Journal of the American Medical Informatics Association, 2024, 1–11
https://doi.org/10.1093/jamia/ocae229
Research and Applications

Downloaded from https://academic.oup.com/jamia/advance-article/doi/10.1093/jamia/ocae229/7747780 by Duke Clinical Research Institute, Bernie Chen on 01 October 2024

Research and Applications

Reliable generation of privacy-preserving synthetic electronic health record time series via diffusion models

Muhang Tian, BS[1], Bernie Chen[2], Allan Guo[1], Shiyi Jiang, MS[2], Anru R. Zhang, PhD[1,3,*]

[1]Department of Computer Science, Duke University, Durham, NC 27708, United States, [2]Department of Electrical & Computer Engineering, Duke University, Durham, NC 27708, United States, [3]Department of Biostatistics & Bioinformatics, Duke University, Durham, NC 27708, United States

*Corresponding author: Anru R. Zhang, PhD, Duke University, Department of Computer Science, Department of Biostatistics & Bioinformatics, 2424 Erwin Rd, Durham, NC 27708, United States (anru.zhang@duke.edu)

B. Chen and A. Guo contributed equally to this work.

Abstract

Objective: Electronic health records (EHRs) are rich sources of patient-level data, offering valuable resources for medical data analysis. However, privacy concerns often restrict access to EHRs, hindering downstream analysis. Current EHR deidentification methods are flawed and can lead to potential privacy leakage. Additionally, existing publicly available EHR databases are limited, preventing the advancement of medical research using EHR. This study aims to overcome these challenges by generating realistic and privacy-preserving synthetic EHRs time series efficiently.

Materials and Methods: We introduce a new method for generating diverse and realistic synthetic EHR time series data using denoizing diffusion probabilistic models. We conducted experiments on 6 databases: Medical Information Mart for Intensive Care III and IV, the eICU Collaborative Research Database (eICU), and non-EHR datasets on Stocks and Energy. We compared our proposed method with 8 existing methods.

Results: Our results demonstrate that our approach significantly outperforms all existing methods in terms of data fidelity while requiring less training effort. Additionally, data generated by our method yield a lower discriminative accuracy compared to other baseline methods, indicating the proposed method can generate data with less privacy risk.

Discussion: The proposed model utilizes a mixed diffusion process to generate realistic synthetic EHR samples that protect patient privacy. This method could be useful in tackling data availability issues in the field of healthcare by reducing barrier to EHR access and supporting research in machine learning for health.

Conclusion: The proposed diffusion model-based method can reliably and efficiently generate synthetic EHR time series, which facilitates the downstream medical data analysis. Our numerical results show the superiority of the proposed method over all other existing methods.

Key words: electronic health records; time series generation; diffusion models.

Introduction

The electronic health record (EHR) is a digital version of the patient's medical history maintained by healthcare providers. It includes information such as demographic attributes, vital signals, and lab measurements that are sensitive and important for clinical research. Researchers have been utilizing statistical and machine learning (ML) methods to analyze EHR for a variety of downstream tasks such as disease diagnosis, in-hospital mortality prediction, and disease phenotyping.[1,2] However, due to privacy concerns, EHR data are strictly regulated, and thus the availability of EHR data for research and education is often limited, creating barriers to the development of computational models in the field of healthcare. Widely used EHR deidentification methods to preserve patient information privacy are criticized for having high risks of reidentification of the individuals.[3]

Instead of applying traditional deidentification methods that can adversely affect EHR data utility,[4] EHR synthetic data generation is one promising solution to protect patient privacy. Realistic synthetic data preserve crucial clinical information in real data while preventing patient information leakage.[5,6] Synthetic data also have the added benefit of providing a larger sample size for downstream analysis than deidentifying real samples.[7] As a result, more research initiatives have begun to consider synthetic data sharing, such as the National COVID Cohort Collaborative supported by the U.S. National Institutes of Health and the Clinical Practice Research Datalink sponsored by the U.K. National Institute for Health and Care Research.[8,9] With the advancement in ML techniques, applying generative models to synthesize high-fidelity EHR data is a popular research of interest.[5] Recent advances in generative models have shown significant success in generating realistic high-dimensional data like images, audio, and texts,[10,11] suggesting the potential for these models to handle EHR data with complex statistical characteristics.

Some representative work utilizing generative models for EHR data synthesis includes medGAN,[12] medBGAN,[13] and EHR-Safe (We could not obtain code implementation for this work even after reaching out to the authors. Therefore, we

Received: December 30, 2023; **Revised:** August 4, 2024; **Editorial Decision:** August 11, 2024; **Accepted:** August 12, 2024
© The Author(s) 2024. Published by Oxford University Press on behalf of the American Medical Informatics Association.
All rights reserved. For permissions, please email: journals.permissions@oup.com

are unable to compare TimeDiff with this work's proposed methods.).[6] However, most approaches to EHR data synthesis are GAN-based, and GANs are known for their difficulties in model training and deployments due to training instability and mode collapse.[14] Recently, diffusion probabilistic models have shown superb ability over GANs in generating high-fidelity image data.[15–17] A few studies thus propose to generate synthetic EHR data via diffusion models given their remarkable data generation performance.[18,19] However, most EHR data synthesis methods, either GAN-based or diffusion-based, focus on binary or categorical variables such as the International Classification of Diseases (ICD) codes. Additionally, there is limited prior work on generating EHR data with temporal information, and most state-of-the-art time series generative models are GAN-based.[20] studied the diffusion models for EHR time series generation with a focus only on continuous-valued time series.1 It resorts to Gaussian diffusion for generating discrete sequences, treating them similarly to real-valued sequences but with further postprocessing of the model output. These observations motivate us to bridge the gap by introducing a novel direct diffusion-based method to generate realistic EHR time series data with mixed variable types.

Specifically, we make the following contributions in this paper:

- We propose TimeDiff, a new diffusion probabilistic model that uses a bidirectional recurrent neural network (BRNN) architecture for realistic privacy-preserving EHR time series generation.
- To the best of our knowledge, TimeDiff is the first work introducing a mixed diffusion approach that combines multinomial and Gaussian diffusion for EHR time series generation. TimeDiff can simultaneously generates both continuous and discrete-valued time series.
- We demonstrate that TimeDiff outperforms state-of-the-art methods for time series data generation by a big margin in terms of data fidelity and privacy. Additionally, our model requires less training effort than GAN-based methods.

Time series generation

Prior sequential generation methods using GANs rely primarily on binary adversarial feedback,[21,22] and supervised sequence models mainly focus on tasks such as prediction,[23] forecasting,[24] and classification.[25] TimeGAN[26] was one of the first methods to preserve temporal dynamics in time series synthesis. The architecture comprises an embedding layer, recovery mechanism, generator, and discriminator, trained using both supervised and unsupervised losses. GT-GAN[27] considers the generation of both regular and irregular time series data using a neural controlled differential equation (NCDE) encoder[28] and GRU-ODE decoder.[29] This framework, combined with a continuous time flow processes generator[30] and a GRU-ODE discriminator, outperformed existing methods in general-purpose time series generation. Recently, Biloš et al[31] proposed to generate time series data for forecasting and imputation using discrete or continuous stochastic process diffusion (DSPD/CSPD). Their proposed method views time series as discrete realizations of an underlying continuous function. Both DSPD and CSPD use either the Gaussian or Ornstein-Uhlenbec process to model noise and apply it to the entire time series. The learned distribution over continuous functions is then used to generate synthetic time series samples.

Diffusion models

Diffusion models[32] have been proposed and achieved excellent performance in the field of computer vision and natural language processing. Ho et al[15] proposed denoizing diffusion probabilistic models (DDPMs) that generate high-quality images by recovering from white latent noise. Gu et al[33] proposed a vector-quantized diffusion model on text-to-image synthesis with significant improvement over GANs regarding scene complexity and diversity of the generated images. Dhariwal and Nichol[34] suggested that the diffusion models with optimized architecture outperform GANs on image synthesis tasks. Saharia et al[35] proposed a diffusion model, Imagen, incorporated with a language model for text-to-image synthesis with state-of-the-art results. Kotelnikov et al[36] introduced TabDDPM, an extension of DDPM for heterogeneous tabular data generation, outperforming GAN-based models. Das et al[37] proposed ChiroDiff, a diffusion model that considers temporal information and generates chirographic data. Besides advancements in practical applications, some recent developments in theory for diffusion models demonstrate the effectiveness of this model class. Theoretical foundations explaining the empirical success of diffusion or score-based generative models have been established.[38–40]

EHR data generation

There exists a considerable amount of prior work on generating EHR data. Choi et al[12] proposed medGAN that generates EHR discrete variables. Built upon medGAN, Baowaly et al[13] suggested 2 models, medBGAN and medWGAN, that synthesize EHR binary or discrete variables on ICD codes. Yan et al[41] developed a GAN that can generate high-utility EHR with both discrete and continuous data. Biswal et al[42] proposed the EHR Variational Autoencoder that synthesizes sequences of EHR discrete variables (ie, diagnosis, medications, and procedures). He et al[18] developed MedDiff, a diffusion model that generates user-conditioned EHR discrete variables. Yuan et al[19] created EHRDiff by utilizing the diffusion model to generate a collection of ICD diagnosis codes. Naseer et al[43] used continuous-time diffusion models to generate synthetic EHR tabular data. Ceritli et al[44] applied TabDDPM to synthesize tabular healthcare data.

