IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 17 May 2023, accepted 7 July 2023, date of publication 18 July 2023, date of current version 26 July 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3296542

RESEARCH ARTICLE

Synthetic ECG Signal Generation Using Probabilistic Diffusion Models

EDMONMD ADIB[1], (Student Member, IEEE), AMANDA S. FERNANDEZ[2], (Senior Member, IEEE), FATEMEH AFGHAH[3], (Senior Member, IEEE), AND JOHN J. PREVOST[1], (Senior Member, IEEE)

[1]Department of Electrical and Computer Engineering, The University of Texas at San Antonio (UTSA), San Antonio, TX 78249, USA
[2]Department of Computer Science, The University of Texas at San Antonio (UTSA), San Antonio, TX 78249, USA
[3]Department of Electrical and Computer Engineering, Clemson University, Clemson, SC 29634, USA

Corresponding author: Edmonmd Adib (edmond.adib@utsa.edu)

This research was supported by the Open Cloud Institute, UTSA.

ABSTRACT Deep learning image processing models have had remarkable success in recent years in generating high quality images. Particularly, the Improved Denoising Diffusion Probabilistic Models (DDPM) have shown superiority in image quality to the state-of-the-art generative models, which motivated us to investigate their capability in the generation of the synthetic electrocardiogram (ECG) signals. In this work, synthetic ECG signals are generated by the Improved DDPM and by the Wasserstein GAN with Gradient Penalty (WGAN-GP) models and then compared. To this end, we devise a pipeline to utilize DDPM in its original 2*D* form. First, the 1*D* ECG time series data are embedded into the 2*D* space, for which we employed the Gramian Angular Summation/Difference Fields (GASF/GADF) as well as Markov Transition Fields (MTF) to generate three 2*D* matrices from each ECG time series, which when put together, form a 3-channel 2*D* datum. Then 2*D* DDPM is used to generate 2*D* 3-channel synthetic ECG images. The 1*D* ECG signals are created by de-embedding the 2*D* generated image files back into the 1*D* space. This work focuses on unconditional models and the generation of *Normal Sinus Beat* ECG signals exclusively, where the Normal Sinus Beat class from the MIT-BIH Arrhythmia dataset is used in the training phase. The *quality*, *distribution*, and the *authenticity* of the generated ECG signals by each model are quantitatively evaluated and compared. Our results show that in the proposed pipeline and in the particular setting of this paper, the WGAN-GP model is consistently superior to DDPM in all the considered metrics.

INDEX TERMS Generative adversarial networks, wasserstein GAN, synthetic ECG generation, probabilistic diffusion model, improved denoising diffusion probabilistic models (DDPM).

I. INTRODUCTION

Electrocardiogram (ECG) is the manifestation of the heart's rhythm and electrical activity. ECG monitoring is a painless, fast, non-invasive, and cheap procedure which can reveal much regarding the heart's health. Automated ECG-based diagnosis is becoming increasingly more popular as it eliminates randomized human error and can be readily available at a patient's bedside using affordable wearable heart monitoring devices. Automatic ECG diagnosis models are usually deep learning models that classify patients' ECG signals according to the morphological pattern of the ECG [1]. The datasets used for training these models are usually highly imbalanced, as normal beats are much more abundant than abnormal cases. Realistic synthetic[1] ECG signals can augment real datasets, enriching them and enabling better class balance. Additionally, ECG data are considered

The associate editor coordinating the review of this manuscript and approving it for publication was Mohammad Zia Ur Rahman.

[1]In this context *synthetic beats* and *generated beats* are the outputs of the generative models and used interchangeably.

private information, therefore their usage is highly regulated, whereas synthetic ECG signals can be used without any restriction.

In general, there are several approaches to addressing imbalances in datasets. One approach is to use oversampling methods such as SMOTE [2] to generate additional synthetic data. Other approaches to address the inadequacy of samples are to utilize new designs of loss functions such as focal loss [3] or using a new training scheme, e.g., few-shot training [4]. However, using deep generative algorithms such as Generative Adversarial Networks (GAN) [5] and Variational Auto-Encoders (VAE) [6] is becoming more popular (e.g. [7] and [8]).

Computer vision deep learning models have been outstandingly successful in *classification* as well as in *generation* tasks. Most prominent models (AlexNet, VGG-16, ResNet-18, ...) are pre-trained and readily available on the internet. Using transfer learning techniques, they need only a minor fine tuning to produce the best-quality results, saving days, weeks, or even months of training time. Additionally, using 2*D* data space provides more augmentation techniques (such as flipping, rotation, and mirroring), which are very beneficial particularly in the classification tasks. Moreover, better classification performances compared to 1*D* space implementations have been reported [9]. Improved Denoising Diffusion Probabilistic Models (DDPM) [10] have proved to generate images with quality superior to GAN models [11]. In this study, we present a pipeline to generate 1*D* synthetic ECG signals using 2*D* DDPM. We also investigate the quality of the generated beats by DDPM and compare it with GAN models. To the best of our knowledge, this is the first time that diffusion models are used for the generation of synthetic ECG signals. The 1*D* ECG time series data are embedded into 3-channel 2*D* data (similar to RGB image files) using Gramian Angular Summation/Difference Fields (GASF/GADF) and Markov Transition Fields (MTF) [12] and then fed to the Improved DDPM as image files. DDPM is trained on the embedded data and then sampled to generate embedded 2*D* ECG data which are then de-embedded and transformed back to 1*D* ECG time series using the inverse transformations (Figure. 1). Three different settings of DDPM hyper-parameters have been considered as three study cases and the fourth study case is the data generated by the WGAN-GP model. The four study cases are compared in terms of the *quality*, *distribution*, and *authenticity* of the generated beats (such as whether they can replace the real data in a classification test). Precision score, Area Under Curve (AUC) of Precision-Recall curves, and also that of the Receiver Operating Characteristic curves (ROC AUC) have been used as the metrics of classification. MIT-BIH Arrhythmia dataset [13], [14] is used as the dataset. Since this research is the first use of DDPM in ECG generation, we employed DDPM in the unconditional form and focused only on the *Normal Sinus* class to investigate the feasibility of the idea. However, it can be also used in the conditional form for generating synthetic arrhythmia signals in various classes.

