Generating Synthetic Electronic Health-Record Data

COMP 400

Alexander Petrov

260927451

McGill University

Dr. Joëlle Pineau

Dr. Marc-André Legault

Dr. Louis-Antoine Mullie

April 28, 2023

Abstract

Denoising diffusion probabilistic models (DDPMs) have shown great potential for generative modeling. Traditionally used for applications in computer vision, they have found their place among many different domains of application, such as healthcare. However, generative models in the medical sector are more difficult to train, because the data is limited and very costly to obtain. Furthermore, much of electronic medical record data is not image data, which traditional diffusion models were originally designed for, but rather are tabular in nature. TabDDPM is an adaptation of the traditional diffusion modeling method in that it enables learning on heterogeneous data consisting of a mix of numerical and categorical tabular data, using the general diffusion framework. In this paper, we apply and adapt the ideas of the TabDDPM model to see whether it can accurately generate synthetic health-record data in the intensive care unit setting. An important component of the TabDDPM model is its neural network. We tested the model on an open data set of ICU patient data, the MIMIC IV data set, using five different neural network architectures. We used 10 laboratory measurements and 4 patient-stay variables for continuous data, and 6 demographic variables for categorical data. Among the five models we have designed, the best model used a ResNet neural network, which managed to preserve the correlation of the measurements from the MIMIC IV data with a mean absolute difference in Spearman cross-correlation coefficients of 0.03736 for the numerical data and of 0.10163 for the categorical data. The high performance of our model's treatment of the numerical data is mostly due to their different treatment in the form of different preprocessing and a different loss function for the categorical variables. The source code for our experiments is available at https://github.com/AlexanderPetrovGH/COMP400

1. Introduction

Generative models in the medical sector are difficult to train, because healthcare data is heterogeneous and access to data is restricted (Strongman et al., 2019). Due to their sensitive nature, access to patient-level electronic health-record data is regulated for legal and ethical reasons, limiting their availability for the development of machine learning models.

Solving the problem of generating synthetic health-record data is thus very important because when data are scarce, or of poor quality, the machine learning models struggle in trying to generalize the patterns of a system, which eventually leads to inaccurate predictions. (Figueira & Vaz, 2022).

In recent literature, machine learning models have been developed with relative success for generative purposes on natural image datasets. For publicly available healthcare datasets, however, few state-of-the-art works have benchmarked the performance of the deep learning models (Purushotam et al., 2018). Thus, synthetic data could find use in machine learning communities by helping establish benchmark datasets, which can then be used to evaluate the performance of different machine learning models. Synthetic data can also help create simulation models to estimate treatment effects or forecast resource usage (Erdemir et al., 2020).

Since the majority of the work on diffusion models deals with homogeneous and abundant data (Kotelnikov et al., 2022), there has been a relative paucity of work focusing on the generation of synthetic data consisting of both numerical and categorical data. Yet, much of real-word data comprises both types of data, warranting the adaptation of standard generative models.

TabDDPM was proposed as a framework to apply diffusion models to mixed data, and showed promising results in several reference datasets. TabDDPM could help accurately model patient data found in electronic medical records, which could be used to explore the impacts of design decisions on database maintenance and extensibility, data quality, accessibility, and analytic capability (Kahn et al., 2012).

In this report, we aim to generate, using the TabDDPM method, synthetic health-record data, that are as close as possible to the original data distribution while providing credible instances that are not represented in the original dataset of MIMIC IV. We believe that these results can not only be used for the development of benchmark datasets for other machine learning models. We further believe that the data can be shared with fewer ethical and legal considerations, as the results would be freed of any ethical and privacy concerns attached to the patients.

2. Methods

2.1 MIMIC-IV

For this project, we used the Medical Information Mart for Intensive Care (MIMIC)-IV dataset. The MIMIC-IV database consists of the healthcare information of over 40 000 de-identified patients residing in the Beth Israel Deaconess Medical Center from 2008 to 2019. It provides information about the inpatients' demographics, vitality measurements, test results, procedures, medication, imaging reports, mortality, and notes from their caregivers (Johnson et al., 2016). The de-identification of the patient data is crucial in making the usage of this data more ethical. Data access to MIMIC-IV requires completing an application process to preserve the participants' privacy. In order to access the dataset, one must complete a training course in research with human participants and sign a data use agreement (Johnson et al., 2016).

The data comes in the form of a character based comma delimited format. To facilitate model development given the computationally intensive nature of diffusion processes, we have focused on a randomly selected subset of 500 patients for model development. For each patient, there were several numerical and categorical features, but we only kept the features that had the least missing data. Ultimately, we ended up having 14 numerical features, and 6 categorical features, for a total of 20 features. Namely, we have days survived pre-admission, days survived post-admission, stay hours, and life days as our patient stay data. Our signals come from the labs of anion, glucose, chloride, CO_2, pH, urea, creatinine, sodium, potassium, and hematocrit. As our categorical features we have gender, race, religion, marital status, admit source, and admit priority.