However, most existing work focuses on discrete or tabular data generation. There is limited literature on EHR time series data generation, and this area of research has not yet received much attention.[45] Back in 2017, RCGAN[22] was created for generating multivariate medical time series data by employing RNNs as the generator and discriminator. Until recently, Yoon et al[6] proposed EHR-Safe that consists of a GAN and an encoder-decoder module. EHR-Safe can generate realistic time series and static variables in EHR with mixed data types. Li et al[46] developed EHR-M-GAN that generates mixed-type time series in EHR using separate encoders for each data type. Theodorou et al[47] suggested generating longitudinal continuous EHR variables using an autoregressive language model. Moreover, Kuo et al[20] suggested utilizing diffusion models to synthesize discrete and continuous EHR time series. However, their approach mainly relies on Gaussian diffusion and adopts a U-Net

Downloaded from https://academic.oup.com/jamia/advance-article/doi/10.1093/jamia/ocae229/7747780 by Duke Clinical Research Institute, Bernie Chen on 01 October 2024

architecture.[48] The generation of discrete time series is achieved by taking argmax of softmax over real-valued one-hot representations. By contrast, our proposed method considers multinomial diffusion for discrete time series generation, allowing the generation of discrete variables directly. He et al,[49] a concurrent work to ours, introduces FLEXGEN-EHR for synthesizing heterogeneous longitudinal EHR data through a latent diffusion method. It also addresses missing modalities by formulating an optimal transport problem to create meaningful latent embedding pairs. In comparison, our work introduces a direct diffusion model to generate heterogeneous EHR data, effectively handling potential missingness directly within the generation process.

Methods

Datasets

We use 4 publicly available EHR datasets to evaluate TimeDiff: Medical Information Mart for Intensive Care III and IV (MIMIC-III/IV)[50,51] and the eICU Collaborative Research Database (eICU).[52] Additionally, to evaluate TimeDiff with state-of-the-art methods for time series generation on non-EHR datasets, we include Stocks and Energy datasets from studies that proposed TimeGAN[26] and GT-GAN.[27]

Metrics

We evaluate our methods and make comparisons on a series of metrics, both qualitative and quantitative, characterizing the authenticity of the synthesized data, the performance for downstream analysis—in-hospital mortality prediction, and the preservation of privacy:

Authenticity

- t-SNE visualization: We flatten the feature dimension and use t-SNE dimension reduction visualization[53] on synthetic, real training, and real testing samples. This qualitative metric provides visual guidance on the similarity of the synthetic and real samples in 2D space. Details are described in A.5.2.
- UMAP visualization: We follow the same procedure as using t-SNE for visualization of distribution similarity between synthetic, real training, and real testing samples. UMAP preserves a better global structure compared to t-SNE,[54] and thus we provide it as a complementary metric.
- Discriminative and Predictive Scores: A GRU-based discriminator is trained to distinguish between the synthetic and real samples. For the predictive score, a GRU-based predictor is trained using synthetic samples and evaluated on real samples for next-step vector prediction based on mean absolute error over each sequence. Details of the score computations are described in A.5.2.

Performance for downstream task (in-hospital mortality prediction)

- Train on Synthetic, Test on Real (TSTR): We train ML models using synthetic data and evaluate them on real test data based on the area under the receiver operating characteristic curve (AUC) for in-hospital mortality prediction. We compare the TSTR score to the Train on Real, Test on Real (TRTR) score, which is the AUC obtained from the model trained on real training data and evaluated on real test data.
- Train on Synthetic and Real, Test on Real (TSRTR): Similar to the TSTR, we train ML models and evaluate them on real test data using AUC. We use 2000 real training data in combination with different proportions of synthetic samples to train ML models. This metric evaluates the impact of synthetic data for training on ML model performance. Note that we use 2000 real training samples to simulate the real-world scenario where clinical researchers struggle to obtain limited real EHR data. In this case, we evaluate the viability of using TimeDiff to generate realistic samples as a data augmentation technique.

Privacy

- Nearest Neighbor Adversarial Accuracy Risk (NNAA): This score measures the degree to which a generative model overfits the real training data.[55] NNAA is an important metric for evaluating the privacy of synthetic data as it quantifies the risk of reidentification by measuring how easily an adversary can distinguish between real and synthetic data points. Thus, this metric effectively indicates the potency of anonymization techniques in protecting sensitive information within the synthetic dataset.
- Membership Inference Risk (MIR): An F1 score is computed based on whether an adversary can correctly identify the membership of a synthetic data sample.[56] MIR provides a precise measurement of the security of synthetic datasets, particularly in assessing the likelihood that individual data points can be traced back to the original dataset, thereby evaluating the robustness of data anonymization techniques.

For all the experiments, we split each dataset into training and testing sets and used the training set to develop generative models. The synthetic samples obtained from trained generative models are then used for evaluation. We repeat each experiment over 10 times and report the mean and SD of each quantitative metric. Further details for our experiments and evaluation metrics are discussed in Supplementary Appendix A.

Baselines

We compare TimeDiff with 9 methods: HALO,[47] EHR-M-GAN,[46] GT-GAN,[27] TimeGAN,[26] RCGAN,[22] C-RNN-GAN,[21] RNNs trained with teacher forcing (T-Forcing)[57,58] and professor forcing (P-Forcing),[59] and DSPD/CSPD with Gaussian (GP) or Ornstein-Uhlenbeck (OU) processes (We also hoped to include comparison with EHR-Safe.[6] However, despite attempts, we were unable to obtain the code implementation.).[31] In addition we compare with standard GRU and LSTM approach, with results in Table 1.

Diffusion process on EHR time series

We first introduce our notations for the generation of both continuous-valued and discrete-valued time series in our framework, as both are present in EHR. Specifically, let $\mathcal{D}$ denote our EHR time series dataset. Each patient in $\mathcal{D}$ has continuous-valued and discrete-valued multivariate time series $\boldsymbol{X} \in \mathbb{R}^{P_r \times L}$ and $\boldsymbol{C} \in \mathbb{Z}^{P_d \times L}$, respectively. L is the number of time steps, and P_r and P_d are the number of variables for continuous and discrete data types.

Downloaded from https://academic.oup.com/jamia/advance-article/doi/10.1093/jamia/ocae229/7747780 by Duke Clinical Research Institute, Bernie Chen on 01 October 2024

Table 1. Predictive and discriminative scores of TimeDiff and the baselines.

Metric	Method	Stocks	Energy	MIMIC-III	MIMIC-IV	eICU
Discriminative Score (↓)	**TimeDiff**	**0.048 ± 0.028**	**0.088 ± 0.018**	**0.028 ± 0.023**	**0.030 ± 0.022**	**0.015 ± 0.007**
	EHR-M-GAN	0.483 ± 0.027	0.497 ± 0.006	0.499 ± 0.002	0.499 ± 0.001	0.488 ± 0.022
	DSPD-GP	0.081 ± 0.034	0.416 ± 0.016	0.491 ± 0.002	0.478 ± 0.020	0.327 ± 0.020
	DSPD-OU	0.098 ± 0.030	0.290 ± 0.010	0.456 ± 0.014	0.444 ± 0.037	0.367 ± 0.018
	CSPD-GP	0.313 ± 0.061	0.392 ± 0.007	0.498 ± 0.001	0.488 ± 0.010	0.489 ± 0.010
	CSPD-OU	0.283 ± 0.039	0.384 ± 0.012	0.494 ± 0.002	0.479 ± 0.005	0.479 ± 0.017
	GT-GAN	0.077 ± 0.031	0.221 ± 0.068	0.488 ± 0.026	0.472 ± 0.014	0.448 ± 0.043
	TimeGAN	0.102 ± 0.021	0.236 ± 0.012	0.473 ± 0.019	0.452 ± 0.027	0.434 ± 0.061
	RCGAN	0.196 ± 0.027	0.336 ± 0.017	0.498 ± 0.001	0.490 ± 0.003	0.490 ± 0.023
	C-RNN-GAN	0.399 ± 0.028	0.499 ± 0.001	0.500 ± 0.000	0.499 ± 0.000	0.493 ± 0.010
	T-Forcing	0.226 ± 0.035	0.483 ± 0.004	0.499 ± 0.001	0.497 ± 0.002	0.479 ± 0.011
	P-Forcing	0.257 ± 0.026	0.412 ± 0.006	0.494 ± 0.006	0.498 ± 0.002	0.367 ± 0.047
	HALO	0.491 ± 0.006	0.500 ± 0.000	0.497 ± 0.003	0.494 ± 0.004	0.370 ± 0.074
	Real Data	*0.019 ± 0.016*	*0.016 ± 0.006*	*0.012 ± 0.006*	*0.014 ± 0.011*	*0.004 ± 0.003*
Predictive Score (↓)	**TimeDiff**	**0.037 ± 0.000**	**0.251 ± 0.000**	**0.469 ± 0.003**	**0.432 ± 0.002**	**0.309 ± 0.019**
	EHR-M-GAN	0.120 ± 0.047	0.254 ± 0.001	0.861 ± 0.072	0.880 ± 0.079	0.913 ± 0.179
	DSPD-GP	0.038 ± 0.000	0.260 ± 0.001	0.509 ± 0.014	0.586 ± 0.026	0.320 ± 0.018
	DSPD-OU	0.039 ± 0.000	0.252 ± 0.000	0.497 ± 0.006	0.474 ± 0.023	0.317 ± 0.023
	CSPD-GP	0.041 ± 0.000	0.257 ± 0.001	1.083 ± 0.002	0.496 ± 0.034	0.624 ± 0.066
	CSPD-OU	0.044 ± 0.000	0.253 ± 0.000	0.566 ± 0.006	0.516 ± 0.051	0.382 ± 0.026
	GT-GAN	0.040 ± 0.000	0.312 ± 0.002	0.584 ± 0.010	0.517 ± 0.016	0.487 ± 0.033
	TimeGAN	0.038 ± 0.001	0.273 ± 0.004	0.727 ± 0.010	0.548 ± 0.022	0.367 ± 0.025
	RCGAN	0.040 ± 0.001	0.292 ± 0.005	0.837 ± 0.040	0.700 ± 0.014	0.890 ± 0.017
	C-RNN-GAN	0.038 ± 0.000	0.483 ± 0.005	0.933 ± 0.046	0.811 ± 0.048	0.769 ± 0.045
	T-Forcing	0.038 ± 0.001	0.315 ± 0.005	0.840 ± 0.013	0.641 ± 0.017	0.547 ± 0.069
	P-Forcing	0.043 ± 0.001	0.303 ± 0.006	0.683 ± 0.031	0.557 ± 0.030	0.345 ± 0.021
	HALO	0.042 ± 0.006	0.299 ± 0.053	0.816 ± 0.020	0.767 ± 0.012	0.378 ± 0.038
	Real Data	*0.036 ± 0.001*	*0.250 ± 0.003*	*0.467 ± 0.005*	*0.433 ± 0.001*	*0.304 ± 0.017*