A. RELATED WORKS

Much research has been done on the generation of ECG in one-dimensional space. Alcaraz and Strodthoff [15] used a combination of structured state space model and diffusion models to generate 10-second 12-lead synthetic ECGs. They generated synthetic digital twins of PTB-XL ECG dataset. Wang et al. [16] augmented their imbalanced dataset using synthetically generated beats by a 1*D* auxiliary classier generative adversarial network (AC-GAN). Delaney et al. [17] studied the generation of realistic synthetic signals using a range of architectures from the GAN family. Esteban et al. [18] employed two-layer BiLSTM architecture in the generator and the discriminator to generate synthetic ECG signals. Adib et al. [19] compared several GAN models (mono-class) in one study, and in another study they compared conditional and unconditional GAN-GP models (multiclass) in the generation of synthetic ECG beats [20].

There are also research studies in which the 1*D* ECG time series are embedded into 2*D* space. Ahmad et al. [21] developed a novel image fusion model (by AlexNet architecture) for the classification task of ECG beats. They convert 1*D* beats into a 3-channel 2*D* image using Gramian Angular Field (GAF), Recurrence Plot (RP), and Markov Transition Field (MTF). Cai et al. [22] also created a 3-channel embedding using Gramian angular field (GAF), recurrent plot (RP), and tiling for their classification task. Diker et al. [23] did 2*D* embedding of heartbeats using spectrograms to classify of them by AlexNet, VGG-16, and ResNet-18. Izci et al. [24] mimicked LeNet CNN architecture for the classification of 2*D* grayscale images (plots) of beats without any embedding. Hao et al. [25] converted one-dimensional ECG signal into spectro-temporal images and used multiple dense convolutional neural networks to capture both beat-to-beat and single-beat information for analysis. Hao et al. [25] used a time-frequency representation of ECG beats (wavelet transform and short-term Fourier transform) for classification by a dense CNN model. Huang et al. [26] used a short-time Fourier transform to embed heartbeats into 2*D* spectrograms for their classification by a 2*D*-CNN model. Oliviera and Nobrega [27] used wavelet transform for 2*D* embedding of beats. Salem et al. [28] employed spectrograms to classify ECG data as well. Mathunjwa et al. [29] and [30] as well as [31] used recurrent plots for classification of 2*D* ECG data. To the best of our knowledge, currently there is no study on using 2*D* models to *generate* synthetic ECG signals.

II. METHODOLOGY

A. DATASET AND SEGMENTATION

The MIT-BIH Arrhythmia dataset [13], [14] is one of the benchmark datasets of ECG analysis. It is a set of 48 Holter recordings, each 30 minutes long and with two channels, that were obtained by the Beth Israel Hospital Arrhythmia Laboratory between 1975 and 1979. The upper channel is consistently the modified limb lead II (MLII) and the lower

one is mostly a modified lead V1 (occasionally V2 or V5, and in one instance V4). The digitization of the analog signals has been done at 360 Hz. In this study, we used the upper channel as it is only from one electrode (MLII).

The *Adaptive Window* method is used for the segmentation of the ECG signals into individual beats. This method adapts to changes in the heart rate. The R-peaks of the QRS complex in heartbeats are annotated in the dataset. Using this information, the distances to the next and previous R-peaks are determined for each individual beat and then 75% of each is used as the *cutoff* to find the boundaries of the individual beat. Then, all the segmented individual beats are resampled to 256. The resulting dataset is comprised of 109, 338 individual beats in 15 classes and is highly imbalanced.

B. DENOISED DIFFUSION PROBABILISTIC MODELS (DDPM)

DDPM, similar to the VAE model, is a variational-based model [10] where the objective is to find the distribution of the dataset explicitly [32]. There are two processes in the DDPM: forward process and backward process. In the forward process, q, noise is added to a datapoint $x_0 \sim q(x_0)$ gradually and in steps. The generated noisy samples (latent variables) are x_1 through x_T. The noise which is added at each time step t is a Gaussian noise with a variance with a specific schedule β_t [33]:

$$q(x_t|x_{t-1}) = \mathcal{N}\left(x_t; \sqrt{1-\beta_t}x_t, \beta_t \boldsymbol{I}\right) \tag{1}$$

$q(x_t|x_0)$ can be expressed in a closed form by defining $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{s=0}^{t} \alpha_s$:

$$\begin{aligned} q(x_t|x_0) &= \mathcal{N}\left(x_t; \sqrt{\bar{\alpha}_t}\, x_0, (1-\bar{\alpha}_t)\, \boldsymbol{I}\right) \\ &= \sqrt{\bar{\alpha}_t}\, x_0 + \epsilon \sqrt{1-\bar{\alpha}_t}, \quad \epsilon \sim \mathcal{N}(0, \boldsymbol{I}) \end{aligned} \tag{2}$$

where, $1 - \bar{\alpha}_t$ is the variance of the noise for any arbitrary step and can be used directly instead of β_t. Using Bayes theorem, it can be proved that the posterior of the reverse process $q(x_{t-1}|x_t, x_0)$ is also a Gaussian distribution [33]:

$$q(x_{t-1}|x_t, x_0) = \mathcal{N}\left(x_{t-1}; \tilde{\mu}_t(x_t, x_0), \tilde{\beta}_t \boldsymbol{I}\right) \tag{3}$$

with:

$$\begin{cases} \tilde{\mu}_t(x_t, x_0) &= \dfrac{\sqrt{\bar{\alpha}_{t-1}}\,\beta_t}{1-\bar{\alpha}} x_0 + \dfrac{\sqrt{\alpha_t}\,(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t} \\ \tilde{\beta}_t &= \dfrac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t \end{cases} \tag{4}$$

