ORIE 4999 Final Report Spring 2025
Synthetic Data Generation for Survival Analysis with Panel Data

Members: Jack Jansons, **Mentors:** Arielle Anderer
Github Repository: https://github.com/jcj59/TimeSurvivalGANs/tree/main

0. Introduction

Last semester we looked at clinical trials for prostate cancer. Our main objective was to identify a way to improve clinical trials with a strong emphasis on efficiency. How can we maximize the impact of a clinical trial while minimizing the time, resources, and patients required to run an effective trial. We began by learning about how a clinical trial is structured and what the data from a clinical trial looks like. In particular, we were interested in clinical trials where a series of measurements are taken for each patient multiple times at different time stamps. We were then interested in predicting the time-to-event for a patient given these readings. This is called survival analysis with panel data. We were interested in this setting because if we can effectively predict the time-to-event outcomes for patients given only a few readings, we can make conclusions from our data about the trial without having to wait to observe the events. This can effectively speed up trials and improve efficiency for testing different interventions.

To predict time-to-event outcomes with panel data, we looked at deep survival machines (DSMs). More particularly, we looked at deep recurrent survival machines (DRSMs) as these utilize a recurrent neural network (RNN) to make predictions from panel data as opposed to a single reading. We identified a dataset that fit our setting: the PBC dataset. This dataset comes from the Mayo Clinic and is widely used in survival analysis literature. It was originally collected as part of a clinical trial on primary biliary cirrhosis of the liver, which is a chronic liver disease that slowly destroys the bile ducts within the liver. We trained a DRSM on the PBC dataset; however, we wanted to improve our results. One of the main limitations of using the PBC dataset is there is only data for 312 patients. Increasing the amount of data we train our DRSM model on would drastically improve its performance.

We left off the semester coming up with ideas to overcome this roadblock. Clinical trial data is very difficult to find and get access to because it usually contains sensitive information, so we looked towards generating synthetic data based on the data we already have. That leads us to our goal for this semester. We want to generate synthetic panel data along with time-to-event data for survival analysis with panel data in order to improve the accuracy of our DRSM model on the PBC dataset.

1. Synthetic Panel Data Generation with TimeGAN

1.1 PBC Dataset Overview

As specified in the introduction, we chose to use the PBC dataset. The main draws to this dataset are that it is well documented, publicly available, and is present in survival analysis literature. It is also one of the few survival analysis datasets that contains panel data, so it fits our setting exactly. This dataset contains various readings for 312 patients over the span of the trial.

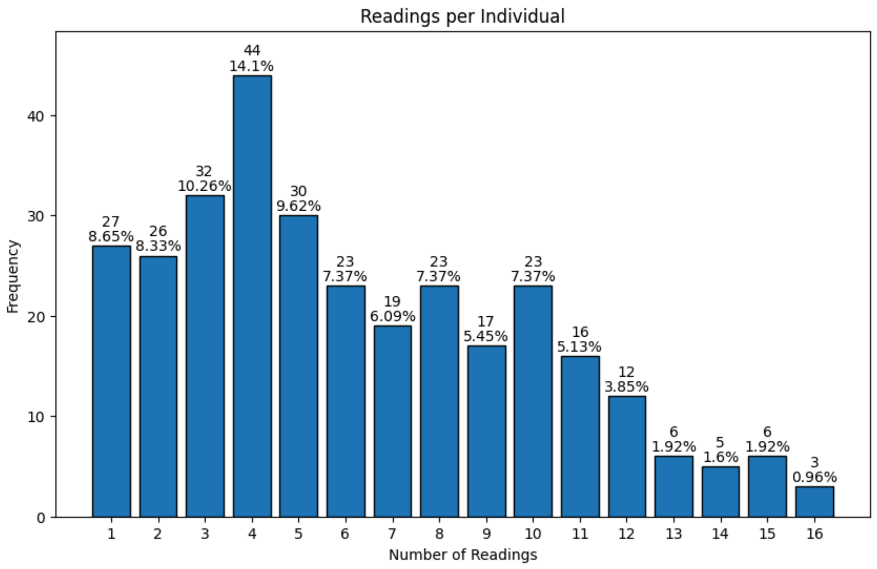

Figure 1: Number of readings per patient in the PBC dataset.

