G-Network for Outcome Prediction Under Dynamic Intervention Regimes

by

Rui Li

Submitted to the Department of Electrical Engineering and Computer Science

in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2020

Author . . . .
Department of Electrical Engineering and Computer Science
December 12, 2019

Certified by . . . .
Roger Mark
Professor
Thesis Supervisor

Certified by . . . .
Li-wei Lehman
Research Scientist
Thesis Supervisor

Accepted by . . . .
Katrina LaCurts
Chair, Master of Engineering Thesis Committee

G-Network for Outcome Prediction Under Dynamic Intervention Regimes

by

Rui Li

Submitted to the Department of Electrical Engineering and Computer Science
on December 12, 2019, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

Counterfactual prediction is useful in settings where one would like to know what would have happened had an alternative regime been followed, but one only knows the outcomes under the observational regime. Typically, the regimes are dynamic and time-varying. In these scenarios, G-computation can be used for counterfactual prediction. This work explores a novel recurrent neural network approach to G-computation, dubbed G-Net. Many implementations of G-Net were explored and compared to the baseline, linear regression. Two independent datasets were used to evaluate the performance of G-Net: one from a physiological simulator, CVSim, and another from the real-world MIMIC database. Results from the CVSim experiments suggest that G-Net outperforms the traditional linear regression approach to G-computation. The best G-Net model found from the CVSim experiments was then evaluated using the MIMIC dataset. The outcomes under a few different counterfactual strategies on the MIMIC cohort were explored and evaluated for clinical plausibility.

Thesis Supervisor: Roger Mark
Title: Professor

Thesis Supervisor: Li-wei Lehman
Title: Research Scientist

Acknowledgments

First and foremost, I would like to thank my direct supervisor on this project, Li-wei Lehman, for all her help in putting together this project, the overall research direction guidance, and the countless meetings and discussions. Thank you Professor Mark as well for the help in defining a meaningful clinical focus for this project, patiently explaining the medical side of things to me, and the discussions and feedback. I've gained much clinical knowledge in the process.

I would also like to thank my mentors on this project, Zach Shahn and Jun Li. Thank you, Zach, for the many lengthy discussions and for your statistical expertise. I learned much about statistical models and evaluation methods from the meetings with you. Thanks, Jun, for your deep learning expertise and the numerous discussions, ranging from kalman filtering to residual networks to memory networks. I learned much about state of the art deep learning techniques from you.

Thanks, Ming Yu Lu, for the work in generating the CVSim datasets used in this project. Thanks to the rest of the IBM team for the discussions.

Last but not least, thanks to my parents for their unconditional love and support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Counterfactual prediction is necessary in many application settings, where one has access to the data under the observational regime, but one wonders what would happen had an alternative regime been followed. For instance, in clinical decision making tasks, it is often the case that researchers know the interventions that were followed and the outcome for the patients under these observed interventions. However, researchers want to know what would have happened to the patients had physicians followed an alternative intervention strategy, also known as the counterfactual strategy. Treatment strategies of interest are usually time-varying and dynamic; there is an intervention at each time step and the intervention at each time step is dependent on the history up until that time point. Counterfactual prediction also generally requires accounting for causal effects between the treatment strategies and the other variables of interest.

Let us consider a particular clinical problem, the administration of fluid in the ICU [4]. The amount of fluid that the patient receives at each time step is dependent on the patient's health history up to and including that time step. For instance, if a patient's blood pressure has been extremely low for the previous few time steps and no fluid had been administered yet, a physician might want to administer a large volume of fluid to help increase blood pressure and blood perfusion to organs, especially in the case of septic patients. However, too much fluid could have negative effects such as increased chances of developing pulmonary edema. It is often difficult to determine the amount of fluid to administer, so it would be immensely helpful to physicians if there were a system to allow them to explore what would happen to a patient under different counterfactual strategies.

The collection of g-methods can be used to estimate the effects of dynamic, time-varying treatment strategies [6]. G-methods include marginal structural models, structural nested models, and g-computation [13, 10, 11, 16, 12, 17, 14, 9]. Of these g-methods, g-computation is the best suited for *general* dynamic treatment strategies with *high dimensional* history [10]. G-computation is a framework that first uses the data under the observational regime to learn the conditional distribution of the covariates at the next time step given the history of covariates and treatments, and then as a second step, estimates the counterfactual outcomes by running Monte Carlo simulations forward in time. Many different statistical models could be used as the regressor to model the conditional distribution of the covariates at the next time step given the history of covariates and treatments.

Traditionally linear regression has been the regressor of choice. However, recent works have shown that recurrent neural networks (RNN) perform well in modeling and predicting high-dimensional multivariate time-series data. This thesis explores a recurrent neural network framework for G-computation, called G-Net. Many variants of the network architecture are explored and compared against each other and against the traditional linear regression approach to G-computation. The performance of these models was evaluated on two datasets; the first one is generated from CVSim[5], a hemodynamic system simulation program, and the second one is extracted from MIMIC-3, a public critical care database [7].

Chapter 2

Related Work

The first application of the parametric g-formula to an epidemiological problem was conducted by Taubman et al. [16]. They use the parametric g-formula to predict the population risk of coronary heart disease under different intervention strategies. As a part of their study, they included some static covariates (age, smoking prior to the start of the study, BMI at the start of the study, etc.) and some time-varying covariates (BMI, high blood pressure, diabetes, cancer, diabetes, etc.). Some interventions they tried included no intervention, quitting smoking, exercising 30 minutes a day, consuming at least 5g alcohol per day, and a few interventions which were combinations of these previous interventions. Using g-formula, they were able to predict the 20 year risk ratio as a result of each of these interventions. In their study, they used traditional linear and logistic regression to predict some of the covariates. Taubman et al. found that the g-formula method does indeed support the idea that a joint intervention of no smoking, increased exercise, improved diet, moderate alcohol intake, and reduced BMI reduces the risk of coronary heart disease. This result lends support to the belief that g-formula is a powerful and accurate framework that could be used to estimate the effects of different counterfactual dynamic, time-varying interventions.

There has been some recent work by Atan et. al., Alaa et.al., and Yoon et. al. to predict the outcome under counterfactual strategies given data under the observational regime [2, 1, 19]. However, their works mainly focus on learning point exposure rather than dynamic, time-varying treatment effects.

Lim et al. used a RNN approach to marginal structural models (MSM) for counterfactual predictions [8]. However, there is a limitation to MSMs; MSMs can only make predictions for *static* time-varying intervention strategies. History adjusted MSMs could make predictions for a strategy like "give 0.5 liter of fluid for the next 4 hours" or "give 1 liter of fluid for the next 2 hours and then give 1 liter for the following 2 hours". Under these strategies, the treatment amount at each hour must be specified up front. MSMs cannot make predictions for *dynamic* intervention strategies, such as "if blood pressure falls below 65 for the next 4 hours, give 1 liter of fluid; else give no fluids" or "if no fluids had been given in the past 2 hours and blood pressure was below 65 in the past hour, give 1 liter of fluid". G-computation, though, can make predictions for dynamic intervention strategies, where the intervention at each timestep could depend on the covariates at that time step, and the covariates and treatments in the past.

Schulam et al. introduced counterfactual Gaussian processes, a form of continuous time g-computation[15]. In their study, they only experimented with static time-varying intervention strategies applied to ICU data. Xu et al. used Gaussian processes to predict individual patient level treatment response curves, though they only evaluated their work on a held out test set from the real-world observational ICU data; they did not evaluate their method for counterfactual prediction [18]. One limitation of Gaussian processes is that they have high time complexity. On the other hand, RNNs are much more scalable and better able to handle larger number of observations and higher dimensional data.

RNNs, in particular LSTMS, have been shown to achieve state-of-the-art performance on predictions involving time-series data. There have not been any works on using RNN for regressors in G-computation, though RNNs seem promising due to their state of the art performance on time-series data. This work explores a RNN based G-computation. Not only is a RNN based G-computation expected to perform well, but also it does not have the limitations the previously mentioned related works have.

Chapter 3

Background on G-Computation

G-computation is a framework that first uses the data under the observational regime to learn the conditional distribution of the covariates at the next time step given the history of covariates and treatments, and then as a second step, estimates the counterfactual outcomes by running Monte Carlo simulations forward in time. Many counterfactual strategies can be tested and the simulation outcomes compared to see the effects of various strategies on the covariates. If one such counterfactual strategy was a strategy where treatment is always withheld, then comparing the simulations from a counterfactual treatment strategy where treatment were administered against the no treatment counterfactual simulations, would give insights on the treatment effect. The Monte Carlo simulations can be analyzed and summarized to give the population level average treatment effects, which can then be used to enable exploration of what strategies are good for a cohort in general. In addition, these Monte Carlo simulations can be analyzed at the individual patient level to see what would have happened to an individual patient if they had been under a particular counterfactual strategy.

Let A_t be the treatment action at time t, Y_t be the outcome of interest at time t, and L_t be the vector of covariates at time t. Let $Y_t(S_m)$ be the counterfactual outcome by following the counterfactual intervention strategy S starting at time m until time t. The goal of counterfactual prediction is to estimate the expected outcome at time t given the history through time m, $H_m = (L_{1:m}, A_{1:m-1})$, and a specific counterfactual strategy S which is started at time m and continues until time t. If written mathematically, this would be:

$$E[Y_t(S)|H_m] \tag{3.1}$$

It is also possible to estimate the counterfactual outcome *distribution*.

$$p(Y_t(S)|H_m) \tag{3.2}$$

In order for g-computation to properly estimate (3.1) and (3.2), there is a set of assumptions which must hold [10].

