Generating Privacy-Preserving Longitudinal Synthetic Data

Robin van Hoorn[1*] **Tom Bakkes**[1] **Zoi Tokoutsi**[2] **Ymke de Jong**[2]
R. Arthur Bouwman[1,3] **Mykola Pechenizkiy**[1]
[1]Eindhoven University of Technology [2]Philips [3]Catharina Hospital, Eindhoven
`r.d.v.hoorn@student.tue.nl`
`{t.h.g.f.bakkes, m.pechenizkiy}@tue.nl`
`{zoi.tokoutsi, ymke.de.jong}@philips.com`
`arthur.bouwman@catharinaziekenhuis.nl`

Abstract

Before synthetic data (SD) generators can produce electronic health records, many challenges remain. One of these challenges is to generate both privacy-preserving and longitudinal SD. This work combines research on longitudinal SD and privacy-preserving static SD and presents a GAN architecture called Time-ADS-GAN. Time-ADS-GAN outperforms current state-of-the-art models on both utility and privacy on three datasets and is able to reproduce the results of a healthcare study significantly better than TimeGAN. As a second contribution, a variation of the ϵ-identifiability metric is introduced and used in the analysis.

1 Introduction

Novel machine learning-based healthcare research is in many cases dependent on privacy-sensitive patient data. Unfortunately, due to rightfully strict privacy regulations, obtaining access to such data is often an arduous undertaking and can severely slow down innovation [3, 17]. Synthetic data (SD) is one approach in a set of privacy-enhancing technologies that promises to solve (part of) the data access problem [12].

Through the development of SD generation techniques, researchers have realized that SD does not necessarily preserve the privacy of patients, even though the data is by definition 'fake' [16]. As such, research is maturing on the generation of privacy-preserving SD generation methods as well as on metrics to analyze such methods [1, 20, 9, 10]. This research on privacy-preserving SD generation methods has, however, been mostly focused on data such as static tabular data and image data.

If SD is to be used for healthcare applications, there is a necessity to be able to also generate privacy-preserving longitudinal SD. Since the adoption of electronic health records and the increase in automated data collection, hospitals have been obtaining large amounts of longitudinal data. Practically all recorded patient data is longitudinal, such as daily measurements for in-patients, heart rate measurements taken during surgery or the by-monthly check-up data during a recovery.

To our knowledge, the research streams of longitudinal SD generation and privacy-preserving (static) SD generation have never been merged. This research merges these streams of research and introduces Time-ADS-GAN, a GAN framework for generating privacy-preserving longitudinal SD.

2 Relevant Literature

Since the introduction of the GAN framework by Goodfellow et al. [8], researchers have sought to adapt it to better suit their needs regarding both privacy and longitudinal data.

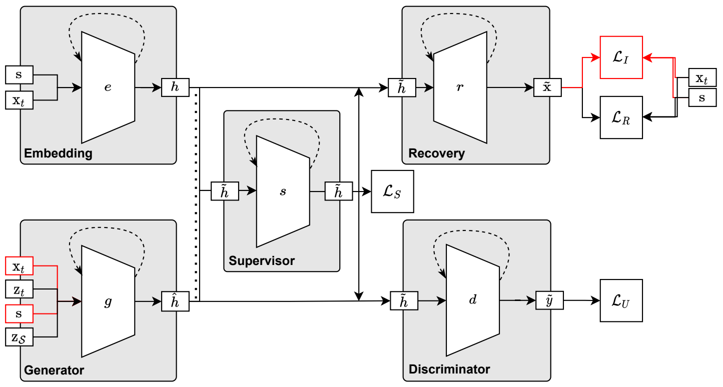


Figure 1: The Time-ADS-GAN architecture. The additions to the TimeGAN model are highlighted in red. Notation follows TimeGAN.

Longitudinal GANs When generating longitudinal data, it is desirable to learn the conditional distribution of the data and to predict the future time series given the past time series. C-RNN-GAN simply replaced the multi-layer perceptrons in the traditional framework with Long-Short Term Memory (LSTM) cells [13]. In a similar style, RCGAN used a unidirectional LSTM discriminator, dropping dependence on the previous output and conditioning on additional input [7].

Both models were outperformed by TimeGAN [19], which generates and discriminates samples in a supervised embedding space. The supervision is done through an autoencoder, which is first pre-trained and subsequently jointly trained with the rest of the model. Intuitively, the autoencoder provides the latent space, the adversarial components work with this latent space to improve the generator, and the latent dynamics are synchronised through the supervised loss.

SD Privacy Unfortunately, there are many examples of unsatisfactory attempts to anonymise datasets [2, 4, 14]. Although some argue that SD protects privacy 'by design' with regards to traditional attacks such as linkage or disclosure [6], Stadler et al. show that these studies "severely underestimate the privacy risks of synthetic data publishing" [16].

Hence, researchers have published works in which SD generators are optimized for both quality and privacy. The first widely tried method employs Differential Privacy (DP) [5]. Unfortunately, all implementations of this utilitarian notion of privacy heavily degrade the quality of the generated SD [9, 10, 16, 20].

ADS-GAN takes a different approach and introduces ϵ-identifiability [16]. Intuitively, ϵ-identifiability assures that a ratio of more than $1-\epsilon$ of the data points is not identifiable because the generated point is further away from the closest real point than the nearest real neighbour of that real point.

3 Approach

To combine the two research streams, we introduce Time-ADS-GAN, visualized in Figure 1. Compared to TimeGAN, Time-ADS-GAN has (1) a conditioned generator that takes in an additional real data sample, and (2) an additional identifiability loss ($\mathcal{L}_I$) after the recovery module. $\mathcal{L}_I$ (see Yoon et al. [20]) can be adapted for longitudinal data as follows:

$$\begin{aligned}\mathcal{L}_I = &\mathbb{E}_{\mathbf{s},\hat{\mathbf{s}}|\mathbf{s}}[-||\mathbf{w}_{\mathcal{S}} \cdot (\mathbf{s}-\hat{\mathbf{s}})||]+ \\ &\mathbb{E}_{\mathbf{x}_{1:T},\hat{\mathbf{x}}_{1:T}|\mathbf{x}_{1:T}}[-\delta(\mathbf{x}_{1:T},\hat{\mathbf{x}}_{1:T},\mathbf{w}_{\mathcal{X}})]\end{aligned} \tag{1}$$

where δ is a distance function that can be adapted based on the context of the data. This research employs the weighted Euclidean distance, which is computationally efficient for training, similar to previous work [20]. A detailed description of the architecture and model training and optimization is provided in Appendix A and Appendix B, respectively, to ensure the reproducibility of the results.