Figure 1 shows the distribution of the number of readings each patient has in the PBC dataset ranging from 1 to 16 readings. This dataset also contains time-to-event data for each patient and whether that event is censored or not.

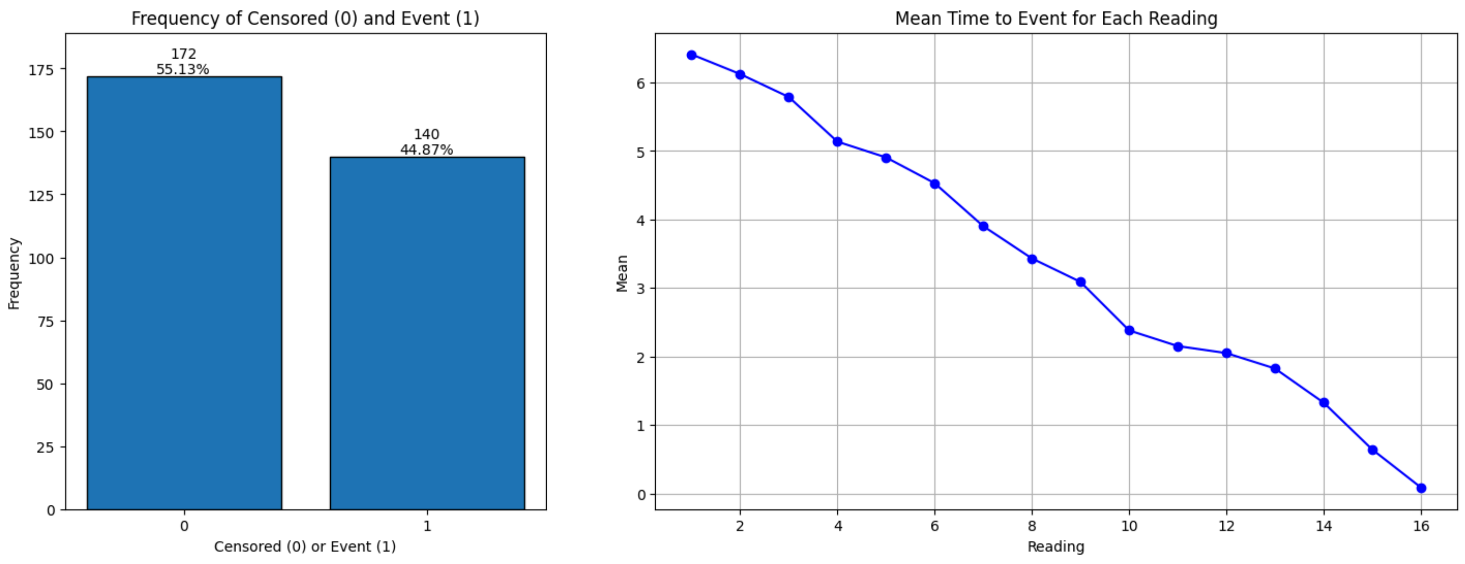

Figure 2: Percentage of censored events in PBC dataset and average time-to-event from each reading.

Figure 2 shows the percentage of censored events in the PBC dataset as well as the average time-to-event from each reading. From the figures we can see that 55.13% of the data is censored and the average time-to-event from when a patient enters the trial is ~6.25 days.

1.2 TimeGan Overview

TimeGANs are a method of generating synthetic panel data. They consist of four main components: encoder network, reconstruction network, generator, and discriminator.