1. **Consistency:** the observed outcome is the same as the counterfactual outcome where the counterfactual strategy is the observed strategy
2. **Sequential Exchangeability:** there is no unobserved confounding of treatment at any time and any future outcome
3. **Positivity:** the counterfactual treatment strategy of interest has some non-zero probability of being followed

With these assumptions in place, (3.2) can be re-written for $t = m$.

$$p(Y_m(S)|H_m) = p(Y_m|H_m, A_m = S(H_m)) \tag{3.3}$$

For times $t > m$, it is necessary to adjust for time-varying confounding.

$$\begin{aligned} p(Y_t(S) = y|H_m) = &\int_{l_{m+1:t}} p(Y_t = y|H_m, L_{m+1:t} = l_{m+1:t}, A_{m:t} = S(H_{m:t})) \\ &\times \prod_{k=m+1}^{t} p(L_k = l_k|H_m, L_{m+1:k-1} = l_{m+1:k-1}, A_{m:k-1} = S(H_m, l_{m+1:k-1})) \end{aligned} \tag{3.4}$$

This integral is difficult to calculate in closed form, so Monte Carlo simulation is typically used to estimate it. In order to use Monte Carlo simulation it is necessary to be able to simulate from the joint conditional distribution:

$$p(L_t|L_{1:t-1}, A_{1:t-1}) \tag{3.5}$$

Training data would be needed in order to fit a model to learn this joint conditional distribution. Once this model has been fit to the training data, it is possible to start simulating from this joint conditional distribution. The outcome, Y_t, can be set as one of the L_t, for simplicity in modeling. The G-computation framework, where Y_t is set to be one of the covariates, L_t, is shown below in pseudocode form.

G-Computation Framework

```
# Step 1: Fit a model to learn $p(L_t|L_{1:t-1}, A_{1:t-1})$
$R \leftarrow$ trainModel(training data)
# Step 2: Simulate under counterfactual strategy $S$
$simulations = []$
for $n \leftarrow 1$ to number of samples do
    for $j \leftarrow 1$ to number of simulations do
        $L_m \leftarrow R(L_{1:m-1}, A_{1:m-1})$
        $A_m \leftarrow S(L_{1:m}, A_{1:m-1})$
        $s_j = []$
        for $t \leftarrow m + 1$ to $T$ do
            $L_t \leftarrow R(L_{1:t-1}, A_{1:t-1})$
            $A_t \leftarrow S(L_{1:t}, A_{1:t-1})$
            $s_j \mathrel{+}= [(L_t, A_t)]$
        end
        $simulations \mathrel{+}= s_j$
    end
end
```

Chapter 4

G-Network Design

Many models can be used to learn the $p(L_t|L_{1:t-1}, A_{1:t-1})$ used for G-computation. G-Net is a neural network framework for G-computation where the model used to fit the conditional probability is a recurrent neural network structure, since recurrent networks have been shown to perform well in predictive tasks involving high-dimensional multivariate time-series data. This chapter explains the general G-Net framework and discusses a few specific implementations of G-Net.

4.1 Separation of Covariates for Sequential Simulation

Often times, the distributions of the covariates are very different from one another, so it may be difficult to simulate from the joint distribution. One solution is to separate the covariates into groups, where the covariates within each group have similar distributions. Then, it is possible to sequentially simulate the covariates, group by group. Separation of the covariates by their distribution types allows for the use of a separate model for each of the distribution types, which in theory should result in better performance.

The conditional probability distribution

$$p(L_t|L_{1:t-1}, A_{1:t-1}) \tag{4.1}$$

can be rewritten as

$$p(L_t^0|L_{1:t-1}, A_{1:t-1}) \times p(L_t^1|L_t^0, L_{1:t-1}A_{1:t-1}) \times ... \times p(L_t^{G-1}|L_t^0, L_t^1, ..., L_t^{G-2}, L_{1:t-1}, A_{1:t-1}) \tag{4.2}$$

where the superscripts denote the group a particular covariate is in. Each variable could be its own group, or all variables could be in one group. The number of groups, G, could vary from 1 to the number of covariates of interest used by model. The mathematical equivalence of (4.1) and (4.2) allows for separate models for different variable types. For this particular project, there were only two major types of covariates used: categorical and continuous variables, so in all the G-Net variants explored in this project $G = 2$.

4.2 G-Net Architecture

With $G = 2$, the G-Net framework would essentially have two regressors, one for the categorical variables and one for the continuous variables, as illustrated in Figure 4-1. The categorical regressor, R^0, takes as input all the current timestep's covariates, (L_t^0, L_t^1) and the intervention A_t at this timestep. It outputs the prediction of the categorical covariates at the next timestep, L_{t+1}^0. The continuous regressor, R^1, takes as input all the current timesteps covariates, (L_t^0, L_t^1), treatment A_t, and the output of R^0, L_{t+1}^0. It outputs the prediction of the continuous covariates at the next timestep, L_{t+1}^1.

Figure 4-1 only illustrates the high level schematic for G-Net. Many possible G-Net implementations could extend this high level architecture. For instance, it is possible to first learn a representation for (L_t^0, L_t^1, A_t) and feed that learned feature into R^0. It is also possible to try many different implementations for the regressors R^0 and R^1, such as a linear layer, a linear layer with ReLu activation, or LSTMs. A

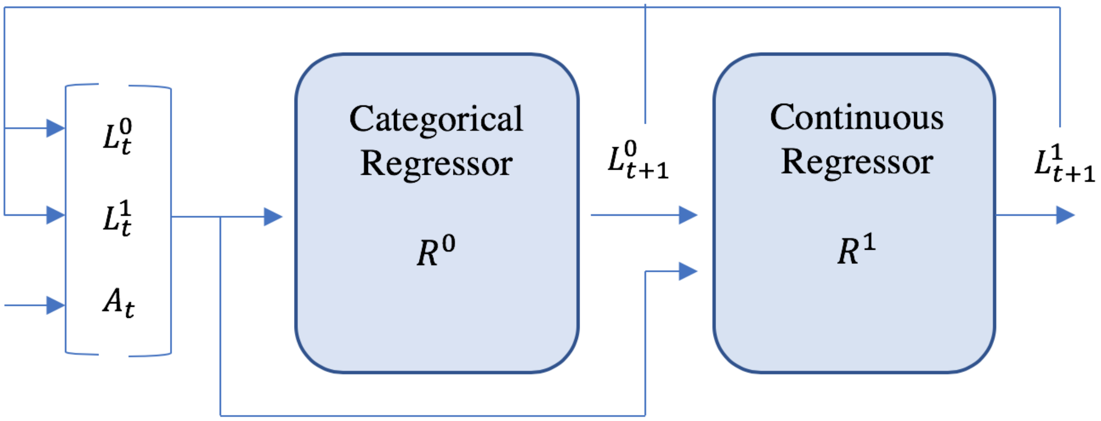

Figure 4-1: Framework for G-Network

few combinations of these different implementations are experimented with. Figures 4-2 through 4-5 illustrate the various G-Net implementations that are experimented with in this thesis.

Model $M1$ (shown in Figure 4-1) is the baseline linear regression model, with one linear layer for the categorical variables and one linear layer for the continuous variables. Models $M2$ and $M4$ (shown in Figures 4-3 and 4-5) include a representational layer, F, to first encode a representation for the inputs. In this project, $F = LSTM$. $M2$ and $M4$ are different in that $M2$ has linear layers for the categorical and continuous regressors, while $M4$ has LSTMs. $M3$ (shown in Figure 4-4) does not have a representational layer, F. $M3$ has two LSTMs to function as the categorical and continuous regressors.

4.3 Training Details

4.3.1 Teacher Forcing During Training

During training time, there are two possible ways to train the second regressor for the continuous covariates R^1. One way is student forcing, the other is teacher forcing. Teacher forcing is when the regressor R^1 is fed as input the *actual* L_{t+1}^0 from the training data. Student forcing, on the other hand, is when the regressor R^1 is fed as input the *predicted* $\hat{L^0_{t+1}}$ from regressor R^0. There are pros and cons to each one of these methods. For this project, teacher forcing was chosen because it would result in a R^1 that is more representative of the conditional distribution in the training set.

4.3.2 Choice of Loss Functions

Many possible loss functions can be used. For this project, the loss function used for the categorical variables was cross entropy loss. The loss function used for the continuous variables was root mean-squared error (RMSE) loss. Both the cross-entropy loss and the RMSE loss were averaged over all timesteps. By averaging over the losses at all the timesteps, the regressors R^0 and R^1 will be less prone to bias due to the timestep number. The total loss that is backpropagated is the sum of the average cross-entropy loss and the average RMSE loss.

4.4 Simulation Details

Regressor R_0 estimates the conditional expectation $E[L^0_t|L_{1:t-1}, A_{1:t-1}]$, and regressor R_1 estimates the conditional expectation $E[L^1_t|L^0_t, L_{1:t-1}, A_{1:t-1}]$. However, the probability distribution $p(L^0_t|L_{1:t-1}, A_{1:t-1})$ and $p(L^1_t|L^0_t, L_{1:t-1}, A_{1:t-1})$ are needed in order to simulate from the categorical and continuous distributions respectively.

The conditional expectation for the categorical variables could be used directly for the simulation process. The output of R_0 is the softmax over the all possible categories for each categorical variable. To simulate a single categorical variable, a sample is generated from the categorical distribution with the probability of each category being equal to the softmax prediction for the category.

Simulating from the continuous distribution is a bit more complicated. Equation (4.3) shows how it is possible to simulate the distribution from the conditional expectation, $\hat{E}[L^1_t|L^0_t, L_{1:t-1}, A_{1:t-1}]$ and the empirical losses, e^1_t.

$$L^1_t|L^0_t, L_{1:t-1}, A_{1:t-1} \sim \hat{E}[L^1_t|L^0_t, L_{1:t-1}, A_{1:t-1}] + e^1_t \tag{4.3}$$

where e_t^1 is a random sample from the distribution of the empirical residuals $L_t^1 - \hat{L_t^1}$

$$e_t^1 \sim L_t^1 - \hat{L_t^1} \tag{4.4}$$

The empirical losses e_t^1 are calculated from a hold-out validation set (from the observational regime dataset) that is not used to train the model. This way of sampling from the continuous distribution is non-parametric, as only the empirical losses are used; there is no assumption of the underlying distributional form. There is no need to limit the application of G-Net to any particular distriution like Gaussian, Poisson, etc. A non-parametric implementation is more general and has the advantage of working on any type of distribution.