To sample from $q(x_0)$, knowing the denoising process $q(x_{t-1}|x_t)$, we can start from $q(x_T)$ and then sample the reverse steps $q(x_{t-1}|x_t)$ till we reach x_0. Under certain assumptions on β_t and T, sampling x_T is trivial, so pure noise is usually used [33]. However, since $q(x_{t-1}|x_t)$ is *intractable*, a neural network is trained to approximate it. Since as $T \to \infty$, $\beta_t \to 0$ and $q(x_{t-1}|x_t)$ approaches a diagonal Gaussian distribution, it would be sufficient to train a neural network to predict the mean μ_θ and the diagonal covariance Σ_θ:

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)) \tag{5}$$

The variational lower-bound loss function of L_{vlb} for $p_\theta(x_0)$ is:

$$L_{vlb} = L_0 + L_1 + \cdots + L_{T-1} + L_T \tag{6}$$

$$L_0 = -\log p_\theta(x_0|x_1) \tag{7}$$

$$L_{t-1} = D_{KL}(q(x_{t-1}|x_t, x_0) \| p_\theta(x_{t-1}|x_t)) \tag{8}$$

$$L_T = D_{KL}(q(x_T|x_0) \| p(x_T)) \tag{9}$$

Ho et al. [33] used a simplified loss function in which a neural network $\epsilon_\theta(x_t, t)$ is trained to predict ϵ from Eq. 2:

$$L_{simple} = E_{t\sim[1,T], x_0\sim q(x_0), \epsilon\sim\mathcal{N}(0,\boldsymbol{I})}\left[\|\epsilon - \epsilon_\theta(x_t, t)\|^2\right] \tag{10}$$

Then $\mu_\theta(x_t, t)$ can be derived from $\epsilon_\theta(x_t, t)$:

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t)\right) \tag{11}$$

L_{simple} does not provide any signal for training $\Sigma_\theta(x_t, t)$. So, instead of learning $\Sigma_\theta(x_t, t)$, it is fixed to a constant, i.e.,: $\Sigma_\theta(x_t, t) = \sigma_t^2 \boldsymbol{I}$ where $\sigma_t^2 = \beta_t$ or $\sigma_t^2 = \tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$ which are the upper and lower bounds for the true reverse process' step variance, respectively.

C. IMPROVED DDPM

Log-likelihood is a metric in generative modeling and is an indication of the coverage of the existing modes in the dataset by the model [34]. Also, it has been shown that it has a great impact on the quality of samples and learned feature representations [35]. Nichol and Dhariwal [10] made a few modifications on the DDPM developed by Ho et al. [33] to achieve better Log-likelihood, which are introduced below very briefly.

1) LEARNED SIGMA

$\Sigma_\theta(x_t, t)$ is the variance of the reverse process. Nichol et al. [10] argue that by increasing the number of the diffusion steps, the role of the mean $\mu_\theta(x_t, t)$ becomes more dominant than $\Sigma_\theta(x_t, t)$ in determining the *distribution* of the data. They also argue that not fixing, rather learning the $\Sigma_\theta(x_t, t)$ would provide a better choice and the Log-likelihood is improved. They suggested an interpolation between β and $\tilde{\beta}$ (the two extremes) for the parameterization of the variance. In fact, their model outputs a vector v which interpolates between the two fixed extreme values (lower and higher bounds) and has the same dimension as the data:

$$\Sigma_\theta(x_t, t) = \exp\left(v \log \beta_t + (1-v)\log \tilde{\beta}_t\right) \tag{12}$$

Since L_{simple} (Eq. 10) is independent of $\Sigma_\theta(x_t, t)$, they suggested a new hybrid objective to learn $\Sigma_\theta(x_t, t)$:

$$L_{hybrid} = L_{simple} + \lambda L_{vlb} \tag{13}$$

in which λ is set to a low value of 0.001 to prevent L_{vlb} from overwhelming L_{simple}.

2) NOISE SCHEDULE

Linear noising schedule is used in the original DDPM by Ho et al. [33], which Nichol and Dhariwal [10] found sub-optimal for lower resolution image processing (i.e. 64×64 or 32×32). Instead, they proposed a *cosine* schedule, which retain information in the noisy images longer in the noising process steps, as opposed to the strong linear schedule in which the noisy images become pure noise much earlier and destroys information more quickly in the noising process (Figures 3 and 5 [10]):

$$\bar{\alpha}_t = \frac{f(t)}{f(0)}, \quad f(t) = \cos\left(\frac{t/T + s}{1+s} \cdot \frac{\pi}{2}\right)^2 \tag{14}$$

with s being an offset parameter set at $s = 0.008$.

3) IMPORTANCE-SAMPLED L_{vlb}

Nichol and Dhariwal [10] found that, contrary to their expectation, optimizing L_{hybrid} achieves better log-likelihood rather than optimizing L_{vlb} directly, which they believe was caused by L_{vlb} and its gradient being more *noisy*. So, in order to reduce the variance of L_{vlb}, they employed *importance sampling* rather than sampling uniformly throughout the samples:

$$L_{vlb} = \mathbb{E}_{t\sim p_t}\left[\frac{L_t}{p_t}\right], \quad \text{where } p_t \propto \sqrt{\mathbb{E}\left[L_t^2\right]} \text{ and } \sum p_t = 1 \tag{15}$$

in which they kept a 10-step history for the evaluation of $\mathbb{E}\left[L_t^2\right]$ which is updated dynamically. Using importance sampling, they could achieve their best log-likelihoods with considerably less noisy objective than the uniformly sampled objective. However, the importance sampling technique does not make any improvement on the less noisy L_{hybrid} objective [10].

D. PIPELINE

The general pipeline used in this study for the generation of synthetic ECG signals is shown in Fig. 1. First, $1D$ real ECG beats (exclusively in class N) are embedded into the $2D$ space and then 3-channel $2D$ image files (similar to RGB image files) are formed. Then, the Improved DDPM [10] is trained and then sampled to generate $2D$ 3-channel synthetic ECG image files. Finally, the $1D$ time series are reconstructed by de-embedding the generated data back into $1D$ space. The stages of the pipeline are discussed in more detail in the following sections.