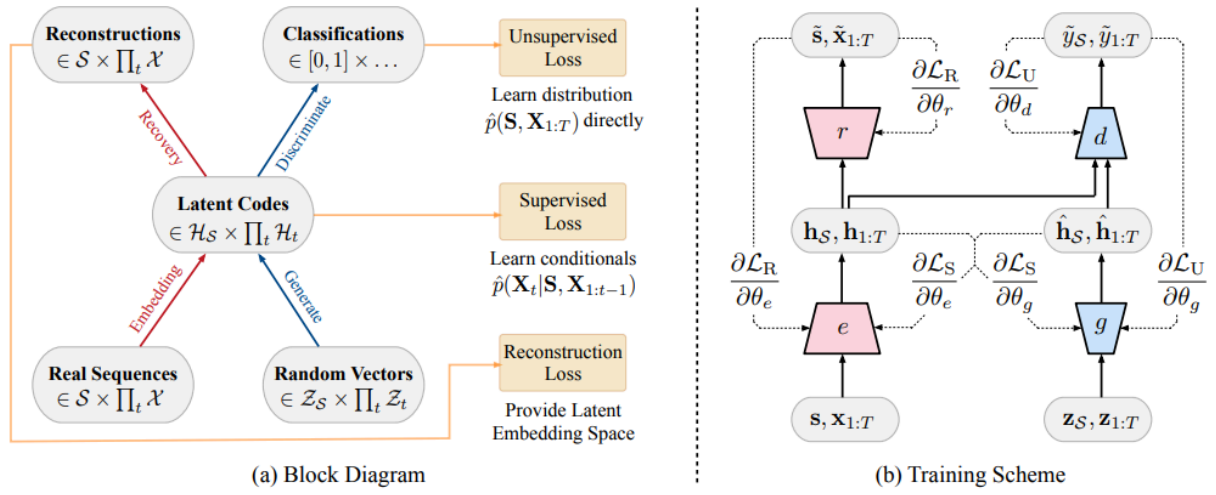
Reconstructions $\in \mathcal{S} \times \prod_t \mathcal{X}$
Classifications $\in [0,1] \times \ldots$
Unsupervised Loss
Learn distribution $\hat{p}(\mathbf{S}, \mathbf{X}_{1:T})$ directly
Recovery
Discriminate
Latent Codes $\in \mathcal{H}_\mathcal{S} \times \prod_t \mathcal{H}_t$
Supervised Loss
Learn conditionals $\hat{p}(\mathbf{X}_t | \mathbf{S}, \mathbf{X}_{1:t-1})$
Embedding
Generate
Real Sequences $\in \mathcal{S} \times \prod_t \mathcal{X}$
Random Vectors $\in \mathcal{Z}_\mathcal{S} \times \prod_t \mathcal{Z}_t$
Reconstruction Loss
Provide Latent Embedding Space
(a) Block Diagram
$\tilde{\mathbf{s}}, \tilde{\mathbf{x}}_{1:T}$
$\tilde{y}_\mathcal{S}, \tilde{y}_{1:T}$
$\frac{\partial \mathcal{L}_\text{R}}{\partial \theta_r}$
$\frac{\partial \mathcal{L}_\text{U}}{\partial \theta_d}$
r
d
$\mathbf{h}_\mathcal{S}, \mathbf{h}_{1:T}$
$\hat{\mathbf{h}}_\mathcal{S}, \hat{\mathbf{h}}_{1:T}$
$\frac{\partial \mathcal{L}_\text{R}}{\partial \theta_e}$
$\frac{\partial \mathcal{L}_\text{S}}{\partial \theta_e}$
$\frac{\partial \mathcal{L}_\text{S}}{\partial \theta_g}$
$\frac{\partial \mathcal{L}_\text{U}}{\partial \theta_g}$
e
g
$\mathbf{s}, \mathbf{x}_{1:T}$
$\mathbf{z}_\mathcal{S}, \mathbf{z}_{1:T}$
(b) Training Scheme

Figure 3: TimeGAN architecture overview showing four main components.