2.2 Overview of diffusion models

DDPMs are likelihood-based generative models that minimize the difference between the observed and learned data distributions by matching their score functions, i.e. the gradients of their log-likelihoods (Yang et al., 2022). DDPMs do so by first adding random noise to observed instances until the data follows a prior distribution, such as an isotropic Gaussian, then learning to iteratively denoise the data to recover the observed instances (Ho et al., 2020).

2.3 The forward process

The forward process of a DDPM is defined by a Markov chain that adds Gaussian noise to the observed data according to a predefined noise schedule. We thus have that the distribution governing the added noise depends only on the state reached in the previous event. (Markov, 10) The forward process is defined by:

$$Q(X_{1:T}|\, X_0) \;=\; \prod_{t=1}^{T} Q(X_t|X_{t-1})$$

where X_0 is the original data input, X_t is the noised data at timestep t, and T is the total number of timesteps in the diffusion chain.

We further have that the random variables follow a normal distribution (Kotelnikov et al., 2022) $X_{t+1} \sim N(\sqrt{1-\beta_t} X_t, \beta_t I))$, for $0 \leq t \leq T$, where T is a fixed number of time steps, and $(\beta_t)_{1 \leq t \leq T}$ describes the variance schedule at each time step controlling the degree of corruption that is sequentially added to the data.

The distribution from which the random variables are sampled governs how the data will change after each time step. Indeed, we can infer that the expected value $\mathbb{E}[X_{t+1}] = \sqrt{1-\beta_t} X_t$. This indicates that we can expect, on average, for the noisy version of the data at time t, will resemble that of the data at time t. Thus, every subsequent noise addition isn't a drastic change to the data. On the other hand, an important remark is that the random variable X_t depends only on the previous X_{t-1} latent variable (Yang, 2019) , and so for t >> 1, we see that the dependencies on the original X_0 input begins to fade away, which allows us to have a sufficiently noisy version of the original data.

2.4 The Beta Scheduler

The Beta scheduler is a sequence of real numbers which scales and adjusts the parameters of the distribution. (Ho et al., 2020) used a linear schedule increasing from $\beta_1 = 10^{-4}$ to $\beta_T = 10^{-2}$, whereas (Nichol & Dhariwal, 2021) showed that employing a cosine schedule works even better. Indeed, this was our approach as well. This allows these scaling parameters to oscillate within a bounded range, taking a wide range of values at each time step. This in turn helps in preventing the noise from having a bias in adding noise in a specific way.

2.5 The Reverse Process

The heart of the diffusion model is the reverse process described in (Ho et al., 2020). The idea is to predict the noise that had been added, and to remove it sequentially. Recall that the noise added on the next step is conditional on the state of the latent variable at the current time step. In other words, since we had that X_t depended on X_{t-1} for the forward process, we have that for the reverse process, it is in fact X_{t-1}which depends on X_t. We thus obtain the following equation by applying Bayes' theorem iteratively.

$$P_{\Theta}(X_{0:T}) = P(X_T) \prod_{t=1}^{T} P_{\theta}(X_{t-1} \,|\, X_t),$$

where $X_{t-1} \sim N\,(\mu_{\theta}(X_t,\ t), \Sigma_{\theta}(X_t,\ t)\,)$, and X_T is the fully noised version of the input data from which the reverse process begins, μ_{θ} is the mean of the distribution, and Σ_{θ} is the standard deviation of the distribution, which both depend on X_t and t.

2.6 The Loss Function

Since we are looking to obtain a good posterior, a good solution would be to use techniques from Bayesian inference.

Recall that the goal of the DDPM is to learn the data generating distribution. The solution to this is to approximate the distribution of the posterior P(X|D) using some other distribution Q(X), which is easier to work with. In particular, we can optimize the parameters of the approximate distribution using neural networks. This is variational inference.

For such purposes, it is common to use the Kullback-Leibler divergence (Bishop & Nasrabadi, 2006), defined as the expectation of the difference of the logarithm of the probabilities, i.e.

$$KL(\,Q(X)\,||\,P(X|D)\,) := \mathbb{E}_{Q}[log\,(Q(X)\;-\;log\,(P(X|D))]$$

$$= \quad \mathbb{E}_{Q}(log(Q(X))\;-\mathbb{E}_{Q}(log(P(X|D))$$

$$= \quad \mathbb{E}_{Q}[log(Q(X)]\;-\mathbb{E}_{Q}[log(\frac{P(X,D)}{P(D)})]$$

$$= \quad \mathbb{E}_{Q}[log(Q(X)]\;-\mathbb{E}_{Q}[log(P(X,\,D))\;-\;log(P(D))]$$

$$= \quad \mathbb{E}_{Q}[log(Q(X)]\;-\mathbb{E}_{Q}[log(P(X,\,D))]\;-\;\mathbb{E}_{Q}[log(P(D))]$$

$$= \quad (\mathbb{E}_{Q}[log(Q(X)]\;-\mathbb{E}_{Q}[log(P(X,\,D)))\;-\;log(P(D))]$$

We define the left-hand term as - ELBO, and by rewriting we get

ELBO = log (P(X)) - KL ($Q(X)$ || $P(X|D)$), and by non-negativity of KL, we get

ELBO $\leq$ $log(P(X)$, which justifies the name Evidence Lower Bound for ELBO.