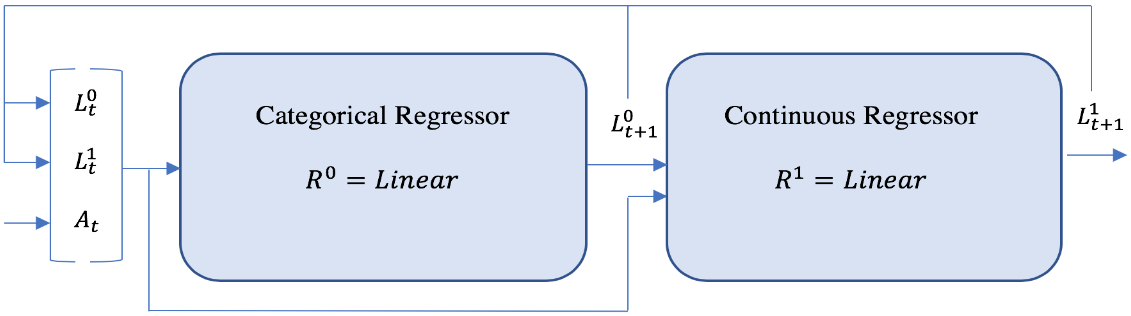

Figure 4-2: M1: Linear Regression Baseline

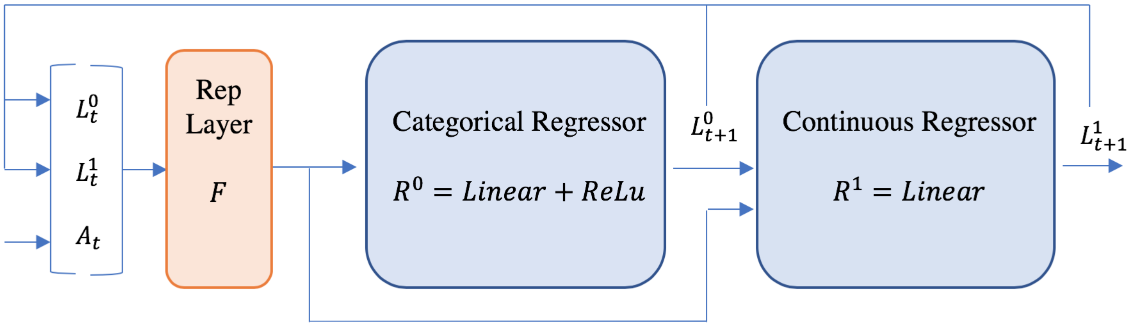

Figure 4-3: M2: Representational Layer + 2 Linear layers

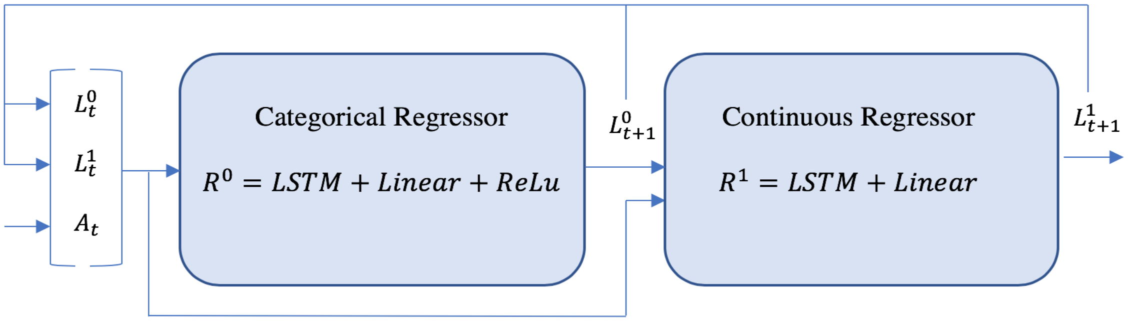

Figure 4-4: M3: 2 LSTMS, one for categorical variables, one for continuous variables

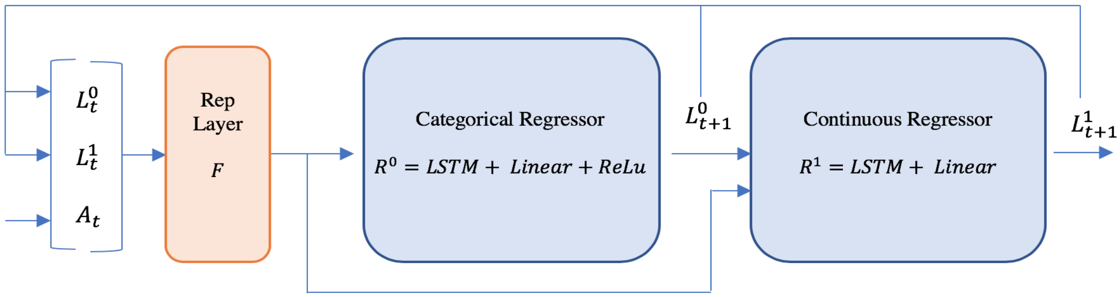

Figure 4-5: M4: 1 Representational Layer + 2 LSTMS, one for categorical variables, one for continuous variables

Chapter 5

CVSim Experiments

Counterfactual prediction is difficult to evaluate on real world datasets because real world datasets only contain the observations under the observational regime. There is no ground truth to evaluate against for the counterfactual policies. For this reason, a physiological simulator was used to generate the datasets under the counterfactual strategies of interest to be used for evaluation of G-Net, in addition to a generated training set under the observational regime. Three datasets were generated using the physiological simulator: one training set, D_O, generated using the observational regime S_O, and two test sets D_{C1}, D_{C2}, generated using the counterfactual policies S_{C1} and S_{C2}, respectively.

5.1 CVSim Dataset Generation

The physiological simulator used to generate the ground-truth is CVSim, a program that simulates the human cardiovascular system. CVSim takes as input a few variables that define the hemodynamic system. Using these input variables as settings for the 6-compartment circulatory model, CVSim deterministically simulates forward 25 output variables under the hemodynamic model. Some of the output variables include arterial pressure (AP), central venous pressure (CVP), total peripheral resistance (TPR), total blood volume (TBV). Four extra variables were derived from the 25 CVSim ouptut variables: systolic blood pressure (SBP), diastolic blood pressure

(DBP), mean blood pressure (MBP), and a pulmonary edema binary indicator. A subset of these 28 CVSim output and derived variables are included in the covariates L_t. Details on the variables included can be found in Appendix A.

In a clinical setting, physicians try to keep AP and CVP at some reasonable level by administering fluid (adjusting TBV) or by giving vasopressors (adjusting TPR). In this project, the intervention of interest is a vector of 2 interventions, $A_t = [A_t^1, A_t^2]$: bolus amount and vasopressor amount. Bolus administration was mimicked by increasing the TBV. Vasopressor administration was mimicked by increasing TPR.

Observational intervention S_O is a *stochastic* intervention regime. Under S_O, the amount of fluid and vasopressor administered at each timestep varies according to a logistic function with respect to the change in CVP and change in MAP. The first counterfactual strategy, S_{C1}, is similar to S_O, except it is a *deterministic* strategy with different coefficients for the function to calculate treatment from the covariates. The second counterfactual strategy, S_{C2}, is a strategy where treatment is always withheld. In other words, no bolus or vasopressor is ever administered under S_{C2}. The exact equations for S_O and S_{C1} are given in Appendix A.

D_O is generated in its entirety using the observational strategy S_O. D_{C1} and D_{C2} were generated under S_O for the first m timesteps. From timestep m and on, D_{C1} and D_{C2} were generated under S_{C1} and S_{C2}, respectively. This method of dataset generation is shown in Figure 5-1. There were a total of 10,000 samples in D_O, and 1000 in each of D_{C1} and D_{C2}, and with outlier removal the final numbers were 8282 samples in D_0, and 895 samples in D_{C1} and 895 samples in D_{C2}. For D_{C1} and D_{C2}, $m = 34$ timesteps. For D_O, D_{C1}, and D_{C2}, the total number of timesteps was 66.

5.2 CVSim Experimental Setup

The four models described in Section 4, $M1 - M4$, are experimented with on the CVSim datasets. Each of $M1 - M4$ were trained on the observational dataset, D_O and then evaluated on both D_{C1} and D_{C2}. During evaluation time on test set D_{C1}, the models were first fit to the first m timesteps of D_{C1}, and then at each time t,

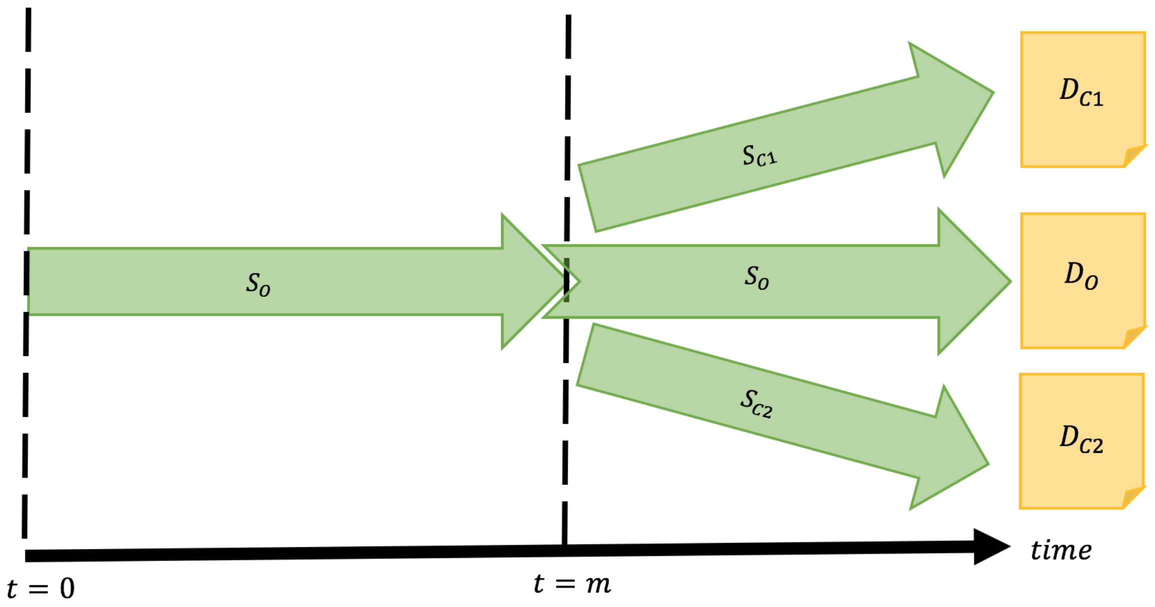

Figure 5-1: CVSim Data Generation Procedure. D_O is the training dataset generated under the observational regime S_O. D_{C1} and D_{C2} are the counterfactual test sets generated under intervention strategies S_{C1} (treatment) and S_{C2}(no treatment).

for $t > m$, the model predicts the covariates $\hat{L}_t$ and computes an intervention, $\hat{A}_t$, under the intervention strategy S_{C1} using $\hat{L_{1:t}}, \hat{A_{1:t-1}}$. The same process is repeated for all models on D_{C2}, except the intervention would set $A_t = [0, 0]$ now since D_{C2} is the no-treatment counterfactual test set. For each sample (conceptually a sample is a patient) from D_{C1} and D_{C2}, $K = 100$ Monte Carlo simulations were generated by the models, using the simulation method outlined in Section 4.4. A total of $K \times N$ simulations were generated for each of S_{C1} and S_{C2}, where N is the total number of patients in the respective test set.