Figure 3 shows the four main components in two different views. Essentially, data from the real dataset is fed into the embedding network which embeds the data into a lower dimensional latent space which is easier to work in. This is a critical step as panel data typically has very high dimensions due to the time aspect. It also allows for coherent data generation across time because the latent space is able to effectively encode time-dependent relationships in the data. The recovery network is then used to try to reconstruct the real data from the latent space. This is critical because it allows us to generate synthetic data in the low dimensional latent space which is trivial using the general GAN architecture. We can then use the reconstruction network to broadcast this synthetically generated latent data into real panel data.

1.3 Model Training and Evaluation

Both the PBC dataset and TimeGANs are readily available in the Synthcity Python package. Using this package, we were able to generate 500 synthetic patients. Figure 4 shows the number of readings per patient in the synthetically generated dataset. Compared to the real dataset, the number of readings is less diverse as it only spans from 2 to 10 readings rather than 1 to 16 readings.

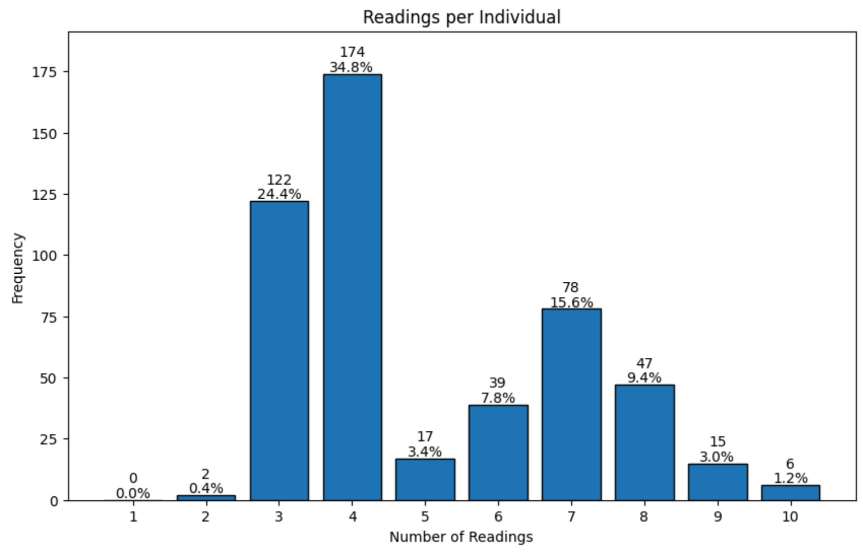

Figure 4: Number of readings per patient in the synthetic data generated by TimeGANs

It appears that the mode is still the same in both distributions which suggests that the TimeGAN possibly takes a greedy approach to tricking the discriminator. This could be an example of literal mode collapse in the training of the TimeGAN. Now that we have generated the panel data, we need to generate time-to-event data that matches this data.

2. Generating Time-to-Event Data

2.1 Deep Recurrent Survival Machines for Survival Modeling

As in the previous semester, we are training a DRSM to predict time-to-event data; however, this time we are using this to generate time-to-event data on the synthetic dataset. We use a DRSM trained on the real PBC dataset to accomplish this task. We can take the model that we trained last semester to generate the pdfs of the time-to-event data for the synthetic dataset. Once we have a pdf for each patient, we can sample times from these pdfs to get the time-to-event data. We are still presented with one issue: how do we determine whether an event is censored or not.

2.2 Simulating Censoring

When determining how to effectively censor our synthetic data, we considered two different possibilities. The first method was from the SurvivalGAN paper which accomplishes that task we have set out on but for single readings of data rather than panel data. They use a time regressor model which takes in the pdf generated by the DSM, the covariates, and whether the event should be censored or not. If the event should be censored, the time regressor predicts the censored time from a combination of the covariates and the pdf. If it is not censored, the time regressor simply samples the event time from the pdf. This is more of a black box machine learning method that appeared hand wavy to us. We would also have to modify the time regressor architecture to a recurrent model to take in panel data instead of a single reading.