E. 1D-2D EMBEDDING

Wang and Oates [12] proposed a novel embedding framework for mapping time series data from one-dimensional space into two-dimensional space, which enabled the utilization of computer vision techniques *as-is* for time series analysis. First, they map the ECG time series from Cartesian to polar coordinates. Then, they use *Gramian Angular Summation/Difference Fields* (GASF/GADF) and *Markov Transition Fields* (MTF) to build 3 separate $2D$ matrix embeddings of the time series and finally put them together to create a 3-channel $2D$ image file similar to an RGB image file. The proposed pipeline in this study (Fig. 1) has been used only for the DDPM, whereas in the WGAN-GP model, all the data (training and generated) and the model itself are in the $1D$ space and no embedding was necessary.

1) POLAR COORDINATES REPRESENTATION

ECG time series are one-dimensional, vector-like data, $X = \{x_1, x_2, \ldots, x_N\}$, which represent the time-progression of the induced voltage to the electrode caused by the motion of the electrical impulse generated by the sinoatrial (SA) node in the heart. When normalized and rescaled, $(\tilde{X})$, all the timestep values in $\tilde{X} = \{\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_N\}$ are between -1 and 1, i.e., $\tilde{x}_i \in [-1, 1]$ for $i = 0, \ldots, N$. Thus, each value can be interpreted as the cosine of an imaginary angle $\varphi_i \in [0, \pi]$ [12]:

$$\tilde{x}_i = \cos(\varphi_i) \quad \text{and} \quad \varphi_i = \arccos(\tilde{x}_i) \tag{16}$$

Thus, the polar coordinates of mapped data will be (Fig. 2):

$$\begin{cases} \varphi_i = \arccos(\tilde{x}_i) & -1 \le \tilde{x}_i \le 1 \\ r_i = \dfrac{i}{N} & i \in [1, N] \end{cases} \tag{17}$$

This mapping is *bijective* (one-to-one correspondence) as $\cos(\varphi)$ is monotonic when $\varphi \in [0, \pi]$. Therefore, the forward as well as reverse mappings are unique. Also, the temporal relations of the timesteps are preserved in the mapping.

2) GRAMIAN ANGULAR FIELDS (GASF/GADF)

After transformation into polar coordinates, the *Gramian Summation Angular Field* (GASF) and the *Gramian Difference Angular Field* (GADF) matrices are defined [12]:

$$GASF = \begin{bmatrix} \cos(\frac{\varphi_1}{2} + \frac{\varphi_1}{2}) & \ldots & \cos(\frac{\varphi_1}{2} + \frac{\varphi_N}{2}) \\ \vdots & \ddots & \vdots \\ \cos(\frac{\varphi_N}{2} + \frac{\varphi_1}{2}) & \ldots & \cos(\frac{\varphi_N}{2} + \frac{\varphi_N}{2}) \end{bmatrix} \tag{18}$$

$$GADF = \begin{bmatrix} \cos(\frac{\varphi_1}{2} - \frac{\varphi_1}{2}) & \ldots & \cos(\frac{\varphi_1}{2} - \frac{\varphi_N}{2}) \\ \vdots & \ddots & \vdots \\ \cos(\frac{\varphi_N}{2} - \frac{\varphi_1}{2}) & \ldots & \cos(\frac{\varphi_N}{2} - \frac{\varphi_N}{2}) \end{bmatrix} \tag{19}$$

GASF and GADF are in fact quasi-Gramian matrices because the defined *cos()* functions do not satisfy the linear property in the inner-product space, however, they do preserve the temporal dependency of the timesteps in the time series [12]. Additionally, the main diagonal of GASF can be used directly in the de-embedding to reconstruct the original time series in the Cartesian coordinates by Eq. 16, as *cos()* is monotonic when $\varphi_i \in [0, \pi]$. In GASF/GADF embedding, half-angles are used because with the full-angles, the

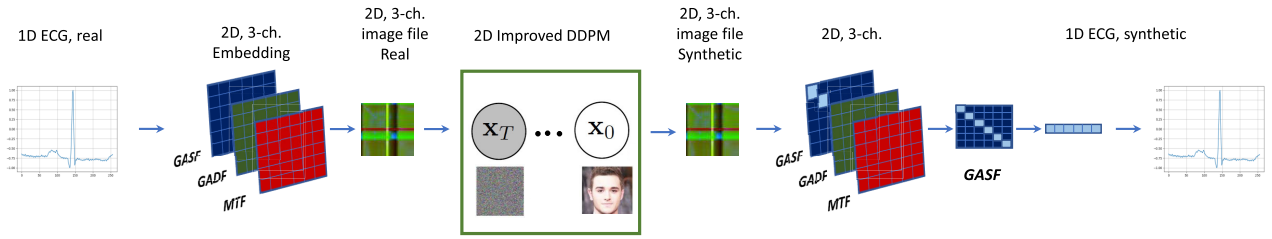


FIGURE 1. Block diagram of the proposed model. First, the normalized ECG time series are transformed into 2*D* space using GASF, GADF, and MTF separately, and 3-channel RGB-like data are generated. Then, the Improved DDPM model is trained and sampled to generate 2*D* ECG data. Finally, the 1*D* ECG time series are reconstructed using the diagonals of the GASF channel.

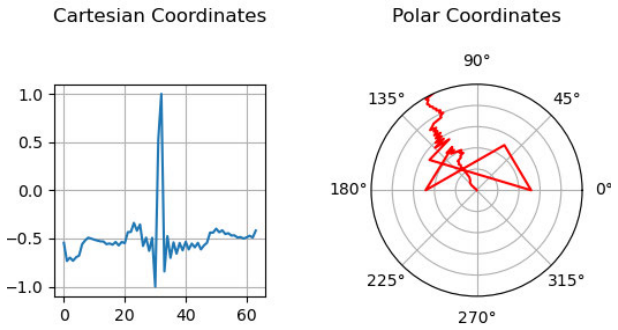


FIGURE 2. *Normal* ECG beat in cartesian and polar coordinates.

mapping in the de-embedding will not be unique ($\cos(2\theta) = 2\cos^2(\theta) - 1 \quad \longrightarrow \quad \cos(\theta) = \pm\frac{1}{2}\sqrt{\cos(2\theta) + 1}$).