5.3 CVSim Evaluation Methods

The $K \times N$ Monte Carlo simulations from the models $M1 - M4$ can be compared against the "ground-truth" generated by CVSim, given in D_{C1} and D_{C2} for timesteps $t > m$. Many different evaluation methods are explored. All of the following evaluations in combination will give an accurate assessment of the performance of these models.

5.3.1 MSE Over Time

The first check will be of how close the models predictions of the covariates, $\hat{L_t}$, are to the ground-truth generated by CVSim, L_t. The metric used to measure the difference between the two is MSE. For each patient, the $K = 100$ simulations are averaged into an average trajectory for that patient. This average trajectory is then compared to the actual trajectory given in D_{C1} and D_{C2} by computing the MSE at each timestep of simulation. The MSE is the average MSE across all covariates at the timestep. The MSE is then averaged across all the patients in the counterfactual test set. Each of $M1 - M4$ has a MSE over time plot. These plots are compared to see which model has the lowest MSE over time.

5.3.2 Calibration Over Time

The second check will be of how well calibrated the simulations are. Ideally, the actual trajectory will be within the ranges of all the Monte Carlo simulations corresponding to that actual trajectory, and near the average of the Monte Carlo simulations. For each simulation timestep, the proportion of trajectories over all variables that fall within the 5th and 95th percentiles of their respective K Monte Carlo simulations is calculated. For a well-calibrated model, 90 percent of the actual averages will fall within the 5th and 95th percentile of all the Monte Carlo simulations. The calibration over time can be plotted for each of these models, $M1 - M4$, to visualize how well calibrated the models are over time.

5.3.3 Population Level Average Trajectory

For each covariate, the actual average population trajectory can be plotted to see if it falls within the 5th and 95th percentile of the $K \times N$ simulations. The predicted averages from G-Net can be plotted as well to see if the predicted average is close to the actual average. The actual average should fall within the 5th and 9th percentiles of the $K \times N$ simulations and be fairly close to the predicted average across the $K \times N$ simulations.

5.3.4 Individual Patient Monte Carlo Visualizations

A third check will be a visual check of the trajectories for each of the covariates for a *single* patient, rather than the population level average trajectory. For a few randomly selected patients from the counterfactual test sets, the actual trajectory was plotted along with the $K = 100$ Monte Carlo simulations for that patient. One plot was created for each covariate. If the actual trajectories fall within the ranges for the $K = 100$ Monte Carlo simulations, then this is another piece of supporting evidence that the models are performing well.

5.3.5 Analysis of Average Treatment Effect

A fourth check is of the average treatment effect to make sure the *predicted* treatment effect is close to the *actual* treatment effect. Treatment effect is the difference between the outcomes under treatment, S_{C1} and no-treatment, S_{C2}. To compute the treatment effect of a single variable, the difference of the mean under S_{C1} and the mean under S_{C2} is computed for each patient at each timestep. Then, for each time t, the treatment effect for a particular variable is the average treatment effect across all patients. This procedure is repeated for each covariate. At the end of this, each covariate will have its own treatment effect over time plot with both the treatment effect from the G-Net predictions, and the actual ground-truth treatment effect computed from the test set. The predicted treatment effect should be similar to the actual treatment effect.

Chapter 6

CVSim Results

In this chapter, CVSim results are presented. The performances of models $M1 - M4$ are compared through MSE and calibration over time plots. The best model found was then evaluated by looking at the population level trajectories, individual patient level Monte Carlo simulations, and treatment effect plots.

6.1 MSE over Time

The performances of the various models, $M1 - M4$, were evaluated by looking at the MSE over time. At each timestep, the MSE between the predictions and the ground truth from CVSim was recorded. This was done for both the counterfactual strategy S_{C1}, where there was treatment administered, and the counterfactual strategy S_{C2}, where treatment was withheld. From Figure 6-1, it is evident that the G-Net LSTM models outperform the traditional linear regression model, since they have lower MSE than linear regression. There is also the trend of increasing gaps in performance as the number of simulation timesteps increases. This is expected because LSTMs are known for their abilities to summarize history and perform better than linear regression in the long run. The best G-Net model in terms of the lowest MSE over time is $M3$, the model with 2 LSTMs, one for the categorical regressor, and one for the continuous regressor.

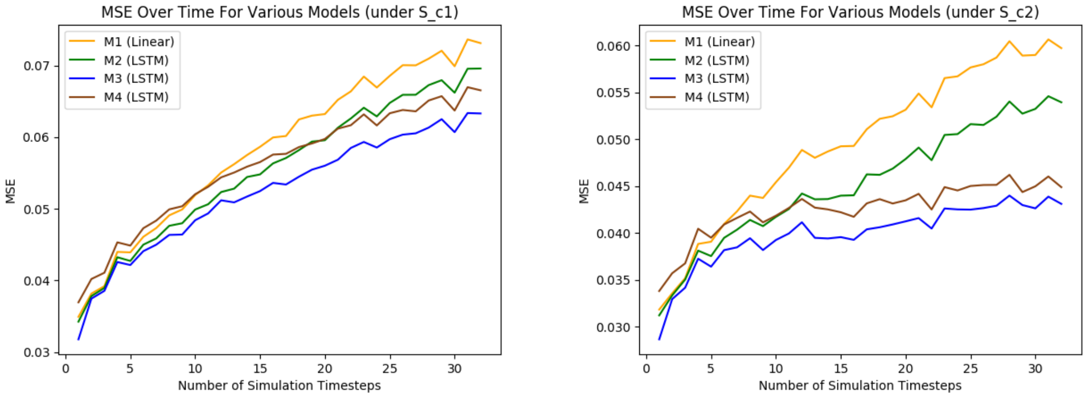

Figure 6-1: MSE Over Time Plots for Models M1-M4 for both S_{C1} (treatment) and S_{C2}(no treatment)

6.2 Calibration Over Time

Besides looking at MSE, it is also useful to look at how well calibrated the models are. For each timestep of simulation, the percentage of actual covariates falling within the 5th and 95th percentile of simulations was recorded. This was done for all models on both counterfactual strategies S_{C1} and S_{C2}. The perfect calibration would be 90 percent. The calibration plots shown in Figure 6-2 show that the G-Net models are better calibrated than the linear regression baseline, in general. The percentage calibrated is not perfect but is reasonable and expected. The calibration is initially higher and then falls as the number of simulation timesteps increases. This behavior

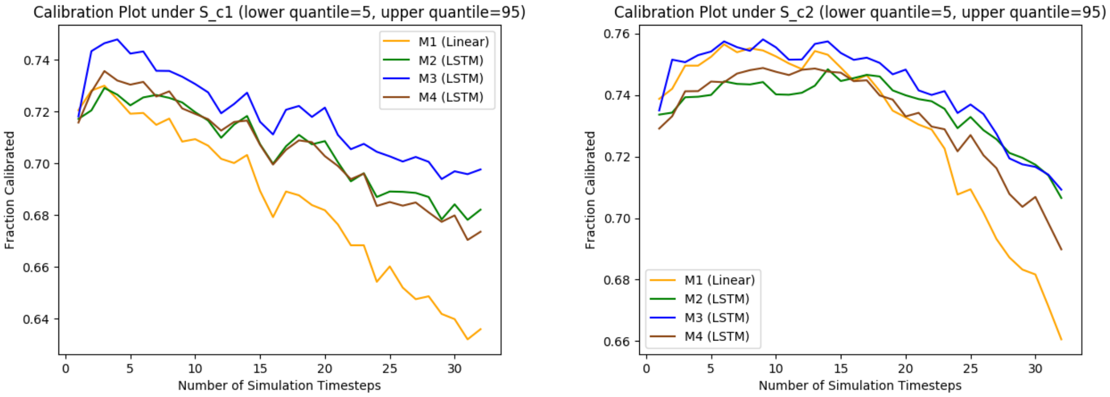

Figure 6-2: Calibration Plots for Models M1-M4 for both S_{C1} (treatment) and S_{C2}(no treatment)

is expected because the longer the simulations run, the more the predictions will diverge from ground truth. The calibration plots also indicate that the G-Net model $M3$ is best due to its higher calibration over time relative to the other models.

6.3 Population Level Trajectory Visualizations

Besides evaluating performance by looking at MSE and calibration, it is also helpful to visualize the population level trajectories for each one of the covariates individually. Figure 6-3 showcases a few population level trajectories for a few selected covariates: heart rate(HR), mean arterial pressure(MAP), left ventricular pressure(LVP), and arterial volume (AV). These results are from the best G-Net model, $M3$, as determined from the previous MSE and calibration plots.