Bolded values are best-performing models, and italic values are for real data.

To generate both continuous-valued and discrete-valued time series, we consider a "mixed sequence diffusion" approach by adding Gaussian and multinomial noises. For continuous-valued time series, we perform Gaussian diffusion by adding independent Gaussian noise similar to DDPM. The forward process is thus defined as:

$$q\left(\boldsymbol{X}^{(1:T)}|\boldsymbol{X}^{(0)}\right)=\prod_{t=1}^{T}\prod_{l=1}^{L}q\left(\boldsymbol{X}_{\cdot,l}^{(t)}|\boldsymbol{X}_{\cdot,l}^{(t-1)}\right), \tag{1}$$

where $q(\boldsymbol{X}_{\cdot,l}^{(t)}|\boldsymbol{X}_{\cdot,l}^{(t-1)})=\mathcal{N}(\boldsymbol{X}_{\cdot,l}^{(t)};\sqrt{1-\beta^{(t)}}\boldsymbol{X}_{\cdot,l}^{(t-1)},\beta^{(t)}\boldsymbol{I})$ and $\boldsymbol{X}_{\cdot,l}$ is the l^{th} observation of the continuous-valued time series. In a similar fashion as eqn (12), we define the reverse process for continuous-valued features as $p_\theta(\boldsymbol{X}^{(0:T)})=p(\boldsymbol{X}^{(T)})\prod_{t=1}^{T}p_\theta(\boldsymbol{X}^{(t-1)}|\boldsymbol{X}^{(t)})$, where

$$\begin{aligned}
p_\theta\left(\boldsymbol{X}^{(t-1)}|\boldsymbol{X}^{(t)}\right)&:=\mathcal{N}\left(\boldsymbol{X}^{(t-1)};\boldsymbol{\mu}_\theta\left(\boldsymbol{X}^{(t)},t\right),\tilde{\beta}^{(t)}\boldsymbol{I}\right),\\
\boldsymbol{\mu}_\theta\left(\boldsymbol{X}^{(t)},t\right)&=\frac{1}{\sqrt{\alpha^{(t)}}}\left(\boldsymbol{X}^{(t)}-\frac{\beta^{(t)}}{\sqrt{1-\bar{\alpha}^{(t)}}}\boldsymbol{s}_\theta\left(\boldsymbol{X}^{(t)},t\right)\right),\\
\tilde{\beta}^{(t)}&=\frac{1-\bar{\alpha}^{(t-1)}}{1-\bar{\alpha}^{(t)}}\beta^{(t)}.
\end{aligned} \tag{2}$$

To model discrete-valued time series, we use multinomial diffusion.[60] The forward process is defined as:

$$q\left(\tilde{\boldsymbol{C}}^{(1:T)}|\tilde{\boldsymbol{C}}^{(0)}\right)=\prod_{t=1}^{T}\prod_{p=1}^{P_d}\prod_{l=1}^{L}q\left(\tilde{\boldsymbol{C}}_{p,l}^{(t)}|\tilde{\boldsymbol{C}}_{p,l}^{(t-1)}\right), \tag{3}$$

$$q\left(\tilde{\boldsymbol{C}}_{p,l}^{(t)}|\tilde{\boldsymbol{C}}_{p,l}^{(t-1)}\right):=\mathcal{C}\left(\tilde{\boldsymbol{C}}_{p,l}^{(t)};\left(1-\beta^{(t)}\right)\tilde{\boldsymbol{C}}_{p,l}^{(t-1)}+\beta^{(t)}/K\right), \tag{4}$$

where $\mathcal{C}$ is a categorical distribution, $\tilde{\boldsymbol{C}}_{p,l}^{(0)}\in\{0,1\}^K$ is a one-hot encoded representation of $C_{p,l}$ (We perform one-hot encoding on the discrete-valued time series across the feature dimension. For example, if our time series is {0, 1, 2}, its one-hot representation becomes $\{[1,0,0]^\top,[0,1,0]^\top,[0,0,1]^\top\}$.), and the addition and subtraction between scalars and vectors are performed element-wise. The forward process posterior distribution is defined as follows, where $\odot$ represents the Hadamard product that returns a matrix with each element being the product of the corresponding elements from the original 2 matrices:

$$q\left(\tilde{\boldsymbol{C}}_{p,l}^{(t-1)}|\tilde{\boldsymbol{C}}_{p,l}^{(t)},\tilde{\boldsymbol{C}}_{p,l}^{(0)}\right):=\mathcal{C}\left(\tilde{\boldsymbol{C}}_{p,l}^{(t-1)};\boldsymbol{\phi}/\sum_{k=1}^{K}\phi_k\right), \tag{5}$$

$$\boldsymbol{\phi}=\left(\alpha^{(t)}\tilde{\boldsymbol{C}}_{p,l}^{(t)}+\left(1-\alpha^{(t)}\right)/K\right)\odot\left(\bar{\alpha}^{(t-1)}\tilde{\boldsymbol{C}}_{p,l}^{(0)}+\left(1-\bar{\alpha}^{(t-1)}\right)/K\right). \tag{6}$$

The reverse process $p_\theta(\tilde{\boldsymbol{C}}_{p,l}^{(t-1)}|\tilde{\boldsymbol{C}}_{p,l}^{(t)})$ is parameterized as $q(\tilde{\boldsymbol{C}}_{p,l}^{(t-1)}|\tilde{\boldsymbol{C}}_{p,l}^{(t)},\boldsymbol{s}_\theta(\tilde{\boldsymbol{C}}_{p,l}^{(t)},t))$. We train our neural network, $\boldsymbol{s}_\theta$, using both Gaussian and multinomial diffusion processes:

$$\mathcal{L}_{\mathcal{N}}(\theta):=\mathbb{E}_{\boldsymbol{X}^{(0)},\boldsymbol{\varepsilon},t}\left[\|\boldsymbol{\varepsilon}-\boldsymbol{s}_\theta\left(\sqrt{\bar{\alpha}^{(t)}}\boldsymbol{X}^{(0)}+\sqrt{1-\bar{\alpha}^{(t)}}\boldsymbol{\varepsilon},t\right)\|^2\right], \tag{7}$$

Downloaded from https://academic.oup.com/jamia/advance-article/doi/10.1093/jamia/ocae229/7747780 by Duke Clinical Research Institute, Bernie Chen on 01 October 2024

$$\mathcal{L}_{\mathcal{C}}(\theta) := \mathbb{E}_{p,l}\left[\sum_{t=2}^{T} D_{\mathrm{KL}}\left(q\left(\tilde{\boldsymbol{C}}_{p,l}^{(t-1)} | \tilde{\boldsymbol{C}}_{p,l}^{(t)}, \tilde{\boldsymbol{C}}_{p,l}^{(0)}\right) || p_\theta\left(\tilde{\boldsymbol{C}}_{p,l}^{(t-1)} | \tilde{\boldsymbol{C}}_{p,l}^{(t)}\right)\right)\right], \tag{8}$$

where $\mathcal{L}_{\mathcal{N}}$ and $\mathcal{L}_{\mathcal{C}}$ are the losses for continuous-valued and discrete-valued multivariate time series, respectively. The training of the neural network is performed by minimizing the following loss:

$$\mathcal{L}_{\text{train}}(\theta) = \lambda \mathcal{L}_{\mathcal{C}}(\theta) + \mathcal{L}_{\mathcal{N}}(\theta), \tag{9}$$

where λ is a hyperparameter for creating a balance between the 2 losses. We investigate the effects of λ in Supplementary Appendix B.6.