Baseline Models Time-ADS-GAN is compared to three baseline models, being RCGAN [7], TimeGAN [19] and R-ADS-GAN. R-ADS-GAN is a recurrent version of ADS-GAN, where the generator and discriminator are taken from the TimeGAN architecture.

As Time-ADS-GAN is heavily based on TimeGAN, the TimeGAN architecture is included as a baseline model. RCGAN is tested because it was shown to perform rather well in the TimeGAN publication [19]. Lastly, R-ADS-GAN is created as a variation on ADS-GAN as ADS-GAN is not suitable for longitudinal data.

Datasets The models are compared on three datasets. First, a synthetically generated sine waves dataset contributes to the academic relevance and is taken directly from Yoon et al. [19]. Second, a preprocessed subsample (1500 samples) of the publically available Philips eICU healthcare dataset [15]. The subsample contains patients suffering from pulmonary sepsis and features six longitudinal signals. Lastly, an in-patient deterioration dataset featuring 50.000+ samples, based on research from Kipnis et al. [11], provides practical relevance as it was obtained from a hospital in the Netherlands and is more complex. This dataset features 19 (categorical and continuous) temporal variables and 8 static variables. Details on the datasets and preprocessing can be found in Appendix C.

Evaluation To assess the quality of the models, the models are utilized for the generation of synthetic datasets which are then compared to the original datasets through metrics. For an initial qualitative analysis, t-SNE visualizations are constructed. This provides a first indication of both the fidelity and diversity of the SD. To quantitatively assess the fidelity of the synthetic datasets, both the discriminative score and the α-precision metric were used [19, 1]. To quantitatively assess the diversity of the synthetic datasets, the predictive score and β-recall metric were utilized [19, 1]. Lastly, to assess the privacy of the synthetic datasets, the ϵ-identifiability metric and authenticity metric were used [20, 1]. Although the α-precision, β-recall and authenticity metrics from Alaa et al. [1] were originally introduced for static tabular data, their publication also shortly specifies how the metrics can be utilized for assessments of longitudinal SD. These same specifications were used.

Apart from assessing the quality of the SD through metrics alone, the SD generated based on the in-patient deterioration dataset was also used to recreate the research results from a healthcare study using that dataset [11]. The model in the study was used to assess whether similar results could be achieved when using the SD of TimeGan and Time-ADS-GAN.

4 Results

The results are summarized visually through Figures 2, 3 and 4, visualizing the t-SNE analysis, the quality-privacy tradeoff and the reproduction study respectively.

Qualitative Results The t-SNE visualizations in Figure 2 can provide an initial idea of model quality. The overlap of synthetic and real data in such a visualization provides an indication of how well the SD matches the real data.

As can be seen in Figure 2, all models can generate the sine wave data well. Generating high-quality Philips eICU data is only problematic for RCGAN. However, Time-ADS-GAN is the only model that is reasonably able to generate the in-patient deterioration data.

Quantitative Results To visualize the quality privacy trade-off, the sum of α-precision and β-recall, measuring SD utility, is plotted against two privacy metrics in Figure 3. RCGAN is excluded because its results were significantly worse.

For the sine wave data and the Philips eICU data, the quality privacy trade-off can be interpreted in a similar manner. Generally, Time-ADS-GAN outperforms both TimeGAN and R-ADS-GAN. In some less privacy-preserving settings, R-ADS-GAN is able to outperform TimeGAN. However, the privacy-quality trade-off of TimeGAN is preferable over that of R-ADS-GAN.

For the sine wave data, it is clear that Time-ADS-GAN performs closer to the original data score than the other models, and would hence be preferable. Surprisingly, the original data score is outperformed for the Philips eICU data. This is likely because the size of the dataset is rather small. As the score is calculated by taking two subsets (in this case half of the original dataset), the sampled subsets likely differ in their distribution more than two larger subsets from, for example, the sine wave data.

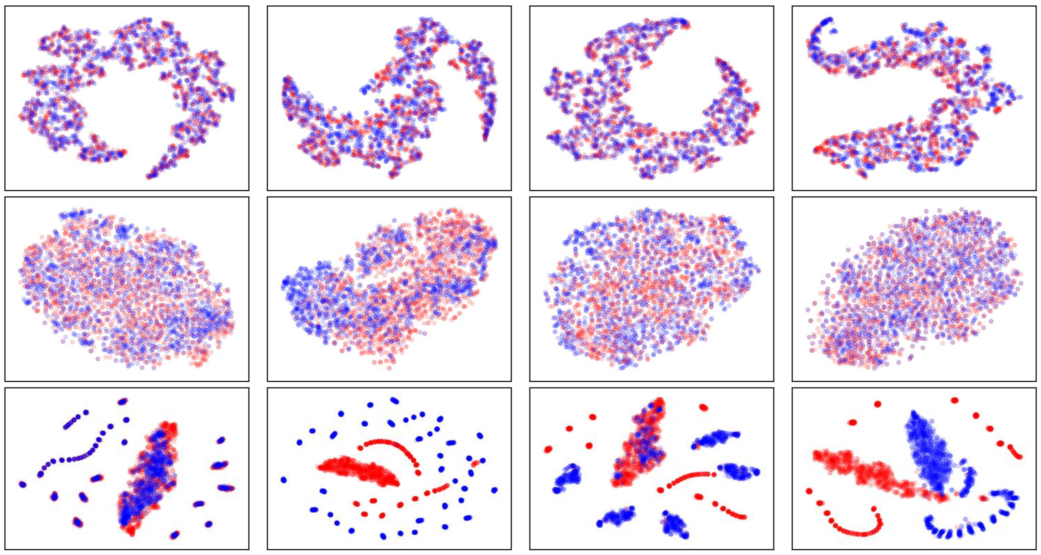

(a) Time-ADS-GAN (b) RCGAN (c) TimeGAN (d) R-ADS-GAN

Figure 2: t-SNE visualization on the sine wave dataset (1st row), the Philips eICU dataset (2nd row) and the in-patient deterioration dataset (3rd row). ● denote real data, and ● denote synthetic data. The columns provide visualizations for each different model.

For the in-patient deterioration dataset, the tradeoff is significantly different from the other two datasets, because most models are not able to converge sufficiently due to dataset complexity. This further confirms what was observed in the qualitative results. As a result of the poor fit, privacy is deemed high. Consequently, both privacy metrics are not very informative. Regarding utility, TimeGAN and Time-ADS-GAN are able to converge to a similar score. Consequently, these models are compared in the reproduction study using the in-patient deterioration study.