3) MARKOV TRANSITION FIELDS (MTF)

GASF and GADF capture the *static* information in the time series elements without any notion on the *dynamic* information, i.e., how the values in timesteps change in time progression. In contrast, MTF captures the *dynamic* information by setting up some Q quantile bins and assigning each x_i to its corresponding bin q_i where $i \in [1, Q]$. For any pair of x_i and x_j, with the corresponding bins q_i and q_j, M_{ij} in *Matrix Transition Field* (MTF) denotes the probability of transitioning from q_i to q_j. Thus, the MTF matrix takes into account the *temporal positions* as well as *temporal changes* of the timesteps in the time series. The main diagonal in the MTF matrix represents the probability of transitioning from the quantile at timestep i to itself (self-transition probability) [12]:

$$GTF = \begin{bmatrix} M_{11} & \dots & M_{1N} \\ \vdots & \ddots & \vdots \\ M_{N1} & \dots & M_{NN} \end{bmatrix} \tag{20}$$

F. DE-EMBEDDING

After the diffusion model is trained on the 2*D* embedded data, the trained model is sampled to generate synthetic 2*D* 3-channel ECG data. Since the elements on the main diagonal of the GASF channel of the generated data consist only of the univariate data, $\bar{X} = \{\cos(\bar{\varphi}_1), \dots, \cos(\bar{\varphi}_N)\}$, we can de-embed the generated 2*D* data back into 1*D* space (i.e., reconstruct the time series) very easily by using Eq. 16, given that the mapping is bijective.

G. PRECISION OR RECALL

In this study we intend to maximize the number of true positives (TP) and at the same time minimize the number of false positives (FP). Thus, $\frac{FP}{TP}$ is to be minimized. Therefore, Precision score ($Pr = \frac{TP}{TP+FP} = \frac{1}{1+FP/TP}$) relates to the purpose of this study more than the Recall score ($Re = \frac{TP}{TP+FN} = \frac{1}{1+FN/TP}$), as in the abundance of the generated data, it is immaterial if some of the *good* samples are overlooked (false negatives).

H. PRECISION - RECALL PLOT AREA UNDER CURVE

One of the metrics used to evaluate the performance of binary classification is the area under the Precision-Recall plot. This plot captures the trade-off between the two scores at different (rather than at one single) thresholds. First, in the classification of the whole set of the test data, $Pr = \frac{TP}{TP+FP}$ is plotted against $Re = \frac{TP}{TP+FN}$ when the threshold of classification probability is varied from zero to one. Then the area under the curve of the plot is measured. The area is equal to 1 for a perfect classification [36].

I. RECEIVER OPERATOR PLOT AREA UNDER CURVE

Receiver Operator Curve (ROC) is the plot of $TPR = \frac{TP}{TP+FN}$ against $FPR = \frac{FP}{FP+TN}$. ROC AUC score is a metric which measures the trade-off between TPR and FPR when the threshold of the *classification probability* is varied from zero to one. A higher ROC AUC value indicates a better classification model [36].

J. AUTHENTICITY OF GENERATED BEATS

One of the metrics we used in our comparison is the authenticity test, in which the generated beats are checked whether they can function as and replace the *real* beats in a classification task. This investigation is done via the so-called *classification* or *authenticity* test. The objective of this binary classification test is to distinguish Normal beats from the anomaly (here we picked the typical Class L as the anomaly).

First, the state-of-the-art classifier (ResNet34) is trained on a totally balanced and all-real training set consisting of two classes: $\boldsymbol{N}$ (Normal Beat) and $\boldsymbol{L}$ (Left Bundle Branch Block Beat) with 7,000 samples in each class. The trained classifier is put to test on an unseen test set and the classification metrics are recorded. This is the *reference case*, and any other study case is compared with this case (Table 1).

Then, the *compromised case* is formed by imbalancing the training set purposely by reducing the number of samples in N class down to 350. The performance of the classification of the compromised case, which has been made poor intentionally, is recorded. Then, the imbalanced training set is augmented/balanced by the synthetic beats in each of the study cases and the classifier is trained on them. Since the training set is augmented/balanced, the classification performance improves significantly relative to the compromised case. The case which produces the best improvements in the classification metrics has produced the most authentic synthetic beats (Table 1). The same test set is used in all cases: 1000 samples of unseen real data in each class with no synthetic beats.

TABLE 1. Authenticity test training set supports.

Cases	Train Set Class N	Train Set Class L
Reference (Balanced, all real)	7000 r	7000 r
Compromised (Imbalanced)	350 r	7000 r
Augmented with case 00	350 r + 6650 s	7000 r
Augmented with case 01	350 r + 6650 s	7000 r
Augmented with case 02	350 r + 6650 s	7000 r
Augmented with case GAN	350 r + 6650 s	7000 r

r: Real Beat *s: Synthetically Generated Beat*

1) CLASSIFIER: ECGResNet34

ResNet34 [37] is the state-of-the-art tool used in the classification of images. It has 34 layers and incorporates *residual* building blocks. Each block is comprised of two 3×3 convolutional layers with a residual stream [37], which reduces the risk of gradient vanishing/exploding. It is pretrained on the ImageNet dataset (more than 100, 000 images in 200 classes). We used its $1D$ implementation [38] for our classification test.

III. EXPERIMENTAL SETUP

A. WGAN-GP MODEL DESIGN

The architectures of the generator and the critic in the WGAN-GP model are comprised of building blocks that are repeated multiple times (Table 2). The details of the architectures of the generator and the critic are shown in Table 3.

TABLE 2. WGAN-GP building blocks.

Layer	Generator	Critic
1	ConvTranspose1d [1]	ConvTranspose1d [1]
2	BatchNorm1d	InstanceNorm1d
3	ReLU	LeakyReLU

1: *kernel size = 4, stride = 2, padding = 1*

B. IMPROVED DDPM MODEL DESIGN

Improved DDPM [10] has been used *as-is* as the diffusion model. Codes of the DDPM are taken from [10] (https://github.com/openai/improved-diffusion).