Missing value representation

In medical applications, missing data and variable measurement times play a crucial role as they could provide additional information and indicate a patient's health status.[61] We thus derive a missing indicator mask $\boldsymbol{M} \in \{0,1\}^{P_r \times L}$ (or alternatively, $\boldsymbol{M} \in \{0,1\}^{P_d \times L}$ if the time series is discrete-valued) for each $\boldsymbol{X} \in \mathcal{D}$ (For simplicity in writing, we refer to $\boldsymbol{X}$ only, but this procedure can also be applied on $\boldsymbol{C}$.):

$$M_{p,l} = \begin{cases} 0, & \text{if } X_{p,l} \text{ is present;} \\ 1, & \text{if } X_{p,l} \text{ is missing.} \end{cases} \tag{10}$$

Then $\boldsymbol{M}$ encodes the measurement time points of $\boldsymbol{X}$. If $\boldsymbol{X}$ contains missing values, we impute them in the initial value of the forward process, that is, $\boldsymbol{X}^{(0)}$, using the corresponding sample mean (Using the sample mean for imputation is a straightforward and computationally efficient method. It helps maintain the central tendencies and distributional characteristics of the original data, minimizing the introduction of biases that might occur with more complex methods.[62,63]). Nevertheless, $\boldsymbol{M}$ retains the information regarding the positions of missing values. Our method generates discrete and continuous-valued time series, allowing us to seamlessly represent and generate $\boldsymbol{M}$ as a discrete time series.

TIMEDIFF architecture

In this section, we describe our architecture for the diffusion model. A commonly used architecture in DDPM is U-Net.[48] However, most U-Net-based models are tailored to image generation tasks, requiring the neural network to process pixel-based data rather than sequential information.[15,17,64] Even its 1D variant, 1D-U-Net, comes with limitations such as restriction on the input sequence length (which must be a multiple of U-Net multipliers) and a tendency to lose temporal dynamics information during down-sampling. On the other hand, TabDDPM[36] proposed a mixed diffusion approach for tabular data generation but relied on a multilayer perceptron architecture, making it improper for multivariate time series generation.

To address this challenge of handling EHR time series, we need an architecture capable of encoding sequential information while being flexible to the input sequence length. The time-conditional BRNN or NCDE[28] can be possible options. After careful evaluation, we found that BRNN without attention mechanism offers superior computational efficiency and have chosen it as the neural backbone s_θ for all of our experiments. A more detailed discussion of NCDE is provided in Supplementary Material SA.4.1.

Diffusion step embedding

To inform the model about the current diffusion time step t, we use sinusoidal positional embedding.[65] The embedding vector output from the embedding layer then goes through 2 fully connected (FC) layers with GeLU activation in between.[66] The embedding vector is then fed to a SiLU activation[66] and another FC layer. The purpose of this additional FC layer is to adjust the dimensionality of the embedding vector to match the stacked hidden states from BRNN. Specifically, we set the dimensionality of the output to be 2 times the size of the hidden dimension from BRNN. We denote the transformed embedding vector as $\boldsymbol{t}_{\text{embed}}$. This vector is then split into 2 vectors, each with half of the current size, namely $\boldsymbol{t}_{\text{embed\_scale}}$ and $\boldsymbol{t}_{\text{embed\_shift}}$. Both vectors share the same dimensionality as BRNN's hidden states and serve to inform the network about the current diffusion time step.

Time-conditional BRNN

In practice, BRNN can be implemented with either LSTM or GRU units. To condition BRNN on time, we follow these steps. We first obtain noisy samples from Gaussian (for continuous-valued data) and multinomial (for discrete-valued data) diffusion. The 2 samples are concatenated and fed to our BRNN, which returns a sequence of hidden states $\{\boldsymbol{h}_l\}_{l=1}^{L}$ that stores the temporal dynamics information about the time series. To stabilize learning and enable proper utilization of $\boldsymbol{t}_{\text{embed}}$, we apply layernorm[67] on $\{\boldsymbol{h}_l\}_{l=1}^{L}$. The normalized sequence of hidden states, $\{\tilde{\boldsymbol{h}}_l\}_{l=1}^{L}$, is then scaled and shifted using $\{\tilde{\boldsymbol{h}}_l \odot (\boldsymbol{t}_{\text{embed\_scale}} + \boldsymbol{1}) + \boldsymbol{t}_{\text{embed\_shift}}\}_{l=1}^{L}$. These scaled hidden states contain information about the current diffusion step t, which is then passed through an FC layer to produce the final output. The output contains predictions for both multinomial and Gaussian diffusions, which are extracted correspondingly and used to calculate $\mathcal{L}_{\text{train}}$ in eqn (9). A visual demonstration of our architecture is shown in Figure 1, where the use of BRNN allows the denoizing of noisy time series samples of arbitrary length L, and the diffusion step embedding is utilized to inform the model about the stage of the reverse diffusion process.

Results

Authenticity of the generated EHR time series

We evaluate the authenticity of the generated synthetic EHR time series both qualitatively and quantitatively. We provide a visualization of the distributions of the synthetic and real data using t-SNE, following,[6] shown in Figure 2. Additionally, we present another visualization metric using UMAP in Figure 3. Both visualization methods indicate the synthesized data generated from TIMEDIFF overlaps with real training and test data, suggesting TIMEDIFF can generate more realistic data compared to other baselines. Visualizations of the raw synthetic and real data per feature are presented in Supplementary Appendix B.1. Note that t-SNE and UMAP are only for qualitative evaluation and are not precise. We next present quantitative metrics for precise evaluation. By comparing the predictive and discriminative scores in Table 1, we observe that TIMEDIFF yields significantly lower scores than all the baseline methods across 6 datasets. For instance, TIMEDIFF yields a 95.4% lower mean discriminative score

Downloaded from https://academic.oup.com/jamia/advance-article/doi/10.1093/jamia/ocae229/7747780 by Duke Clinical Research Institute, Bernie Chen on 01 October 2024

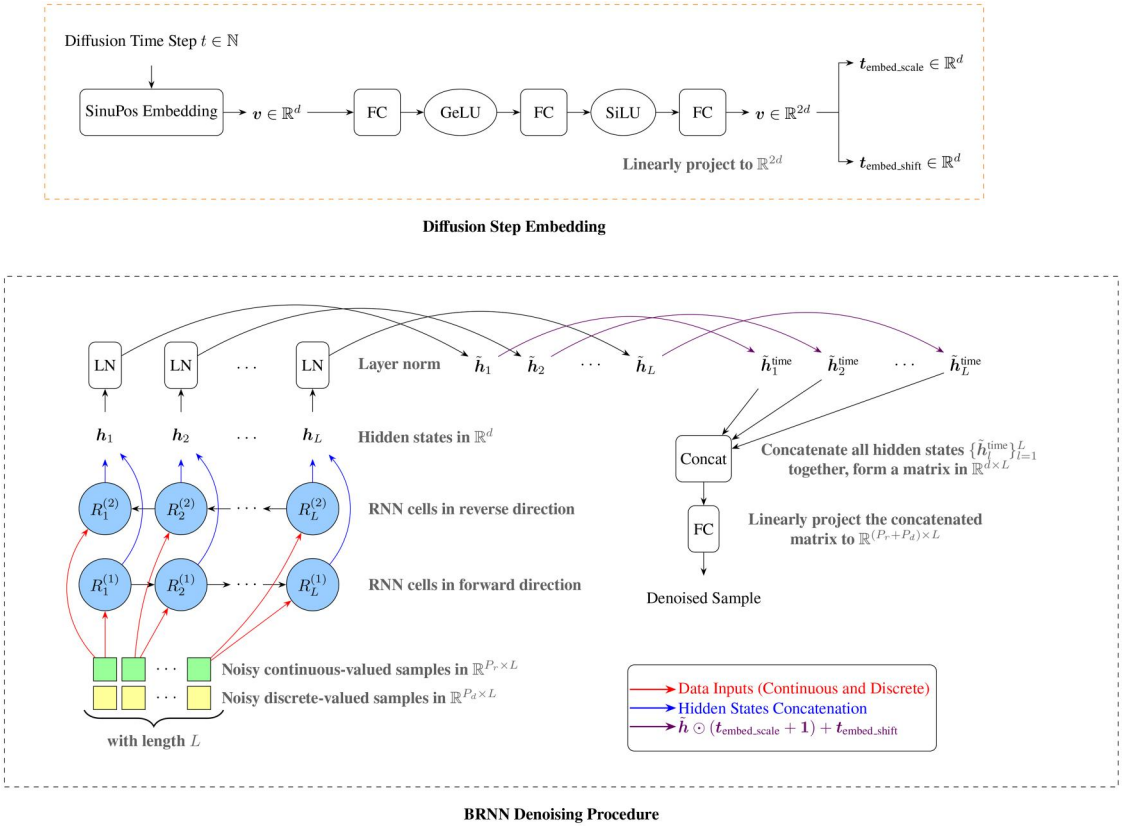

Figure 1. Visualization of TimeDiff architecture. FC, fully connected layer; SiLU, sigmoid linear unit activation; SinuPos Embedding, shorthand for sinusoidal positional embedding; GeLU, Gaussian error linear unit activation.