Reproduction of Research In the original in-patient deterioration study, Kipnis et al. [11] create a model to predict the deterioration of in-patients in the next 12 hours. The quality of the prediction model is assessed through three distinct metrics, namely the ROC curve, PR curve and the relationship and difference of the predicted value between deteriorating and non-deteriorating patients.

To further assess the quality of the SD of TimeGAN and Time-ADS-GAN, the prediction model was fitted on the synthetic datasets generated using the models that achieved the highest utility according to the metrics. The results of the reproduction study are visualized in Figure 4.

The results show abundantly clear that the reproduction value of the SD generated using Time-ADS-GAN is much better than of the SD generated using TimeGAN. For the TimeGAN-based results, both the ROC and PR curves show minimal performance. In addition, the relationship between event (a deteriorating patient) and non-event (a non-deteriorating patient) data is non-existent. This is in stark contrast with the results from the Time-ADS-GAN generated SD. Those results show both a decent ROC and PR curve, as well as a clearly distinct mean prediction value between the event and non-event data. Note that the results from the Time-ADS-GAN SD are clearly distinct from the original results and that the prediction model identifies a much weaker relationship. However, the prediction model can identify a clear relationship for the Time-ADS-GAN SD where it could not do so for the TimeGAN SD.

4.1 Adapting ϵ-Identifiability

The analysis thus far gives a nuanced understanding of the various models. One of the main limitations of the overall analysis is the inconsistency and sometimes contradictory nature of the metrics. There has not been sufficient effort put into research towards proper metrics for the utility and privacy of longitudinal SD. The metrics used in this research are currently state-of-the-art for these topics but

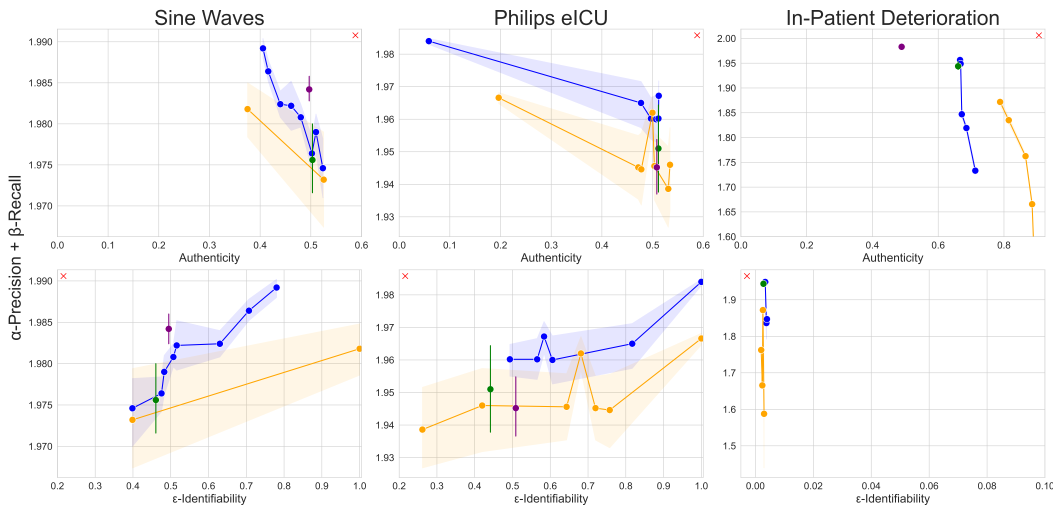
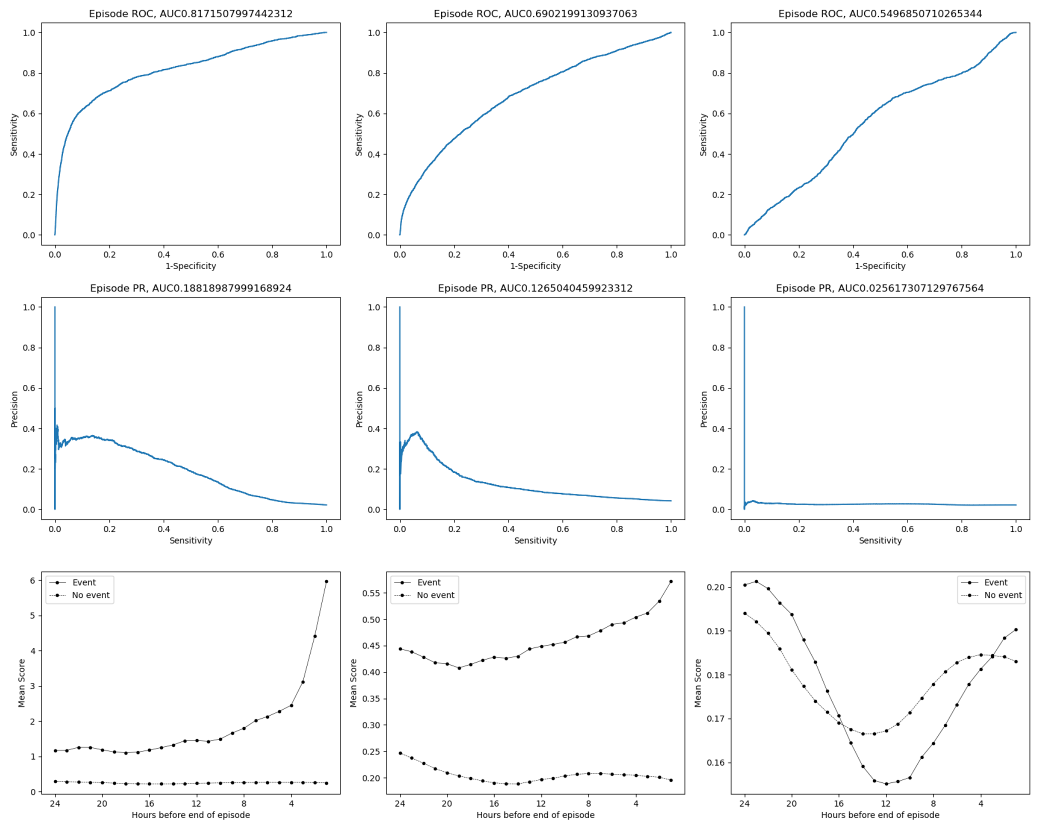


Figure 3: Tradeoff between utility (y-axis) and privacy (x-axis) of Time-ADS-GAN ●, R-ADS-GAN ●, TimeGAN ●, and original data score ●. The original data score is calculated using two subsets of the original data. ✕ indicate the target direction for optimization.