TABLE 3. WGAN-GP architecture.

Layer	Generator	Critic
Input	$16 \times 100 \times 1$	$16 \times 1 \times 64$
1	Block	Vonv1d, LeakyReLU
2	Block	Block
3	Block	Block
4	Block	Block
5	ConvTranspose1d	Conv1d
6	FC	FC
7	tanh	-
Output	$16 \times 1 \times 64$	$16 \times 1 \times 1$

C. PLATFORM

The training of and sampling from the DDPMs are done on the Arc (the HPC cluster at the University of Texas at San Antonio (UTSA)). Currently, Arc can run programs with two V-100 GPUs on each node. For this study, one node from the cluster with two parallel GPUs has been used.

For the rest of the computations, a desktop and a laptop have been used: a Dell Alienware desktop with Intel i9-9900k at 3.6 GHz (8 cores, 16 threads) microprocessor, 64 GB RAM, and NVIDIA GeForce RTX 2080 Ti graphics card with 24 GB RAM, and a personal Dell G7 laptop with an Intel i7-8750H at 2.2 GHz (6 cores, 12 threads) microprocessor, 20 GB of RAM, and NVIDIA GeForce 1060 MaxQ graphics card with 6 GB. Our codes are available on the GitHub page of the paper (https://github.com/mah533/Synthetic-ECG-Signal-Generation-using-Probabilistic-Diffusion-Models).

D. STUDY CASES

Three different hyperparameter settings for the Improved DDPM have been considered which are shown in Table 4. The rest of the parameters are the same for all cases.

TABLE 4. DM case studies.

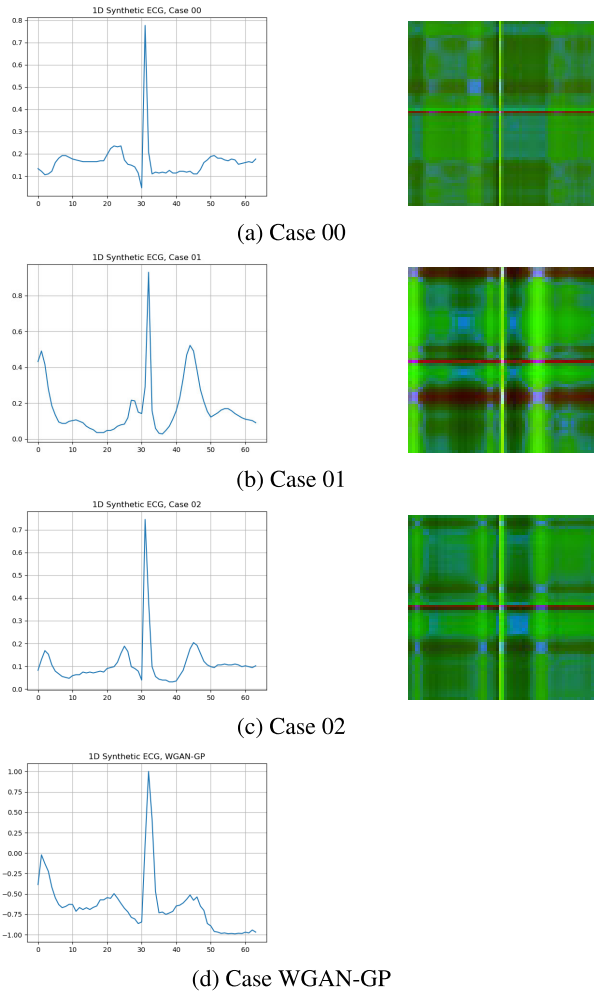
Cases	Learn Sigma	Noise Schedule	Use KL [1]	Schedule Sampler
00	False	Linear	False	Uniform
01	True	Cosine	True	Uniform
02	True	Cosine	True	loss second moment

1: *Kullback-Leibeler (KL) Divergence*

The above three study cases have been compared to the 4^{th} case, which is the synthetic ECG beats generated by the WGAN-GP. Since the ultimate goal is to generate realistic synthetic beats that resemble and function like real beats as closely as possible, an additional case (rl) is considered for reference, in which real beats have been used instead of synthetic ones in the corresponding comparison.

IV. RESULTS

Samples of the generated synthetic ECG signals (1D and 2D) are shown in the Figure 3. The aforementioned four case studies are compared by the *quality*, *distribution* and *authenticity* of the generated beats in each case.



(a) Case 00

(b) Case 01

(c) Case 02

(d) Case WGAN-GP

FIGURE 3. Samples of synthetically generated ECG signals.

A. QUALITY

When discussing the quality of the beats, we refer to how much the generated synthetic beats resemble the real ones in appearance and morphology. For assessing the quality of the heartbeats quantitatively, *the average distances of the generated beats from a randomly selected template* is measured using two distance functions (AKA similarity measures): Dynamic Time Warping (DTW) and Fréchet distance functions. Both distance functions consistently show that the generated beats by the WGAN-GP model are by far closer to the reference case (rl), in terms of quality (Table 5).

B. DISTRIBUTION

Maximum Mean Discrepancy (MMD) is a kernel based statistical tool to measure the distance between two *distributions*

TABLE 5. Quality of generated beats.

Cases	Ave. DTW Distance	Ave. Fréchet Distance
00	6.67	1.074
01	6.95	1.117
02	6.36	1.042
GAN	2.12	0.723
Real (rl)	2.09	0.718

(Eq. 21) [39]. To compare the distributions of generated beat sets, equal number of samples (7000, i.e., the total number of generated beats) from the real data (rl) and the generated data in each case are selected randomly, ($m = n = 7000$) and the MMD value between them is measured utilizing the linear kernel. For reference, the MMD value between two disjoint sets of real samples is shown as well (Table 6).

$$
\begin{aligned}
&MMD^2\,(p, q)\\
&= \mathbb{E}_{x,x'}\left[k\left(x_i, x_j'\right)\right] - 2\mathbb{E}_{x,y}\left[k\left(x_i, y_j\right)\right] + \mathbb{E}_{y,y'}\left[k\left(y_i, y_j'\right)\right]\\
&= \frac{1}{m(m-1)}\sum_{i=1}^{m}\sum_{j\neq i}^{m} k\left(x_i, x_j\right) - \frac{2}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} k\left(x_i, x_j\right)\\
&\quad + \frac{1}{n(n-1)}\sum_{i=1}^{n}\sum_{j\neq i}^{n} k\left(y_i, y_j\right) \qquad (21)
\end{aligned}
$$

In terms of the distribution of the generated beats, the WGAN-GP model generates beats much closer to the real beats than the diffusion models do.

