	41.6	0	40.8

LOC treatment options at stage t ($t = 1, 2$) are inpatient ($A_t = 1$) and outpatient ($A_t = 2$); stage 2 also includes no treatment ($A_t = 3$). These treatment options report whether patients have received corresponding levels of care during each three-month period. Inpatient or residential is the most intensive treatment, where youth are admitted for at least one night to a residential, inpatient or hospital program for substance use problems. Outpatient records youth who have been admitted to a regular (1-8 hours per week) or intensive outpatient (more than 8 hours per week) program. The outcome of interest for this study is Substance Frequency Scale (SFS) at 12 months after intake, which is coded as $Y = -1 \times SFS$ to ensure that the higher value is preferable.

We constrain the optimization over $A_2 \leq 2$ given $A_1 = 1$ by not allowing "treat-

ment discontinuation" for adolescents starting with inpatient treatment. According to NIH, since relapse often occurs for adolescents with substance abuse, more than one episode of treatment may be necessary [27]. Also, inpatient treatment is recommended for a severe level of addiction. Staying in treatment for an adequate period of time is especially important for patients with $A_1 = 1$.

We randomly split the data 3:1 for training and evaluation. The training data contain randomly selected 7,599 records, and the remaining 2,532 records form the evaluation data. This facilitates inference in the comparison of the Restricted T-RL with naïve T-RL and non-dynamic regimes such as $A_1 = A_2 = 1$ and $A_1 = A_2 = 2$. The naïve T-RL excludes 404 patients who received the inapplicable treatment route $A_1 = 1,\ A_2 = 3$. Better interpretability of $\widehat{g}^{\text{opt}}$ provided by the T-RL methods can be assessed with lower probability of recommending the restricted route to patients in the evaluation set. Moreover, we compare $\widehat{E}\left(Y|\widehat{g}^{\text{opt}}\right) - \widehat{E}\left(Y\middle|g^{\text{obs}}\right)$ and the percent of patients who have $\widehat{E}\left(Y|\widehat{g}^{\text{opt}}\right) > \widehat{E}\left(Y\middle|g^{\text{obs}}\right)$ among estimated $\widehat{g}^{\text{opt}}$ to examine the ability to provide optimal treatments of each method.

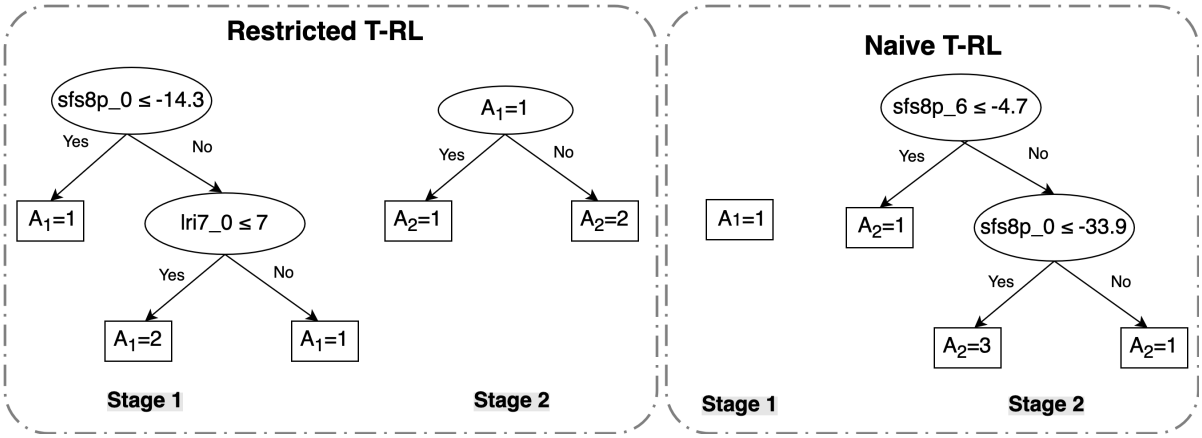

Figure 3.2: Estimated DTR from Restricted T-RL and Naïve T-RL. $sfs8p\_t$: -1* SFS at t months post intake, higher value preferred; lri: living risk index, lower value preferred; dss: depressive symptom scale, lower value preferred.

According to **Figure** 3.2, the Restricted T-RL provides interpretable DTR based on SFS at intake (sfs8p_0), living risk index at (lri7_0) intake and first stage treatment

(A_1). It recommends $A_1 = A_2 = 1$ to 947 (37.4%) patients and $A_1 = A_2 = 2$ to 1,585 (62.6%) patients in the evaluation data. Although deleting all patient records in the restricted arm, the naïve T-RL allows "treatment discontinuation" for patients who greatly improved their SFS at month 6, which leads to 674 (26.6%) patients being recommended with the inapplicable arm in the evaluation set.

Table 3.2 shows that the Restricted T-RL provides a regime in the desired class with the greatest improvement (0.7 points) in estimated substance use score at 12 months post intake. It is expected to help 56% of patients improve their substance use outcomes compared to the observed treatment. The naïve T-RL is unable to provide an interpretable and well-performing regime due to the bias introduced by the deletion of patients.

Table 3.2: Treatment regime performance on evaluation data. % improved refers to the percent of patients who are expected to benefit from the estimated regime compared to the observed treatment.

	Restricted T-RL	Naive T-RL	A_1=1, A_2=1	A_1=2, A_2=2
$\widehat{E}\left(Y\middle\vert\widehat{g}^{\text{opt}}\right) - \widehat{E}\left(Y\middle\vert g^{\text{obs}}\right)$ (SD)	0.70 (1.4)	-0.08 (1.9)	-0.03 (1.9)	0.14 (1.1)
% improved	56.0	38.6	39.8	42.3

3.5 Discussion

The proposed method Restricted T-RL searches for interpretable DTR within clinically meaningful treatment trajectories. It allows user specified restrictions for observed treatment routes. According to simulation and GAIN data application results, Restricted T-RL avoids selection bias by using partial information from patients in the restricted arm to estimate optimal treatment decisions for stages before their first problematic treatment. We achieve this by "correcting" the counterfactual rewards of the patients in the restricted arms using information from similar patients who received applicable treatments. The idea of stratifying based on patient history and

truncating treatment options is not limited to tree-based methods. Such sub-optimal DTR estimation framework can be applied to other methods when appropriate.

The naïve T-RL method suffers from information loss and possible selection bias, yet still recommends inapplicable treatment routes to new patients. This happens because the rules are estimated stagewise and the naive method considers all observed treatments in the current stage as possible treatments. Without conditioning on previous treatment sequences and without restricting the final stage treatment space, it is not guaranteed that the recommended treatment is different from $b_{j,m,j}$ when the previous recommendations are $\overline{\mathbf{a}}_{j-1} = (b_{1,m,j}, .., b_{j-1,m,j})$.

In this study, we focused on constraints with inapplicable treatment sequences. This provides a solution for many observational studies with a relatively large sample size. However, when we have little observational data in any of the 0^{th} to M_j^{th} group, the Restricted DTR is hard to estimate. We might consider parametric models for these subgroups or bootstrap techniques to handle insufficient data. Moreover, the treatment sequence constrains can also come from patient characteristics or stagewise rewards (R_j). For example, if future treatments must consider whether a patient is a responder to current treatment at stage j, all DTR ignoring the responder status at stage j should be restricted. Further studies are necessary to solve this issue.

CHAPTER IV

DataSifterText: Partially Synthetic Text Generation for Sensitive Clinical Notes

4.1 Introduction

The broad adoption of electronic health records (EHRs) responds to a significant public demand for rapid data-driven scientific investigations using the wealth of newly available data and the spectrum of innovative data science methods [13, 20, 21]. However, regulatory policies and social norms regarding concerns for data privacy hinder efficient data sharing, which results in a scarcity of available data for research purposes.

EHR data broadly falls into two categories: structured and unstructured data, structured data being numerical or categorical information, while unstructured data include physician notes, such as free-text or voice notes. Clinical narratives provide valuable information and have become an essential component in many biomedical data sources. The recent success in natural language processing (NLP) techniques has generated much interest in processing and analyzing unstructured clinical data.

Safe and rapid sharing of unstructured medical information is difficult to achieve. Generalizing and suppressing names, addresses, and other protected health information (PHI) is a common practice in creating anonymized clinical notes [25, 38, 9].

Recent studies have automated the procedure of searching for PHI and suppressing information by applying deep learning techniques [16, 98]. However, on the one hand, sensitive personal identity information in EHRs may still be obtained from detailed descriptions in narrative reports. On the other hand, when completely suppressing all sensitive PHI, de-identified medical records contain little analytical value.

In 2018, Jiaqi Guan et al. developed medGAN [29], a synthetic EHR text generation tool based on Generative Adversarial Network that utilizes long short-term memory (LSTM) as the text generator and fastText, convolutional neural network, and bidirectional recurrent neural networks as candidate discriminators. MedGAN can create fully synthetic EHR text from pre-specified disease features extracted from original text data. Yet despite little concern in disclosure risk, the data governor may face multiple challenges when generating fully synthetic text data to meet analytic goals. First, since text generation from medGAN is based on disease features, the generated fully synthetic data of "fake" patients might not represent the target population. Second, the LSTM model might suffer from insufficient data so that it fails to capture the conditional distribution of the real-world EHR text. In this scenario, fully synthetic text documents have limited data utility.

The first limitation is also applicable to fully synthetic structured datasets. In 1993, Rubin first proposed partially synthetic data generation for structured data [70], which provides a set of multiple-imputed data replacing sensitive data values with imputations. This method retains the distribution of the target population but provides data privacy by replacing sensitive values with a group of similar alternatives [57, 58]. In a recent report, the DataSifter technique (DataSifter I) [45] provides a semi-automated procedure to create a single partially synthetic dataset that disguises the replaced value locations and provides better privacy protection than the multiple imputation methods. The DataSifter I technique depends on two major steps: (1) introducing random artificial missingness to the original data and imputing the

missingness back using a robust iterative imputation method; (2) classifying neighbor cases for each record and randomly swapping a subset of feature values between similar records. The first operation masks true information sporadically and the second operation guarantees partial obfuscation for each record while preserving the data's geometrical information in feature space.

The second challenge of limited EHR text data can be handled by Bidirectional Encoder Representation from Transformers (BERT). The BERT model is the state-of-the-art language representation trained on 3,300M words that depends on multiple attention layers to provide a deep bidirectional language representation that suits multiple downstream tasks[17]. It has excellent performance on small datasets under different NLP tasks with its transformers model architecture stacking multiple attention layers.

However, there are no existing methods for generating partial synthetic unstructured data. In this chapter, we propose a free-text data anonymization tool, DataSifterText, which generates partial synthetic free-text data that protects patients' PHIs. Analogous to DataSifter I, the proposed method depends on two major steps: artificially masking/predicting sensitive tokens and replacing content with neighboring documents. To fill in the masked tokens (words, phrases or punctuation), the DataSifterText technique utilizes the BERT model. We applied the DataSifterText protocol to work injury records and clinical notes from EHR data to demonstrate the use of DataSifterText on sensitive free-text data.

4.2 Free-text Data Structure

Let us assume we have a corpus of text data consisting of n documents. The corpus can be one-hot encoded by representing each token (word or punctuation) as a vector of binary indices. We denote the i^{th} document by $W_i = (w_1, w_2, w_3, \ldots, w_{T_i}) \in \mathcal{W}$, where w_1 is the "start of sentence" one-hot vector, w_{T_i} is the "end of sentence" one-

hot vector, T_i is the length of the document, and $w_t, \forall\ t \in \{2, \ldots, (T_i - 1)\}$, is the one-hot vector for the matching word token at location t.

One common goal of a text mining algorithm is to derive corresponding classification labels ($\mathbf{L} = (L_1, \ldots, L_n)$) associated with each text document in the corpus. For example, in EHRs, each patient's clinical notes are linked with International Classification of Diseases (ICD) diagnostics, from which we can calculate the comorbidity score level as labels to represent the patient's health condition at the hospital visit [11].

Our goal is to utilize the joint distribution of tokens for documents in the corpus to create the partially synthetic (sifted) data. For instance, the distribution of the i^{th} document denoted by $W_i = (w_1, w_2, w_3, \ldots, w_{T_i})$ may be modeled as:

$$P(w_1, w_2, w_3, \ldots, w_{T_i}) = P(w_1) \prod_{k=2}^{T_i} P(w_k | w_1, \ldots, w_{k-1})$$
$$= P(w_1) P(w_1) P(w_1, w_2) \ldots P(w_{T_i} | w_1, \ldots, w_{T_i - 1}) \ .$$

Similarly, we denote the resulting sifted data (DataSifterText output) by

$$W_i^* = \left(w_1^*, w_2^*, w_3^*, \ldots, w_{T_i}^*\right),$$

which shares a similar distribution with W_i. When the perturbation is significant, the sifted text will move away from the original document to protect subject information disclosure.