(a) Original (b) Time-ADS-GAN (c) TimeGAN

Figure 4: A visualization of the results of the reproduction study for the original data, the Time-ADS-GAN data, and the TimeGAN data. The rows present the ROC curve (1st row), the Precision-Recall curve (2nd row), and the mean deterioration prediction in the last 24 hours before the end of the episode for both event (deteriorating) samples and non-event (recovering) samples (3rd row).

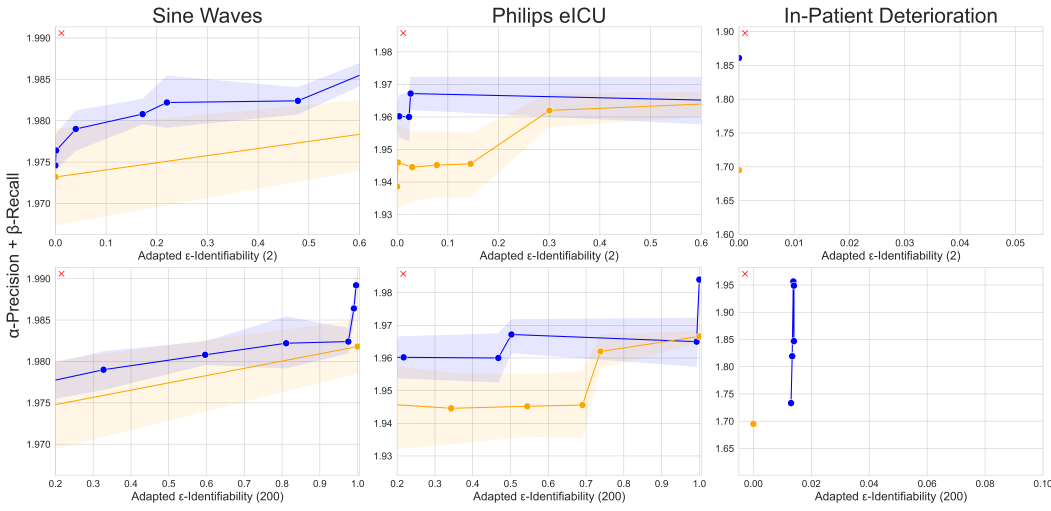


Figure 5: Tradeoff between utility (y-axis) and privacy (x-axis) of Time-ADS-GAN ● & R-ADS-GAN ● using the adapted ϵ-identifiability metric. × indicate the target direction for optimization.

suffer from several flaws, including, but not limited to, (1) being difficult to interpret, and (2) being difficult to calculate by being dependent on several trained neural networks.

The ϵ-Identifiability metric, as originally proposed by Yoon et al. [20], also suffers from difficult interpretability. First, when using the ϵ-identifiability metric, two identically and independently distributed datasets will evaluate to a score of 0.5. As the original goal, framed by Yoon et al. [20], is to minimize this metric, it is not intuitive what a privacy evaluation lower than 0.5 is supposed to mean. In addition, the ϵ-Identifiability metric does not take into account differences in density within the datasets. As such, a synthetic sample based on an identical twin will be considered more privacy-preserving than a similarly distant sample based on a singleton child.

To counteract these drawbacks, two changes are proposed to adapt the ϵ-identifiability metric. First, instead of calculating the distance from a synthetic point to its nearest neighbor, the adapted ϵ-identifiability metric calculates only the distance from a synthetic point to the real point it was based on. To perform this calculation, every SD sample must be conditioned on a real sample, as is the case in ADS-GAN and Time-ADS-GAN. Second, to alleviate the inconsistency concerning the data distribution, the metric averages over the privacy scores obtained by calculating the distance deemed different enough by calculating the distance up to J nearest neighbors. A formalization of the ϵ-Identifiability metric is provided in Appendix D.

The changes made to the metric can be identified through the quality-privacy trade-off results using the novel metric, visualized in Figure 5. First, the metric results extend all the way from 0.0 until 1.0, where 0.0 means perfect privacy given the metric settings (distance metric and J). Second, the influence of the parameter J is clearly visible through the distribution of points, particularly for the Philips eICU data.

5 Conclusions

This study suggests an approach towards privacy for longitudinal SD, outperforming previous models on quality (similarly to the work of ADS-GAN [20]). By combining TimeGAN and ADS-GAN, Time-ADS-GAN is able to outperform the state-of-the-art models on both privacy and utility. According to the metrics, Time-ADS-GAN outperforms all other models on all datasets and on most of the metrics. Similarly, Time-ADS-GAN shows significant superiority over TimeGAN in reproducing the results of a healthcare study.

Nevertheless, significant hurdles still have to be tackled before successful adoption within the healthcare sector can be expected. First, better metrics for longitudinal data need to be developed. Second, better baseline comparisons need to be established. Lastly, approaches and standards for SD privacy-preservation need to be further developed and tested on additional clinical datasets.

6 Acknowledgements

This research was conducted as part of a master's thesis through a collaborative effort between the Eindhoven University of Technology, Philips Eindhoven, the Catharina Hospital, and their collective consortium, the Eindhoven MedTech Innovation Centre (e/MTIC). We extend our gratitude to our colleagues from these institutions for their valuable insights, help and support. Our appreciation also goes out to the reviewers, whose feedback enhanced the quality and clarity of our work.

References

[1] A. Alaa, B. Van Breugel, E. S. Saveliev, and M. van der Schaar, "How faithful is your synthetic data? sample-level metrics for evaluating and auditing generative models," in *International Conference on Machine Learning*. PMLR, 2022, pp. 290–306.

[2] C. Culnane, B. I. Rubinstein, and V. Teague, "Health data in an open world," *arXiv preprint arXiv:1712.05627*, 2017.

[3] J. W. de Kok, M. Á. A. de la Hoz, Y. de Jong, V. Brokke, P. W. Elbers, P. Thoral, A. Castillejo, T. Trenor, J. M. Castellano, A. E. Bronchalo *et al.*, "A guide to sharing open healthcare data under the general data protection regulation," *Scientific Data*, vol. 10, no. 1, p. 404, 2023.

[4] Y.-A. De Montjoye, L. Radaelli, V. K. Singh, and A. S. Pentland, "Unique in the shopping mall: On the reidentifiability of credit card metadata," *Science*, vol. 347, no. 6221, pp. 536–539, 2015.

[5] C. Dwork, A. Roth *et al.*, "The algorithmic foundations of differential privacy," *Foundations and Trends® in Theoretical Computer Science*, vol. 9, no. 3–4, pp. 211–407, 2014.

[6] M. Elliot, K. O'hara, C. Raab, C. M. O'Keefe, E. Mackey, C. Dibben, H. Gowans, K. Purdam, and K. McCullagh, "Functional anonymisation: Personal data and the data environment," *Computer Law & Security Review*, vol. 34, no. 2, pp. 204–221, 2018.