Since we want a minimal KL divergence, this implies we want to minimize the - ELBO, and our loss becomes L(θ) $= \; -\,ELBO$

$$= \mathbb{E}_{Q}[log(P(X,\,D))]\;-\;\mathbb{E}_{Q}[log(Q(X))]$$

$$= \;\mathbb{E}_{Q}[\frac{log(P(X,D)}{Q(X)}]$$

The way we optimize this loss function is via a neural network.

2.7 The Neural Networks Used

The neural networks that we have used are the multilayer perceptron (MLP) and the residual neural network (ResNet) architecture.

Our first model is of type ResNet, of layer architecture 128 x 128, and with a first dropout rate of 0.5 and a second dropout rate of 0.5. Our second model is of type ResNet, of layer architecture 128 x 128, with a first dropout rate of 0.2 and a second dropout rate of 0.1. Our third model is of type MLP, of layer architecture 256x128x256, with a dropout rate of 0.2. Our fourth model is of type MLP, of layer architecture 128x128x128, with a dropout rate of 0.2. Finally, as a control for layer depth, we used an MLP with a layer architecture 32x32, with a dropout rate of 0.2.

The idea of the MLP is that each layer of nodes is completely connected to the next layer of nodes. The idea of the fully connected ResNet is to have several blocks, i.e. portions of the neural network which are fully connected like an MLP, but the connections between the blocks are not fully connected. This is realized by feedforward neural networks with shortcut connections, i.e. the connections which skip one or more layers in the network. The outputs of these shortcut connections are added to the outputs of the stacked layers. (He et al., 2016).

We apply adam gradient descent to train these networks, and we use dropout to avoid overfitting.

2.8 Multinomial Diffusion

In addition to the traditional diffusion models on a normal distribution, extensions have also been proposed for categorical random variables over a multinomial distribution (Kotelnikov et al., 2022). Indeed, for categorical data, the distributions are described as follows. Let K denote the number of categories, and let X_t denote a random variable taking values in $\{0,\ 1\}^K$.

The forward process is similar to the traditional approach but here we use uniform noise instead.

Thus, we have $X_{t+1} \sim Cat((1 - \beta_t)X_t + \frac{\beta_t}{K})$, for $0 \leq t \leq T - 1$, and $X_T \sim Cat(1/K)$.

For the reverse direction, as with Gaussian diffusion, we use a neural network to optimize the ELBO, and learn the parameters which characterize the probability distribution.

However, the key way in which TabDDPM differentiates itself from the traditional diffusion models is in its treatment of the mixed data. This leads to two independent diffusions, and thus to two independent losses : one of them being the mean squared error, and the other being the multinomial loss.

2.9 Data pre-processing

Input features were split into numerical and categorical variables. Numerical features were normalized using the uniform quantile transform, while categorical variables were one-hot encoded.

We applied a feature-wise uniform quantile transform to the numerical features in the input data. We use that if a random variable X has a continuous distribution function F, then the random variable $Y := F(X) \sim U(0, 1)$ (Casella & Berger, 2002). This serves as a way to squeeze the data in a more compact set. In our experiments, we used N=50 quantiles for each feature.

We apply one-hot encoding to the categorical variables. If a categorical variable is of a certain category j out of K values it can possibly take, we encode the category as a zero vector everywhere, but taking the value 1 at the jth component, where $1 \leq j \leq K$.

2.10 Model Inputs and Outputs

The transformed numerical and categorical features for individual data instances are fused as one vector. These vectors are then used as input to train the TabDDPM model, which learns the parameters in the probability distribution as described above for the reverse process. Once the parameters are learned, we sample from this distribution to obtain our generated synthetic data. This output vector must then be transformed back to its original form, i.e. we reverse the uniform quantile transform for the numerical data, and we reverse the one-hot encoding of the categorical values. This is the output of our model (Kotelnikov et al., 2022.

2.11 Model Assessment

We measured the performance of the TabDDPM model using five different neural network architectures for the denoising function. As described in Section 2.7, the first two are ResNet architectures, which vary in size and dropout rate, and the last three are MLPs with varying architectures. We compared the training dynamics of each model over 10 000 training steps, and assessed the performance of the models using 500 samples. To measure how similar our synthetic and original data are, we plot them side-by-side using kernel density estimation (KDE) (Węglarczyk, 2018) for continuous data, and bar graphs for categorical data. In addition, we assess the ability of the generative model to preserve data ranks by computing Spearman cross-correlation matrices (Schober, 2018). We take the absolute difference of the original and synthetic Spearman cross-correlation matrices, and we take the mean over all the entries of the matrix. We call this value the mean distance (**MD),** and we refer to it as such throughout this report. We do the same for the categorical variables, giving us a 6 x 6 matrix. Ideally, we want the feature ranks to be preserved in the synthetic data. To better visualize this, we also plot the entry-wise absolute difference of the two correlation matrices.

3. Results

When observing the MD for the different neural networks in Figure 1, we see that the lowest MD for continuous data results are in model 2, i.e. a 128x128 ResNet neural network with a dropout rate of 0.2 and 0.1. It is thus this model's performance which we report.

Table 1: Summary of the neural network architectures and hyperparameter configurations we considered to train the TabDDPM model.