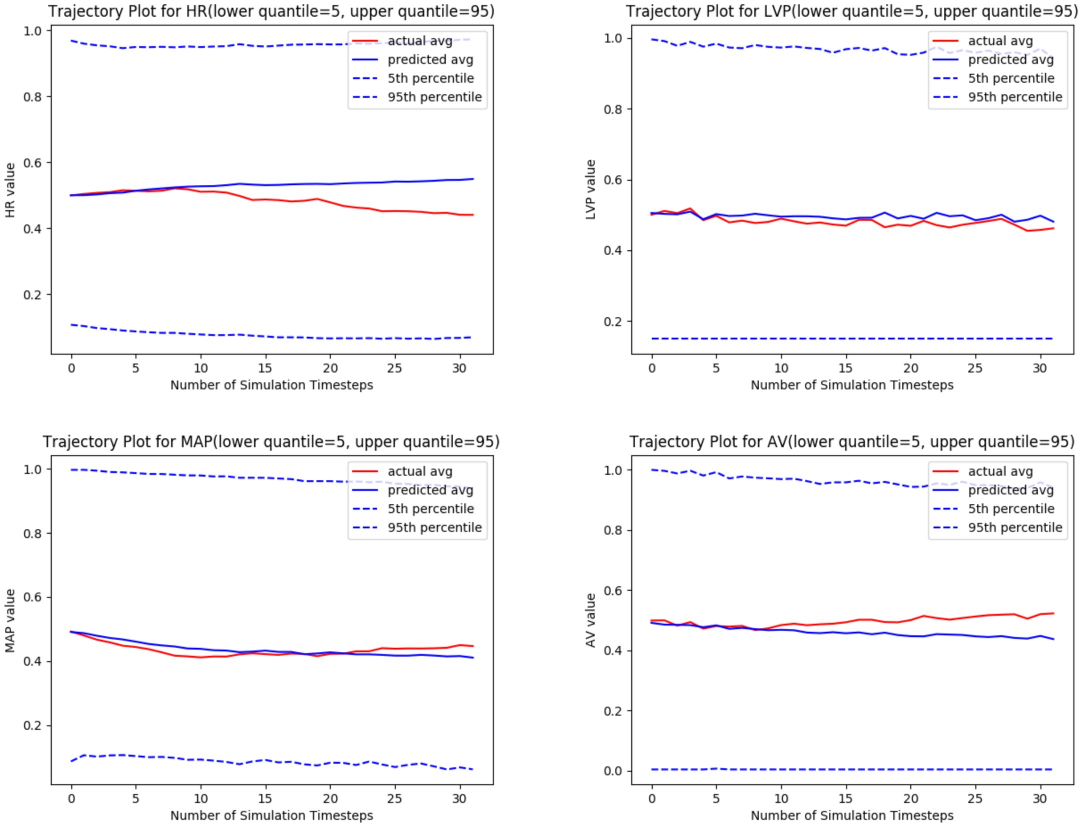

Figure 6-3: Population level trajectory plots for a few selected variables. S_{C1} (treatment).

These results show the average of the trajectories predicted by G-Net with the average of the actual trajectories generated by CVSim. The predicted average and the actual average are fairly close, another indication that the models are performing well. The 5th and 95th percentiles are included in the plots as well. The actual averages fall between the 5th and 95th percentile lines, as expected.

6.4 Individual Patient Level Monte Carlo Simulation Visualizations

It is also possible to view the 100 Monte Carlo simulations for a single patient. The best G-Net model, $M3$, was used to generate the Monte Carlo simulations shown in Figure 6-4. Figure 6-4 shows the Monte Carlo simulations for left ventricular pressure (LVP) and arterial flow (AQ) for a patient selected at random from the test set. The light, semi-transparent blue lines are single Monte Carlo simulations. The solid blue line is the average over all 100 simulations. The red line is the actual trajectory for the variable for this particular patient. The actual trajectory should fall within the ranges of the Monte Carlo simulations, and they do, as expected.

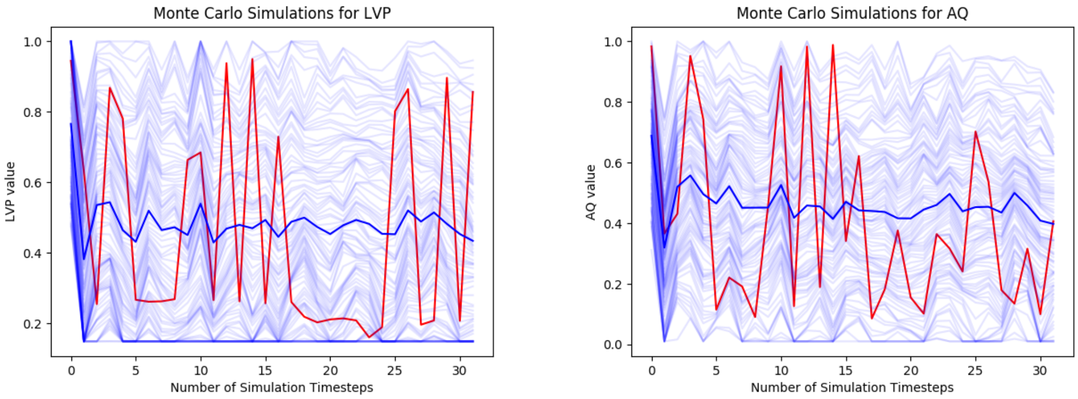

Figure 6-4: Visualizations of 100 Monte Carlo Simulations generated by the best G-Net model.

6.5 Analysis of Average Treatment Effect

Figure 6-5 shows the average treatment effect between S_{C1} (treatment) and S_{C2} (no treatment) for a few variables. The best G-Net model, $M3$, was used to generate the results given in Figure 6-5.

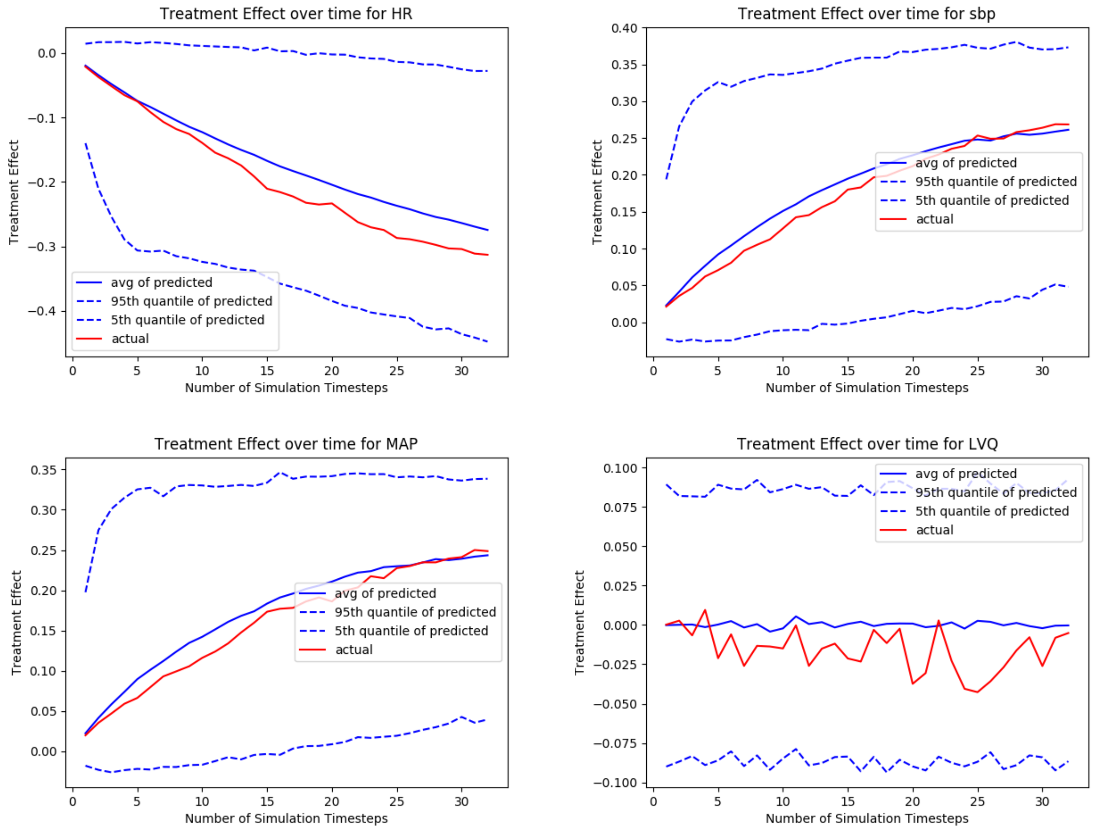

Figure 6-5: Comparison of the average treatment effect predicted by the best G-Net model and the actual average treatment effect.

The actual treatment effect, shown in red, is compared with the predicted treatment effect, shown in blue. The 5th and 9th percentiles of the predicted average treatment effects are shown as well. The predicted treatment effect is close to the actual treatment effect, and the average treatment effect falls within the 5th and 95th percentile lines as expected.

The treatment effect curves seem to suggest that a joint intervention of fluids and vasopressors will decrease the heart rate, increase blood pressure, and not have much effect on variables like left ventricle flow (LVQ).

Chapter 7

MIMIC Experiments

Previously, the performance of the G-Net was evaluated on a dataset generated by a simulator, CVSim. However, ideally, G-Net would be used on real-world datasets and be able to predict what would happen to patients under counterfactual strategies after training on a real-world observational dataset. This chapter explores a real world clinical use case, where the cohort for the clinical study was extracted from MIMIC.

7.1 A Clinical Application: Effects of Fluid Administration

For patients in the ICU, physicians typically administer fluids to patients to increase blood pressure if it is too low. However, too much fluid has adverse effects, such as pulmonary edema. It would be useful to know what would happen to the patient under different fluid intervention strategies. This would allow physicians to test out different strategies to find the one that would result in greater survival rate and better health trends for the patients, without having to experiment on the patients through trial and error in real life.

Fluid administration strategies are typically dynamic and time varying. At each time step, the amount of fluid administered is dependent on the patient's current health status and the patient's health trends in the previous few time steps. The counterfactual prediction problem involving dynamic, time-varying strategies calls for the use of g-computation. In this clinical setting, the intervention, A_t, is the amount of fluid administered at each timestep.

7.2 MIMIC Data Extraction and Processing

The cohort of this clinical study of fluid administration strategies was limited to patients who had a pre-ICU fluid amount recorded under Metavision from the MIMIC database. (Data extraction was limited to Metavision because it has better documented pre-ICU fluid information.) After this filter was applied the total cohort size was 11,919 patients.

The variables of interest included in this study included both static and time-varying variables. Static variables include demographic information (gender, age, etc.) and previous health history information (end stage renal failure, congestive heart failure, diabetes, etc.). Time varying variables include patient vital signs (heart rate, blood pressure, temperature, etc.) and lab values (blood urea nitrogen, lactate, etc.). The full list of variables included is charted in Appendix B.

Time varying variables were binned by the hour. Vital signs are typically more frequent than lab values. There may be more than one measurement recorded for a particular variable in a single hour. If that is the case, the average over all measurements within that hour is summarized. If no measurements were charted in a particular hour, a NA is recorded. The NA values were then prorated by filling forward the most recent measurement. If a patient trajectory starts with NA values, the NA values are replaced with zeros.