[7] C. Esteban, S. L. Hyland, and G. Rätsch, "Real-valued (medical) time series generation with recurrent conditional gans," *arXiv preprint arXiv:1706.02633*, 2017.

[8] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," 2014. [Online]. Available: https://arxiv.org/abs/1406.2661

[9] J. Jordon, J. Yoon, and M. Van Der Schaar, "Pate-gan: Generating synthetic data with differential privacy guarantees," in *International conference on learning representations*, 2018.

[10] J. Jordon, J. Yoon, and M. van der Schaar, "Differentially private bagging: Improved utility and cheaper privacy than subsample-and-aggregate," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[11] P. Kipnis, B. J. Turk, D. A. Wulf, J. C. LaGuardia, V. Liu, M. M. Churpek, S. Romero-Brufau, and G. J. Escobar, "Development and validation of an electronic medical record-based alert score for detection of inpatient deterioration outside the icu," *Journal of biomedical informatics*, vol. 64, pp. 10–19, 2016.

[12] D. McDuff, T. Curran, and A. Kadambi, "Synthetic data in healthcare," *arXiv preprint arXiv:2304.03243*, 2023.

[13] O. Mogren, "C-rnn-gan: Continuous recurrent neural networks with adversarial training," *arXiv preprint arXiv:1611.09904*, 2016.

[14] A. Narayanan and V. Shmatikov, "De-anonymizing social networks," in *2009 30th IEEE symposium on security and privacy*. IEEE, 2009, pp. 173–187.

[15] T. J. Pollard, A. E. Johnson, J. D. Raffa, L. A. Celi, R. G. Mark, and O. Badawi, "The eicu collaborative research database, a freely available multi-center database for critical care research," *Scientific data*, vol. 5, no. 1, pp. 1–13, 2018.

[16] T. Stadler, B. Oprisanu, and C. Troncoso, “Synthetic data–anonymisation groundhog day,” in *31st USENIX Security Symposium (USENIX Security 22)*, 2022, pp. 1451–1468.

[17] P. J. Thoral, J. M. Peppink, R. H. Driessen, E. J. Sijbrands, E. J. Kompanje, L. Kaplan, H. Bailey, J. Kesecioglu, M. Cecconi, M. Churpek *et al.*, “Sharing icu patient data responsibly under the society of critical care medicine/european society of intensive care medicine joint data science collaboration: the amsterdam university medical centers database (amsterdamumcdb) example,” *Critical care medicine*, vol. 49, no. 6, p. e563, 2021.

[18] J. Yoon, “Timegan,” https://github.com/jsyoon0823/TimeGAN, 2019.

[19] J. Yoon, D. Jarrett, and M. Van der Schaar, “Time-series generative adversarial networks,” *Advances in neural information processing systems*, vol. 32, 2019.

[20] J. Yoon, L. N. Drumright, and M. Van Der Schaar, “Anonymization through data synthesis using generative adversarial networks (ads-gan),” *IEEE journal of biomedical and health informatics*, vol. 24, no. 8, pp. 2378–2388, 2020.

Appendices

A Detailed Architecture

The following section elaborates upon the Time-ADS-GAN architecture in detail, including the aspects that are analogous to Yoon et al. [19], namely the embedding, recovery, supervisor and discriminator modules. Time-ADS-GAN only changed the specifics of the generator and added an additional loss function, as described in Section 3.

Embedding and Recovery The embedding and recovery modules form an autoencoder which provides Time-ADS-GAN with an easy-to-learn latent space and allows the model to generate the data in a potentially lower dimensional space. Let $\mathcal{H}_{\mathcal{S}}, \mathcal{H}_{\mathcal{X}}$ represent the latent spaces corresponding to the features spaces $\mathcal{S}$ and $\mathcal{X}$ for static and temporal features respectively. The embedding function e takes in the feature space and maps it to the embedding space, formally denoted as $\mathcal{S} \times \prod_t \mathcal{X} \to \mathcal{H}_{\mathcal{S}} \times \prod_t \mathcal{H}_{\mathcal{X}}$, therewith producing latent codes $\mathbf{h}_{\mathcal{S}}, \mathbf{h}_{1:T} = e(\mathbf{s}, \mathbf{x}_{1:T})$. The embedded static features are used to embed the temporal features. The embedding function e in [19] is implemented through a recurrent neural net,

$$\mathbf{h}_{\mathcal{S}} = e_{\mathcal{S}}(\mathbf{s}), \qquad \mathbf{h}_t = e_{\mathcal{X}}(\mathbf{h}_{\mathcal{S}}, \mathbf{x}_t, \mathbf{c}_{e_{\mathcal{X}},t-1}), \tag{2}$$

where $e_{\mathcal{S}} : \mathcal{S} \to \mathcal{H}_{\mathcal{S}}$ and $e_{\mathcal{X}} : \mathcal{H}_{\mathcal{S}} \times \mathcal{X} \times \mathcal{C}_{e_{\mathcal{X}}} \to \mathcal{H}_{\mathcal{X}}$ represent the embedding functions for the static and temporal features respectively, and $\mathbf{c}_{e_{\mathcal{X}},t-1}$ represents the carry state (often called the hidden state) of the autoregressive cell. Note that this formalization differs slightly from the formalization presented by Yoon et al. [19]. They assume that the output of $e_{\mathcal{X}}$ is equal to the carry state, but this does not need to be, i.e. if the autoregressive cell used is an LSTM. $\mathcal{C}_{e_{\mathcal{X}}}$ represents the carry state feature space, subscripted to denote the function it is referring to.