Model	MD for Continuous Data	MD for Discrete Data	Number of Parameters	Type	Dropout Rate	Architecture
1	0.04508	0.09662	178052	Resnet	0.5 and 0.5	128 x 128
2	0.03736	0.10163	178052	Resnet	0.2 and 0.1	128 x 128
3	0.11573	0.11532	132740	MLP	0.2	256x128x256
4	0.11058	0.10467	77700	MLP	0.2	128x128x128
5	0.11461	0.10402	29124	MLP	0.2	32x32

In Figure 1, we see the training curve reach a loss of 0.3966 by the 10 000th training step. We can see that Gaussian loss accounts for 0.2403 of the total loss, whereas the multinomial loss accounts for 0.1563. We can quickly deduce that the categorical features achieve a lower loss than the loss of the numerical features. However, we also see that the Gaussian loss drops significantly more than the multinomial loss.

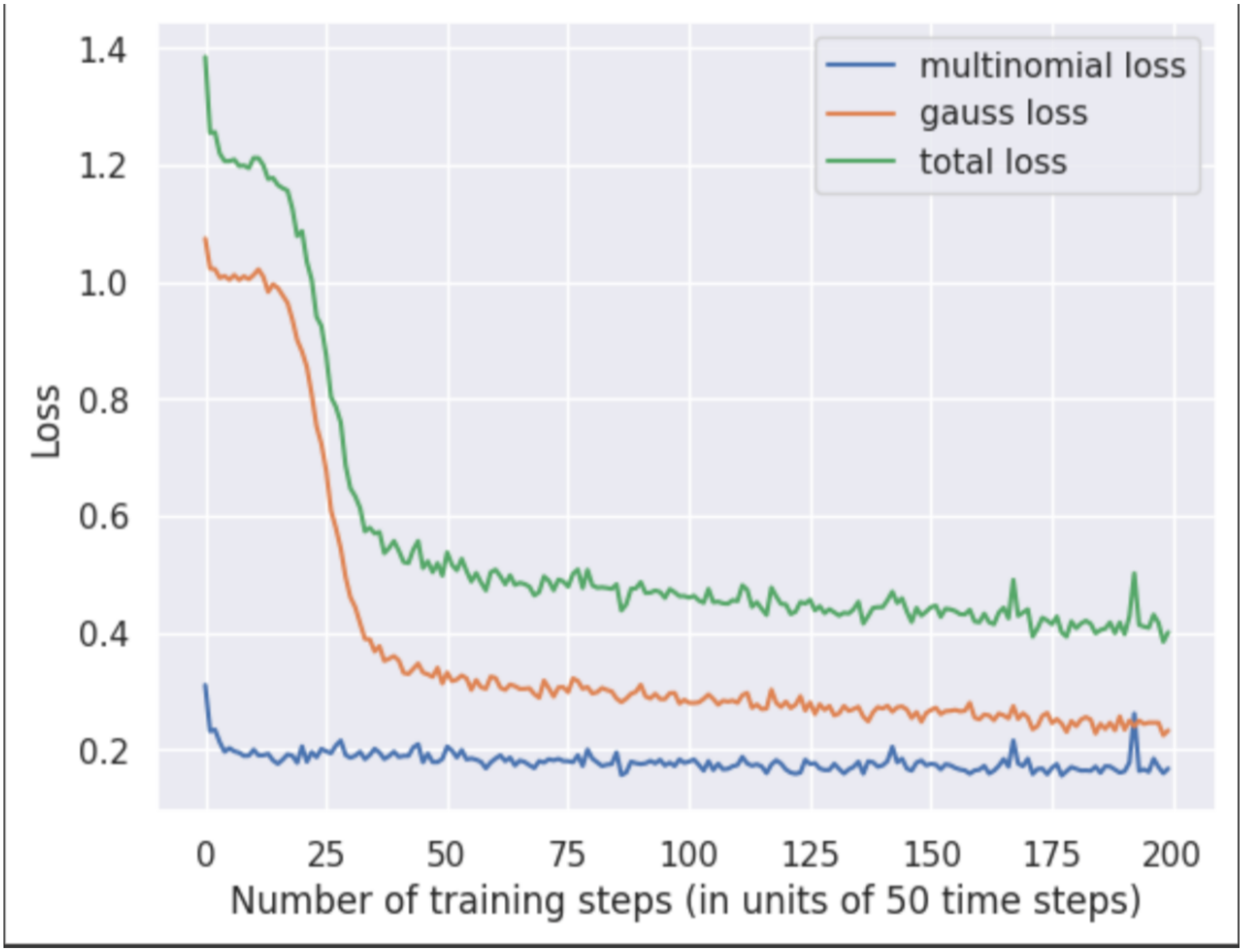

Figure 1: Training curves for a TabDDPM model using a ResNet architecture as the backbone trained for 10 000 time steps on the MIMIC-IV data

Visually, one can see in the KDE plots, in Figures 2 and 3, the similarity between the blue curve, representing the original data, and the orange curve, representing the synthetic data. Figure 2 compares the KDE curves for the 10 laboratory measurements, whereas Figure 3 shows the continuous patient-stay data. Notably, we see the blue and orange curves both peaking at the same value.