4.3 Privacy and Utility Definitions for Partially Synthetic Text Data

Data privacy for each document is quantified using the cosine similarity between the Document-term Matrix (DTM) [72, 97] entry of the original and the sifted doc-

uments. If we define $\underbrace{D_1}_{n\times m}$ as the DTM for original data considering most frequent m terms and $\underbrace{D_2}_{n\times m}$ for the sifted data, data privacy for the i^{th} document, P_i, is derived using

$$P_i = \frac{D_{1i}D_{2i}}{\parallel D_{1i} \parallel \parallel D_{2i} \parallel},$$

where D_{1i}, D_{2i} are the i^{th} row in the original and sifted DTM.

Data utility is measured by the label prediction results from a BERT model [17] denoted by $f : \mathcal{W} \to \mathcal{L}$, where $\mathcal{W}$ is the set of possible documents and $\mathcal{L}$ denotes the set of labels. Using the original free-text data, we fine-tune the BERT model (multilingual cased BERT-Base, which is pre-trained by text in 104 languages, using 12 layers, 768 hidden layers, 12 heads, and 110M parameters) using label prediction as the downstream task. We denote the original prediction results as $f(W_i) \in \mathcal{L}$ and the sifted data prediction results as $f(W_i^*) \in \mathcal{L}$ for document $i = 1, \ldots, n$. Data utility is calculated by the mean label prediction accuracy $\frac{1}{n}\sum_{i=1}^{n} I[f(W_i^*) = L_i]$ and the mean label prediction agreements $\frac{1}{n}\sum_{i=1}^{n} I[f(W_i) = f(W_i^*)]$, where both measurements are preferable with higher values. When we have informative data and, as a result, an accurate BERT model, the utility of the partially synthetic dataset can be directly assessed by label prediction accuracy. We can rely more on the mean label prediction agreements when the original BERT model cannot successfully capture the map between documents and labels. Note that if W_i and W_i^* share the same class label, the subsequent prediction results under the same BERT model should be similar. For $f(W_i)$, the training and prediction data may be shared, but overfitting is not a concern since the goal is to compare label prediction results before and after the DataSifterText procedure.

4.4 DataSifterText Technique

4.4.1 Masking and Prediction

Analogous to DataSifter I imputation step, we create artificial missingness in each document and impute the missingness back with a trained language model. The goal of this step is to substitute sensitive tokens with similar phrases to protect personal information. The artificial missingness was created by replacing original text with the "[mask]" tokens. Then, the "[mask]" tokens will be imputed by BERT trained from the original data. We have chosen BERT as our language model for mask prediction because one of its embedded tasks for parameter training is predicting words in a blank. Moreover, the pre-trained BERT model provides a sufficient language model for relatively small data.

Generally, personal information in free-text documents is stored only in specific terms. Masking words in a random manner might be inefficient to protect privacy. Thus, we embed a blacklist and a whitelist for masking. The algorithm avoids masking word tokens in the blacklist (usually includes standard punctuation or stop words), while the word tokens in the white list are masked with higher probabilities. For EHRs, it is suitable to include medical-related terms in the whitelist. Deep learning techniques can also be applied for constructing a data specific whitelist that covers more PHI. The proposed algorithm caps the number of "[mask]" tokens to be assigned in a document by (number of words)$\times$0.5 and avoids consecutively masking by dynamically adjusting masking probabilities. Both settings provide context for the language model to make accurate predictions for masked locations.

4.4.2 Document Obfuscation

In the obfuscation step, we replace part of each document with text in similar documents. To find the contents in each document that needs to be replaced, we apply

Rapid Automatic Keyword Extraction (RAKE) algorithm [67] or TextRank [49] to identify the most important key phrase or key sentence. The RAKE approach parses each document with stop words and punctuation to search for candidate key phrases. To quantify the importance of each key phrase, RAKE derives a summarized score using the word frequency and degree or co-occurrence score. TextRank is a graph-based ranking model for text summary. It uses content overlap between sentences to rank the importance of each sentence.

With shorter documents, we provide two possible methods to find the contents for replacement. The first method only considers replacing the top q important key phrases in the document with that of a similar document. In the second method, we replace the contents starting from the first index of the most important key phrase recognized by RAKE to the end of the sentence, which offers a higher level of obfuscation. When longer documents are observed, TextRank is utilized to identify the most important sentences in the documents for replacement.

The similarity of two documents is measured by the cosine distance of their entries in the Document-term Matrix (DTM) [72] using the term frequency-inverse document frequency (TF-IDF) scheme [97]. For data with a larger number of documents, we apply Mini Batch K-means on DTM to group documents into smaller clusters and search for neighboring documents within each cluster.

4.5 Implementation of DataSifter Unstructured

The input text and corresponding labels are denoted by $\mathbf{W} = (W_1, \ldots, W_n)$ and $\mathbf{L} = (L_1, \ldots, L_n)$, respectively. In the masking and prediction step, we use parameters p_w and p_n to control for the masking probability in whitelist and blacklist. We denote $D(W, m)$ as the function to create a DTM using the TF-IDF scheme and most frequent m terms with documents in $\mathbf{W}$. $D(W, m)_i$ denotes the i^{th} row of the DTM. We denote $RAKE(W)$ and $TR(W)$ as the RAKE algorithm and the TextRank algo-

rithm applied to text corpus $\mathbf{W}$, correspondingly. We further denote $RAKE\left(W_i\right)_q$ as the top q important key phrases and $TR\left(W_i\right)_q$ as the top q important sentences for W_i. The input replacement method is a tuple with four options $(RAKE\ keyphrase, q)$, $(RAKE\ index, 1)$, $(TextRank, q)$ and (No obfuscation, 0) that specifies the content extraction method for the obfuscation step. Our DataSifterText protocol is implemented in Python 3.6.

Algorithm 4 DataSifterText - Masking and Prediction

Input: $\mathbf{W}$, $\mathbf{L}$, blacklist, whitelist, p_w, p_n, and replacement method.
Construct BERT on documents with downstream task as blank words prediction.
for $i = 1, .., n$ **do**
    Set nmask = 0, coef = 1.2.
    while nmask $\leq 0.5 * T_i$ **do**
        for $t = 1, ..., T_i$ **do**
            if $w_t \in$ whitelist **then**
                Set $p = 1 - p_w *$ coef.
            else if $w_t \in$ blacklist **then**
                Set $p = 0$.
            else
                Set $p = 1 - p_n *$ coef.
            end if
            Mask w_t with probability p.
            if w_t is masked **then**
                nmask = nmask+1
                coef = 1.2
            else if coef > 0.05 **then**
                coef = coef-0.05
            end if
        end for
    end while
end for
for $i = 1, .., n$ **do**
    for $t = 1, ..., T_i$ **do**
        if $w_t =$ [MASK] **then**
            Use trained BERT model to generate $P(w_t|W_i)$.
            Sample one token with the obtained distribution $P(w_t|W_i)$ and replace the [MASK] token at location t.
        end if
    end for
end for

Algorithm 4 DataSifterText (continued) - Document Obfuscation

if Replacement method $\neq$ (No obfuscation,0) **then**
  Construct $D(\mathbf{W}, m)$.
  Use Mini Batch K-means to classify documents into K clusters.
  Obtain $RAKE(\mathbf{W})$ or $TR(\mathbf{W})$ based on replacement method specification.
  for $i = 1, ..., n$ **do**
    Sample $min(1,000, n_{W_i})$ documents in the same cluster as W_i, where n_{W_i} is the number of documents in the same cluster as W_i, such that the document index $j \neq i$.
    for $j = 1, ..., 1,000$ **do**
      $dist(i,j) = \frac{D(\mathbf{W},m)_i D(\mathbf{W},m)_j}{\|D(\mathbf{W},m)_i\| \|D(\mathbf{W},m)_j\|}$
    end for
    Sample one document from $\{W_j : dist(i,j)$ within the smallest top 10% $\forall j\}$ as the replacement partner for document i and denote as W_{j*}.
    if Replacement method = $(RAKE\ keyphrase, q)$ **then**
      Replace $RAKE(W_i)_q$ with $RAKE(W_{j*})_q$
    else if Replacement method = $(RAKE\ index, 1)$ **then**
      Replace all tokens from $RAKE(W_i)_1$ to W_{T_i} with $RAKE(W_{j*})_1$ to $W_{T_{j*}}$
    else
      Replace $TR(W_i)_q$ with $TR(W_{j*})_q$.
    end if
  end for
end if
Output: $\mathbf{W}^* = \mathbf{W}$, and $\mathbf{L}^* = \mathbf{L}$

4.6 Application

4.6.1 CDC Data

Work-related injury records are generated every day. In 2019, the Centers for Disease Control and Prevention (CDC) National Institute for Occupational Safety and Health (NIOSH) launched a text classification challenge to automatically classify injury records according to the Occupational Injury and Illness Classification System (OIICS) [23]. There are 153,956 injury records in the CDC data with injury descriptions in text, gender, age, and OIICS labels. We applied DataSifterText to a subset of records (n=86,666) with the five most frequent OIICS labels, including overexertion involving outside sources (71), struck by object or equipment (62), falls

on same level (42), exposure to other harmful substances (55), and struck against object or equipment (63). The average length of the injury description is 18 words with a minimum of 3 words and a maximum of 37 words. The entire corpus contains 24,004 distinct terms.

We used the DataSifterText technique to construct partially synthetic datasets and examined their privacy and utility performances. In the application, we considered three obfuscation levels of the DataSifterText using the following parameter settings: low obfuscation ($p_w = 0.5$, $p_n = 0.75$, and replacement method = (No obfuscation, 0)); medium obfuscation ($p_w = 0.5, p_n = 0.75$, and replacement method = $(RAKE\, keyphrase, 1)$); and high obfuscation ($p_w = 0.5, p_n = 0.75$, and replacement method = $(RAKEindex, q)$). As a result, during the masking and prediction step, the tokens in the whitelist were masked with probability $1 - 0.5 \times coef$, whereas the normal tokens excluded from the whitelist and blacklist were masked with probability $1 - 0.75 \times coef$. We did not consider using TextRank as the replacement method in the obfuscation step, since the CDC data contains only one sentence per document. We compared the DataSifterText method with a naive method that suppresses the masked information such that the artificially masked or obfuscated tokens are not replaced. To measure data utility, we first constructed a BERT language model ($f(\cdot)$) with the downstream task as label prediction based on the original CDC data and obtain OIICS label predictions $f(W_i)$ for $i = 1, ..., n$. After generated the sifted data, we assessed the utility of the synthetic data by calculating label prediction accuracy ($\frac{1}{n}\sum_{i=1}^{n} I\left[f\left(W_i^*\right) = L_i\right]$) and label prediction agreement ($\frac{1}{n}\sum_{i=1}^{n} I\left[f\left(W_i\right) = f\left(W_i^*\right)\right]$), using the constructed BERT model. The data privacy protection for each sifted document was measured by the cosine distance between the original and sifted DTM entries. We chose m= 5,000 most frequent terms to limit the size of DTM for original and sifted text corpus.

The application results imply that the text in the sifted data was significantly

altered, yet it maintained a high level of data utility. The BERT model for label prediction ($f(\cdot)$) constructed with the original data predicts OIICS labels with 95.7% accuracy on a n = 1,000 test data, which indicates the language model has a good understanding of the injury description. The distribution of OIICS labels and prediction accuracies are shown in **Table** 4.1. For most labels, BERT had excellent prediction accuracy (around 95%) on the test sample. As an exception, for label 63 (struck against object or equipment), the BERT prediction had 86.1% accuracy, probably due to the lack of data.

Table 4.1: Distribution and BERT prediction accuracy for OIICS labels.

Label	**42**	**55**	**62**	**63**	**71**
Count (%)	15,624 (18.0)	11,672 (13.5)	24,402 (28.2)	9,058 (10.5)	25,910 (29.9)
Prediction Accuracy %	97.8	98.2	94.0	86.1	98.6

Examples of the original, sifted, and naively suppressed texts under different obfuscation levels are demonstrated in **Table** 4.2. The masking and prediction step with the specified parameter settings masked 3 tokens on average or 18.57% of the tokens in each document. In addition to the masking and prediction step, the medium and high obfuscation applied document obfuscation step with RAKE keyphrase and RAKE index methods, respectively. It is noticeable that higher obfuscation levels are associated with more replacements among words in the sifted text.

Table 4.2: Examples of original and sifted partially synthetic data. The bolded words were obfuscated by the masking and prediction step. The italic words in square brackets were created by the obfuscation step. Abbreviations: FX is short for bone fracture, and DX is short for diagnostics.