The recovery module functions very similarly but in reverse. Formally, the recovery function $r : \mathcal{H}_{\mathcal{S}} \times \prod_t \mathcal{H}_{\mathcal{X}} \to \mathcal{S} \times \prod_t \mathcal{X}$ takes the embedded static and temporal features and maps them back to their feature space $\tilde{\mathbf{s}}, \tilde{\mathbf{x}}_{1:T} = r(\tilde{\mathbf{h}}_{\mathcal{S}}, \tilde{\mathbf{h}}_{1:T})$. The $\sim$ symbol notation above a variable denotes that the variable can be either constructed using a real sample or generated using a random prior. In this case, the recovery module can have as input an embedding of a real sample or a generated embedding of the generator. According to the paper, r is implemented through a feedforward network at each step,

$$\tilde{\mathbf{s}} = r_{\mathcal{S}}(\tilde{\mathbf{h}}_{\mathcal{S}}), \qquad \tilde{\mathbf{x}}_t = r_{\mathcal{X}}(\tilde{\mathbf{h}}_t), \tag{3}$$

where $r_{\mathcal{S}} : \mathcal{H}_{\mathcal{S}} \to \mathcal{S}$ and $r_{\mathcal{X}} : \mathcal{H}_{\mathcal{X}} \to \mathcal{X}$ are recovery networks for the static and temporal features respectively. However, Yoon et al. [19] continue to state later that both the embedding and recovery functions need to be autoregressive. In their implementation, they also use a recurrent network for the recovery function [18]. As such, the recovery function $r_{\mathcal{X}}$ can better be described as

$$\tilde{\mathbf{x}}_t = r_{\mathcal{X}}(\tilde{\mathbf{h}}_t, \tilde{\mathbf{c}}_{r_{\mathcal{X}},t-1}), \tag{4}$$

where $r_{\mathcal{X}} : \mathcal{H}_{\mathcal{X}} \times \mathcal{C}_e \to \mathcal{X}$ is the new formalization of $r_{\mathcal{X}}$ and $\tilde{\mathbf{c}}_{r_{\mathcal{X}},t-1}$ is again representing the carry state.

Generator and Discriminator The changed generator is conditioned on the real data, as is done in ADS-GAN [20]. The conditional data is concatenated with the random vectors before they are inserted in the generator networks. Hence the generator function $g : (\mathcal{Z}_{\mathcal{S}} \times \prod_t \mathcal{Z}_{\mathcal{X}}) \times (S \times \prod_t \mathcal{X}) \to \mathcal{H}_{\mathcal{S}} \times \prod_t \mathcal{H}_{\mathcal{X}}$ takes a tuple of tuples of static and temporal random and real vectors and generates synthetic latent codes $\hat{\mathbf{h}}_{\mathcal{S}}, \hat{\mathbf{h}}_{1:T} = g(\mathbf{z}_{\mathcal{S}}, \mathbf{z}_{1:T}, \mathbf{s}, \mathbf{x}_{1:T})$. Similar to TimeGAN, g is implemented through a recurrent network,

$$\hat{\mathbf{h}}_{\mathcal{S}} = g_{\mathcal{S}}(\mathbf{z}_{\mathcal{S}}, \mathbf{s}), \qquad \hat{\mathbf{h}}_t = g_{\mathcal{X}}(\hat{\mathbf{h}}_{\mathcal{S}}, \mathbf{z}_t, \mathbf{x}_t, \hat{\mathbf{c}}_{g_{\mathcal{X}},t-1}), \tag{5}$$

where $g_{\mathcal{S}} : \mathcal{Z}_{\mathcal{S}} \times \mathcal{S} \to \mathcal{H}_{\mathcal{S}}$ is a static feature generator network and $g_{\mathcal{X}} : \mathcal{H}_{\mathcal{S}} \times \mathcal{Z}_{\mathcal{X}} \times \mathcal{X} \times \mathcal{C}_{g_{\mathcal{X}}} \to \mathcal{H}_{\mathcal{X}}$ is a recurrent generator for temporal features.

Like the recovery module, the discriminator module operates from the embedding space. The discriminator function $d : \mathcal{H}_{\mathcal{S}} \times \prod_t \mathcal{H}_{\mathcal{X}} \to [0, 1] \times \prod_t [0, 1]$ transforms the embedding space to a classification $\tilde{y}_\mathcal{S}, \tilde{y}_{1:T} = d(\tilde{\mathbf{h}}_\mathcal{S}, \tilde{\mathbf{h}}_{1:T})$. Yoon et al. [19] state that d is implemented through a bidirectional recurrent network with a feedforward output layer,

$$\tilde{y}_\mathcal{S} = d_\mathcal{S}(\tilde{\mathbf{h}}_\mathcal{S}), \qquad \tilde{y}_t = d_\mathcal{X}(\vec{\mathbf{u}}_t, \overleftarrow{\mathbf{u}}_t), \tag{6}$$

where $\vec{\mathbf{u}}_t = \vec{b}_\mathcal{X}(\tilde{\mathbf{h}}_\mathcal{S}, \tilde{\mathbf{h}}_t, \tilde{\mathbf{c}}_{\vec{b}_\mathcal{X},t-1})$ and $\overleftarrow{\mathbf{u}}_t = \overleftarrow{b}_\mathcal{X}(\tilde{\mathbf{h}}_\mathcal{S}, \tilde{\mathbf{h}}_t, \tilde{\mathbf{c}}_{\overleftarrow{b}_\mathcal{X},t-1})$ denote bidirectional outputs of the recurrent neural nets, $\vec{b}_\mathcal{X}, \overleftarrow{b}_\mathcal{X}$ are recurrent functions, and $d_\mathcal{S}, d_\mathcal{X}$ are classification functions. However, the author again specified on GitHub that they saw no performance gain or loss by switching to a unidirectional discriminator [18].

Supervisor The supervisor takes in the embedding space and uses the input to predict the next time step in the embedding space. The supervisor function $s : \mathcal{H}_\mathcal{S} \times \prod_t \mathcal{H}_\mathcal{X} \to \prod_t \mathcal{H}_\mathcal{X}$ transforms the embedding space of both the static and temporal features into an embedding space of only the temporal features (the next time step). Logically, the static features do not need to be transformed, as they are the same at every time step. Like most other modules, s is implemented through a recurrent network with a feedforward output layer,

$$\tilde{\mathbf{h}}_{t,sup} = s(\tilde{\mathbf{h}}_\mathcal{S}, \tilde{\mathbf{h}}_{t-1}, \tilde{\mathbf{c}}_{s,t-1}), \tag{7}$$

where $\tilde{\mathbf{h}}_{t,sup}$ represents the supervised (predicted) embedded sample or random prior at time step t, and $\tilde{\mathbf{c}}_{s,t-1}$ represents the carry state of the recurrent cell from time-step $t-1$. $\tilde{\mathbf{h}}_{t,sup}$ should not be confused with $\tilde{\mathbf{h}}_t$ which has not been processed by the supervisor.

Lastly, it should also be noted that although the authors use GRUs and LSTMs to construct their TimeGAN models, there are no architectural restrictions on any of the modules as long as they are autoregressive.

B Model Training and Optimization

B.1 Model Training

The training of the architecture is a structured process that closely follows the methodology of TimeGAN, and it unfolds in three sequential phases. In the initial phase, the training process trains the embedding and recovery models to create a latent space. The second phase involves training the generator and supervisor modules jointly to allow the supervisor module to converge to the previously mentioned latent space. The final phase is an integrative training phase where all components, including the generator, are trained simultaneously. Additionally to the normal generator training as implemented for TimeGAN, a novel component in this phase is the integration of the identifiability loss [18]. This loss is added to the other losses. To calibrate the impact of the identifiability loss on the convergence of the model, an additional parameter is introduced, drawing on the approach used by Yoon et al. [20].