Both time-varying variables and static variables can be continuous or categorical. Continuous variables were normalized by subtracting the mean and dividing by the standard deviation for that variable: $x = \frac{x-\overline{x}}{\sigma(x)}$. Categorical variables were one hot encoded.

Unlike the CVSim experiments, the experiments with the MIMIC cohort included censoring variables. There are four censoring variables: death, mechanical ventilation (MV) initiation, initiation of diuretics administration, and discharge from hospital. These variables are binary indicators. As soon as a 1 is encountered for a patient for any one of these censoring variables, the patient is censored, meaning no training data is available for that patient from the censoring timepoint and onward. Patient data was limited to a maximum of 240 hours after ICU admission. If a patient dies within 240 hours, their death indicator is set to 1 for the hour of death. The diuretics indicator is set to 1 when diuretics is initiated. The MV indicator is set to 1 when MV is initiated. If the patient record is less than 240 hours, and the patient has not yet been censored due to death, diuretics initiation, or MV initation, the release indicator was set to 1 for the final hour of data, meaning the patient was discharged from hospital.

7.3 MIMIC Evaluation Methods

An issue with using a real-world clinical datasets is that only the outcomes under the observational regime are available. It is not clear what the ground-truth is for different *counterfactual* strategies. In this case, it is impossible to evaluate the *counterfactual* predictive abilities of G-Net, since there is no ground-truth for the counterfactual strategies to compare against. However, it is still possible to evaluate the predictive abilities of G-Net on the *observational* dataset. If the G-Net has accurate predictions for the observational dataset, it should do reasonably well in predicting the outcomes under counterfactual strategies, assuming the conditional distributions under the observational regime and the counterfactual regimes are similar.

It is also useful to try a few real-world counterfactual policies on the MIMIC cohort to see what would have happened had physicians followed a specific alternative fluid administration strategy. Two counterfactual strategies were experimented with in this study: one aggressive fluid administration strategy, S_{C1}, and one conservative fluid administration strategy, S_{C2}. The aggressive counterfactual strategy is to give 1000 mL of fluid if blood pressure is below 65. The conservative counterfactual strategy is to give 200 mL if blood pressure is below 55. The results are then analyzed to see if they are reasonable in a clinical sense.

Aggressive Counterfactual Regime: S_{C1}

```
if $L_t^{MBP} \leq 65$ then
    $A_t = 1000$ mL
else
    $A_t = 0$
end
```

Conservative Counterfactual Regime: S_{C2}

```
if $L_t^{MBP} \leq 55$ then
    $A_t = 200$ mL
else
    $A_t = 0$
end
```

7.4 MIMIC Experimental Setup Details

Only the best G-Net model, $M3$, was used for the MIMIC experiments. Its predictive performance on the observational dataset was evaluated. Then, it was used to illustrate the effects of the two counterfactual strategies on the covariates, L_t.

The cohort of patients with pre-ICU fluid charted totaling 11,919 individuals was split into a training set, a validation set, and a test set, using a 60-20-20 split, resulting in 7151 in the training set, 2383 in the validation set, and 2385 in the test set. The G-Net model was first trained using the training set, with the hyper-parameters tuned for using the validation set.

The experimental setup during testing is different from the CVSim setup. With the MIMIC experiments, the treatment, A_t, is also predicted along with L_t at each timestep, since only the predictive performance of the model is evaluated on the test set from the observational regime; there is no counterfactual strategy to define A_t at each timestep. During test time, the G-Net model is first fit to the b timesteps of data as the baseline ($b = 5$ for the results presented in the next chapter). Then, starting at time $t = b + 1$, K trajectories were simulated for each patient ($K = 5$ for the results presented in the next chapter), giving a total of $K \times N$ simulations, where N is the total number of patients in the test set. If at any time during the simulations, a censoring variable becomes 1, the trajectory is censored and not included in the population level analysis from that censoring time step and onward.

For each variable, the $K \times N$ simulations from G-Net are averaged for each time point of the simulation window. This results in a population average predicted trajectory curve for each variable. This curve is plotted against the actual population average trajectory curve calculated from the test set. The actual average for each variable at each timestep is calculated by taking the average over the N patient trajectories in the test set. These predicted and actual population average trajectory curves are compared to see if they are close.

Using the trained G-Net model, counterfactual strategies S_{C1} and S_{C2} were experimented with. Unlike previously with the predictive check on the observational test set, now A_t is no longer predicted by the model. Rather at each timestep, A_t is computed based on the counterfactual strategy of interest. These A_t are then used along with $\hat{L_t}$ to predict L_{t+1}. The simulation procedure is the same as the simulations for the predictive check, where population level averages are computed resulting in population level trajectory curves. The population level trajectory curves under the aggressive counterfactual strategy, S_{C1}, and the conservative counterfactual strategy, S_{C2}, are compared to see if the difference between the two makes clinical sense.

Results of the predictive check of the best G-Net model on the observational test set and the counterfactual illustrations are shown in the next chapter.

Chapter 8

MIMIC Results

8.1 Check of Predictive Performance

Figure 8-1 shows the predictive abilities of the best G-Net model, $M3$, under the observational regime on a few of the censoring variables. These variables are categorical(binary), taking the values 0 or 1. These plots show the fraction of the test set population that have been censored due to death or diuretics initiation at each time step, starting at the first simulation timestep and continuing through until the end of the simulation window. The predicted curve is very close to the actual curve, which shows that the G-Net model performs well on predicting categorical variables.

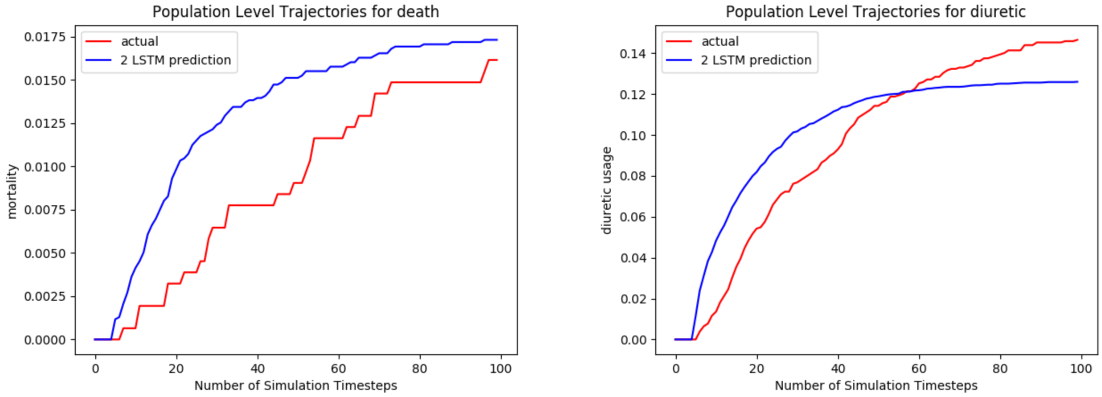

Figure 8-1: The population level average trajectory plots for a few censoring variables: death and diuretics initiation.

Figure 8-2 shows the best G-Net model's predictive performance on a few continuous variables: systolic blood pressure, diastolic blood pressure, respiratory rate, and temperature.

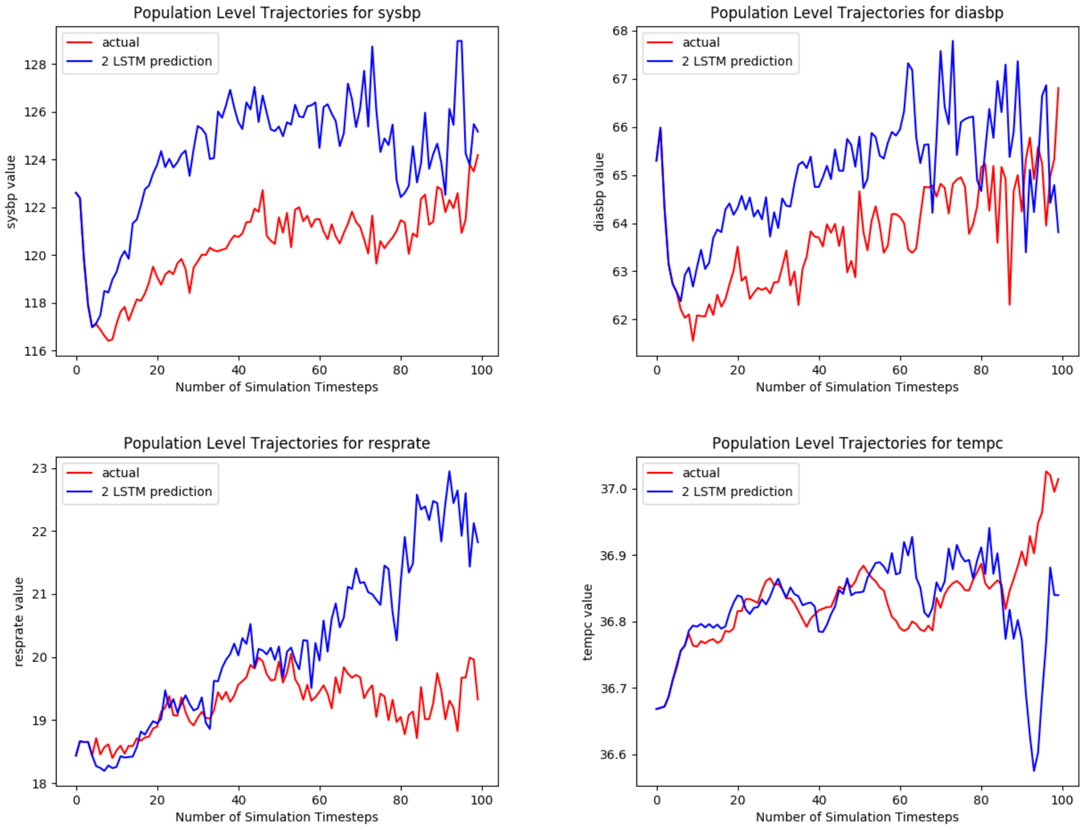

Figure 8-2: The population level average trajectory plots for a few time-varying covariates: systolic blood pressure, diastolic blood pressure, respiratory rate, and temperature.

The predicted trajectory and the actual trajectory are very close, upon inspection of the clinical ranges for these values. The clinical differences between the two curves are almost negligible, and shows that the G-Net model can perform well on continuous variables as well.