Data Type	**Injury Description**	**Naive suppressing description**
Original	Slipped and fell onto shoulder at work FX shoulder	NA
Sifted low obfuscation	Slipped and **fell** onto **something** at work **today**	Slipped and **[mask]** onto **[mask]** at work **[mask]**
Sifted medium obfuscation	Slipped and [*break leg*] at work **today**	Slipped and **[mask]** **[mask]** **[mask]** at work **[mask]** **[mask]**
Sifted high obfuscation	Slipped and [*DX ankle foot pain at work fell*]	Slipped and **[mask]** **[mask]** **[mask]** **[mask]** **[mask]** **[mask]** **[mask]**

The data utility of the partially synthetic CDC datasets is summarized in **Table** 4.3. Label prediction accuracy was calculated by comparing the predicted label for each document from the BERT model with the ground truth. The original data had a 98.5% accuracy suggesting nearly perfect training accuracy. The partially synthetic data achieved 86.6%, 74.5%, and 60.9% of label prediction accuracy from low to high obfuscation level. We observed a clear separation of utility preservation among the three levels, and the sifted datasets maintained the high prediction accuracy in a five-class classification problem. Compared to the naive suppressing method, the DataSifterText method provided consistently better accuracy across different scenarios and the differences between label accuracies are larger under higher obfuscation levels. This implies that our replaced tokens in the mask and prediction step provide informative content that improves prediction accuracy. The label prediction agreement was similar to the accuracy of all synthetic datasets.

Table 4.3: Data utility for original, sifted and naive suppressing CDC text (n=86,666) according to the constructed BERT model ($f(\cdot)$) that maps the CDC injury records to 5-class OIICS labels. The label prediction accuracy records $\frac{1}{n}\sum_{i=1}^{n} I\left[f\left(W_i^*\right) = L_i\right]$ and the label prediction agreement records $\frac{1}{n}\sum_{i=1}^{n} I\left[f\left(W_i\right) = f\left(W_i^*\right)\right]$.

Data Type	**DataSifterText label prediction accuracy**	**Naive suppressing label prediction accuracy**	**Label prediction agreement**
Original	98.5%	NA	NA
Sifted low obfuscation	86.5%	85.7%	87.3%
Sifted medium obfuscation	74.5%	71.1%	75.0%
Sifted high obfuscation	60.9%	48.8%	61.1%

The data privacy assessment for CDC synthetic datasets was performed using the cosine similarity between the original text and sifted text DTM entries. As illustrated in **Figure** 4.1, the mean cosine similarity for low, medium and high obfuscation levels were 0.73, 0.54, and 0.35, respectively. Under low and medium obfuscation, the distributions of the cosine similarities were approximately normal. The high

obfuscation level provided significant alteration in text such that a great proportion of documents were completely altered from the original text (cosine similarity = 0). The naive suppressing method was expected to provide similar cosine similarity as the sifted dataset. However, the unmasked tokens in the naïve synthetic data revealed actual information about the corresponding patients, which yielded higher reidentification risk than the low-obfuscation DataSifterText method.

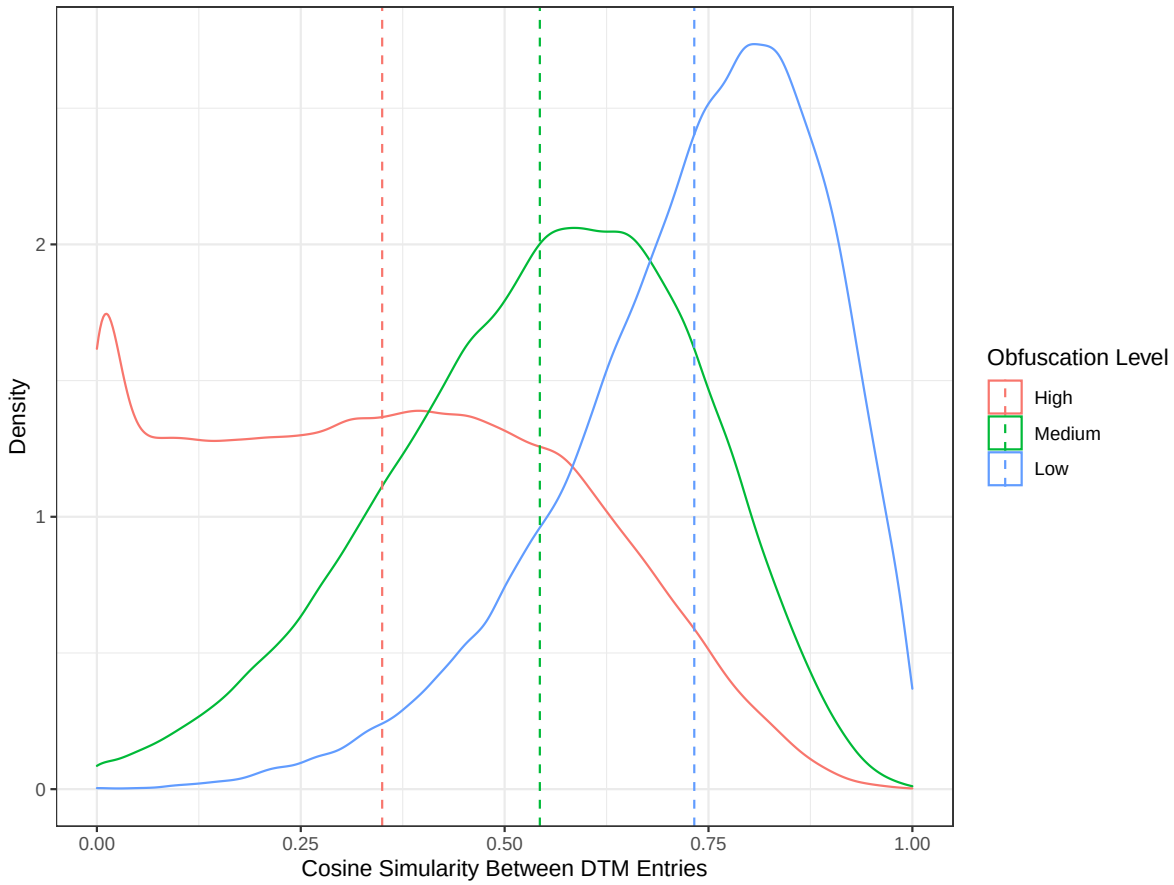

Figure 4.1: Data privacy for partially synthetic CDC datasets generated by DataSifterText. The cosine similarities were calculated by comparing the DTM entries of the original and sifted text documents. The DTM was constructed using the original text corpus with 5,000 most frequent terms. The vertical lines indicate the means of the cosine similarities among different obfuscation levels.

4.6.2 Clinical narratives in MIMIC III

The Medical Information Mart for Intensive Care III (MIMIC III) provides electronic health records from a large tertiary care hospital between 2001 and 2012 [34]. MIMIC III data stores information related to patients' hospital visits, including caregiver notes, vital signs, medications, laboratory measurements, imaging reports, mortality, and more. In this study, we focused on patient discharge summaries that describe the reason for the hospital visit, medical procedures performed, medication, and other clinical narratives related to the hospital stays. The discharge summaries are on average 220 words long (SD = 238) with minimal 4 words and maximum 3,135 words. To control model complexity and time complexity, we specified the maximum token length for BERT as 512 tokens per document. Around 9% of the documents had exceeded the handling limit. Thus, we provided two versions of preprocessing for the MIMIC III data. In the first version, "original text," our free-text documents were capped at 512 tokens deleting exceeding contents. For the second version, "summarized text," we first used TextRank to rank the sentences and then truncated them to 512 tokens. The first version retains the normal order of the sentences in the documents, while the second version keeps the most informative sentences within the word limit.

We considered a subset of 44,423 hospital visits that covers a variety of disease categories. Each discharge summary was paired with a Charlson Comorbidity Index (CCI) calculated from the corresponding International Classification of Diseases Version 9 (ICD-9) codes for diagnosis12. The ICD-9 codes were assigned after each hospital stay for billing purposes. In the study cohort, patients can be classified into 3 classes of CCIs: 0, 1-2, and $\geq$3. In the full dataset, 29% of patients belonged to the 0 class, 22% belonged to the 1-2 class, and the remaining patients belonged to the $\geq$3 CCI class. The MIMIC III clinical narratives contained de-identified data that masks name, address, dates and other protected health information with generalized

tags. However, detailed clinical procedures and information including the reason for the hospital visit, treatment history and allergies may lead to high reidentification risk.

We aimed to generate partially synthetic datasets using DataSifterText and examine the corresponding data privacy and utility. For the MIMIC III data, since multiple sentences are contained per document, we used two obfuscation levels: low obfuscation ($p_w = 0.5, p_n = 0.75$, and replacement method = (No obfuscation, 0)) and high obfuscation ($p_w = 0.5, p_n = 0.75$, and replacement method = $(TextRank, 1)$). We applied DataSifterText to both versions of preprocessed clinical text. Like the CDC application, we calculated the label accuracy and agreement to compare the data utility and obtained the cosine similarity of DTM entries to compare the data privacy.

The application data utility results are summarized in **Table** 4.4. The BERT model for label prediction constructed using "original text" and "summarized text" had a training accuracy of 62%, which indicates a fair understanding of the connection between the free-text and their labels. The two preprocessing methods yielded similar results under the low (58%) and high obfuscation levels (52%) in terms of label prediction accuracy. We observed that the reduction in accuracy in the MIMIC III application was not as significant as the CDC application when the original BERT model was less accurate. Moreover, we observed high label agreement in all settings compared to label accuracy, which showed the effectiveness of our obfuscation method. Specifically, the "summarized text" provided a significant higher label agreement.

We examined the data privacy of partially synthetic data with the cosine similarities of the original and sifted data DTM entries considering the 5,000 most frequent terms. **Figure** 4.2 illustrates the distribution of the cosine similarities across different preprocessing methods and levels of obfuscation. The results were similar for the two preprocessing methods that low obfuscation provided relatively similar documents

Table 4.4: Data utility for original and sifted MIMIC III text (n=44,423) according to the constructed BERT model ($f(\cdot)$) that maps patient discharge summary to 3-class CCIs. Original text was preprocessed by truncating long documents to 512 tokens. Summarized text was preprocessed with the TextRank algorithm and truncating the ranked summaries to 512 tokens per document. The label prediction accuracy records $\frac{1}{n}\sum_{i=1}^{n} I\left[f\left(W_i^*\right) = L_i\right]$ and the label prediction agreement records $\frac{1}{n}\sum_{i=1}^{n} I\left[f\left(W_i\right) = f\left(W_i^*\right)\right]$.

Data Type	**Original text**			**Summarized text**		
Obfuscation level	None	Low	High	None	Low	High
Label prediction accuracy	63.1%	57.9%	50.5%	62.4%	58.8%	53.3%
Label prediction agreement	NA	75.6%	61.9%	NA	80.9%	71.6%

with the raw text yielding an average cosine similarity of 0.87 while high obfuscation altered more information offering an average cosine similarity of 0.63.

4.7 Discussion

In this chapter, we proposed the DataSifterText technique to generate partially synthetic free-text that enables data-sharing by providing data privacy protection while maintaining data utility. We also derived measures to quantify data privacy and data utility for free-text data using the BERT model and DTM. According to our clinical data applications, the proposed technique protects the distribution of the original text corpus, offers individual level data obfuscation, and enables collaborative data analytics without compromising personally identifiable information. The DataSifterText algorithm provides sufficient privacy protection by disguising the location of true and obfuscated tokens. The proposed method can be implemented on multicore parallel programming environments to address scalability issues.