B.2 Hyperparameter Optimization

The computational costs of training a Time-GAN or Time-ADS-GAN instance are heavily dependent on the size of the data, the size of the network, and the number of epochs that the model trains for. Therefore, initial hyperparameter optimization was done using Bayesian hyperparameter optimization. To evaluate parameter configurations, the predictive score is used. Although not a perfect metric, it gives an indication of the utility of the synthetic samples and a limited ability to detect mode collapse. The metric is calculated 3-5 times, depending on the size of the dataset, to approximate the performance of a parameter configuration.

The Bayesian hyperparameter optimization process assists in finding the best hyperparameters for each dataset and architecture combination. However, the best-performing parameter set, according to the Bayesian hyperparameter optimization process, is not necessarily the optimal parameter set for various reasons. First, it is computationally infeasible to train all models optimally during the Bayesian optimization process. To save time, models can only train up to 12000 epochs. In addition, model parameter configuration can only be trained once, even though multiple random initializations might result in vastly different results. Training a model under these constraints gives an indication of possible performance but not necessarily an absolute representation of the best-performing model. Second, the evaluation of a parameter set during hyperparameter optimization is based on the predictive score. Trivially, the predictive score does not assess the full capability of the model but provides an initial indication. Therefore, the Bayesian optimization process helps find a good set of hyperparameters, but this set still needs to be fine-tuned further.

Fine-tuning a parameter set is a flexible process. First, the results of the Bayesian optimization process need to be assessed. If there are trends in the data from the Bayesian optimization process, then these need to be considered. From there, fine-tuning is done by hand on the basis of domain knowledge and experience in training GANs.

After the best parameter configuration is determined, a final instance needs to be trained. Unfortunately, due to the adversarial nature of GANs, the model's performance can change significantly throughout the training process. Therefore, to train a final instance, several models are initialized with the best parameter configuration and trained for 50000 epochs. Every 2000 epochs, the model performance is assessed, and only the best of each model training procedure is kept. After all initialized models have completed training, their best-performing models are compared, and one is kept as the best-performing model overall.

C Datasets - Details and Preprocessing

C.1 Sine Waves Dataset

Sine waves can be easily generated and occupy a non-linear feature space. In addition, it is very easy to inspect this data visually. The sine waves are generated with varying frequency η and phase θ. For each dimension $i \in \{1, ..., 5\}$, $x_i(t) = sin(\eta * t + \theta)$ for $t \in \{1, ..., 2\}$, $\eta \sim \mathcal{U}[0, 0.1]$ and $\theta \sim \mathcal{U}[0, 0.1]$.

This data is also used in the original TimeGAN publication Yoon et al. [19]. Although the paper specifies a different approach to generating sine waves, issue #40 on their GitHub specifies that the details on the data generation as presented in the paper are not correct and that the details as specified here are what they used to obtain their results [18].

C.2 Philips eICU Dataset

The Philips eICU database contains the health data of patients who resided in critical care units across the United States. The database consists of information of over 200.000 patients and includes data regarding "vital sign measurements, care plan documentation, the severity of illness measures, diagnosis information, treatment information and more" [15].

The Philips eICU data provides highly complex patient dossiers, divided over 31 tables filled with diagnoses, medication prescriptions, measurements, and much more. Although it would be incredible to synthesize such specific patient dossiers, it is illogical to go from synthesizing sine waves to generating 31 tables of data. To get started in a simpler setting, it is interesting to look at a small subgroup of people admitted to the ICU for a particular diagnosis and synthesize the measurements most often associated with that diagnosis. Taking a subgroup drastically reduces the variability of the data as it reduces the typically taken measurements and responses that patients exhibit.

In this case, the focus subgroup is patients admitted to the ICU with pulmonary sepsis. This is the largest group of patients in the Philips eICU database, consisting of 8862 patients. The most commonly measured longitudinal signals present in the database for this subgroup are non-invasive systolic blood pressure (SBP), non-invasive diastolic blood pressure (DBP), temperature (T), heart rate (HR), respiratory rate (RR), and oxygen saturation (OS). In addition to these temporal signals, several static factors are included, being the sex of the patient (binary categorical variable), as well as the height and age.

Preprocessing To feed the model consistent data, the samples need to be preprocessed. First, the samples are filtered. Patients are included if they were in the ICU longer than two days, and each of the included signals was measured at least 12 times. In addition, the first signal had to be within the first four hours of the first day, and the last signal had to be in the last four hours of the second day. After filtering on these inclusion criteria, 1584 patients remained. Continuing, all signals are linearly subsampled to intervals of four hours, giving a total sample length of 13. After subsampling, outliers are classified and removed. The sex of the patient is one-hot encoded. The age and height variables are kept as-is. Lastly, to be able to generate all features with the GAN frameworks, all variables need to reside in the [0, 1] interval, and thus feature-wise min-max scaling is applied. The final dataset contains 1503 samples.

Min-max scaling is applied because the GAN framework implementations cannot generate numbers outside the [0, 1] range as they operate with a sigmoid function as the last activation of the generator modules. However, as min-max scaling is applied, single outliers (or small groups of outliers) can disproportionately affect the scaling of features. If less than 2% of samples occupied 30% of the scaled feature space, they are counted as outliers. In total, about 5% of points were classified as an outlier and removed from the dataset.

C.3 In-patient Deterioration Dataset

The third dataset included in this study comprises data from the Catharina Hospital in Eindhoven, Netherlands. It was created to replicate a study conducted in the United States, which involved the development and validation of an electronic medical record-based alert score for detecting inpatient deterioration outside the ICU [11]. The primary objective of this study is to predict patient deterioration events up to twelve hours in advance. Such events include admission to the ICU for more than six hours, mortality after ICU admission, mortality despite medical attention, or acute surgery followed by an ICU admission. The dataset encompasses approximately 52,000 patients, and its application aims to reproduce and expand on the findings of Kipnis et al. [11]. The data employed in this research is derived from the reproduction study, which is ongoing and has yet to reach a conclusion.

Episodes Patients admitted to the hospital go through different episodes. An episode is any period of time that a patient resides in the hospital (non-ICU) before either being discharged or having an incident. If a patient has an incident and is admitted to the ICU for an extended period, the patient often returns to normal care after their health improves. This marks the start of a new episode, which can again lead to an incident or a discharge. The dataset hence can contain multiple episodes per patient. Logically, most patients do not deteriorate, making the dataset highly imbalanced, as is common in medical datasets.