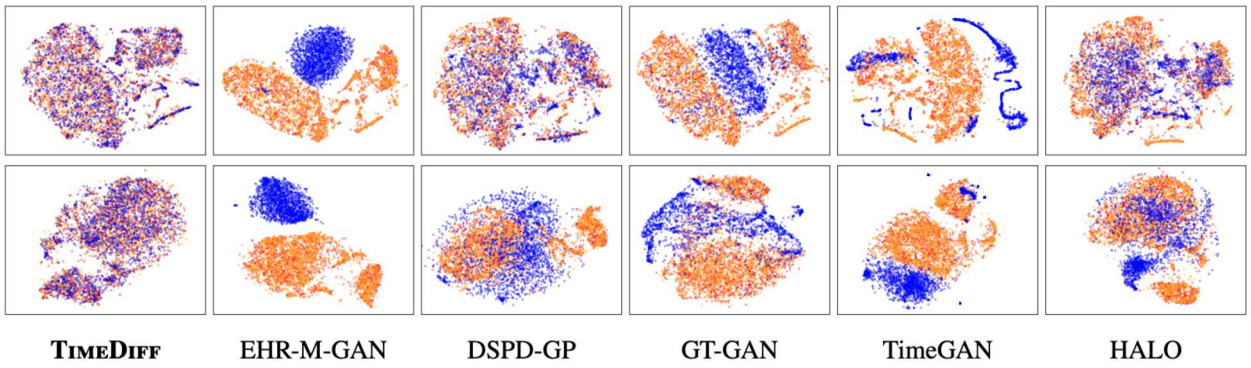

Figure 2. t-SNE visualization of the eICU (first row) and the MIMIC-IV (second row) datasets. Synthetic samples in blue, real training samples in red, and real testing samples in orange. We observe that there is a significant overlap between synthetic samples from TimeDiff and real testing samples, suggesting TimeDiff produces realistic synthetic EHR data. DSPD-GP and HALO also yield noticeable overlap.

compared to DSPD-GP and obtains a 1.6% higher mean predictive score than real testing data on the eICU dataset. For non-EHR datasets, TimeDiff achieves a 37.7% lower and a 60.2% lower mean discriminative scores on the Stocks and Energy datasets than GT-GAN while having similar mean predictive scores as using real testing data.

Data performance on in-hospital mortality prediction

We evaluate the data performance of the generated synthetic EHR time series on one common downstream task: in-hospital mortality prediction.[68,69] We use 6 ML algorithms: XGBoost (XGB),[70] Random Forest (RF),[71] AdaBoost (AB),[72] and ℓ_1 and ℓ_2 regularized Logistic Regression (LR L1/L2).[73] Additionally, to simulate the practical scenario where synthetic samples are used for data augmentation, we compute the TSRTR score for each ML model. The prediction models are trained using synthetic samples from TimeDiff and assessed on real test data.

From Figure 4, we observe that the TSTR scores obtained from models trained using synthetic EHR time series are close to the TRTR scores yielded from models trained using real data. We also notice a nondecreasing trend in the TSRTR scores as the percentage of synthetic EHR data increases for

Downloaded from https://academic.oup.com/jamia/advance-article/doi/10.1093/jamia/ocae229/7747780 by Duke Clinical Research Institute, Bernie Chen on 01 October 2024

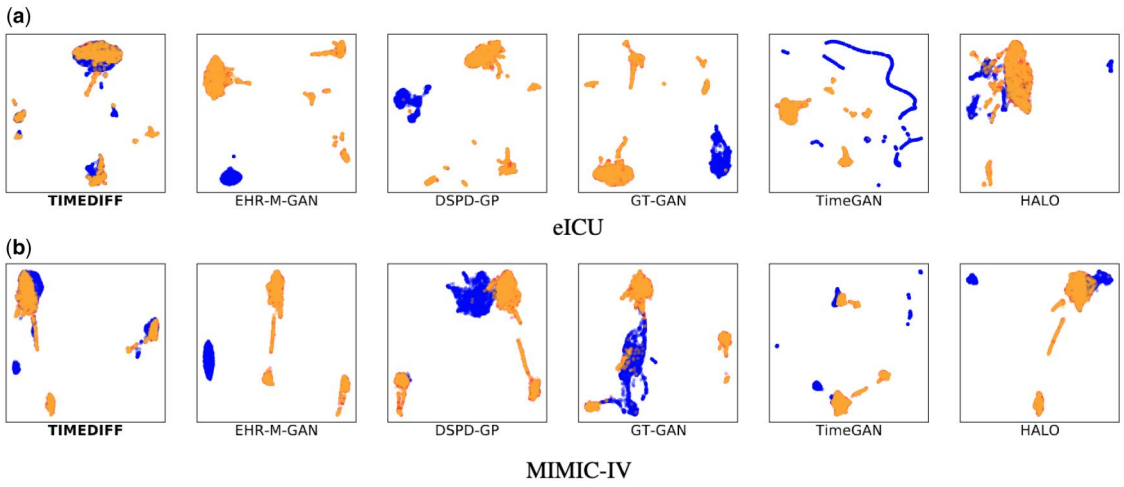

Figure 3. UMAP visualization of the eICU and the MIMIC-IV datasets. Synthetic samples in blue, real training samples in red, and real testing samples in orange. We observe a similar result as the t-SNE visualizations, where there is an overlap between synthetic and real testing samples for TimeDiff. The overlap for other models is less significant.

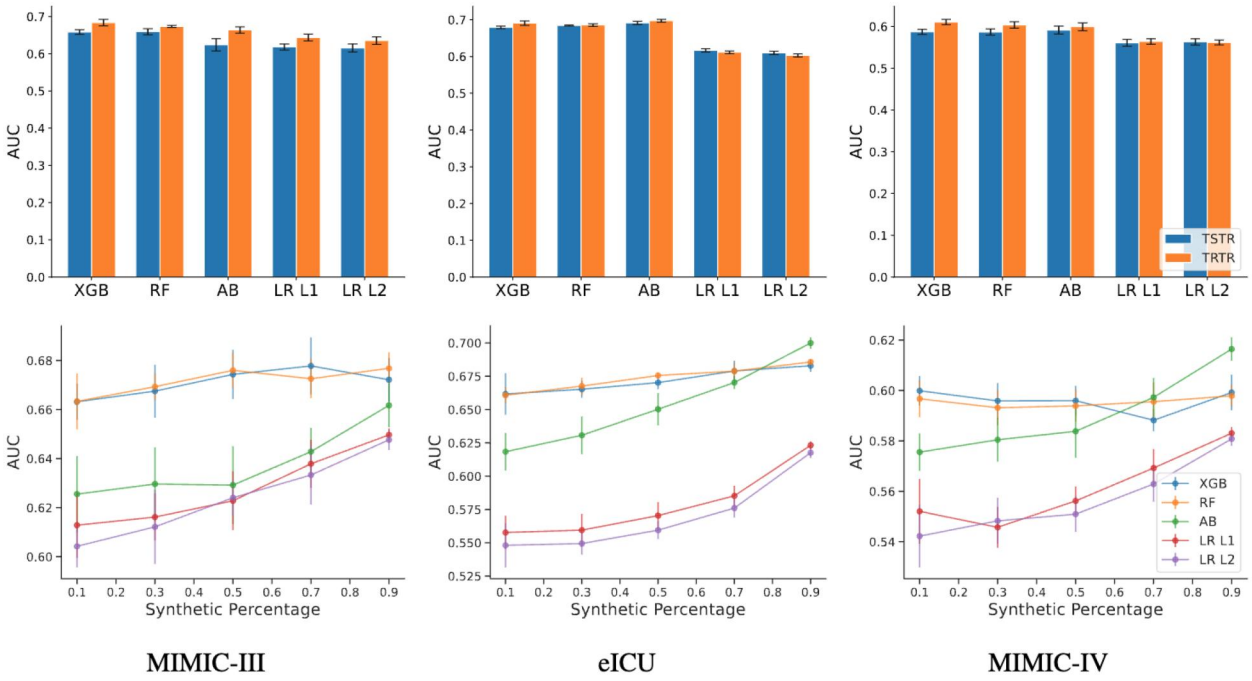

Figure 4. TSTR scores compared to TRTR scores (Top) and TSRTR scores (Bottom).

ML model training. Additional TSTR and TSRTR evaluations for all baseline generative models can be found in Supplementary Material SB.4.2.

Note that in addition to the 6 ML classifiers mentioned above, we utilize GRU and LSTM for prediction due to their ability in handling sequential data. We present the TSTR and TRTR scores obtained from RNNs in Supplementary Material SB.4.3, Table S10. We observe that they achieve lower scores compared to the conventional classifiers.