After controlling the maximum document length in BERT models, we preprocessed the original clinical text that exceeds this limit by truncation or summarization. In the MIMIC III data application, we recommend using the second version of

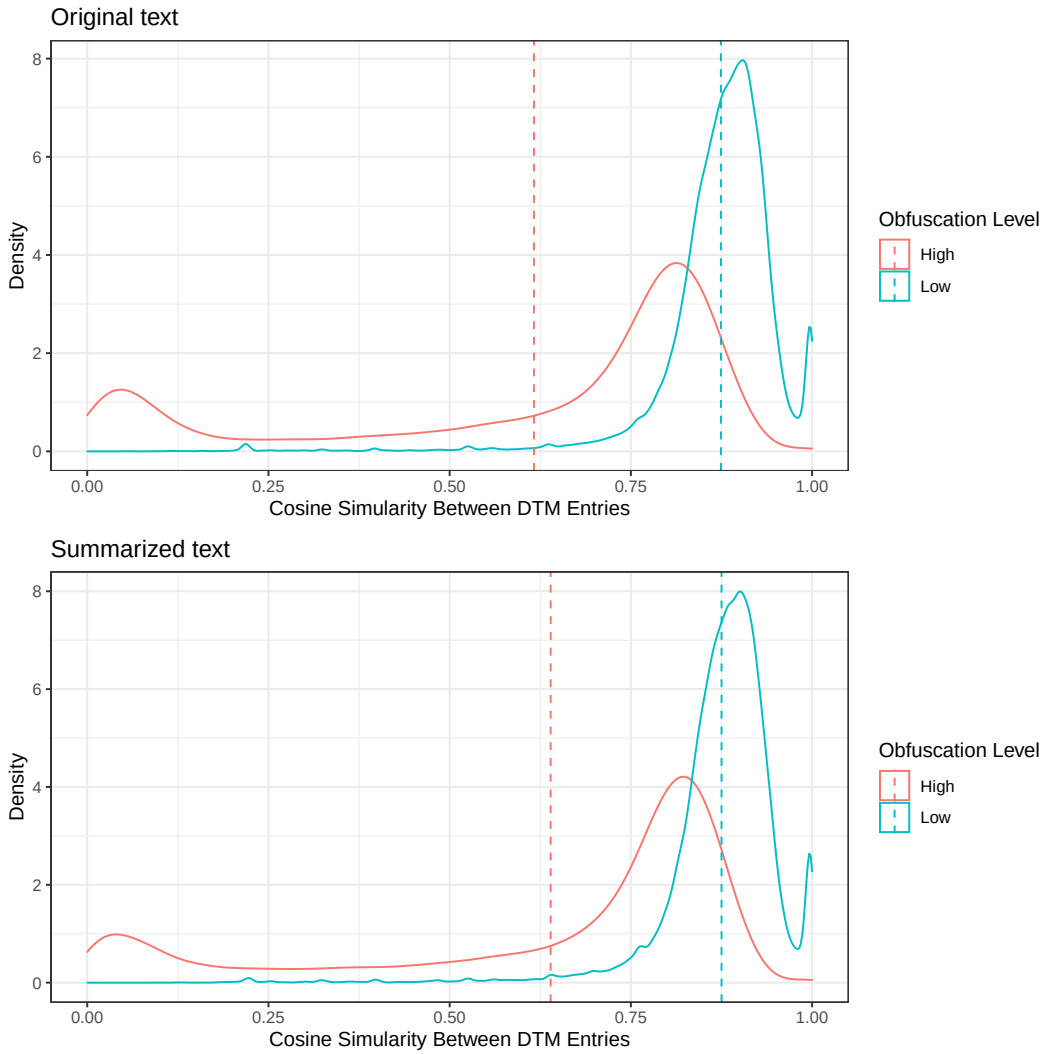

Figure 4.2: Data privacy for partially synthetic MIMIC III datasets generated by DataSifterText. The cosine similarities were calculated by comparing the DTM entries of the original and sifted text documents. The DTM was constructed using the original text corpus with 5,000 most frequent terms. The vertical lines indicate the means of the cosine similarities among different obfuscation levels.

preprocessing for long text, since it contains less noise when modeling the association between text and labels. However, for some datasets, the original order of the sentences might contain more information than the rank of the sentences. Under such settings, the data governor could consider using the "original text" preprocessing format. Future studies might also consider splitting the original text into subtexts and combining the results back to one document to avoid the truncation of long docu-

ments. This approach may require extra steps during the synthetic data evaluation phase, since the BERT model cannot handle long text.

We chose BERT as a language modeling tool because it represents a state-of-the-art technique with significantly broad applications and good scalability. The DataSifterText protocol can also be reimplemented using other language models to decrease model complexity or handle longer text objects.

CHAPTER V

Clinical Free-text Information Extraction in Dynamic Treatment Regime Estimation

5.1 Introduction

Personalized medicine (PM) provides individualized patient treatment recommendations tailoring heterogeneous patient characteristics with specific treatments, unlike the one-size-fits-all model of care. The dynamic treatment regime (DTR) is an effective vehicle under PM's umbrella that offers adaptive treatment strategies [10]. Especially for chronic conditions, a course of medical intervention containing multiple treatment stages is often needed for patients. In this paper, we demonstrate the proposed method using a motivating example, where the treatment strategy is adjusted and adapted over time to control patients' blood pressure during the first two days in intensive care units (ICU). We consider four potential classes of antihypertensive agents – angiotensin-converting enzyme inhibitors (ACEI), beta-blockers, calcium channel blockers (CCB), and diuretics. After a patient is admitted to ICU, one treatment will be assigned to the patient immediately to control the elevated blood pressure based on his/her own medical history and clinical evidences. If it controls the blood pressure, no treatment is needed on day two. Otherwise, we recommend a subsequent treatment to non-responders. In this example, we have two

sequential treatment decision stages, and one possible DTR can be: Treat patients younger than 55 years of age with ACEI and treat the rest of patients with CCB on day one; then provide CCB for non-responders on day two.

To guide evidence-based effective treatment decisions at each stage, researchers are developing statistical methods to evaluate and identify the optimal DTR, which tailor the optimal treatment choice to each individual that maximizes the expected clinical outcome given the individual's current disease status and medical history. Abundant patient information is needed to obtain accurate estimations for the desired clinical outcomes and provide various candidate tailoring variables while constructing treatment rules.

Electronic Health Records (EHRs) are major data sources for observational studies in the biomedical field, including detailed information about patients' medical histories (e.g., diagnoses, medications, and test results). As EHRs are adopted widely across different healthcare institutions, massive amounts of health information are available to develop, assess and fine-tune the optimal DTRs. Various statistical methods have been developed for identifying optimal DTRs using observational data in EHR. Commonly used parametric and semi-parametric methods include marginal structural models with inverse probability weighting (IPW) [51, 31, 90], G-estimation of structural nested mean models [63, 64, 65], and targeted maximum likelihood estimations [88]. These methods provide high interpretability but require correct modeling assumptions for a sequence of conditional models, which is practically unattainable for a large number of covariates. To alleviate modeling assumptions and maintain interpretability, Laber and Zhao proposed a tree-based method for estimating optimal treatment regimens [39]. Tao and Wang generalized the method using the doubly robust approach and developed Tree-based Reinforcement Learning (T-RL) that supports multi-stage treatment decision making [84, 85]. However, all existing methods only consider using information from structured EHR data, which is designed for the

management of care or billing purposes. When studying a specific disease, critical patient characteristics might not be available in the structured data, or partial missingness can occur in such variables. For example, tobacco use is a risk factor for many chronic diseases, including vascular diseases and lung cancer [3, 22], but structured EHR data does not include smoking status as a regular entry. Moreover, manual transcription errors occur in 1-10% of the structured EHR data [46]. For these two reasons, it is essential to collect various and accurate patient characteristics to select the correct tailoring variables when constructing DTR. Thus, we consider additional information from resources other than structured EHR data.

Narrative content in EHR, like clinical notes, represents a reliable supplementary data resource for critical patient characterization. Information extraction (IE) techniques are well studied in many disease-specific investigations for identifying unique disease conditions [92]. Most of the IE techniques are rule-based, depending on regular expression matching. cTAKES is among the most popular IE method facilitating biomedical studies based on clinical free-text data [73]. It consists of many individual tools, including sentence boundary detector, tokenizer, named entity recognizer, and negation recognizer. These individual components can be grouped to handle the extraction of patient-specific characteristics.

Nevertheless, no existing work has evaluated the benefit of performing clinical IE using narrative contents in EHR to enhance the accuracy of the estimated DTR systematically. For this study, we developed a protocol for extracting patient characteristics from the EHR narrative contents to improve DTR estimation accuracy. The protocol adopts T-RL as a robust and scalable DTR estimation approach. Our extended simulations demonstrate the benefits of utilizing IE in estimating DTR under different circumstances. Applying the protocol to the MIMIC-III database, we exhibit the effective use of IE on estimating an optimal two-stage DTR guiding hypertensive drug use among critically ill patients with severe acute hypertension.

5.2 Methods

5.2.1 Information Extraction from EHRs

In this chapter, IE is used to derive structured data from clinical free-text. We consider using rule-based IE tools that rely on Regular Expressions (REs) [36], which provide a standard mechanism to select specific strings using a bit pattern from a set of character strings. With REs, we can successfully search for patient information with bit patterns consisting of specific keywords and punctuation. Specifically, the proposed protocol closely follows the cTAKES system [73] and employs the following individual components: (1) Named entity recognition, which is the core component to identify and locate the target pieces of information; (2) Boundary detector, which detects the start and end location of the desired informative substring; and (3) Negation annotator, which can help determine the yes or no status, when referring to an identified named entity. The extraction procedure is different for different variable types. Numeric variables can be extracted using named entity recognition and boundary detector; named entity recognition and negation annotator extracts binary and categorical variables.

We take four steps to extract patient characteristics from EHR clinical notes using cTAKES components and REs.

(1) Find sections or headings in the clinical note that might contain the target information. For example, hospital visit-related information might be located in the "discharge summary" or "physician notes," and medication use information may be found in the "pharmacy" sections.

(2) Detect boundaries of the target information with the boundary detector. Patient information is usually contained in one sentence or a number surrounded by word tokens. Identifying the boundaries of the target information helps to further fine prune the target string. Punctuation including periods and colons and units like

"lb," "cm," and "kg" can indicate the beginning of or the end of the target string.

(3) Search for named entities or target keywords. After obtaining the candidate strings in specific sections of the clinical notes, we search for relevant keywords in those strings. For example, if we are extracting patient height, we can search for "height|ht|hgt," where "|" is the "or" notation in RE.

(4) Convert target strings into structured data. For numerical information, we use an ad hoc process by analyzing the possible structures of the target strings and use REs to extract the numerical values and corresponding units. For binary or categorical information, we employ the negation annotator to search for negative tags around (within five tokens) the target keyword. The default status is truth. Once a negative tag is identified, we determine that the status of the corresponding condition is false.

5.2.2 Notations and Data Representation

We consider an EHR dataset containing n patients, T treatment stages, and $K_j\,(K_j \geq 2)$ potential treatment options at the j^{th} treatment stage, $j = 1, \ldots, T$. Patients are observed to follow one of the treatments available at each stage. For the ICU blood pressure management example, we have $T = 2, K_1 = K_2 = 4$. Let A_j denote the treatment at the j^{th} treatment stage that may take a value a_j, where $a_j \in A_j = 1, \ldots, K_j$. Let $\bar{\mathbf{A}}_T \equiv (A_1, \ldots, A_T)$ denote the sequence of treatment indicators until stage T. Similarly, we denote the observed treatment routes with $\bar{a}_T \equiv (a_1, \ldots, a_T)$. We use R_j to denote the clinical outcome observed following A_j, which varies under different patient characteristics $\mathbf{X}_j$ and prior treatments received $\bar{\mathbf{A}}_{j-1}$. The overall clinical outcome at stage T is considered a functional of the reward history such that $Y \equiv f\,(R_1, \ldots, R_T)$, where $f\,(\cdot)$ is a pre-specified function (e.g., sum). We assume Y is bounded and preferable with larger values. Stage-wise individualized treatment recommendations are inferred from the observed final

outcome Y, the current candidate treatments $a_j \in \mathcal{A}_j$ and the patient medical history $\mathbf{H}_j = \left(\bar{\mathbf{A}}_{j-1}, \mathbf{X}_j^T\right)^T \in \mathcal{H}_j$. Using EHR data and our IE procedure, patient characteristics can be observed from two data components: structured and free-text data. We denote the patient characteristics at stage j observed in structured EHR data as $\mathbf{S}_j = \mathbf{X}_j^S \in \mathbb{R}^{n_s \times q_s}$, where $\mathbf{S}_j$ might contain sporadic missing values. Let $\mathbf{T}_j = \mathbf{X}_j^t \in \mathbb{R}^{n_t \times q_t}$ denote the patient characteristics at stage j extracted from the free-text data. After combining the two components we obtain $\mathbf{X}_j \in \mathbb{R}^{n \times q}$ such that $n \geq \max(n_s, n_t)$ and $q \geq \max(q_s, q_t)$ with potentially more patients and covariates than $\mathbf{X}_j^S$. The addition of $\mathbf{T}_j$ can help handle missingness and adding extra variables that are not observed from $\mathbf{S}_j$. With the observed data, we aim to find a sequence of personalized treatment rules $\mathbf{g} = (g_1, \ldots, g_T)$ that maximizes the expected counterfactual clinical outcome if the $\mathbf{g}$ is followed to make treatment decisions, where g_j maps from patient history $\mathbf{H}_j$ to potential treatments $a_j \in \mathcal{A}_j$.

5.2.3 The Estimation of DTR using Tree-based Reinforcement Learning (T-RL)

We utilize T-RL [85] to estimate the optimal DTR using structured and free-text EHR data. T-RL is a non-parametric optimization procedure that outputs an unsupervised decision tree for treatment guidance at each stage, where each fork is a split in a tailoring variable and each end node contains a recommended treatment for the corresponding patient subgroup.