8.2 Illustration of Effects of Counterfactual Strategies

Two counterfactual strategies, S_{C1}(aggressive policy) and S_{C2}(conservative policy), were experimented with to see what would have happened to the MIMIC cohort under those strategies. Details on S_{C1} and S_{C2} are in Chapter 7. The plots in Figure 8-3 show the population level effects of an aggressive and a conservative fluid administration policy on certain variables. Usually, if more fluids are administered, the blood pressure would be higher, the urine output should increase, and the blood urea nitrogen (BUN) score should decrease. The plots in Figure 8-3 supports this. The fact that the results make clinical sense indicates that the G-Net model is performing well on counterfactual prediction tasks.

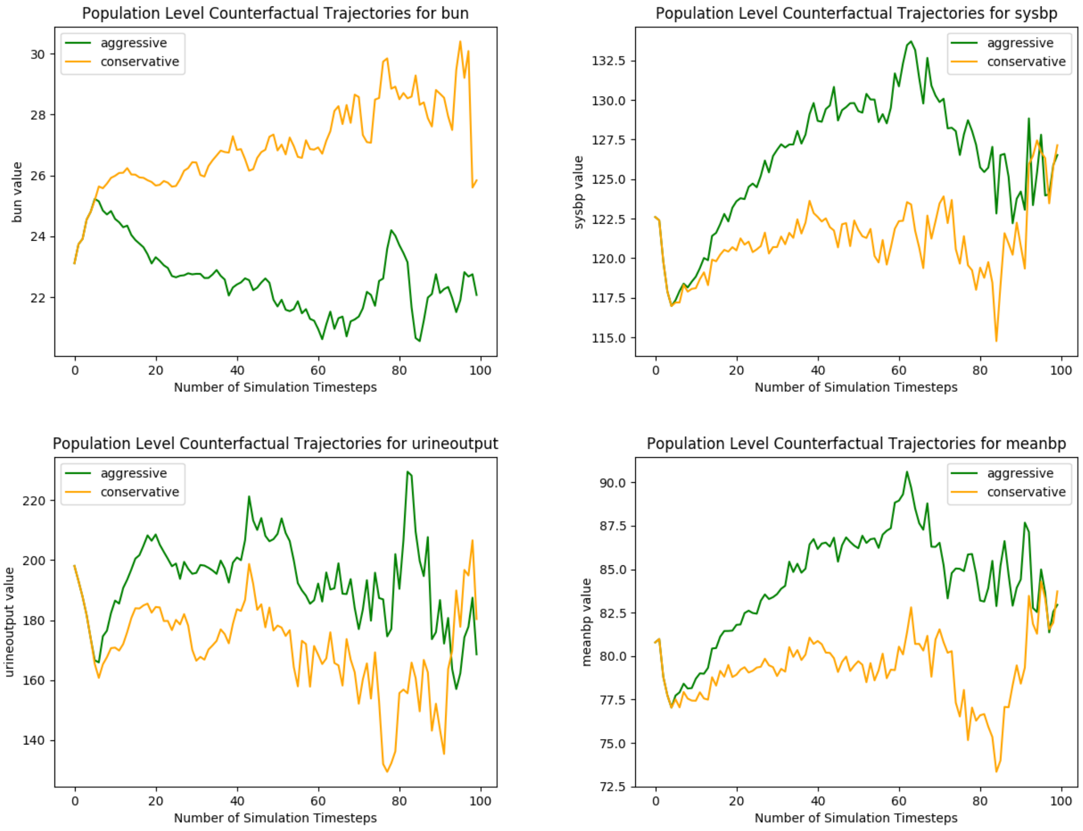

Figure 8-3: The effect of the counterfactual strategies on some variables of interest using the best G-Net model

Only two simple counterfactual policies S_{C1} and S_{C2} were explored in this project, however, any complicated dynamic, time-varying fluid administration strategy could be implemented. Physicians can use G-Net to test various counterfactual strategies of interest to see what would have happened to the population.

Chapter 9

Discussions and Future Work

In this project, a *non-parametric* approach to G-computation using recurrent neural networks was explored. However, it is possible to assume different parametic forms for the conditional distributions, such as Gaussian, Poisson, exponential, etc. One potential area for future work is to explore parametric implementations of G-Network. Parametric implementations might work well in the case that strong assumptions could be made about the distribution in the dataset because they are more tailored to the specific distribution.

Another potential area of future work is to explore different models for the G-Net framework. In this project, only a few models were explored. However, many improvements can be made to these models, such as the addition of a temporal attention mechanism to improve performance on datasets where there may be long range dependencies.

Chapter 10

Conclusion

This thesis introduces a novel recurrent neural network approach to G-computation, dubbed G-Net. A few specific implementations were explored for the G-Net framework. The CVSim results show that G-Nets perform better in counterfactual prediction tasks than the traditional linear regression implementation of G-computation. The MIMIC results illustrate the accurate predictive abilities of G-Net under the observational regime, and the clinical plausibility of the counterfactual predictions from G-Net.

G-Net can be used for counterfactual prediction tasks where the strategies of interest are time-varying and dynamic, and aid in the decision of which intervention strategy is ideal and leads to the best outcome. Though this thesis only explored clinical datasets and clinical interventions, G-Net can be used in any real-world setting where it is useful to know counterfactual outcomes under various dynamic, time-varying intervention strategies.

Appendix A

CVSim Dataset Details

CVSim is an open-source simulator of the human cardiovascular system [5]. The dataset generated for this study was derived from the waveforms output of the 6 compartment CVSim model. CVSim is a deterministic model, where the output at the next instance is fixed given the input to the CVSim model. Since a distribution of possible values was needed for this project, the dataset created using the CVSim model had some stochasticity injected during the dataset generation procedure.

A.1 CVSim Variables

A.1.1 Inputs Variables to CVSim

CVSim takes as input some values for certain hemodynamic parameters. To simulate many trajectories, input values were sampled randomly from the range of values allowed by CVSim. The input variables and their corresponding ranges are given in Table A.1.

A.1.2 Output Variables from CVSim

At each time point, the 6 compartment CVSim model generates 25 hemodynamic outputs. For this project, 4 additional outputs were derived from the 25 hemodynamic outputs: diastolic blood pressure(DBP), systolic blood pressure (SBP), mean blood

Table A.1: CVSim Input Variables and Ranges

Input Variables	**Range**
Arterial Compliance	0.4 - 1.1
Nominal Heart Rate	40 - 60
Pulmonary Arterial Compliance	0.1 - 19.9
Pulmonary Microcirculation Resistance	0.4 - 1.0
Total Blood Volume	1500 - 1600
Total Peripheral Resistance	0.1 - 1.4
Total Zero-Pressure Filling Volume	500 - 3500

pressure(MBP), and pulmonary edema indicator flag. The DBP is the *lowest* arterial blood pressure measurement over a time interval. The SBP is the *highest* arterial blood pressure measurement over a time interval. The MAP is average pressure during one cardiac cycle. It defined mathematically as $MAP = \frac{2}{3}DBP + \frac{1}{3}SBP$. The pulmonary edema flag is 1 if pulmonary venous pressure (PVP) is above 25, else 0. The pulmonary edema indicator is a categorical(binary) variable.

Table A.2 lists the various variables generated as output by CVSim. The final column in the table indicates whether the variable was included as a covariate L_t in the project. Some output variables were dropped to satisfy the assumptions necessary for G-computation. A total of 20 variables were included in L_t.

The output variables were normalized by the generalized logistic (GL) normalization method, since the GL normalization method tends to be robust to outliers, and has better performance on smaller datasets [3].

A.2 Dataset Generation Procedure

A.2.1 Introducing Instability and Stochasticity

CVSim is deterministic, meaning once the system is initialized with a set of input variables, the trajectory will be fixed. Realistically though, there would be unpredictability due to instabilities in a patient, such as disease. For this study, some probability of disease is introduced at each timestep. There are two types of disease: *sepsis* and *blood loss*. At each timestep, there can only be sepsis or blood loss, not

Table A.2: CVSim Output Variables and Abbreviations. The final column indicates whether the particular variable was included as one of the covariates, L_t, in the study.

Output Variables	**Abbreviation**	**Included in L_t**
Arterial Pressure	AP	Yes
Arterial Flow	AQ	Yes
Arterial Volume	AV	Yes
Arteriolar Resistance	AR	Yes
Central Venous Pressure	CVP	Yes
Central Venous Flow	CVQ	No
Central Venous Volume	CVV	No
Diastolic Blood Pressure	DBP	Yes
Heart Rate	HR	Yes
Intra-thoracic Pressure	PTH	Yes
Left Ventricle Pressure	LVP	Yes
Left Ventricle Flow	LVQ	Yes
Left Ventricle Volume	LVV	No
Left Ventricle Contractility	LVC	Yes
Mean Arterial Pressure	MAP	Yes
Pulmonary Arterial Pressure	PAP	No
Pulmonary Arterial Flow	PAQ	No
Pulmonary Arterial Volume	PAV	No
Pulmonary Edema	PE	Yes
Pulmonary Venous Pressure	PVP	No
Pulmonary Venous Flow	PVQ	No
Pulmonary Venous Volume	PVV	No
Right Ventricle Pressure	RVP	Yes
Right Ventricle Flow	RVQ	Yes
Right Ventricle Volume	RVV	Yes
Right Ventricle Contractility	RVC	Yes
Systolic Blood Pressure	SBP	Yes
Total Blood Volume	TBV	Yes
Venous Tone	VT	Yes

both. The probabilities of the diseases are given below:

1. Probability of Disease, $P(disease) = 0.05$ at each time t

2. Probability of Blood Loss Given Disease, $P(blood\ loss|disease) = 0.5$

3. Probability of Sepsis Given Disease, $P(sepsis|disease) = 0.5$

If there is sepsis at a given time t, then $TPR_{t+1} = r_s \times TPR_t$, where $0 < r_s < 0.7$. In words, if there is sepsis at time t, the total peripheral resistance is reduced by a factor r_s, which is sampled from a uniform distribution between 0 and 0.7. If there is blood loss at a given time t: then $TBV_{t+1} = r_b \times TBV_t$, where $0 < r_b < 0.95$. In words, if there is blood loss at time t, the total blood volume is reduced by a factor of r_b, which is sampled from a uniform distribution between 0 and 0.95.