Data privacy of synthetic EHR time series

We also assess the risks of the generated synthetic EHR time series being attacked by malicious entities using the NNAA and the MIR scores. These metrics allow us to evaluate whether our approach can produce privacy-preserving synthetic EHR samples. As presented in Table 2, we observe that TimeDiff yields the AA_{test} and AA_{train} scores around 0.5 across all 4 EHR datasets. TimeDiff also obtains low NNAA and MIR scores compared to baseline methods. Note that the full results are presented in Supplementary Material SB.4.

Model runtime comparison

We compare the number of hours to train TimeDiff with EHR-M-GAN, TimeGAN, and GT-GAN presented in Table 3. As shown in Table 3, TimeDiff requires less training time compared to GAN-based approaches.

Note that in our experiments, to demonstrate that our proposed approach outperforms GANs in terms of training time, we primarily compared the training time of our proposed diffusion model with GANs. Most of the GANs primarily involve pretraining the embedding layer and subsequently training with adversarial feedback. This staged procedure

Downloaded from https://academic.oup.com/jamia/advance-article/doi/10.1093/jamia/ocae229/7747780 by Duke Clinical Research Institute, Bernie Chen on 01 October 2024

Table 2. Privacy scores of the synthesized EHR time series yielded from TimeDiff and the baseline methods.

Metric	Method	MIMIC-III	MIMIC-IV	eICU
AA_{test} (∼0.5)	**TimeDiff**	**0.574 ± 0.002**	**0.517 ± 0.002**	**0.537 ± 0.001**
	EHR-M-GAN	0.998 ± 0.000	1.000 ± 0.000	0.977 ± 0.000
	DSPD-GP	0.974 ± 0.001	0.621 ± 0.002	0.888 ± 0.000
	DSPD-OU	0.927 ± 0.000	0.804 ± 0.003	0.971 ± 0.000
	CSPD-GP	0.944 ± 0.001	0.623 ± 0.002	0.851 ± 0.001
	CSPD-OU	0.967 ± 0.001	0.875 ± 0.002	0.982 ± 0.000
	GT-GAN	0.995 ± 0.000	0.910 ± 0.001	0.981 ± 0.000
	TimeGAN	0.997 ± 0.000	0.974 ± 0.001	1.000 ± 0.000
	RCGAN	0.983 ± 0.001	0.999 ± 0.000	1.000 ± 0.000
	HALO	0.698 ± 0.002	0.709 ± 0.002	0.653 ± 0.001
	Real Data	*0.552 ± 0.002*	*0.497 ± 0.002*	*0.501 ± 0.002*
AA_{train} (∼0.5)	**TimeDiff**	**0.573 ± 0.002**	**0.515 ± 0.002**	**0.531 ± 0.002**
	EHR-M-GAN	0.999 ± 0.000	1.000 ± 0.000	0.965 ± 0.002
	DSPD-GP	0.968 ± 0.002	0.620 ± 0.003	0.888 ± 0.001
	DSPD-OU	0.928 ± 0.001	0.788 ± 0.003	0.971 ± 0.000
	CSPD-GP	0.940 ± 0.002	0.629 ± 0.005	0.852 ± 0.001
	CSPD-OU	0.966 ± 0.001	0.880 ± 0.003	0.983 ± 0.000
	GT-GAN	0.995 ± 0.001	0.907 ± 0.002	0.981 ± 0.000
	TimeGAN	0.997 ± 0.000	0.969 ± 0.003	1.000 ± 0.000
	RCGAN	0.984 ± 0.001	0.999 ± 0.000	1.000 ± 0.000
	HALO	0.696 ± 0.001	0.717 ± 0.002	0.653 ± 0.002
	Real Data	*0.286 ± 0.003*	*0.268 ± 0.004*	*0.266 ± 0.002*
NNAA (↓)	**TimeDiff**	0.002 ± 0.002	0.002 ± 0.002	0.006 ± 0.002
	EHR-M-GAN	0.000 ± 0.000	0.000 ± 0.000	0.012 ± 0.003
	DSPD-GP	0.005 ± 0.003	0.003 ± 0.003	0.001 ± 0.001
	DSPD-OU	0.001 ± 0.001	0.016 ± 0.004	0.000 ± 0.000
	CSPD-GP	0.004 ± 0.002	0.007 ± 0.005	0.001 ± 0.001
	CSPD-OU	0.001 ± 0.001	0.005 ± 0.003	0.001 ± 0.001
	GT-GAN	0.001 ± 0.000	0.004 ± 0.002	0.000 ± 0.000
	TimeGAN	0.000 ± 0.000	0.005 ± 0.003	0.000 ± 0.000
	RCGAN	0.001 ± 0.000	0.000 ± 0.000	0.000 ± 0.000
	HALO	0.002 ± 0.002	0.008 ± 0.002	0.002 ± 0.001
	Real Data	*0.267 ± 0.004*	*0.229 ± 0.003*	*0.235 ± 0.003*
MIR (↓)	**TimeDiff**	0.191 ± 0.008	0.232 ± 0.048	0.227 ± 0.021
	EHR-M-GAN	0.025 ± 0.007	0.435 ± 0.031	0.049 ± 0.006
	DSPD-GP	0.032 ± 0.021	0.050 ± 0.009	0.000 ± 0.000
	DSPD-OU	0.060 ± 0.032	0.007 ± 0.006	0.000 ± 0.000
	CSPD-GP	0.060 ± 0.028	0.034 ± 0.017	0.000 ± 0.000
	CSPD-OU	0.066 ± 0.046	0.016 ± 0.020	0.000 ± 0.000
	GT-GAN	0.005 ± 0.002	0.046 ± 0.013	0.000 ± 0.000
	TimeGAN	0.010 ± 0.002	0.173 ± 0.020	0.000 ± 0.000
	RCGAN	0.013 ± 0.002	0.277 ± 0.049	0.000 ± 0.000
	HALO	0.189 ± 0.007	0.019 ± 0.012	0.036 ± 0.040
	Real Data	*0.948 ± 0.000*	*0.929 ± 0.005*	*0.927 ± 0.001*

Bolded values are for best-performing models, and italic values are for real data.

Table 3. Runtime comparisons between the TimeDiff and baseline methods (hours).

Dataset	TimeDiff	EHR-M-GAN	TimeGAN	GT-GAN
MIMIC-III	**2.7**	18.9	10.8	21.8
MIMIC-IV	**2.7**	28.8	29.5	47.3
eICU	**8.7**	87.1	110	59.1

Bolded values are the best-performing models.

made GANs more computationally heavy to train than diffusion models (which only requires optimizing one loss function for one neural network in our proposed approach).

Ablation study

We further investigate the effect of utilizing multinomial diffusion in TimeDiff on missing indicators for EHR discrete sequence generation. We compare it with TimeDiff using Gaussian diffusion, with the following 2 methods applied to the resulting output as transformations to discrete sequences: (1) direct rounding; (2) applying argmax to the softmax output of real-valued, one-hot encoded representations (The synthetic one-hot encoding is not discrete since we use Gaussian diffusion. This method is also adopted by Ref.[20] for generating discrete time series with diffusion models.).

We present the discriminative and predictive scores obtained using the aforementioned methods on the MIMIC-III/IV and the eICU datasets in Table 4. We notice that TimeDiff using multinomial diffusion obtains lower discriminative and predictive scores across all 3 datasets.

Discussion

The synthetic samples generated by TimeDiff exhibit remarkable overlap with real training and testing data (see Figure 2), indicating that the generated samples preserve similar data distribution to real data. Note that we obtain the same

Downloaded from https://academic.oup.com/jamia/advance-article/doi/10.1093/jamia/ocae229/7747780 by Duke Clinical Research Institute, Bernie Chen on 01 October 2024

Table 4. Ablation study on generating missing indicators using multinomial diffusion.

Metric	Method	MIMIC-III	MIMIC-IV	eICU
Discriminative score (↓)	with Gaussian and rounding	0.355 ± 0.020	0.121 ± 0.025	0.030 ± 0.018
	with Gaussian and softmax	0.088 ± 0.023	0.155 ± 0.032	0.042 ± 0.045
	with multinomial	**0.028 ± 0.023**	**0.030 ± 0.022**	**0.015 ± 0.007**
Predictive score (↓)	with Gaussian and rounding	0.486 ± 0.005	0.433 ± 0.003	0.312 ± 0.031
	with Gaussian and softmax	0.472 ± 0.004	0.434 ± 0.002	0.320 ± 0.035
	with multinomial	**0.469 ± 0.003**	**0.432 ± 0.002**	**0.309 ± 0.019**

Bolded values are the best-performing method.

observation across all datasets, with the rest of the visualizations presented in Supplementary Material SB.2.

The discriminative and predictive scores in Table 1 suggest that the synthetic time series generated by TimeDiff has the closest data distribution to real data compared to samples generated by other baseline methods. We notice that the DSPD/CSPD baseline method from a recent work does not yield good performance on EHR datasets. This observation can be attributed to its temporal modeling, which treats time series as discrete realizations of an underlying continuous process. This continuity assumption may not hold for EHR time series data, which are highly discontinuous.