When estimating the optimal DTR, we adopt the counterfactual framework for causal inference defined in Robins, 1986 [62]. Under the three standard assumptions: consistency, no unmeasured confounding and positivity, we link the counterfactual outcomes with observed information. At stage T, we denote $Y^*(A_1, ..., A_{T-1}, a_T)$ or $Y^*(a_T)$ as the counterfactual outcome for patients receiving treatment a_T given previous treatments. We aim to search for optimized treatment regime g_T^{opt} that maxi-

mizes expected counterfactual outcome. Using backward induction, at stage $j(j < T)$, we maximize the counterfactual outcome when all future treatments are optimized, which is denoted by $Y^*(\overline{\mathbf{A}}, a_j, g_{j+1}^{\text{opt}}, ..., g_T^{\text{opt}})$. However, such counterfactual outcome is not observable for all patients. We estimate stage-wise pseudo-outcome denoted as $PO_j = \hat{E}[Y(A_1, ..., A_j, g_{j+1}^{\text{opt}}, ..., g_T^{\text{opt}})]$ to approximate the target outcome and obtain g_j^{opt}. T-RL uses doubly robust AIPW estimates for patients' counterfactual outcome and pseudo outcome under all possible treatments.

The T-RL algorithm seeks the optimal regime with a sequence of treatment decision at each stage by constructing a binary tree. At any stage, we use $\hat{E}[Y_i(a)]$ to denote the estimated pseudo outcome for patient $i = 1, .., n$ given treatment $a \in \mathcal{A}$. For considering each split that separates patient group Ω into ω and ω^c, the T-RL compares

$$\mathcal{P}(\Omega, \phi) = max_{a \in \mathcal{A}} \frac{1}{n} \sum_{i=1}^{n} \hat{E}[Y_i(a)]$$

with

$$\mathcal{P}(\Omega, \omega) = max_{a_1, a_2 \in \mathcal{A}} \frac{1}{n} \{ \sum_{i=1}^{n} \hat{E}[Y_i(a')] I(i \in \omega) + \sum_{i=1}^{n} \hat{E}[Y_i(a'')] I(i \in \omega^c) \}$$

to decide if split is needed. When $\mathcal{P}(\Omega, \omega) - \mathcal{P}(\Omega, \phi)$ is meaningfully large, the algorithm will make the partition with the corresponding tailoring variable and optimal treatments (a' and a''). To avoid overfitting by pruning the tree, we use stopping rules which consider the minimal node size, minimal improvement for $\mathcal{P}(\Omega, \omega) - \mathcal{P}(\Omega, \phi)$, and maximum depth of the tree.

5.3 Simulation

We simulated a 2 stage ($T = 2$), 3 treatments per stage ($K_1 = K_2 = 3$) observational study and its corresponding EHR data to evaluate the benefit of using IE in estimating optimal DTRs. The full data contains the stage-wise rewards (R_1, R_2),

treatment received (A_1, A_2), 3 complete structured patient characteristics X_1, X_2, X_3, and 2 patient characteristics that can be extracted from the clinical narratives: weight in pounds (X_4) and current smoking status (X_5), where X_4 is in pounds and X_5 is binary. In the simulation, we considered cases where X_4 contains missing values or has entry errors and X_5 is not observed in the structured data. Then, we compare the DTR estimation performance with and without IE under above cases.

To create a scenario with simulated clinical text, we first sampled 27,707 discharge summaries from the Medical Information Mart for Intensive Care III (MIMIC-III) data [34] that contain information about weight but do not contain smoking status. Then, we randomly inserted information in a random set of text documents to create simulated clinical notes with patients' smoking information. The inserted information included smoker tags like "10 pack-year smoking", "cigars daily", "heavy smoking" and non-smoker tags like "tobacco: denies" and "former smoker". As a result, we simulated 45% smokers in our study population.

For all simulation cases, X_1, X_2, X_3 were sampled independently from $N(0, 1)$. When we observed weight information in the text, the true X_4 value agreed with the text information, otherwise it was sampled from $N(195, 51)$, which approximates the weight distribution observed in the clinical text. The true X_5 is 1 when we assign a smoker tag to a patient record and 0 otherwise. We set $\mathcal{A}_1 = \mathcal{A}_2 = \{0, 1, 2\}$. In the first stage, the treatments A_1 followed a Multinomial$(\pi_{01}, \pi_{11}, \pi_{21})$ distribution where

$$\pi_{01} = \frac{1}{\exp(0.005X_4 + 0.5X_5) + exp(0.5X_3 - 0.5X_5) + 1},$$
$$\pi_{11} = \frac{\exp(0.005X_4 + 0.5X_5)}{\exp(0.005X_4 + 0.5X_5) + exp(0.5X_3 - 0.5X_5) + 1},$$
$$\pi_{21} = \frac{\exp(0.5X_3 - 0.5X_5)}{\exp(0.005X_4 + 0.5X_5) + exp(0.5X_3 - 0.5X_5) + 1}.$$

We considered a tree structured true optimal regime such that

$$g_1^{opt}(\mathbf{H}_1) = I(X_5 > 0) \times I(X_1 > -0.5) + I(X_1 > 0.5).$$

Correspondingly, the stage-wise reward was generated by

$$R_1 = \exp\{1.5 + 0.003X_4 - |1.5X_5 - 2| \times [A_1 - g_1^{opt}(\mathbf{H}_1)]^2\} + \epsilon_1,$$

where $\epsilon_1 \sim N(0, 1)$. For stage 2, the treatments were distributed as Multinomial$(\pi_{02}, \pi_{12}, \pi_{22})$, where

$$\begin{aligned}
\pi_{02} &= \frac{1}{\exp(0.2R_1 - 0.5) + \exp(0.5X_1) + 1}, \\
\pi_{12} &= \frac{\exp(0.2R_1 - 0.5)}{\exp(0.2R_1 - 0.5) + \exp(0.5X_1) + 1}, \\
\pi_{22} &= \frac{\exp(0.5X_1)}{\exp(0.2R_1 - 0.5) + \exp(0.5X_1) + 1}.
\end{aligned}$$

The optimal decision rule for the second stage depends on the first treatment response

$$g_2^{opt}(\mathbf{H}_2) = I(X_2 > -1) \times [I(R_1 > 0) + I(R_1 > 2)].$$

The second stage reward was generated by

$$R_2 = \exp\{1.18 + 0.2X_1 - |1.5X_2 + 2| \times [A_2 - g_2^{opt}(\mathbf{H}_2)]^2\}.$$

The target clinical outcome is the sum of two stage rewards $Y = R_1 + R_2$.

We next considered two cases of observed data. For the first case, some of the X_4 entries contained error that its observed values were 100 times larger than the true

values. The errors were generated with the following probability

$$P(X_4 \text{ contains entry error}) = 0.1I(X_5 = 0) \times I(\text{weight observed in text}).$$

Under case one, X_4 contained entry error and X_5 is not observed in structured EHR. When using IE, we combined the information and obtained the full complete data. In the second case, our structured data contained missing values such that X_4 was missing at random (MAR). The probability of missing was assigned based on X_5 and if weight information was included in the clinical notes:

$$P(X_4 = NA) = 0.1 + 0.5I(X_5 = 0) \times I(\text{weight observed in text}).$$

Under case two, X_4 was MAR in the structured data, but missing completely at random with 10% missing when combining information extracted the from clinical text. Similarly, X_5 was not observed in structured data.

We considered a training sample of size $n = 500$ or $n = 1,000$ and a test sample of size $n = 1,000$ for each type of data (with or without IE) under both cases. We had 1,000 replications for each scenario. The estimated DTRs ($\hat{g}^{opt}$) were evaluated by percentage of optimal two-stage treatment recommendations (opt%) given to the patients in the test set. We also compared the estimated expected counterfactual outcome denoted by $\hat{E}\{Y^*(\hat{g}^{opt})\}$ in the test set using the true rewards model.

Table 5.1 summarizes the performance of our proposed method in the simulation studies. The cTAKES IE method successfully extracted the true smoking status from the text messages for all patients. Thus, all scenarios with IE contains the true smoking status variable, whereas the scenarios without IE does not have X_5. Case 1 mimicked the scenario where entry error is present for some of the structured variables. According to **Table** 5.1, when $n = 500$, the data with IE under case 1 had significantly higher opt% (91.5% vs 58.8%) and has improved the estimated

mean counterfactual outcome by 13% comparing to the data containing entry error without IE. When the sample size increased to 1,000, the performance of both $\hat{g}^{opt}$ has improved while the difference remained similar. For case 2, where missingness is involved in the body weight variable, the data with IE provides slightly smaller opt% and $\hat{E}\{Y^*(\hat{g}^{opt})\}$ compared to the full data due to 10% random missing in X_4. However, the data with IE still significantly outperforms the data without IE across different sample sizes.

Scenario	n = 500		n = 1,000	
	opt %	$\hat{E}\{Y^*(\hat{g}^{opt})\}$	opt %	$\hat{E}\{Y^*(\hat{g}^{opt})\}$
Case 1 with IE	91.5 (13.4)	11.10 (0.76)	97.2 (7.4)	11.35 (0.44)
Case 1 without IE	58.8 (8.3)	9.08 (0.39)	63.1 (7.1)	9.31 (0.30)
Case 2 with IE	90.1(13.9)	10.93 (0.93)	96.3 (8.8)	11.23 (0.62)
Case 2 without IE	57.4 (10.2)	8.75 (0.77)	64.5 (7.0)	9.25 (0.53)

Table 5.1: Simulation Results. The weight variable (X_4) in structured EHR contains entry errors in case 1 and has missing values in case 2. Under both cases, the current smoking status (X_5) is not observed in structured EHR data. opt% is the percentage of optimal treatment combinations recommended to the test sample. $\hat{E}\{Y^*(\hat{g}^{opt})\}$ denotes the estimated expected counterfactual outcome. Standard deviations are recorded in parenthesis.

The empirical distribution of $\hat{E}\{Y^*(\hat{g}^{opt})\}$ is shown in **Figure** 5.1. We observe that for the datasets including information extracted from clinical text, most of $\hat{E}\{Y^*(\hat{g}^{opt})\}$ were close to their optimal values. With a larger sample size ($n = 1,000$), the values were more centralized towards the optimal counterfactual outcome. The values of $\hat{E}\{Y^*(\hat{g}^{opt})\}$ for datasets without using IE spread out around 9 and failed to approach the optimal counterfactual outcome under the larger sample size.

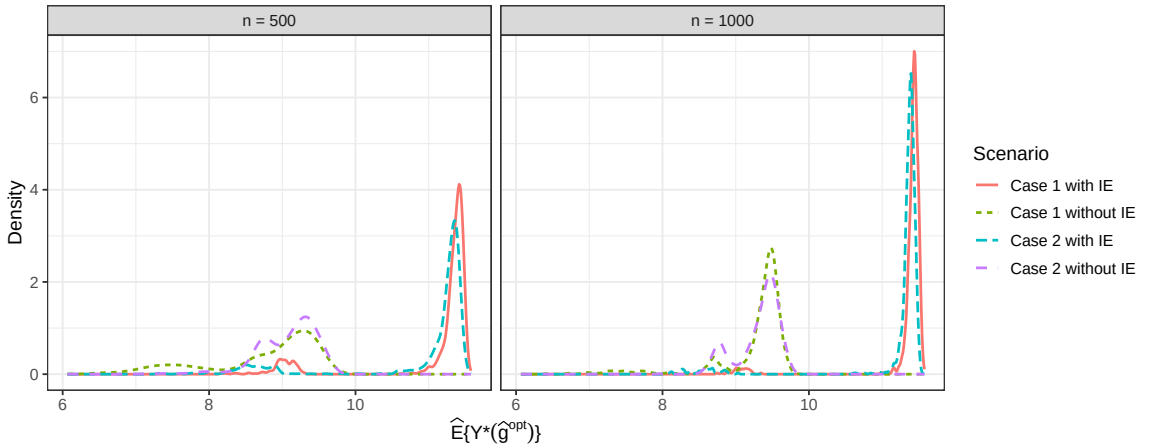

Figure 5.1: Empirical distribution of $\hat{E}\{Y^*(\hat{g}^{opt})\}$ among different simulation scenarios. The weight variable (X_4) in structured EHR contains entry errors in case 1 and has missing values in case 2. Under both cases, the current smoking status (X_5) is not observed in structured EHR data.