Instead of this machine learning time regressor model, we decided to create a simulation model which simply simulated patients entering the clinical trial. Patients entered uniformly throughout the time horizon of the trial. Their predicted uncensored time-to-event value that was sampled from the predicted pdfs are then compared to the time between the patient's arrival and the end of the trial. If the event occurs before the end of the trial, then it is uncensored. However, if the event occurs after the trial ends, the event is censored and the time is reduced to the difference between when the clinical trial ended and when the patient arrived.

We observed in the real data that the longest time horizon for an event was just under 15 days. We ran this simulation multiple times for time horizons of the clinical trial ranging from 5 to 15 days. This resulted in censoring percentages between 22.8% and 44.4%. Now that we had successfully generated synthetic panel data based on the PBC dataset with realistic time-to-event data with censoring, we were ready to test the quality of our synthetic data by training a DRSM model on it.

3. Experiments and Results

We trained a DRSM on our synthetic data using the exact same parameters that we used when training on the real PBC dataset. We kept 20% of the PBC dataset aside as a test set which was not trained on by any of our DRSM models. This allowed us to compare the performance of our models on the PBC dataset. We trained DRSM models for our varying censor rates in our synthetic dataset.

To evaluate the performance of our DRSM on the synthetic data, we used three commonly accepted metrics in survival analysis: Time-Dependent Concordance Index (TD-CI), Brier Score, and Time-Dependent Area Under the ROC Curve (TD-AUC). These metrics assess different aspects of the model's predictive accuracy at various time horizons. The time-dependent concordance index considers whether patients with shorter observed survival times have higher predicted risk at specific time points. Values range from 0.5 (random chance) to 1 (perfect discrimination). A higher TD-CI indicates better performance in distinguishing between high-risk and low-risk individuals over time. The Brier Score evaluates the accuracy of predicted survival probabilities. It is a mean squared error between the predicted survival probability and the actual survival status at a given time. Finally, the time-dependent ROC AUC measures the ability of the model to correctly classify individuals who experience the event by a given time horizon versus those who do not. It evaluates the tradeoff between sensitivity and specificity at each horizon. As with standard AUC, a value of 1 reflects perfect classification, while 0.5 indicates no discriminative ability. Together, these three metrics provide a comprehensive picture of model performance: TD-CI evaluates ranking accuracy, Brier Score measures calibration, and TD-AUC quantifies discriminative ability at various time horizons.