A.2.2 Treatment Under the Observational and Counterfactual Regimes

The treatment in this study is a vector of two treatments $[A_t^1, A_t^2]$: $A_t^1 = fluids$ and $A_t^2 = vasopressors$. To mimic the administration of fluids using CVSim, the total blood volume is increased. To mimic the administration of vasopressors using CVSim, the arterial resistance is increased. For this project, the goal of the treatment is to stabilize blood pressure. Therefore, the treatment strategies explored are functions on the mean arterial pressure (MAP) and central venous pressure (CVP). Under the observational regime S_O, the treatment strategy is stochastic function of MAP and CVP. Under the counterfactual treatment regime S_{C1} the treatment is deterministic function of MAP and CVP. (There is no treatment under S_{C2}, so treatment is always 0 at all simulation timepoints).

Observational Regime: S_O

```
$\Delta_t^{MAP} \leftarrow 65 - L_t^{MAP}$
$\Delta_t^{CVP} \leftarrow 10 - L_t^{CVP}$
$P(A_t|L_t) \leftarrow \frac{1}{1+e^{-x}}$, where $x = \alpha_1 \times \Delta_t^{MAP} + \alpha_2 \times \Delta_t^{CVP} + \alpha_0$
$A_t^1 \leftarrow max(0, \beta_1^1 \times \Delta_t^{MAP} + \beta_2^1 \times \Delta_t^{CVP} + \beta_0^1)$, where $\beta_0^1 \sim N(1500, 1000)$
$A_t^2 \leftarrow max(0, \beta_1^2 \times \Delta_t^{MAP} + \beta_2^2 \times \Delta_t^{CVP} + \beta_0^2)$, where $\beta_0^2 \sim N(0, 1)$
```

The parameter values used for the CVSim dataset generation process used in this project are $\alpha_0 = 0.02, \alpha_1 = 0.06, \alpha_2 = 0.24, \beta_1^1 = 10, \beta_2^1 = 60, \beta_1^2 = 0.1, \beta_2^2 = 0.15$.

Counterfactual Regime: S_{C1}

```
$\Delta_t^{MAP} \leftarrow 65 - L_t^{MAP}$
$\Delta_t^{CVP} \leftarrow 10 - L_t^{CVP}$
$shock\_index \leftarrow \frac{L_t^{HR}}{L_t^{SBP}}$
if $L_t^{SBP} \leq 100$ and $shock\_index \geq 0.8$ then
    $P(A_t|L_t) \leftarrow 1$
else
    $P(A_t|L_t) \leftarrow 0$
end
$A_t^1 \leftarrow max(0, \beta_1^1 \times \Delta_t^{MAP} + \beta_2^1 \times \Delta_t^{CVP})$
$A_t^2 \leftarrow max(0, \beta_1^2 \times \Delta_t^{MAP} + \beta_2^2 \times \Delta_t^{CVP})$
```

Appendix B

MIMIC Dataset Details

B.1 Variables Included

B.1.1 Static Variables

Certain static variables were included in the prediction of the covariates $\hat{L}_t$, but were not time-varying, so were not predicted by the model. Some static variables are health history information, such as whether the patient has metastiatic cancer, diabetes, end stage renal failure, etc. Some others are demographic information such as gender, and age. The full list of these static variables included in the study is given in Table B.1.

B.1.2 Time Varying Variables

Vital signs and lab values are examples of time-varying variables, though vital sign measurements are more frequent than lab values. Time varying variables fall into two categories: continuous and categorical. The full list of time-varying variables, along with their units of measurement, included in the study are given in Table B.2.

Table B.1: MIMIC Static Variables Included

Static Variables	**Variable Type**	**Units**
Age	Continuous	years
Pre-ICU Fluid Amount	Continuous	mL
Gender	Categorical(Binary)	N/A
Valvular Disease	Categorical(Binary)	N/A
Peripheral Vascular	Categorical(Binary)	N/A
Hypertension	Categorical(Binary)	N/A
Paralysis	Categorical(Binary)	N/A
Chronic Pulmonary	Categorical(Binary)	N/A
Diabetes Uncomplicated	Categorical(Binary)	N/A
Diabetes Complicated	Categorical(Binary)	N/A
Hypothyroidism	Categorical(Binary)	N/A
Liver Disease	Categorical(Binary)	N/A
AIDS	Categorical(Binary)	N/A
Lymphoma	Categorical(Binary)	N/A
Metastatic Cancer	Categorical(Binary)	N/A
Solid Tumor	Categorical(Binary)	N/A
Rheumatoid Arthritis	Categorical(Binary)	N/A
Coagulopathy	Categorical(Binary)	N/A
Obesity	Categorical(Binary)	N/A
Deficiency Anemias	Categorical(Binary)	N/A
Alcohol Abuse	Categorical(Binary)	N/A
Drug Abuse	Categorical(Binary)	N/A
End Stage Renal Failure	Categorical(Binary)	N/A
Congestive Heart Failure	Categorical(Binary)	N/A

Table B.2: MIMIC Time-Varying Variables Included. An asterisk next to variable means it is the treatment variable, A_t. The rest of the variables listed are the covariates, L_t.

Time-Varying Variables	**Variable Type**	**Units**
Base Excess	Continuous	mmol/L
Bicarbonate	Continuous	mmol/L
Total CO2	Continuous	mEq/L
Hemoglobin	Continuous	g/dL
Lactate	Continuous	mmol/L
SO2	Continuous	%
PCO2	Continuous	mmHg
PH	Continuous	Numerical[1,14]
PO2	Continuous	mmHg
Potassium	Continuous	mEq/L
BUN	Continuous	mmol/L
Temperature	Continuous	°C
Bilirubin	Continuous	mg/dL
Creatinine	Continuous	mg/dL
Platelet	Continuous	counts/10^9 L
Urine Output	Continuous	ml/kg/hr
Heart Rate	Continuous	beats/min
Systolic Blood Pressure	Continuous	mmHg
Diastolic Blood Pressure	Continuous	mmHg
Mean Blood Pressure	Continuous	mmHg
Respiratory Rate	Continuous	breaths/min
SpO2	Continuous	%
Weight	Continuous	lbs
Fluid Amount*	Continuous	mL
GCS	Treated as Continuous	Numerical[3-15]
GCS-Motor	Treated as Continuous	Numerical[1-6]
GCS-Verbal	Treated as Continuous	Numerical[1-5]
GCS-Eyes	Treated as Continuous	Numerical[1-4]
Fluid Amount Indicator*	Categorical(Binary)	N/A
Mechanical Ventilation(MV) Indicator	Categorical(Binary)	N/A
Vasopressor Indicator	Categorical(Binary)	N/A
Death Indicator	Categorical(Binary)	N/A
Release Indicator	Categorical(Binary)	N/A

Bibliography

[1] M Ahmed Alaa, Michael Weisz, and Mihaela van der Schaar. Deep counterfactual networks with propensity-dropout. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 2017.

[2] Onur Atan, James Jordan, and Mihaela van der Schaar. Deep-treat: Learning optimal personalized treatments from observational data using neural networks. In *Proceedings of AAAI*, 2018.

[3] Xi Hang Cao, Ivan Stojkovic, and Zoran Obradovic. A robust data scaling algorithm to improve classification accuracies in biomedical data. *BMC Bioinformatics*, 2016.

[4] Simon Finfer, John Myburgh, and Renaldo Bellomo. Intravenous fluid therapy in critically ill adults. *Nature Reviews Nephrology*, 14(9):541, 2018.

[5] T Heldt, R Mukkamala, GB Moody, and RG Mark. CVSim: An open-source cardiovascular simulator for teaching and research. *Open Pacing, Electrophysiol & Ther J*, 3:45–54, 2010.

[6] Miguel Hernan and James Robins. *Causal Inference*. Chapman and Hall, Forthcoming.

[7] Alistair E. Johnson, Tom J. Pollard, Lu Shen, Li-wei H. Lehman, Mornin Feng, Mohammad Ghassemi, Benjamin Moody, Pete Szolovits, Leo Celi, and Roger G. Mark. MIMIC-III: a freely accessible critical care database. *Scientific Data*, 3(160035), 2016.

[8] Bryan Lim, Ahmed Alaa, and Mihaela van der Schaar. Forecasting treatment responses over time using recurrent marginal structural networks. In *Neural Information Processing Systems (NIPS)*, 2018.

[9] Liliana Orellana, James Robins, and Andrea Rotnitzky. Dynamic regime marginal structural mean models for estimation of optimal dynamic treatment regimes. *The International Journal of Biostatistics*, 2008.

[10] 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*, 1986.

[11] James Robins. A graphical approach to the identification and estimation of causal parameters in mortality studies with sustained exposure periods. *Journal of Chronic Diseases*, 1987.

[12] James Robins. Correcting for non-compliance in randomized trials using structural nested mean models. *Communications in Statistics-Theory and Methods*, 1994.

[13] James Robins and Miguel Hernan. Estimation of the causal effects of time varying exposures. In Garrett Fitzmaurice, Marie Davidian, Geert Verbeke, and Geert Molenberghs, editors, *Longitudinal Data Analysis*, pages 553–599. Chapman and Hall, 2009.

[14] James Robins, Miguel Hernan, and Babette Brumback. Marginal structural models and causal inference in epidemiology. *Epidemiology*, 2000.

[15] Peter Schulam and Suchi Saria. Reliable decision support using counterfactual models. In *Neural Information Processing Systems (NIPS)*, 2017.

[16] Sarah Taubman, James Robins, Murray Mittleman, and Miguel Hernan. Intervening on risk factors for coronary heart disease: An application of the parametric g-formula. *International Journal of Epidemiology*, 2009.

[17] Stijn Vansteelandt and Marshall Joffe. Structural nested models and g-estimation: The partially realized promise. *Statistical Science*, 2014.

[18] Yanbo Xu, Yanxun Xu, and Suchi Saria. A Bayesian Nonparametric Approach for Estimating Individualized Treatment-Response Curves. In *Proceedings of the 1st Machine Learning for Healthcare Conference, PMLR 56:282-300*, 2016.

[19] Jinsung Yoon, James Jordan, and Mihaela van der Schaar. Ganite: Estimation of individualized treatment effects using generative adversarial nets. In *ICLR*, 2018.