TABLE 6. MMD value of synthetic and real beats.

Cases	00-rl	01-rl	02-rl	GAN-rl	rl-rl
MMD	39.8	44	35.9	1.00	0.0

C. AUTHENTICITY

Here we quantitatively measure how much the generated beats can replace (i.e., function as) the real ones in a classification test. The metrics used for the authenticity tests are $(\boldsymbol{A})$ Average Precision scores, $(\boldsymbol{B})$ the Area Under the Curve (AUC) of the Precision-Recall Curves, as well as $(\boldsymbol{C})$ the AUC of the Receiver Operating Characteristic curves (AUC ROC score).

The Average Precision scores show that WGAN-GP model outperforms the DDPM in correctly classifying the beats, minimizing FP and maximizing TP. It should be noted that micro- and macro-averages are the same as the test set is balanced (Table 7).

1) PRECISION - RECALL CURVES

The Precision-Recall curves' Area Under Curve, (PR AUC), score is a better metric than the Average Precision score, which is calculated at only one threshold. By checking the

TABLE 7. Authenticity of generated beats.

Cases	Ave Precision	PRC [1] AUC [2] Score	ROC [3] AUC Score
00	0.90	0.95	0.96
01	0.55	0.68	0.63
02	0.76	0.76	0.81
GAN	0.96	0.99	0.99
Real (rl)	0.98	1.00	1.00

1: *Precision-Recall Curve*
2: *Area Under Curve*
3: *Receiver Operating Characteristic Curve*

Precision-Recall curves visually, it can be seen that the WGAN-GP model (Figure 7) produces a graph very close to the real case (Figure 8). However, case 00 has the best PR curve among the DDPM cases (Figure 4). Also, the Precision-Recall AUC score (Table 7) confirms the visual check. The same argument holds for the ROC AUC score.

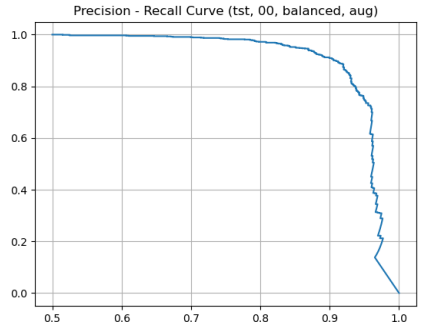


FIGURE 4. Precision-Recall curve, Case 00.

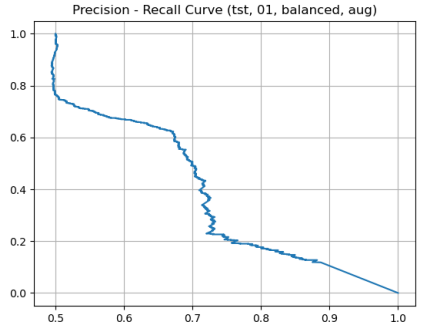


FIGURE 5. Precision-Recall curve, Case 01.

2) CONFUSION MATRIX

The elements on the main diagonal of the confusion matrix show the percentage of number of times the beats are classified correctly. Again, the synthetic beats generated by the WGAN-GP model behave much more like real beats than the beats generated by DDPM do.

V. DISCUSSION

In a typical classification task, since there is no need to map the data back into their original space, no *invertibility* condition is required for the embedding. As long as there is one *and only one* corresponding element in the embedded space, and no two datapoints in the original space have the same mapped datapoint in the destination space (i.e., *injective* mapping), mapping is acceptable. However, in generation tasks, the generated data must be de-embedded and mapped back into the original space. Therefore, the mapping must be *bijective*. That is why the spectrograms are useful in classification tasks but not in generation tasks.

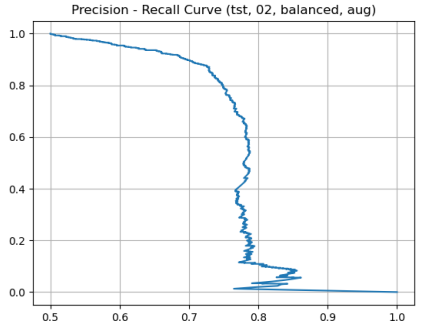


FIGURE 6. Precision-Recall curve, Case 02.

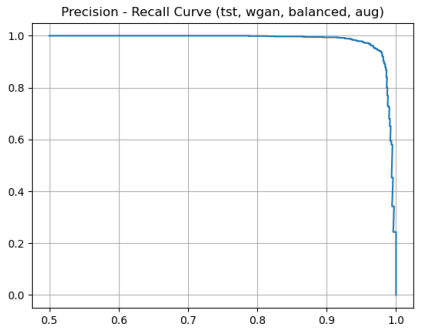


FIGURE 7. Precision-Recall curve, Case GAN.

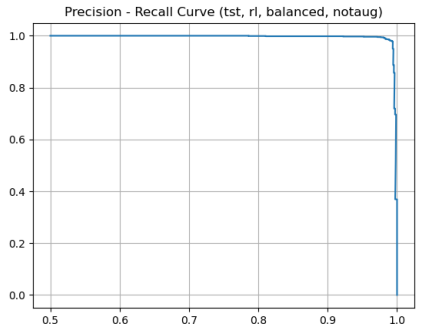


FIGURE 8. Precision-Recall curve, Case real.

Each ECG beat in our dataset has 64 timesteps. Therefore with the WGAN-GP model, each time 64 pieces of information are generated per heartbeat, 100% of which are *useful information*, from the user's perspective. Whereas in DDPM, $3 \times 64 \times 64 = 12,288$ pieces of information are generated per beat, from which only 64 pieces (5.2%) are useful/extracted and the rest are discarded. In the training of the deep learning models, the optimization process finds the optimum values of the trainable parameters in a way that each piece of the generated information are in some certain

TABLE 8. Confusion matrices (all values are in %).