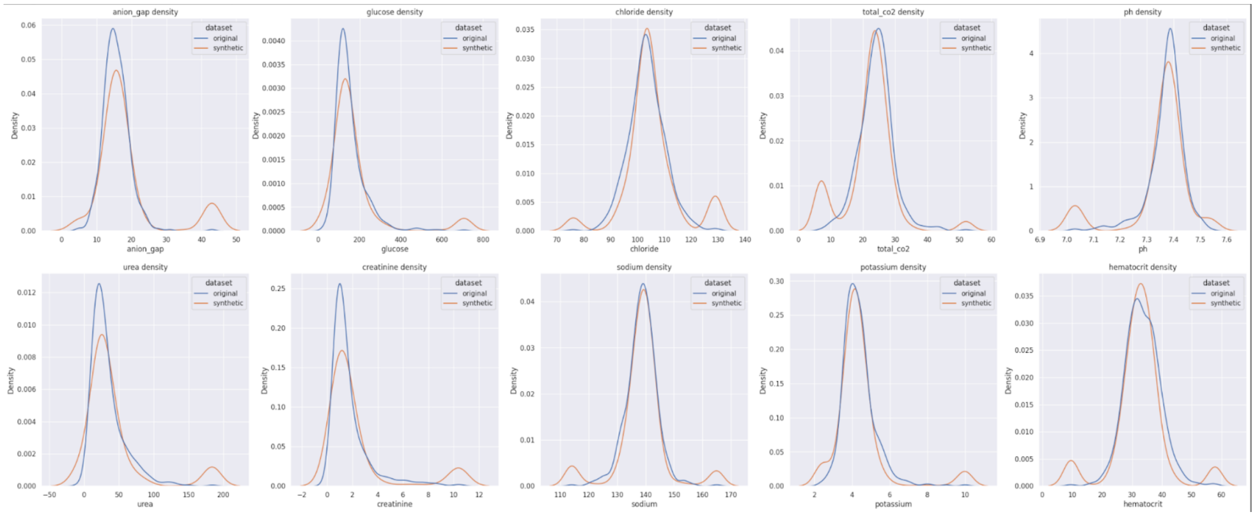

Figure 2: Plots of the KDE of each of the 10 lab signals, for the original distribution (blue) and for the synthetic distribution (orange)

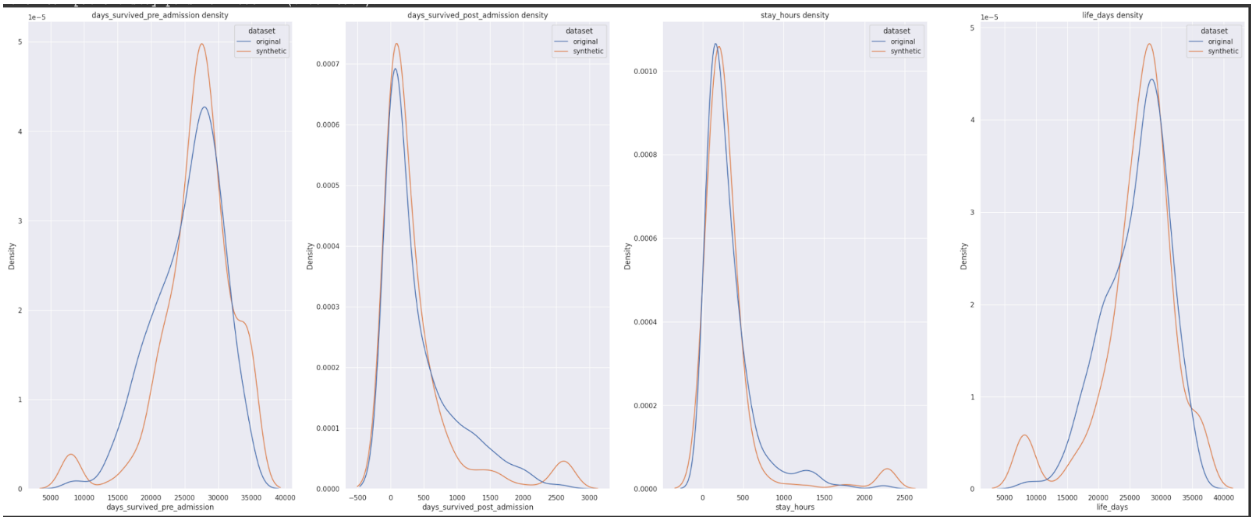

Figure 3: Plots of the KDE of each of the 4 numerical patient-stay variables for the original distribution (blue) and for the synthetic distribution (orange)

In Figure 4, we can see the bar graph representations of the original versus the synthetic data. We see that the gender plot has the least non-overlapping portions between the blue (original) and orange (synthetic) bars. The remaining 5 bar graphs contain substantial non-overlapping regions between blue and orange curves. We further notice that all of the orange bars are of similar heights, yet the blue bars vary in height much more, typically having one clear mode.