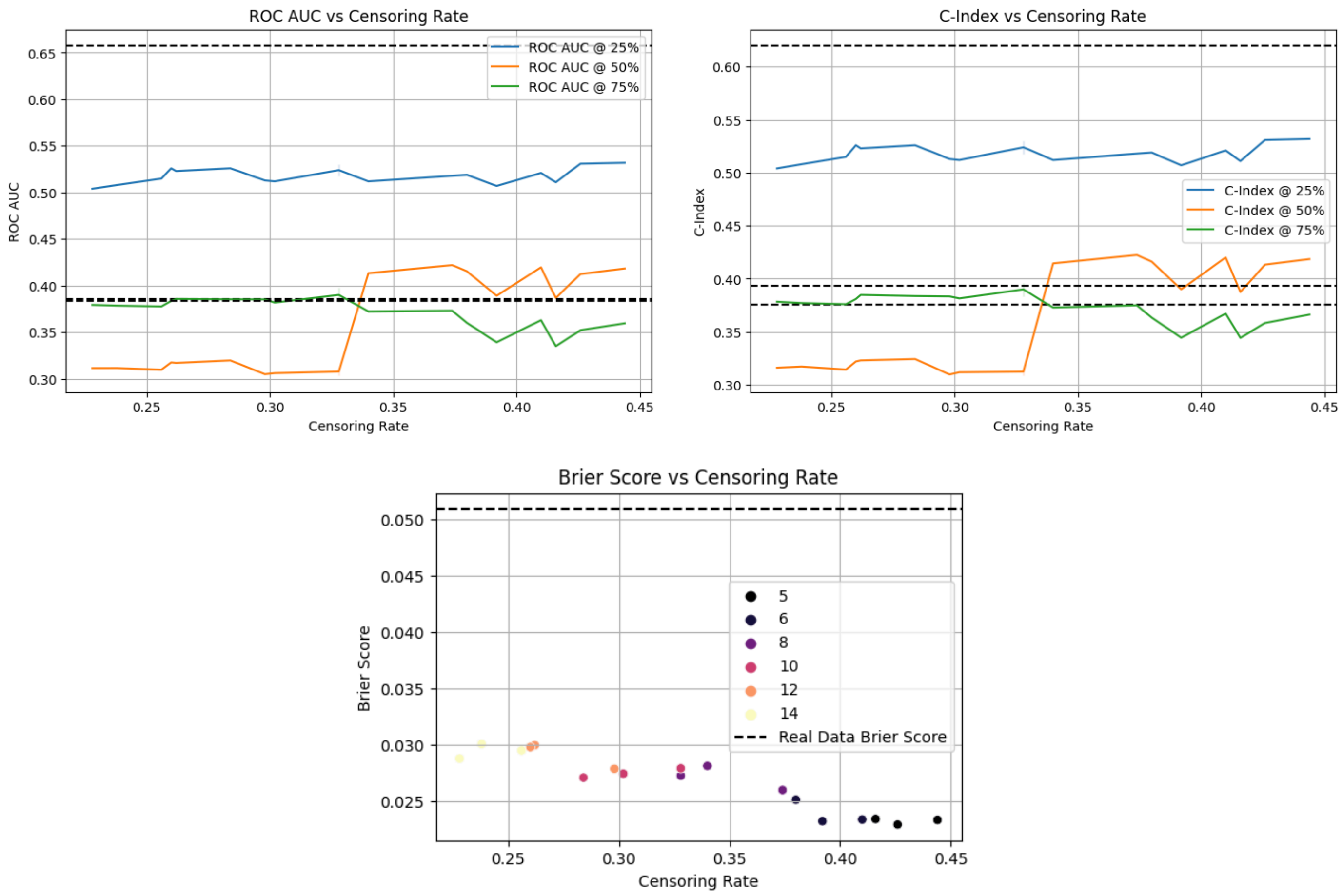

Figure 5: C-Index, ROC AUC, and Brier Scores across different censoring rates in the synthetic data. Black dashed lines correspond to results from real PBC dataset.

Figure 5 summarizes the results of our experiments across different censoring values. Time horizons are split at 25%, 50%, and 75% of the time horizon of time-to-event observations. It is clear in the graphs that the DRSM trained on the original PBC dataset outperforms the models trained on the synthetic data for all censoring rates at 25%. The most significant of these instances is in the brier score graph where the PBC model significantly outperformed all other models. The brier score appears to be most sensitive to censoring rate showing a strong negative linear relationship between censor rate and score while C-Index and ROC AUC appear to be unaffected by censor rate. All models perform similarly at 50% and 75% time horizons. This suggests that the original model is highly accurate for short horizon predictions; however, long horizon predictions are extremely noisy. The models trained on our synthetic dataset have noisy predictions for all time horizons hence why performance is similar at 50% and 75% time horizons but significantly worse at 25%. Likely sources of this error could be inherent to the DRSM model itself which struggles with long horizon predictions in general. The lack of accuracy in short term predictions is likely a result of the quality of the synthetic dataset. Improving data generation quality would likely cause the performance of the model at the 25% time horizon to approach that of the model trained on the real data.