5.4 Personalized Antihypertensive Agents for Critically Ill Patients with Severe Acute Arterial Hypertension

Severe acute arterial hypertension can have significant consequences on various organs, including the heart, kidneys, brain, and lungs [81, 83], leading to life-threatening complications. Acute arterial hypertension is commonly encountered in ICU and acute care settings [78]. Patients with a marked increase in blood pressure and acute severe target-organ injuries (hypertensive emergencies) often require hospitalization in an ICU for immediate blood pressure reduction to safer levels [71]. In "hypertensive emergencies", the therapeutic strategy requires achieving careful and staged blood pressure lowering goals within 24 hours in order to avoid sudden, excessive reductions. "Hypertensive urgencies" describes the scenario when patients have severely elevated blood pressure but are not in danger of immediate acute end-organ injury. In this scenario, while blood pressure reduction is warranted, there are no specific evidence-based guidelines on treatment goals. As such, clinical recommendations typically suggest lowering blood pressure less aggressively and to aim for control over

the ensuing few days. However, there are no absolute blood pressure thresholds that define hypertensive emergencies or urgencies, as the actual levels differ among individuals depending on a number of characteristics including prior hypertension and pre-existing cardiovascular health status. In general, a clinically ill patient with systolic blood pressure (SBP) levels greater than 180 mmHg or diastolic blood pressures greater than 110 mmHg may require intervention [44].

Four classes of antihypertensive agents, including angiotensin-converting enzyme inhibitors (ACEI), beta-blockers, calcium channel blockers (CCB), and diuretics, are commonly used to treat hypertension. Studies suggest that antihypertensive drug responses are heterogeneous across patients [43]. We estimated a two-stage dynamic treatment regime to guide antihypertensive treatment for critically ill patients with severe acute hypertension using IE and T-RL. The DTR was constructed using The Medical Information Mart for Intensive Care III (MIMIC-III) data [34], a de-identified EHR data with over 40,000 patients who stayed in critical care units at a large tertiary care hospital between 2001 and 2012.

Patients with the following conditions were included in the study population: (1) admitted to the ICU for at least 3 days; (2) had a first-day maximum SBP higher than 180 mmHg; and (3) had been prescribed only one type of antihypertensive agent during each stage. These inclusion criteria were selected to exclude patients with shock, significant hypotension, and those who do not require or cannot tolerate antihypertensive therapies. Also, we limited our population to single hypertensive agent receivers to remove the interaction of background blood pressure medications. Although patients were not specifically admitted with the diagnosis of a hypertensive emergency, they had severely elevated blood pressure levels and received antihypertensive treatments. Thus, we assumed that achieving tighter BP control following the intervention is a more successful outcome.

We selected the decrease in SBP as our target clinical endpoint, which is preferable

with a higher value. We also assumed excessive reductions of patients' blood pressure are not achievable by any single antihypertensive agent. We considered ACEI, beta-blockers, CCB, and diuretics as possible treatments for each patient in both stages, where ACEI, beta-blockers, and CCB were introduced orally and diuretics were given intravenously (IV). The estimated two-stage DTR guides patients' blood pressure control during their first two days in ICU. Once a patient with severe acute arterial hypertension is admitted to ICU, the DTR can recommend the most effective antihypertensive class for the individual to use on day one (stage 1). If the maximum SBP is still over 140 mmHg in day two, the DTR further adjusts the hypertension treatment based on past history and further examines the patients' SBP on day 3 (stage 2).

Studies have shown that many clinical factors, including age, family history, race, smoking status and weight, are salient predictors of systolic blood pressure (SBP) and significant risk factors for developing hypertension [91]. However, the MIMIC-III structured data had no information regarding patients' smoking status. In addition, many patients had missing values for their bodyweight upon hospital admission. Without controlling for smoking and bodyweight, the drug effects towards SBP reduction might be biased in the counterfactual outcome model when estimating DTRs. Thus, we utilized the proposed IE method to extract smoking and bodyweight information from physician notes, discharge summaries, and general notes. We detected common patterns in the notes for smoking status by using the named boundary detector, named entity recognition, and negation annotator. For example, "X years smoking history" and "encouraged smoking cessation" indicates current smokers, "quit smoking X years ago" indicates former smokers, and "does not smoke" and "denies any smoking" suggests non-smokers. We assumed patients to be non-smokers when the smoking status was not mentioned in notes. For bodyweight, we extracted numerical information from patterns like "weight (lb)," "wt," and "(current): X kg."

After adding supplemental information from clinical notes, the study population was summarized with 778 complete observations (see **Table** 5.2). The majority of patients were in their 60s or 70s when admitted to the hospital. During the first day, beta-blockers (42.4%) was the most commonly prescribed hypertension drug class, followed by diuretics (31.0%), ACEI (17.6%), and CCB (9.0%). Four hundred and forty-two patients had their blood pressure successfully controlled or stopped taking antihypertensive drugs by the end of the first day. During the second day, a larger proportion of the remaining patients had IV diuretics compared to the first day.

All patient characteristics listed in **Table** 5.2 were considered as potential tailoring variables for the dynamic treatment regime and were included in the counterfactual outcome model. With this study cohort, for stage 1 (day one in ICU), we found that the optimal treatment was oral CCB for patients with maximum baseline SBP larger than 190 mmHg and minimal creatinine larger than 2 Mg/dL. Otherwise, stage 1 optimal treatment strategy should be ACEI. If we failed to control patients' blood pressure during the first day in ICU, patients no older than 70 years of age should receive oral ACEI during the second day while beta-blockers were the best hypertensive agents for patients older than 70 years of age. The estimated personalized treatment decision tree is illustrated in **Figure** 5.2. In fact, the DTR aligns with the guidelines for the treatment of hypertension by the British Hypertension Society that ACEIs are the most recommended step one antihypertensive agent for younger patients [94]. Younger patients often respond better to ACEI therapy, potentially due to several factors (e.g., high renin status). Additionally, high creatinine indicates possible acute or ongoing kidney function decline, and in this setting, it is not surprising that ACEI therapy might be less effective or safe for acute blood pressure-lowering given their potential to further drop glomerular filtration rate [75]. Thus, our results show that patients with creatinine higher than normal levels and baseline SBP higher than 190 mmHg might benefit more from CCB. These results can inform clinical

Table 5.2: Descriptive Statistics of the variables among the study cohort. Note: * mean (standard deviation) for continuous variables. Otherwise listed as n (%)

Stage	1	2
Total number of patients	778	336
Treatment		
Oral ACEI	137 (17.6)	38 (11.3)
Oral beta-blockers	330 (42.4)	128 (38.1)
Oral CCB	70 (9.0)	23 (10.5)
IV diuretics	241 (31.0)	147 (43.8)
Age at Admission*	68.1 (14.8)	68.8 (13.9)
Female	380 (48.8)	175 (52.1)
Black	138 (17.7)	59 (17.6)
Current or Former Smoker	506 (65.1)	217 (64.6)
Weight (lb)*	176.5 (59.7)	173.2 (55.1)
Kidney disease	140 (18.0)	59 (17.6)
Diabetes	336 (43.2)	142 (42.3)
COPD	38 (4.9)	20 (6.0)
Chronic Hypertension	369 (47.4)	150 (44.6)
Daily Max Systolic BP*	196.6 (16.0)	181.6 (26.7)
Daily Max Diastolic BP*	100.6 (24.1)	91.9 (24.2)
Heart Rate*	80.8 (15.3)	81.3 (14.8)
Temperature (C)*	36.9 (0.6)	37.0 (0.6)
Oxygen Saturation*	97.1 (1.9)	97.0 (1.8)
Daily Maximum Hemoglobin*	11.7 (2.0)	11.6 (13.9)
Daily Minimum creatinine (Mg/dL)*	1.8 (2.0)	1.6 (1.6)

practice, pending the results of randomized clinical trials. We further compared the decrease in SBP for the study population under the estimated dynamic treatment regime versus the observed treatment experiences. If the estimated treatment regime were followed, 67.8% of the patients in our study sample would have better control of SBP during their first two days in ICU.

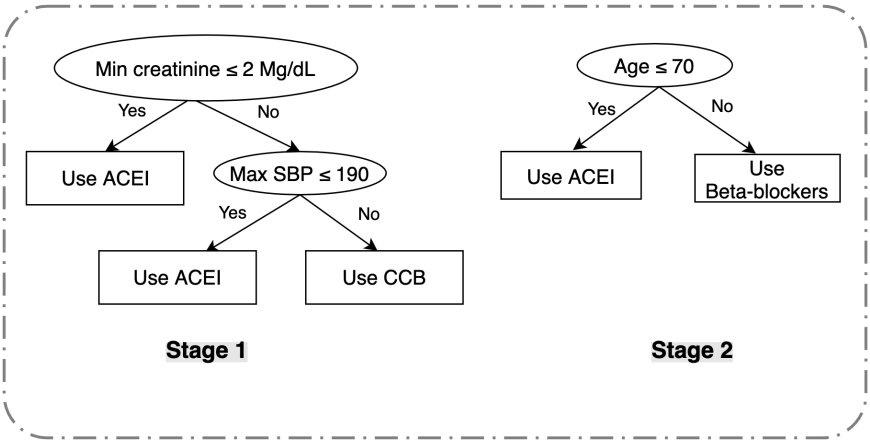

Figure 5.2: Estimated optimal dynamic treatment regime for blood pressure management among critically ill patients with severe acute hypertension. The optimal DTR was estimated using T-RL with extra information extracted from IE. Stage 1 indicates time from the first day to the second in ICU and stage 2 indicates time from the second day to the third day in ICU.

We further compared the counterfactual outcomes for our test set under the estimated optimal treatment regime versus the observed treatment experiences. Overall, the improvement in estimated mean counterfactual systolic blood pressure is 1.40 mmHg if the patient received the optimal treatment according to our estimated treatment decision rule compared to the observed data. If the estimated treatment regime were followed, 64.5% of patients in our test set would have their outcome improved .

5.5 Discussion

We have evaluated the benefit of using IE techniques to extract information from free-text clinical observations when estimating optimal DTRs. Our approach can

effectively alleviate problems in the structured EHR data, including missing values, erroneous entries, and unobserved risk factors. Our experiments show that the T-RL method significantly benefits from the use of IE. This strategy enables clinical decision support for larger study populations, provides more accurate counterfactual outcome modeling, and supports a wider pool of candidate tailoring variables.

However, the improvement of DTR estimation largely depends on the quality of clinical free-text and the IE technique. The benefit of IE may be limited when there is little additional informative content embedded in the clinical free-text. We employed a rule-based IE technique in this chapter. Since none of the IE techniques offers perfect accuracy, some of the extracted information may introduce bias to the final treatment regime. Moreover, for the rule-based IE techniques, scalability might be a potential issue when the number of variables to extract is large. Future studies might consider robust methods like deep learning techniques for extensive information extraction in a large text corpus when estimating optimal DTRs.

CHAPTER VI

Summary and Future Work

This dissertation has addressed three challenges in biomedical researches involving complex observational data: secure data sharing, viable optimal DTR estimation, and handling free-text data. The proposed techniques are designed to provide robust and scalable solutions.

The DataSifter framework proposed in Chapter II (DataSifter I and II) and Chapter IV (DataSifterText) is an essential addition to the literature of partially synthetic data generation. It provides a single synthetic dataset with global data desensitization and enables user-specified obfuscation levels. For structured data, compared to the state-of-the-art multiple imputation method, DataSfiter achieves better protection in data privacy disguising the location of obfuscated and untreated data elements. For unstructured data, DataSifterText offers better data utility by filling in similar contents in the "masked" locations compared to the conventional token suppression method. The DataSifter and DataSifterText algorithms are implemented in R, allowing flexible parallel computing setup. The corresponding R packages are available on The Statistics Online Computational Resource (SOCR) GitHub directory https://github.com/SOCR.

In Chapter III and Chapter V, we have explored practical methods for improving the quality of optimal DTR estimation using observational data. The RT-RL

method proposed in Chapter III accommodates restrictions on feasible treatment combinations in observational studies. It offers a constrained optimization procedure for effectively seeking viable optimal DTR among a subset of possible DTRs, facilitating the interpretability of DTRs for physicians to understand and use in practice. Chapter V advocates the utilization of unstructured information in optimal DTR estimation. We evaluated the benefit of IE on enlarging sample size, handling missing data, and widening the pool of candidate tailoring variables.

Some extensions can further enhance the flexibility and performance of the proposed methods. The current version of DataSifter II assumes that the time intervals between visits are similar across patients. It is of interest to handle the scenario when we have unequally spaced time-varying data across patients. Future studies may also consider introducing different variable importance among variables during the DataSifter obfuscation procedure. In this case, artificial missingness and obfuscation can occur with a higher probability in user-defined important variables to adapt various data de-identification tasks. Moreover, for free-text obfuscation, a valuable extension is to improve the semantic similarities between the observed and sifted text given a specific obfuscation level. This extension would involve training a complex machine that understands both individual word tokens and grammar structures of the original text. Finally, due to the global surge of COVID-19 cases and our limited knowledge about the virus, patient data privacy protection could be an important issue that affects both the treatment development and the patients' right. Obfuscating patient records with the DataSifter algorithm could be a potentially impactful application to alleviate the problem.