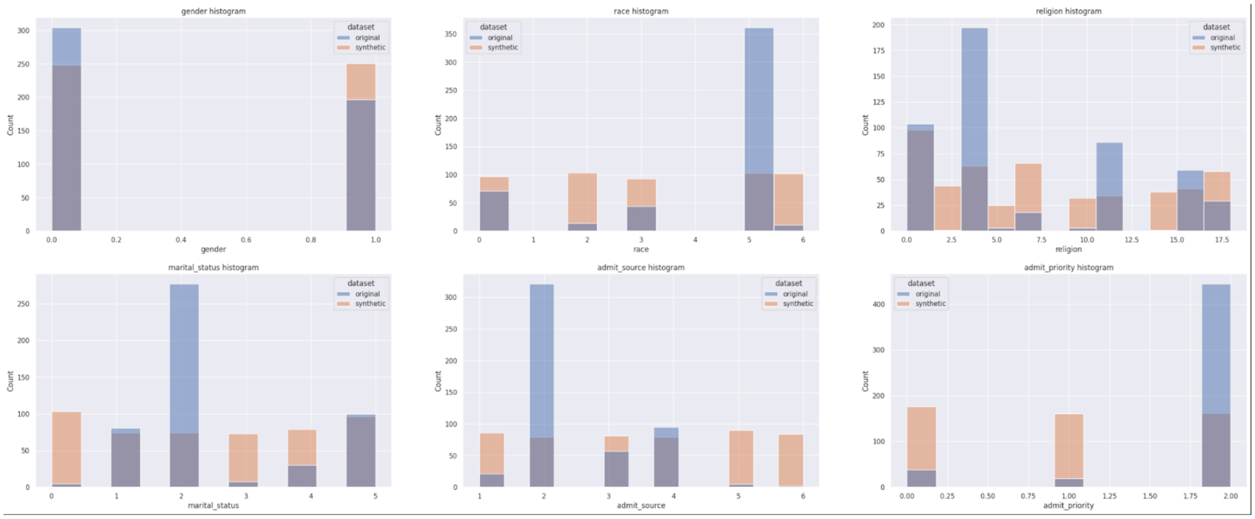

Figure 4: Plots of the bar graphs of the categorical features for the original distribution (blue) and for the synthetic distribution (orange)

In Table 1, we see that the model 2 MD has a value of 0.0376 for the continuous data, whereas the value for the discrete data is 0.10163. We can see this difference visually in Figures 5 and 6. Indeed, Figure 5 has two matrices of similar colour, indicated by a pale difference matrix. On the other hand, Figure 6 shows matrices with more different colours, and a darker difference matrix.

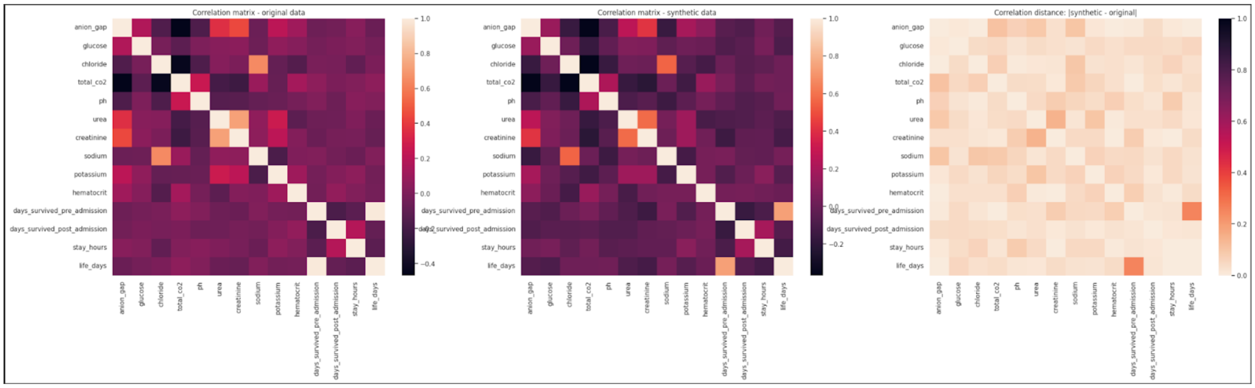

Figure 5: Spearman cross-correlation matrix for the numerical variables: the original data (left), the synthetic data (middle), and their entry-wise absolute difference (right)

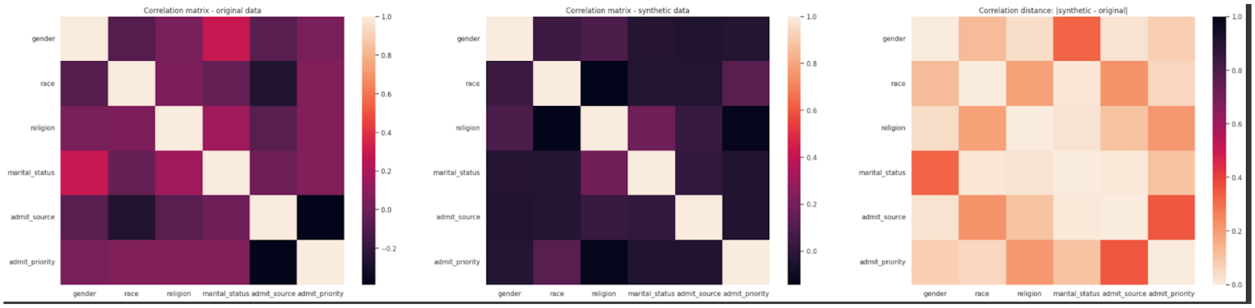

Figure 6: Spearman cross-correlation matrix for the categorical variables: the original data (left), the synthetic data (middle), and their entry-wise absolute difference (right)

A noteworthy result from Table 1 is that the models with an MLP network show MDs about three times higher than those of ResNet for the continuous data. Yet, the MDs for the discrete data are about the same regardless of the neural network.