4. Limitations and Challenges

The primary limitations and challenges we faced in our work this semester were related to the quality of the generated synthetic panel data. The quality of data generation is crucial for the quality of our results and is the primary limiter in the accuracy of the models we trained compared to the model trained on the real data. The biggest limitation of TimeGANs is the relationship between values generated over multiple time steps. TimeGANs attempt to construct these relationships by training an embedding and recovery network to represent high dimensional complex time series data in a low dimensional format that is able to implicitly capture these relationships. However, this method is not always effective. The primary example of this can be seen in categorical variables in the data that vary over timesteps. For example, one of the columns in the PBC dataset is the stage of the disease ranging between values 1-4. Typically, the stage stays constant throughout the different readings, but sometimes it will progress to a worse stage or a better stage. However, when we generate the synthetic data using TimeGANs, the stage of disease generated tends to change erratically across different timesteps thus showing that the TimeGAN is unable to capture the true time dependent relationship between stages of disease. Another prominent example is the timestamp of each reading. Typically, these timestamps will range from 0 to a value less than the time-to-event reading. However, in our generated data, these values tend to be constant throughout. The TimeGAN is unable to know that these values should be increasing throughout the readings for a given patient, again showing that it is unable to effectively capture these temporal relationships.

For our primary objective of ultimately speeding up clinical trials, another limitation would be the effectiveness of deep survival machines at long horizon predictions. We have discussed previously how these models are very accurate within the first 25% time horizon but lose accuracy across the 50% and 75% horizons. This is an inherent problem with the DSM model, so a new method that is more effective at long horizon time-to-event predictions would allow our results to improve.

Finally, another major limitation in the generation of synthetic time-series data is there is no satisfactory method of validating this data once it is generated. We have not come across a method in our research that is effective at determining whether data is similar to the real data it is based on. Current methods rely on comparing synthetic and real datasets visually using t-SNE plots or PCA. We can also look at trends that are easily interpreted by humans as discussed above and see if the generated data obeys these trends too.

5. Future Work

We leave this project in a state where much progress has been made, but the results are still unsatisfying. In future work, it would be interesting to do a deeper dive into time-series data generation. That could probably be an entire project in itself unrelated to this problem that we are trying to tackle. Finding a better method than DSMs for long horizon survival analysis prediction is likely a more difficult task to tackle. Data in clinical trial settings is limited, but it would maybe be more interesting to try to find effective methods for similar settings where data is more readily available and then applying this to clinical trials. Finally, more research in validating time-series data once it is generated would also be an interesting path forward for further research as this would facilitate improvements in generating time-series data.

6. References

[1] Huang, Y., et al. (2023). “Application of machine learning in predicting survival outcomes involving real-world data: a scoping review.” BMC Med Res Methodol 23, 268 https://doi.org/10.1186/s12874-023-02078-1

[2] Ishwaran, H., et al. (2008). “Random survival forests.” Ann. Appl. Stat. 2 (3) 841 - 860. https://doi.org/10.1214/08-AOAS169

[3] Nagpal, C., et al. (2021). “Deep Survival Machines: Fully Parametric Survival Regression and Representation Learning for Censored Data.”

[4] Nagpal, C., et al. (2021). “Deep Parametric Time-to-Event Regression with Time-Varying Covariates.”

[5] Nagpal, C. et al. (2022). “Auton-survival: an open-source package for regression, counterfactual estimation, evaluation and phenotyping with censored time-to-event data.”

[6] “Primary Biliary Cirrhosis, Sequential Data.” Education and Research at Mayo Clinic - Education and Research at Mayo Clinic. www.mayo.edu/research/documents/pbcseqhtml/doc-10027141.

[7] “R: Mayo Clinic Primary Biliary Cirrhosis, Sequential Data.” stat.ethz.ch/R-manual/R-devel/library/survival/html/pbcseq.html.

[8] Norcliffe, A., et al. (2023). “Survivalgan: Generating time-to-event data for survival analysis.” International Conference on Artificial Intelligence and Statistics.

[9] Yoon, J., et al. (2019). “Time-series generative adversarial networks.” Advances in neural information processing systems.

[10] Qian, Z., et al. (2023). “Synthcity: facilitating innovative use cases of synthetic data in different data modalities.”