One important future research direction on RT-RL is to extend the restrictions on observed patient characteristics in addition to treatment combinations. Thus, we can impose corresponding restrictions on the viable treatment set for patients with specific characteristics (e.g., lab test results). Furthermore, when applying IE tech-

niques in optimal DTR estimation, we may consider automated information extraction methods, including deep learning, rather than rule-based algorithms, to alleviate the scalability issue when the number of variables to extract is large. We used T-RL to make interpretable treatment decisions in a clinical setting with extra information provided by IE. Future studies can also consider combining deep reinforcement learning and IE to make adaptive decisions when interpretability is not a concern. For example, in a mobile health setting where wearable devices collect vocal and physical information from patients, a deep reinforcement learning machine can cooperate with IE to understand vocal information and suggest proper interactions with the users.

The general theoretical formulation of the methods leads to the design of tools and direct applications that are expected to go beyond the biomedical and health analytics domains. For instance, the DataSifter framework can be generalized to handle sensitive datasets in socioeconomic, environmental, and insurance domains. In addition, viable optimal DTR estimating procedures have possible extensions on managing climate change, factory production and inventory, air traffic control, firefighting, and autonomous vehicles that require dynamic decision-making.

APPENDICES

APPENDIX A

Proofs for Chapter III

Proposition 1 (Double Robustness). Assume patient observations $\{\mathbf{X}_i, \overline{\mathbf{A}}_{i,(T-1)}, A_{i,T}, Y_i\}_{i=1}^n$ are independent and identically distributed that follows certain multivariate distribution $\mathbf{p}$. A subset of $n^{res,T}$ patients have viable past treatment routes until stage T. We define viable patient observations as $\{\mathbf{H}_{iT}^{res}, A_{iT}, Y_i\}_{i=1}^{n^{res,T}} \equiv \{\mathbf{X}_i, \overline{\mathbf{A}}_{i,(T-1)}, A_{iT}, Y_i\}_{i=1}^{n^{res,T}}$ such that $\overline{\mathbf{A}}_{i,(T-1)} \in \overline{\mathcal{A}}_{T-1}^{res}$. $\mathbb{P}_{n^{res,T}}\{\hat{\mu}_{T,a_T}^{\text{res,AIPW}}(\mathbf{H}_T^{res})\}$ is a consistent estimator of $E\{Y^*(a_T)\}$ if either the propensity score model $\hat{\pi}_{T,a_T}(\mathbf{H}_T^{res})$ or the conditional mean model $\hat{\mu}_{T,a_T}^{res}(\mathbf{H}_T^{res})$ is correctly specified.

Proof According to week law of large numbers, when $n^{res,T} \to \infty$, we have

$$\mathbb{P}_{n^{res,T}}\{\hat{\mu}_{T,a_T}^{\text{res,AIPW}}(\mathbf{H}_T^{res})\} \xrightarrow{p} E\left[\frac{I(A_T = a_T)}{\hat{\pi}_{T,a_T}(\mathbf{H}_T^{res})}Y + \left\{1 - \frac{I(A_T = a_T)}{\hat{\pi}_{T,a_T}(\mathbf{H}_T^{res})}\right\}\hat{\mu}_{T,a_T}^{res}(\mathbf{H}_T^{res})\right].$$

Given the 3 assumptions in Section 2, we further derive

$$\begin{aligned}
&E\{\hat{\mu}_{T,a_T}^{\text{res,AIPW}}(\mathbf{H}_T^{res})\} \\
&= \frac{Pr(A_T = a_T)}{E\{\hat{\pi}_{T,a_T}(\mathbf{H}_T^{res})\}}Y^*(a_T) + \left\{1 - \frac{Pr(A_T = a_T)}{E\{\hat{\pi}_{T,a_T}(\mathbf{H}_T^{res})\}}\right\}E\{\hat{\mu}_{T,a_T}^{res}(\mathbf{H}_T^{res})\} \\
&= E_{\mathbf{H}_T^{res}}\left[\frac{Pr(A_T = a_T|\mathbf{H}_T^{res})}{E\{\hat{\pi}_{T,a_T}(\mathbf{H}_T^{res})\}}E\{Y^*(a_T)|\mathbf{H}_T^{res}\} + \left\{1 - \frac{Pr(A_T = a_T|\mathbf{H}_T^{res})}{E\{\hat{\pi}_{T,a_T}(\mathbf{H}_T^{res})\}}\right\}E\{\hat{\mu}_{T,a_T}^{res}(\mathbf{H}_T^{res})\}\right]
\end{aligned}$$

(1) If $\hat{\pi}_{T,a_T}(\mathbf{H}_T^{res})$ is correctly specified, we have $E\{\hat{\pi}_{T,a_T}(\mathbf{H}_T^{res})\} = Pr(A_T = a_T|\mathbf{H}_T^{res})$.

Therefore,

$$E\{\hat{\mu}^{\text{res,AIPW}}_{T,a_T}(\mathbf{H}^{res}_T)\} = E_{\mathbf{H}^{res}_T}\left\{E\{Y^*(a_T)|\mathbf{H}^{res}_T\}\right\} = E\{Y^*_T(a_T)\}.$$

(2) The expectation of the proposed estimator is equivalent to

$$E\{\hat{\mu}^{\text{res,AIPW}}_{T,a_T}(\mathbf{H}^{res}_T)\} =$$
$$E_{\mathbf{H}^{res}_T}\left[\frac{Pr(A_T = a_T|\mathbf{H}^{res}_T)}{E\{\hat{\pi}_{T,a_T}(\mathbf{H}^{res}_T)\}}\Big[E\{Y^*(a_T)|\mathbf{H}^{res}_T\} - E\{\hat{\mu}^{res}_{T,a_T}(\mathbf{H}^{res}_T)\}\Big] + E\{\hat{\mu}^{res}_{T,a_T}(\mathbf{H}^{res}_T)\}\right].$$

If $\hat{\mu}^{res}_{T,a_T}(\mathbf{H}^{res}_T)$ is correctly specified, $E\{\hat{\mu}^{res}_{T,a_T}(\mathbf{H}^{res}_T)\} = E\{Y^*(a_T)|\mathbf{H}^{res}_T\}$. Then,

$$E\{\hat{\mu}^{\text{res,AIPW}}_{T,a_T}(\mathbf{H}^{res}_T)\} = E_{\mathbf{H}^{res}_T}\left[E\{\hat{\mu}^{res}_{T,a_T}(\mathbf{H}^{res}_T)\}\right] = E\{Y^*(a_T)\}.$$

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] Gbadebo Ayoade, Amir El-Ghamry, Vishal Karande, Latifur Khan, Mohammed Alrahmawy, and Magdi Zakria Rashad. Secure data processing for iot middleware systems. *The Journal of Supercomputing*, 75(8):4684–4709, 2019.

[2] Sumeet Bajaj and Radu Sion. Trusteddb: A trusted hardware-based database with privacy and data confidentiality. *IEEE Transactions on Knowledge and Data Engineering*, 26(3):752–765, 2013.

[3] M Bartal. Health effects of tobacco use and exposure. *Monaldi archives for chest disease*, 56(6):545–554, 2001.

[4] John Bather. *Decision theory: An introduction to dynamic programming and sequential decisions*. John Wiley & Sons, Inc., 2000.

[5] Andrew Baumann, Marcus Peinado, and Galen Hunt. Shielding applications from an untrusted cloud with haven. *ACM Transactions on Computer Systems (TOCS)*, 33(3):8, 2015.

[6] S van Buuren and Karin Groothuis-Oudshoorn. mice: Multivariate imputation by chained equations in r. *Journal of statistical software*, pages 1–68, 2010.

[7] Joan A Casey, Brian S Schwartz, Walter F Stewart, and Nancy E Adler. Using electronic health records for population health research: a review of methods and applications. *Annual review of public health*, 37:61–81, 2016.

[8] Timothy Caulfield, Shawn HE Harmon, and Yann Joly. Open science versus commercialization: a modern research conflict? *Genome medicine*, 4(2):17, 2012.

[9] Venkatesan T Chakaravarthy, Himanshu Gupta, Prasan Roy, and Mukesh K Mohania. Efficient techniques for document sanitization. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 843–852, 2008.

[10] Bibhas Chakraborty and Susan A Murphy. Dynamic treatment regimes. *Annual review of statistics and its application*, 1:447–464, 2014.

[11] Mary E Charlson, Peter Pompei, Kathy L Ales, and C Ronald MacKenzie. A new method of classifying prognostic comorbidity in longitudinal studies: development and validation. *Journal of Clinical Epidemiology*, 40(5):373–383, 1987.

[12] Rui Chen, Qian Xiao, Yu Zhang, and Jianliang Xu. Differentially private high-dimensional data publication via sampling-based inference. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 129–138. ACM, 2015.

[13] Guido Dartmann, Houbing Song, and Anke Schmeink. *Big data analytics for cyber-physical systems: machine learning for the internet of things*. Elsevier, 2019.

[14] Philip M Davis. Open access, readership, citations: a randomized controlled trial of scientific journal publishing. *The FASEB Journal*, 25(7):2129–2134, 2011.

[15] Michael L Dennis, Janet C Titus, Michelle K White, Joan I Unsicker, and D Hodgkins. Global appraisal of individual needs: Administration guide for

the gain and related measures. *Bloomington, IL: Chestnut Health Systems*, 2003.

[16] Franck Dernoncourt, Ji Young Lee, Ozlem Uzuner, and Peter Szolovits. De-identification of patient notes with recurrent neural networks. *Journal of the American Medical Informatics Association*, 24(3):596–606, 2017.

[17] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[18] Adriana Di Martino, Chao-Gan Yan, Qingyang Li, Erin Denio, Francisco X Castellanos, Kaat Alaerts, Jeffrey S Anderson, Michal Assaf, Susan Y Bookheimer, Mirella Dapretto, et al. The autism brain imaging data exchange: towards a large-scale evaluation of the intrinsic brain architecture in autism. *Molecular psychiatry*, 19(6):659–667, 2014.

[19] Ivo D Dinov. Volume and value of big healthcare data. *Journal of medical statistics and informatics*, 4, 2016.

[20] Ivo D Dinov. *Data science and predictive analytics: Biomedical and health applications using R*. Springer, 2018.

[21] Ivo D Dinov. Modernizing the methods and analytics curricula for health science doctoral programs. *Frontiers in Public Health*, 8, 2020.

[22] Richard Doll. Uncovering the effects of smoking: historical perspective. *Statistical methods in medical research*, 7(2):87–117, 1998.

[23] Centers for Disease Control and Prevention. Niosh announces competition for artificial intelligence programmers. `https://www.cdc.gov/niosh/updates/upd-10-24-19.html`. Accessed: 2020-10-11.

[24] Jerome Friedman, Trevor Hastie, and Rob Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, 33(1):1, 2010.

[25] Benjamin CM Fung, Ke Wang, Rui Chen, and Philip S Yu. Privacy-preserving data publishing: A survey of recent developments. *ACM Computing Surveys (Csur)*, 42(4):1–53, 2010.

[26] Craig Gentry and Dan Boneh. *A fully homomorphic encryption scheme*, volume 20. Stanford University Stanford, 2009.

[27] Mark D Godley, Susan H Godley, Michael L Dennis, Rodney R Funk, and Lora L Passetti. The effect of assertive continuing care on continuing care linkage, adherence and abstinence following residential treatment for adolescents with substance use disorders. *Addiction*, 102(1):81–93, 2007.

[28] John C Gower. Some distance properties of latent root and vector methods used in multivariate analysis. *Biometrika*, 53(3-4):325–338, 1966.

[29] Jiaqi Guan, Runzhe Li, Sheng Yu, and Xuegong Zhang. Generation of synthetic electronic medical record text. In *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 374–380. IEEE, 2018.

[30] Daniel E Hall, Barbara H Hanusa, Roslyn A Stone, Bruce S Ling, and Robert M Arnold. Time required for institutional review board review at one veterans affairs medical center. *JAMA surgery*, 150(2):103–109, 2015.

[31] Miguel A Hernán, Babette Brumback, and James M Robins. Marginal structural models to estimate the joint causal effect of nonrandomized treatments. *Journal of the American Statistical Association*, 96(454):440–448, 2001.

[32] Zhen Hong, Zinan Li, and Yubin Xia. Sdvisor: Secure debug enclave with hypervisor. In *2019 IEEE International Conference on Service-Oriented System Engineering (SOSE)*, pages 209–2095. IEEE, 2019.

[33] Xuelin Huang, Sangbum Choi, Lu Wang, and Peter F Thall. Optimization of multi-stage dynamic treatment regimes utilizing accumulated data. *Statistics in medicine*, 34(26):3424–3443, 2015.

[34] 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*, 3:160035, 2016.