4. Discussion

Our results suggest that the theory developed in the TabDDPM paper (Kotelnikov et al., 2022) turned out better for the generation of synthetic numerical values, than for the generation of the categorical variables that we had chosen from the MIMIC-IV data.

The results from Figure 2 suggest that the overall model's loss may be dominated by the Gaussian term and that the multinomial loss may not be trained adequately by the neural network. Even though the loss for the categorical data is lower than that of the numerical one, we note that since it did not decrease, but rather stayed constant throughout the training, there is a problem with the training of the categorical variables. This suggests that the model converged too early. Since the training curve was constant throughout most of the training, it is unlikely that this is due to insufficient training steps.

Upon inspecting the correlation matrices in Figures 5 and 6, we notice the greatest difference, of 0.3, to occur between the entries of the correlation between life days and days survived pre-admission. However, since the lab measurements do not show any outlier correlations, this could simply be due to an insufficient number of training steps.

On the other hand, in the categorical data, the Spearman cross-correlation matrices differ much more than they did for the numerical data. In particular, we observe a great disparity of 0.4 in the correlation coefficients between the admit-source and admit-priority, as well as the marital status and gender correlation. This shows that the correlation isn't as well preserved for the categorical data, as it is for the numerical data.

Overall, the results from Figures 5 and 6 suggest that our synthetic data from the numerical measurements preserve inter-feature correlations better than the categorical patient-stay data.

Indeed, the MDs for the discrete data is $\frac{0.10163}{0.03735} = 2.72$ times worse than that of our continuous data.

Moreover, when comparing the MDs in Table 1, we found that both of our ResNet neural networks outperform all three of our MLPs for the numerical data, but the performance was largely the same for the categorical variables.

One reason for which our model performs so much better with the ResNet architectures, rather than with the MLP is because the ResNet architecture is able to skip some of the connections (He et al., 2016), which perhaps translated into not being affected by the other features, or simply because the information is too delicate to be modified too much throughout the many layers. It is surprising that a simple MLP is unable to provide good results, as an MLP in some cases should be able to even overfit the data by the fundamental approximation theorem of MLPs (Pinkus, 1999). One factor which could explain this is that we only use 500 data instances during our training, and our model could have benefited from a greater data size.

To further assess how the expressiveness of the model behaves with respect to the complexity of the architecture of the model, the stability in performance of the MLP's in Table 1 supports the idea that the complexity of the architecture is not so important to the expressiveness of our model. Indeed, we are always obtaining an MD within the range of [0.11058, 0.11573] for the numerical data and within the interval of [0.10402, 0.11532] for the categorical data. This is despite our MLP ranging from a 32 x 32 architecture to a 256 x 128 x 256 architecture. Thus, we can get away with using a simpler MLP architecture, while not sacrificing performance.

Figures 3 and 4 visually suggest that the KDE plot for the original and synthetic data are very similar. In particular, the modes match up at the highest peak. This suggests that the synthetic numerical data has been generated properly with respect to the distribution of the original data.

This is further supported by the lower MDs associated with this data. However, there is still room for improvement, since we see that there are some smaller oscillations in the synthetic distribution which should not be there.

If we have a look at the bar graph comparison for the categorical data in Figure 4, we see that only the gender has similar original and synthetic distributions, since it has the fewest non-overlapping regions. In comparison, the race, religion, marital status, admit status, and admit priority have original and synthetic distributions which are much less alike, since there are a lot more non-overlapping regions. In fact, it seems that the categorical synthetic data tends to be similar throughout all categories, and does not model the data distributions with a prominent mode. It is likely that this is an artifact of our choice to use a uniform quantile transformation (Casella & Berger, 2002) as a preprocessing step for our model. This is perhaps the reason for which it modeled the gender accurately, since the gender distribution happens to be very even in our data, and there is no prominent mode. This tells us that the model needs more work for accurately performing with the categorical features which are not evenly distributed. This can be achieved by implementing some of the theory developed in (Song et al., 2020).

5. Conclusion

In conclusion, we have found that our ResNet models perform better than our MLP models, and that the MIMIC-IV numerical data shows better synthetic results than the categorical data when applying the TabDDPM method.

In a future project, we would like to find a single loss function which would govern both the numerical and categorical data. This can be done by treating the one-hot variables as numerical data. We would further like to generalize this project to extend to a progression over time. We would like to be able to model a patient's ICU stay over a period of time of 30 days. This brings us to the idea of using convolutional neural networks, similar to (Wibawa et al., 2022) in order to model this data with an extra time dimension. This could be particularly useful for the logistics of the ICU, since this would provide information on the reserves of certain substances: when to refill them, where to prioritize, etc. We would also like to explore how this project can scale to general hospital admissions, and not just ICU cases.