Static data The static data comprises primarily of control variables, including the season of hospital admission, daytime upon hospital admission (categorical), patient sex, and age. Additionally, the static data includes medical variables, namely the admission category and the Laboratory Acute Physiology score (LAPS) upon admission. Critical to the study is whether an incident occurred at the conclusion of the episode, which is hence included as a static variable. While this was not included in the original classification study, it is crucial to incorporate it as a static variable when generating synthetic data to ensure that the resulting time-series data is dependent on whether the patient deteriorates. Lastly, as deterioration occurs towards the end of an episode, and recurrent neural networks have no perception of when a sequence will be ending, the sequence length is added as a static variable.

Temporal data The temporal data pertains to the patient's state on an hourly basis and includes both categorical and continuous temporal data. The categorical variables primarily comprise the patient's care order (indicating whether a patient has a limited treatment plan e.g. do not resuscitate), their current AVPU scale assessment (indicating their level of consciousness), and whether the patient is currently in the operating room (OR) or ICU. Note that visiting the OR or the ICU for a short period of time does not count as an incident.

The continuous temporal variables encompass laboratory tests and measurements of vitals, totaling 19 variables. These variables include measurements of heart rate, respiratory rate, temperature, sodium levels, and oxygen saturation, among others. A full overview of the variables included in the dataset used in training the generative models is provided in Appendix C Table 1.

Preprocessing The laboratory tests and vital measurements of patients are not assessed every hour during their hospital stay. To account for this, Kipnis et al. [11] retains vital measurements up to 24 hours and lab results up to 72 hours after their initial assessment, after which the value is marked as missing. Median imputation is used to replace all missing data points.

In addition to the preprocessing steps taken from the original and reproduction studies, a box cox transformation is applied to several variables to center the weight of their distribution. When the center of mass of a variable's distribution is more centered within the entire distribution, generative models can more easily learn the subtleties of the distribution. Specifically, as the last activation of the generative models is a sigmoid function, it would severely limit the convergence of the model if the majority of the data it generates is near the limits of the activation function. Lastly, min-max scaling is again applied for the same reasons as in the Philips eICU dataset.

Sequence lengths In contrast to the previous two datasets, this dataset contains sequences that vary a lot in length. In particular, the shortest sequence is only 12 hours, whilst the longest is over 8000 hours (more than 2 months). Although the implementation can handle the varying sequence lengths, the high imbalance, together with the extremely long sequences, makes it very unlikely for the models to converge to any usable state. To resolve these issues, the dataset is simplified by only taking the last 50 hours of every episode. This drastically reduces both the size and complexity of the dataset while retaining the most important data, namely whether a patient deteriorates in the last 12 hours.

Table 1: An overview of all variables in the In-Patient Deterioration dataset, including their value space, context, and variable category.
^{a} ED - emergency department; Surgical - the initial hospital unit to which the patient was admitted was the operating room [11].

Category	Variable	Value space	Context
Static Categorical Variables	Season	1, 2, 3	1: [11, 12, 1, 2] 2: [3, 4, 5, 6] 3: [7, 8, 9, 10]
	Daytime	1, 2, 3, 4	[1-6, 7-12, 13-18, 19-24]
	Admit	1, 2, 3, 4	ED^a, Surgical Non-ED, Surgical ED, Medical Non-ED, Medical
	Sex	0, 1	
	Target	0, 1	Whether an incident occurred after the episode.
Static Continuous Variables	LAPS2_HET	≥ 0	Acute Physiologic Instability upon admission in hospital
	Age	≥ 0	
	Sequence Length	≤ 50	

Continued on next page

Table 1: Continuation of the overview of all variables in the In-Patient Deterioration dataset.

Category	Variable	Value space	Context
Temporal Categorical Variables	Care Order	1, 2, 3	Do not resuscitate, Full-code, partial
	AVPU	1, 2, 3, 4	Alert, Verbal, Pain, Unresponsive
	Oxy-sup	0, 1	
	I_OR	0, 1, 2, 3	Non-acute OR Acute OR Unknown OR IC
	MISS_TROP	0, 1	Whether a troponin value is present
Temporal Continuous Variables	RR	≥ 0	Respiratory Rate
	LAPS2	≥ 0	
	Heart Rate	≥ 0	
	BUN	≥ 0	Blood Urea Nitrogen
	Diastolic BP	≥ 0	
	Creatine	≥ 0	
	Glucose	≥ 0	
	WBC	≥ 0	White blood cell count
	Troponin	≥ 0	
	Lactate	≥ 0	
	Oxy Sat	$[0, 100]$	Oxygen Saturation
	Hematocrit	$[0, 100]$	
	Sodium	≥ 0	
	Temperature	≥ 0	
	Anion Gap	≥ 0	
	Systolic BP	≥ 0	
	Bicarbonate	≥ 0	

D Formalization of Adapted ϵ-Identifiability

Given a training dataset $\mathcal{D}$, a generative model $\mathcal{G}$ is trained. $\mathcal{G}$ generates synthetic samples conditioned on samples $x \in \mathcal{D}$ and a prior noise vector **z** (i.e. $\hat{\mathbf{x}}_i = \mathcal{G}(\mathbf{x}_i, \mathbf{z})$). The identifiability for a synthetic dataset $\hat{\mathcal{D}}$ with respect to $\mathcal{D}$ (both of size N) and a number J of nearest neighbours is then calculated as:

$$\mathcal{I}_{adapted}(\mathcal{D}, \hat{\mathcal{D}}) = \frac{1}{J}\sum_{j=1}^{J}\frac{1}{N}\mathbb{I}(\hat{s}_i < s_{i,j}), \tag{8}$$

where $\mathbb{I}$ represents the identity function and

$$s_{i,j} = ||\mathbf{w} \cdot (\mathbf{x}_i - nn_j(\mathbf{x}_i))||, \tag{9}$$

$$\hat{s}_i = ||\mathbf{w} \cdot (\mathbf{x}_i - \hat{\mathbf{x}}_i)||, \tag{10}$$

where $nn_n(\mathbf{x})$ is the n-th nearest neighbour of **x**, and $\mathbf{x}_i$ is again the i^{th} sample in the dataset. The weight vector **w** provides the necessary discrimination between factors, as some characteristics are more identifiable than others and should therefore be weighed more heavily [20]. The definition returns a number between 0 and 1, where 0 means the synthetic data is perfectly private, and 1 means the synthetic data is not private. Trivially, the Euclidean distance can be substituted with any distance metric of choice.