[35] G Prabu Kanna and V Vasudevan. A fully homomorphic–elliptic curve cryptography based encryption algorithm for ensuring the privacy preservation of the cloud data. *Cluster Computing*, pages 1–9, 2018.

[36] Lauri Karttunen, Jean-Pierre Chanod, Gregory Grefenstette, and Anne Schille. Regular expressions for language engineering. *Natural Language Engineering*, 2(4):305–328, 1996.

[37] Theresa HM Keegan, Allison W Kurian, Kathleen Gali, Li Tao, Daphne Y Lichtensztajn, Dawn L Hershman, Laurel A Habel, Bette J Caan, and Scarlett L Gomez. Racial/ethnic and socioeconomic differences in short-term breast cancer survival among women in an integrated health system. *American journal of public health*, 105(5):938–946, 2015.

[38] Dimitrios Kokkinakis and Anders Thurin. Anonymisation of swedish clinical data. In *Conference on Artificial Intelligence in Medicine in Europe*, pages 237–241. Springer, 2007.

[39] EB Laber and YQ Zhao. Tree-based methods for individualized treatment regimes. *Biometrika*, 102(3):501–514, 2015.

[40] Andy Liaw, Matthew Wiener, et al. Classification and regression by randomforest. *R news*, 2(3):18–22, 2002.

[41] Roderick JA Little. A test of missing completely at random for multivariate data with missing values. *Journal of the American statistical Association*, 83(404):1198–1202, 1988.

[42] Roderick JA Little. Statistical analysis of masked data. *Journal of Official statistics*, 9(2):407, 1993.

[43] Azra Mahmud and John Feely. Choice of First Antihypertensive: Simple as ABCD?: . *American Journal of Hypertension*, 20(8):923–927, 08 2007.

[44] Paul E Marik and Joseph Varon. Hypertensive crises: challenges and management. *Chest*, 131(6):1949–1962, 2007.

[45] Simeone Marino, Nina Zhou, Yi Zhao, Lu Wang, Qiucheng Wu, and Ivo D Dinov. Hdda: Datasifter: statistical obfuscation of electronic health records and other sensitive datasets. *Journal of statistical computation and simulation*, 89(2):249–271, 2019.

[46] James A Mays and Patrick C Mathias. Measuring the rate of manual transcription error in outpatient point-of-care testing. *Journal of the American Medical Informatics Association*, 26(3):269–272, 2019.

[47] Charles E McCulloch and John M Neuhaus. Generalized linear mixed models. *Encyclopedia of biostatistics*, 4, 2005.

[48] Erin C McKiernan, Philip E Bourne, C Titus Brown, Stuart Buck, Amye Kenall, Jennifer Lin, Damon McDougall, Brian A Nosek, Karthik Ram, Courtney K

Soderberg, et al. Point of view: How open science helps researchers succeed. *Elife*, 5:e16800, 2016.

[49] Rada Mihalcea and Paul Tarau. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 404–411, 2004.

[50] Susan A Murphy. Optimal dynamic treatment regimes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 65(2):331–355, 2003.

[51] Susan A Murphy, Mark J van der Laan, James M Robins, and Conduct Problems Prevention Research Group. Marginal mean models for dynamic regimes. *Journal of the American Statistical Association*, 96(456):1410–1423, 2001.

[52] Office of the National Coordinator for Health Information Technology. Individuals' perceptions of the privacy and security of medical records and health information exchange, 2019.

[53] Liliana Orellana, Andrea Rotnitzky, and James M Robins. Dynamic regime marginal structural mean models for estimation of optimal dynamic treatment regimes, part i: main content. *The international journal of biostatistics*, 6(2), 2010.

[54] Heather A Piwowar, Roger S Day, and Douglas B Fridsma. Sharing detailed research data is associated with increased citation rate. *PloS one*, 2(3):e308, 2007.

[55] Trivellore E Raghunathan, Jerome P Reiter, and Donald B Rubin. Multiple imputation for statistical disclosure limitation. *Journal of official statistics*, 19(1):1, 2003.

[56] David Reinsel, John Gantz, and John Rydning. The digitization of the world: from edge to core. *IDC White Paper*, 2018.

[57] Jerome P Reiter. Inference for partially synthetic, public use microdata sets. *Survey Methodology*, 29(2):181–188, 2003.

[58] Jerome P Reiter and Satkartar K Kinney. Inferentially valid, partially synthetic data: Generating from posterior predictive distributions not necessary. *Journal of Official Statistics*, 28(4):583, 2012.

[59] Jerome P Reiter and Trivellore E Raghunathan. The multiple adaptations of multiple imputation. *Journal of the American Statistical Association*, 102(480):1462–1471, 2007.

[60] Jerome P Reiter, Quanli Wang, and Biyuan Zhang. Bayesian estimation of disclosure risks for multiply imputed, synthetic data. *Journal of Privacy and Confidentiality*, 6(1), 2014.

[61] Daniel E Rivera, Michael D Pew, and Linda M Collins. Using engineering control principles to inform the design of adaptive interventions: A conceptual introduction. *Drug and alcohol dependence*, 88:S31–S40, 2007.

[62] James Robins. A new approach to causal inference in mortality studies with a sustained exposure period—application to control of the healthy worker survivor effect. *Mathematical modelling*, 7(9-12):1393–1512, 1986.

[63] James M Robins. Correcting for non-compliance in randomized trials using structural nested mean models. *Communications in Statistics-Theory and methods*, 23(8):2379–2412, 1994.

[64] James M Robins. Causal inference from complex longitudinal data. In *Latent variable modeling and applications to causality*, pages 69–117. Springer, 1997.

[65] James M Robins. Optimal structural nested models for optimal sequential decisions. In *Proceedings of the second seattle Symposium in Biostatistics*, pages 189–326. Springer, 2004.

[66] James M Robins and Miguel A Hernán. Estimation of the causal effects of time-varying exposures. *Longitudinal data analysis*, 553:599, 2009.

[67] Stuart Rose, Dave Engel, Nick Cramer, and Wendy Cowley. Automatic keyword extraction from individual documents. *Text mining: applications and theory*, 1:1–20, 2010.

[68] Sara Rosenbaum. Data governance and stewardship: designing data stewardship entities and advancing data access. *Health services research*, 45(5p2):1442–1455, 2010.

[69] Donald B Rubin. Inference and missing data. *Biometrika*, 63(3):581–592, 1976.

[70] Donald B Rubin. Statistical disclosure limitation. *Journal of official Statistics*, 9(2):461–468, 1993.

[71] Diamantino Ribeiro Salgado, Eliezer Silva, and Jean-Louis Vincent. Control of hypertension in the critically ill: a pathophysiological approach. *Annals of intensive care*, 3(1):17, 2013.

[72] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523, 1988.

[73] Guergana K Savova, James J Masanz, Philip V Ogren, Jiaping Zheng, Sunghwan Sohn, Karin C Kipper-Schuler, and Christopher G Chute. Mayo clinical text analysis and knowledge extraction system (ctakes): architecture, compo-

nent evaluation and applications. *Journal of the American Medical Informatics Association*, 17(5):507–513, 2010.

[74] Joseph L Schafer and Recai M Yucel. Computational strategies for multivariate linear mixed-effects models with missing values. *Journal of computational and Graphical Statistics*, 11(2):437–457, 2002.

[75] Anton C Schoolwerth, Domenic A Sica, Barbara J Ballermann, and Christopher S Wilcox. Renal considerations in angiotensin converting enzyme inhibitor therapy: a statement for healthcare professionals from the council on the kidney in cardiovascular disease and the council for high blood pressure research of the american heart association. *Circulation*, 104(16):1985–1991, 2001.

[76] Phillip J Schulte, Anastasios A Tsiatis, Eric B Laber, and Marie Davidian. Q-and a-learning methods for estimating optimal dynamic treatment regimes. *Statistical science: a review journal of the Institute of Mathematical Statistics*, 29(4):640, 2014.

[77] Rebecca J Sela and Jeffrey S Simonoff. Re-em trees: a data mining approach for longitudinal and clustered data. *Machine learning*, 86(2):169–207, 2012.

[78] Tariq Shafi. Hypertensive urgencies and emergencies. *Ethnicity & Disease*, 14(4):S2–32, 2004.

[79] Phil Simon. *Analytics: The Agile Way*. John Wiley & Sons, 2017.

[80] Daniel J Stekhoven and Peter Bühlmann. Missforest—non-parametric missing value imputation for mixed-type data. *Bioinformatics*, 28(1):112–118, 2011.

[81] Joseph F Styron, Preeti Jois-Bilowich, Randall Starling, Robert E Hobbs, Michael C Kontos, Peter S Pang, and W Frank Peacock. Initial emergency

department systolic blood pressure predicts left ventricular systolic function in acute decompensated heart failure. *Congestive Heart Failure*, 15(1):9–13, 2009.

[82] Richard S Sutton, Andrew G Barto, et al. *Introduction to reinforcement learning*, volume 135. MIT press Cambridge, 1998.

[83] Lynda A Szczech, Christopher B Granger, Joseph F Dasta, Alpesh Amin, W Frank Peacock, Peter A McCullough, John W Devlin, Matthew R Weir, Jason N Katz, Frederick A Anderson, et al. Acute kidney injury and cardiovascular outcomes in acute severe hypertension. *Circulation*, 121(20):2183, 2010.

[84] Yebin Tao and Lu Wang. Adaptive contrast weighted learning for multi-stage multi-treatment decision-making. *Biometrics*, 73(1):145–155, 2017.

[85] Yebin Tao, Lu Wang, and Daniel Almirall. Tree-based reinforcement learning for estimating optimal dynamic treatment regimes. *The annals of applied statistics*, 12(3):1914, 2018.

[86] Jonathan P Tennant, François Waldner, Damien C Jacques, Paola Masuzzo, Lauren B Collister, and Chris HJ Hartgerink. The academic, economic and societal impacts of open access: an evidence-based review. *F1000Research*, 5, 2016.

[87] Vernon Turner, John F Gantz, David Reinsel, and Stephen Minton. The digital universe of opportunities: Rich data and the increasing value of the internet of things. *IDC Analyze the Future*, 16, 2014.

[88] Mark J Van Der Laan and Daniel Rubin. Targeted maximum likelihood learning. *The international journal of biostatistics*, 2(1), 2006.

[89] 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.

[90] Lu Wang, Andrea Rotnitzky, Xihong Lin, Randall E Millikan, and Peter F Thall. Evaluation of viable dynamic treatment regimes in a sequentially randomized trial of advanced prostate cancer. *Journal of the American Statistical Association*, 107(498):493–508, 2012.

[91] Wenyu Wang, Elisa T Lee, Richard R Fabsitz, Richard Devereux, Lyle Best, Thomas K Welty, and Barbara V Howard. A longitudinal study of hypertension risk factors and their relation to cardiovascular disease: the strong heart study. *Hypertension*, 47(3):403–409, 2006.

[92] Yanshan Wang, Liwei Wang, Majid Rastegar-Mojarad, Sungrim Moon, Feichen Shen, Naveed Afzal, Sijia Liu, Yuqun Zeng, Saeed Mehrabi, Sunghwan Sohn, et al. Clinical information extraction applications: a literature review. *Journal of biomedical informatics*, 77:34–49, 2018.

[93] Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.

[94] Bryan Williams, Neil R Poulter, Morris J Brown, Mark Davis, Gordon T McInnes, John F Potter, Peter S Sever, and Simon McG Thom. British hypertension society guidelines for hypertension management 2004 (bhs-iv): summary. *Bmj*, 328(7440):634–640, 2004.

[95] Russ Wolfinger and Michael O'connell. Generalized linear mixed models a pseudo-likelihood approach. *Journal of statistical Computation and Simulation*, 48(3-4):233–243, 1993.

[96] Alexander Wood, Vladimir Shpilrain, Kayvan Najarian, Ali Mostashari, and Delaram Kahrobaei. Private-key fully homomorphic encryption for private clas-

sification. In *International Congress on Mathematical Software*, pages 475–481. Springer, 2018.

[97] Ho Chung Wu, Robert Wing Pong Luk, Kam Fai Wong, and Kui Lam Kwok. Interpreting tf-idf term weights as making relevance decisions. *ACM Transactions on Information Systems (TOIS)*, 26(3):1–37, 2008.

[98] Shweta Yadav, Asif Ekbal, Sriparna Saha, and Pushpak Bhattacharyya. Deep learning architecture for patient data de-identification in clinical records. In *Proceedings of the clinical natural language processing workshop (ClinicalNLP)*, pages 32–41, 2016.

[99] Jun Zhang, Graham Cormode, Cecilia M Procopiuc, Divesh Srivastava, and Xiaokui Xiao. Privbayes: Private data release via bayesian networks. *ACM Transactions on Database Systems (TODS)*, 42(4):25, 2017.

[100] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the royal statistical society: series B (statistical methodology)*, 67(2):301–320, 2005.