6. Acknowledgements

I would like to deeply thank Dr. Marc-André Leagult and Dr. Louis-Antoine Mullie for their continuous guidance and patience throughout this semester.

7. References

1. Bishop, C. M., & Nasrabadi, N. M. (2006). *Pattern recognition and machine learning* (Vol. 4, No. 4, p. 738). New York: springer.
2. Casella, G. & Berger, R.L.**2**, (Duxbury Pacific Grove, CA: 2002).
3. Erdemir, A., Mulugeta, L., Ku, J. P., Drach, A., Horner, M., Morrison, T. M., ... & Myers, J. G. (2020). Credible practice of modeling and simulation in healthcare: ten rules from a multidisciplinary perspective. *Journal of translational medicine*, *18*(1), 1-18.
4. Figueira, A., & Vaz, B. (2022). Survey on synthetic data generation, evaluation methods and GANs. *Mathematics*, *10*(15), 2733.
5. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).
6. Ho, J., Jain, A., & Abbeel, P. (2020). Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, *33*, 6840-6851.
7. Hoogeboom, E., Nielsen, D., Jaini, P., Forré, P., & Welling, M. (2021). Argmax flows and multinomial diffusion: Learning categorical distributions. *Advances in Neural Information Processing Systems*, *34*, 12454-12465.
8. Johnson, A. E., Bulgarelli, L., Shen, L., Gayles, A., Shammout, A., Horng, S., ... & Mark, R. G. (2023). MIMIC-IV, a freely accessible electronic health record dataset. *Scientific data*, *10*(1), 1.
9. Kahn, M. G., Batson, D., & Schilling, L. M. (2012). Data model considerations for clinical effectiveness researchers. *Medical care*, *50*.
10. Kotelnikov, A., Baranchuk, D., Rubachev, I., & Babenko, A. (2022). TabDDPM: Modelling Tabular Data with Diffusion Models. *arXiv preprint arXiv:2209.15421*.

11. Nichol, A. Q., & Dhariwal, P. (2021, July). Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning* (pp. 8162-8171). PMLR.
12. Pinkus, A. (1999). Approximation theory of the MLP model in neural networks. *Acta numerica*, *8*, 143-195.
13. Purushotham, S., Meng, C., Che, Z., & Liu, Y. (2018). Benchmarking deep learning models on large healthcare datasets. *Journal of biomedical informatics*, *83*, 112-134.
14. Schober, P., Boer, C., & Schwarte, L. A. (2018). Correlation coefficients: appropriate use and interpretation. *Anesthesia & analgesia*, *126*(5), 1763-1768.
15. Song, J., Meng, C., & Ermon, S. (2020). Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*.
16. Strongman, H., Williams, R., Meeraus, W., Murray‐Thomas, T., Campbell, J., Carty, L., ... & Valentine, J. (2019). Limitations for health research with restricted data collection from UK primary care. *Pharmacoepidemiology and drug safety*, *28*(6), 777-787.
17. Węglarczyk, S. (2018). Kernel density estimation and its application. In *ITM Web of Conferences* (Vol. 23, p. 00037). EDP Sciences.
18. Wibawa, A. P., Utama, A. B. P., Elmunsyah, H., Pujianto, U., Dwiyanto, F. A., & Hernandez, L. (2022). Time-series analysis with smoothed Convolutional Neural Network. *Journal of big Data*, *9*(1), 44.
19. Yang, X. (2019). Markov Chain and its applications. *Available at SSRN 3562746*.
20. Yang, L., Zhang, Z., Song, Y., Hong, S., Xu, R., Zhao, Y., ... & Yang, M. H. (2022). Diffusion models: A comprehensive survey of methods and applications. *arXiv preprint arXiv:2209.00796*.

8. Appendix

Bayesian Overview (Casella & Berger, 2002)

Recall that in Bayesian inference, we start with a prior distribution, then we observe data, and finally we deduce an improved distribution, called the posterior, which reflects the new knowledge gained from the data.

Mathematically, those quantities are related by Bayes' rule.

$P(A|B) \; = \; P(A)\, P(B|A) \,/\, P(B)$

If we let A be the inverse image of our random variable of interest X, and let B represent the inverse image of the random variable D associated to our data, then we get the familiar form

$P(X|D) \; = \; P(X)\, P(D|X) \,/\, P(D)$

We can marginalize over the random variable D to obtain

P(D) = $\int_{x} P(X \,=\, x,\; D)\; dx$

It is precisely this marginal probability which leads us to complications, as it is difficult to approximate.

Spearman Cross-Correlation (Schober, 2018)

The Spearman cross-correlation is defined as the usual Pearson correlation of the respective ranks of the inputs X, Y.

Let $\rho_{X,\,Y} = \frac{Cov(X,\,Y)}{\sigma X\; \sigma Y}$

Then, the Spearman cross-correlation $S \; = \; \rho_{Rank(X),\; Rank(\,Y)}$, where Rank() denotes the ordering of the sorted data X, Y.