Additionally, Table 4 from the ablation study indicates that the synthesized EHR time series is more realistic when using multinomial diffusion in TimeDiff. Evaluation using TSTR and TSRTR metrics is also performed to compare TimeDiff with the "with Gaussian and softmax" alternative, where we observe that TimeDiff outperforms the alternative. The result is presented in Supplementary Material SB.4.2, Figures S26 and S35.

We observe from Figure 4 that models trained using synthetic time series yield similar AUC scores compared to those trained on the real data, indicating that the synthetic EHR time series obtained from TimeDiff maintains high data utility for performing downstream tasks. Additionally, we notice that most of the ML models yield increasing AUC scores with the increase in the number of synthetic samples added to model training. This observation is consistent with our previous findings, indicating the high utility of our synthetic data.

The close to 0.5 scores of AA_{test} and AA_{train} computed from TimeDiff shown in Table 2 suggest that TimeDiff generates high-fidelity synthetic time series and does not overfit its training data. By contrast, although most of the baseline methods have low NNAA and MIR scores, they all have higher AA_{test} and AA_{train} scores, which implies that there may be overfitting on the training data for baseline methods.

Lastly, to assess the effects of EHR time series generation, most features selected in our study are frequent measurements such as vital signs. This design choice enables us to evaluate the ability of TimeDiff to generate sequential measurements without interference from measurement frequencies. Thus, our study does not focus on infrequent time series measurements or static features. Nevertheless, we acknowledge that this is a limitation in our study and have conducted additional experiments on the ability of TimeDiff to generate static and infrequent measurements. The results can be found in Supplementary Material SB.7.

Conclusion

We propose TimeDiff for synthetic EHR time series generation by using mixed sequence diffusion and demonstrate its superior performance compared with all state-of-the-art time series generation methods in terms of data utility. We also demonstrate that TimeDiff can facilitate downstream analysis in healthcare while protect patient privacy. Thus, we believe TimeDiff could be a useful tool to support medical data analysis by producing realistic, synthetic, and privacy-preserving EHR data to tackle data scarcity issues in healthcare. However, it is important to acknowledge the limitations of our study. While our results suggest that TimeDiff offers some degree of patient privacy protection, it should not be seen as a replacement for official audits, which may still be necessary prior to data sharing. It is also interesting to investigate TimeDiff within established privacy frameworks, eg, differential privacy. Additionally, to provide better interpretability and explainability of TimeDiff, subgroup analysis and theoretical analysis are to be developed. While we utilized sample mean imputation for computational efficiency, more advanced missing value imputation techniques could be considered to further evaluate TimeDiff's behavior. Lastly, it would also be meaningful to investigate the modeling of highly sparse and irregular temporal data, such as lab tests and medications. We leave the above potential improvements of TimeDiff for future work.

Author contributions

Muhang Tian, Bernie Chen, and Allan Guo were primarily responsible for the design and execution of the experiments and for writing the manuscript. Shiyi Jiang and Anru R. Zhang were responsible for overseeing the project and writing the manuscript.

Supplementary material

Supplementary material is available at *Journal of the American Medical Informatics Association* online.

Funding

This work was supported by CS+ program in 2023 at the Department of Computer Science, Duke University. S.J. and A.R.Z. were also partially supported by NSF (grant number CAREER-2203741) and NIH (grant numbers R01HL169347 and R01HL168940).

Conflicts of interest

None declared.

Data availability

The EHR datasets utilized during the current study are available in the following repositories: the MIMIC repository and

Downloaded from https://academic.oup.com/jamia/advance-article/doi/10.1093/jamia/ocae229/7747780 by Duke Clinical Research Institute, Bernie Chen on 01 October 2024

the eICU Collaborative Research Database. We also consider the following non-EHR time-series dataset for comparisons. Stocks: the dataset is available online and can be accessed from the historical Google stock price on Yahoo; Energy: this dataset can be obtained from UCI machine learning repository.

Code availability

Our code is available at https://github.com/MuhangTian/TimeDiff.

References

1. Shickel B, Tighe PJ, Bihorac A, Rashidi P. Deep EHR: a survey of recent advances in deep learning techniques for electronic health record (EHR) analysis. *IEEE J Biomed Health Inform*. 2018;22(5):1589-1604.
2. Goldstein BA, Navar AM, Pencina MJ, Ioannidis JPA. Opportunities and challenges in developing risk prediction models with electronic health records data: a systematic review. *J Am Med Inform Assoc*. 2017;24(1):198-208.
3. Benitez K, Malin BA. Evaluating re-identification risks with respect to the HIPAA privacy rule. *J Am Med Inform Assoc*. 2010;17(2):169-177.
4. Janmey V, Elkin PL. Re-identification risk in HIPAA de-identified datasets: the MVA attack. *AMIA Annu Symp Proc*. 2018;2018:1329-1337.
5. Yan C, Yan Y, Wan Z, et al. A multifaceted benchmarking of synthetic electronic health record generation models. *Nat Commun*. 2022;13(1):7609.
6. Yoon J, Mizrahi M, Ghalaty NF, et al. EHR-Safe: generating high-fidelity and privacy-preserving synthetic electronic health records. *NPJ Digit Med*. 2023;6(1):141.
7. Gonzales A, Guruswamy G, Smith SR. Synthetic data in health care: a narrative review. *PLoS Digit Health*. 2023;2(1):e0000082.
8. Haendel MA, Chute CG, Bennett TD, et al.; N3C Consortium. The National COVID Cohort Collaborative (N3C): rationale, design, infrastructure, and deployment. *J Am Med Inform Assoc*. 2020;28(3):427-443.
9. Herrett E, Gallagher AM, Bhaskaran K, et al. Data resource profile: clinical practice research datalink (CPRD). *Int J Epidemiol*. 2015;44(3):827-836.
10. Gui J, Sun Z, Wen Y, Tao D, Ye J. A review on generative adversarial networks: algorithms, theory, and applications. *IEEE Trans Knowl Data Eng*. 2023;35(4):3313-3332.
11. Yi X, Walia E, Babyn PS. Generative adversarial network in medical imaging: a review. *Med Image Anal*. 2018;58:101552.
12. Choi E, Biswal S, Malin B, Duke J, Stewart WF, Sun J. Generating multi-label discrete patient records using generative adversarial networks. In *Proceedings of the 2nd Machine Learning for Healthcare Conference*. Vol. 68. PMLR; 2017:286-305.
13. Baowaly MK, Lin C-C, Liu C-L, Chen K-T. Synthesizing electronic health records using improved generative adversarial networks. *J Am Med Inform Assoc*. 2019;26(3):228-241.
14. Saxena D, Cao J. Generative adversarial networks (GANs): challenges, solutions, and future directions. *ACM Comput Surv*. 2021;54(3):1-42.
15. Ho J, Jain A, Abbeel P. Denoising diffusion probabilistic models. *Adv Neural Inf Process Syst*. 2020;33:6840-6851.
16. Nichol A, Dhariwal P. Improved denoising diffusion probabilistic models. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*. Vol. 139. PMLR; 2021.
17. Rombach R, Blattmann A, Lorenz D, Esser P, Ommer B. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society; 2022:10684-10695.
18. He H, Zhao S, Xi Y, Ho JC. 2023. MedDiff: generating electronic health records using accelerated denoising diffusion model. arXiv, arXiv:2302.04355, preprint: not peer reviewed.
19. Yuan H, Zhou S, Yu S. 2023. EHRDiff: exploring realistic EHR synthesis with diffusion models. arXiv, arXiv:2303.05656, preprint: not peer reviewed.
20. Kuo NI-H, Jorm LR, Barbieri S. 2023. Synthetic health-related longitudinal data with mixed-type variables generated using diffusion models. arXiv, arXiv:abs/2303.12281, preprint: not peer reviewed.
21. Mogren O. 2016. C-RNN-GAN: continuous recurrent neural networks with adversarial training. arXiv, arXiv:1611.09904 [cs.AI], preprint: not peer reviewed.
22. Esteban C, Hyland SL, Rätsch G. 2017. Real-valued (medical) time series generation with recurrent conditional GANs. arXiv, arXiv:1706.02633 [stat.ML], preprint: not peer reviewed.
23. Dai AM, Le QV. 2015. Semi-supervised sequence learning. arXiv, arXiv:1511.01432 [cs.LG], preprint: not peer reviewed.
24. Lyu X, Hueser M, Hyland SL, Zerveas G, Raetsch G. 2018. Improving clinical predictions through unsupervised time series representation learning. arXiv, arXiv:1812.00490 [cs.LG], preprint: not peer reviewed.
25. Srivastava N, Mansimov E, Salakhutdinov R. 2016. Unsupervised learning of video representations using LSTMs. arXiv, arXiv:1502.04681 [cs.LG], preprint: not peer reviewed.
26. Yoon J, Jarrett D, van der Schaar M. Time-series generative adversarial networks. *Adv Neural Inf Process Syst*. 2019;32
27. Jeon J, Kim J, Song H, Cho S, Park N. GT-GAN: general purpose time series synthesis with generative adversarial networks. *Adv Neural Inf Process Syst*. 2022;35:36999-37010.
28. Kidger P, Morrill J, Foster J, Lyons T. Neural controlled differential equations for irregular time series. *Adv Neural Inf Process Syst*. 2020;33:6696-6707.
29. De Brouwer E, Simm J, Arany A, Moreau Y. GRU-ODE-Bayes: continuous modeling of sporadically-observed time series. *Adv Neural Inf Process Syst*. 2019;32
30. Deng R, Chang B, Brubaker MA, Mori G, Lehrmann A. 2021. Modeling continuous stochastic processes with dynamic normalizing flows. arXiv, arXiv:2002.10516 [cs.LG], preprint: not peer reviewed.
31. Biloš M, Rasul K, Schneider A, Nevmyvaka Y, Günnemann S, et al. Modeling temporal data as continuous functions with stochastic process diffusion. In: Krause A, ed. *Proceedings of the 40th International Conference on Machine Learning*. Vol. 202. PMLR; 2023:2452-2470. https://proceedings.mlr.press/v202/bilos23a.html
32. Sohl-Dickstein J, Weiss E, Maheswaranathan N, Ganguli S. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*. PMLR; 2015:2256-2265.
33. Gu S, Chen D, Bao J, et al. Vector quantized diffusion model for text-to-image synthesis. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society; 2022:10686-10696.
34. Dhariwal P, Nichol A. Diffusion models beat GANs on image synthesis. *Adv Neural Inf Process Syst*. 2021:8780-8794.
35. Saharia C, Chan W, Saxena S, et al. Photorealistic text-to-image diffusion models with deep language understanding. In *36th Conference on Neural Information Processing Systems*. 2022.
36. Kotelnikov A, Baranchuk D, Rubachev I, Babenko A. 2022. TabDDPM: modelling tabular data with diffusion models. arXiv, arXiv:abs/2209.15421, preprint: not peer reviewed.
37. Das A, Yang Y, Hospedales TM, Xiang T, Song Y-Z. 2023. ChiroDiff: modelling chirographic data with diffusion models. arXiv, arXiv:abs/2304.03785, preprint: not peer reviewed.
38. Song Y, Ermon S. Generative modeling by estimating gradients of the data distribution. *Adv Neural Inf Process Syst*. 2019;32
39. Song Y, Ermon S. Improved techniques for training score-based generative models. *Adv Neural Inf Process Syst*. 2020;33:12438-12448.

Downloaded from https://academic.oup.com/jamia/advance-article/doi/10.1093/jamia/ocae229/7747780 by Duke Clinical Research Institute, Bernie Chen on 01 October 2024

40. Chen S, Chewi S, Li J, et al. Sampling is as easy as learning the score: theory for diffusion models with minimal data assumptions. In *The Eleventh International Conference on Learning Representations*. 2022.
41. Yan C, Zhang Z, Nyemba S, Malin BA. Generating electronic health records with multiple data types and constraints. *AMIA Annu Symp Proc*. 2020;2020:1335-1344.
42. Biswal S, Ghosh S, Duke J, et al. EVA: generating longitudinal electronic health records using conditional variational autoencoders. In *Proceedings of the 6th Machine Learning for Healthcare Conference*. Vol. 149. PMLR; 2021:260-282.
43. Naseer AA, Walker B, Landon C, et al. ScoEHR: generating synthetic electronic health records using continuous-time diffusion models. In *Machine Learning for Healthcare Conference*. 2023.
44. Ceritli T, Ghosheh GO, Chauhan VK, Zhu T, Creagh AP, Clifton DA. 2023. Synthesizing mixed-type electronic health records using diffusion models. arXiv, arXiv:2302.14679, preprint: not peer reviewed.
45. Koo H, Kim TE. 2023. A comprehensive survey on generative diffusion models for structured data. arXiv, arXiv:abs/2306.04139v2, preprint: not peer reviewed.
46. Li J, Cairns BJ, Li J, Zhu T. Generating synthetic mixed-type longitudinal electronic health records for artificial intelligent applications. *NPJ Digit Med*. 2023;6(1):98.
47. Theodorou B, Xiao C, Sun J. Synthesize high-dimensional longitudinal electronic health records via hierarchical autoregressive language model. *Nat Commun*. 2023;14(1):5305.
48. Ronneberger O, Fischer P, Brox T. U-net: convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015: 18th International Conference*, Munich, Germany, October 5-9, 2015, Part III 18. 2015:234-241.
49. He H, Xi Y, Chen Y, Malin B, Ho J, et al. A flexible generative model for heterogeneous tabular EHR with missing modality. In *The Twelfth International Conference on Learning Representations*. 2024.
50. Johnson AEW, Pollard TJ, Shen L, et al. MIMIC-III, a freely accessible critical care database. *Sci Data*. 2016;3(1):160035-160039.
51. Johnson AEW, Bulgarelli L, Shen L, et al. MIMIC-IV, a freely accessible electronic health record dataset. *Sci Data*. 2023;10(1):1.
52. Pollard TJ, Johnson AEW, Raffa JD, et al. The eICU Collaborative Research Database, a freely available multi-center database for critical care research. *Sci Data*. 2018;5(1):1-13.
53. Van der Maaten L, Hinton G. Visualizing data using t-SNE. *J Mach Learn Res*. 2008;9:2579-2605.
54. McInnes L, Healy J. 2018. UMAP: uniform manifold approximation and projection for dimension reduction. arXiv, arXiv:abs/1802.03426, preprint: not peer reviewed.
55. Yale A, Dash S, Dutta R, et al. Generation and evaluation of privacy preserving synthetic health data. *Neurocomputing*. 2020;416:244-255.
56. Liu G, Wang C, Peng K, et al. SocInf: membership inference attacks on social media health data with machine learning. *IEEE Trans Comput Soc Syst*. 2019;6(5):907-921.
57. Graves A. 2013. Generating sequences with recurrent neural networks. arXiv, arXiv:1308.0850, preprint: not peer reviewed. [Database]
58. Sutskever I, Martens J, Hinton GE. Generating text with recurrent neural networks. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*. PMLR; 2011:1017-1024.
59. Goyal A, Lamb A, Zhang Y, Zhang S, Courville A, Bengio Y. Professor forcing: a new algorithm for training recurrent networks. *Adv Neural Inf Process Syst*. 2016;29
60. Hoogeboom E, Nielsen D, Jaini P, Forré P, Welling M. Argmax flows and multinomial diffusion: learning categorical distributions. In: Ranzato M, Beygelzimer A, Dauphin Y, et al., eds. Advances in Neural Information Processing Systems. Vol. 34. Curran Associates, Inc.; 2021:12454-12465. https://proceedings.neurips.cc/paperfiles/paper/2021/file/67d96d458abdef21792e6d8e590244e7-Paper.pdf
61. Zhou Y, Shi J, Stein R, et al. Missing data matter: an empirical evaluation of the impacts of missing EHR data in comparative effectiveness research. *J Am Med Inform Assoc*. 2023;30(7):1246-1256.
62. Little RJ, Rubin DB. *Statistical Analysis with Missing Data*. John Wiley & Sons; 2019.
63. Enders CK. *Applied Missing Data Analysis*. Guilford Publications; 2022.
64. Song Y, Sohl-Dickstein J, Kingma DP, Kumar A, Ermon S, Poole B. 2020. Score-based generative modeling through stochastic differential equations. arXiv, arXiv:2011.13456, preprint: not peer reviewed.
65. Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need. *Adv Neural Inf Process Syst*. 2017;30
66. Hendrycks D, Gimpel K. 2016. Gaussian error linear units (GELUs). arXiv, arXiv:1606.08415, preprint: not peer reviewed.
67. Ba JL, Kiros JR, Hinton GE. 2016. Layer normalization. arXiv, arXiv:1607.06450, preprint: not peer reviewed.
68. Sadeghi R, Banerjee T, Romine W. Early hospital mortality prediction using vital signals. *Smart Health (Amst)*. 2018;9-10:265-274.
69. Sheikhalishahi S, Balaraman V, Osmani V. 2019. Benchmarking machine learning models on eICU critical care dataset. arXiv, arXiv:1910.00964, preprint: not peer reviewed.
70. Chen T, Guestrin C. XGBoost: a scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM; 2016:785-794.
71. Breiman L. Random forests. *Mach Learn*. 2001;45(1):5-32.
72. Freund Y, Schapire RE. A decision-theoretic generalization of on-line learning and an application to boosting. *J Comput Syst Sci*. 1997;55(1):119-139.
73. Friedman J, Hastie T, Tibshirani R. Regularization paths for generalized linear models via coordinate descent. *J Stat Softw*. 2010;33(1):1-22.

© The Author(s) 2024. Published by Oxford University Press on behalf of the American Medical Informatics Association.
All rights reserved. For permissions, please email: journals.permissions@oup.com
Journal of the American Medical Informatics Association, 2024, 00, 1–11
https://doi.org/10.1093/jamia/ocae229
Research and Applications

Downloaded from https://academic.oup.com/jamia/advance-article/doi/10.1093/jamia/ocae229/7747780 by Duke Clinical Research Institute, Bernie Chen on 01 October 2024