	Pred. N	*Pred. L*
Real N	85.7	14.3
Real L	5.4	94.6

(a) Case 00

	Pred. N	*Pred. L*
Real N	37.1	62.9
Real L	28.2	71.8

(b) Case 01

	Pred. N	*Pred. L*
Real N	73.6	26.4
Real L	21.0	79.0

(c) Case 02

	Pred. N	*Pred. L*
Real N	93.9	6.1
Real L	1.8	98.2

(d) Case GAN

	Pred. N	*Pred. L*
Real N	99.4	0.6
Real L	3.4	96.6

(e) Case real

neighborhood. The total error in the loss function is *typically* comprised of 64 elements in the WGAN-GP and 12, 288 elements in DDPM per beat. Thus, a tighter neighborhood in WGAN-GP (closer to the real beats) and a more relaxed one in DDPM is quite natural.

VI. CONCLUSION

A. WHAT WE DID

In this paper, we presented a pipeline to generate synthetic $1D$ ECG time series using the $2D$ probabilistic diffusion model (Improved DDPM).

B. WHY WE DID IT

With the remarkable success of $2D$ computer vision models, specifically the Improved DDPM and its superiority to the GAN models [10], [11]) and the widespread availability of their pretrained version, it makes sense to apply them to $1D$ time series. One of the benefits of the processing of the data in $2D$ space is providing additional data augmentation techniques (such as flipping, rotation, and mirroring), which are helpful specifically in classification tasks.

C. HOW WE DID IT

In this study, we used unconditional models and used only the N class (*Normal Sinus Beat*) from the MIT-BIH Arrhythmia dataset [13], [14]. The general pipeline used in this study is shown in Figure. 1. First, the $1D$ ECG time series are transformed into the polar coordinates. Then, they are embedded into the 3-channel $2D$ space, similar to RGB image files. *Gramian Summation/Difference Fields* (GASF/GADF) and *Markov Transition Field* (MTF) are used to produce the three $2D$ matrices, which are then put together to form one single image file for each beat. The Improved DDPM model [10] is trained and sampled to generate $2D$ ECG signals, which are then de-embedded to reconstruct a $1D$ ECG signal. The generated data by DDPM are in 3 study cases with 3 different settings in hyperparameters. They are compared to the synthetically generated data by WGAN-GP, as well as to the *real beats*, to see how close the cases are to reality, since the ultimate goal is to generate realistic synthetic ECG beats. The comparison is done in terms of the *quality*, *distribution*, and *authenticity* of the data (i.e., up to what extent they can replace the *real* beats in data augmentation for classification tasks).

D. RESULTS SUMMARY

The results show that the synthetic beats generated by the WGAN-GP model are consistently closer to the real beats than the beats from DDPM, in all cases and by all the metrics used. The average distances of the beats from a template (measured by the DTW and Fréchet Distance functions) as well as the distribution of generated beats (measured by MMD) respectively reveal that the *quality* and the *distribution* of the beats from the WGAN-GP are much closer to the corresponding real ones than those of the DDPM. For quantifying the *authenticity* of the generated beats, we used a classification test using the-state-of-the-art classifier ResNet34 [37] and several standard classification metrics, namely average precision score, AUC of Precision-Recall curves, and AUC of ROC curves, all of which constantly show that the WGAN-GP model outperforms the Improved DDPM in this regard.

VII. LIMITATIONS AND FUTURE WORKS

The Improved DDPM [10] developed by OpenAI is a $2D$ model, i.e., the input/output data and the processing, are in $2D$ space and it has been used in this study *as-is* with almost no changes. We proposed the pipeline in Figure 1 for this application, where the $1D$ ECG data are mapped into a $2D$ space, converted into image files, and fed to a DDPM model. The processing takes place in the $2D$ space and the generated $2D$ data is de-embedded back into $1D$ space, where the $1D$ ECG data are reconstructed, whereas the WGAN-GP model developed and used in this study is inherently $1D$, i.e., the input/output data and the processing are all in $1D$ space, with no embedding necessary. It should be emphasized that the conclusions drawn here apply only to the setting and the pipeline used in this research, and the results might be different in any other setting.

Although the probabilistic diffusion model is being applied to images and $2D$ data mostly, the general concept can be applied to $1D$ data as well, i.e., the model can take $1D$ data, perform the noising/denoising processes in $1D$ space, and generate synthetic data in $1D$ without any embedding. In this case, the model would be a better representative of the diffusion concept and the comparison would be more realistic.

APPENDIX A
GAN, WGAN AND WGAN-GP MODELS

Generative Adversarial Networks are *two-player zero-sum minimax* games with the following loss function:

$$\min_G \max_D \mathbb{E}_{x\sim p_{\text{data}}(x)}\left[\log D(x)\right] + \mathbb{E}_{z\sim p_z(z)}\left[1-\log D(G(z))\right] \tag{A.1}$$

Unlike variational method models, in which the objective is to find the true distribution of the data [32], in GAN models the constraints are on the expectation of the outputs of the generator and the discriminator. Therefore, it is quite possible that the GAN model focuses only on a few modality of the

distribution and still the loss function is satisfied (mode collapse). Moreover, the parameters oscillate and convergence is not achieved always.

Using the the Wasserstein distance function (Kontorovich-Rubenstein Duality) [40] improves the convergence:

$$W_1\left(p_r, q_g\right) = \sup_{\|f\|_L \leq 1} \mathbb{E}_{x\sim p_r}\left[f(x)\right] - \mathbb{E}_{x\sim p_g}\left[f(x)\right] \quad \text{(A.2)}$$

However, the constraint $\|f\|_L \leq 1$ requires $f(.)$ be 1-Lipschitz. Then the loss function of WGAN becomes:

$$\min_G \max_D \mathbb{E}_{x\sim p_{\text{data}}(x)}\left[D(x)\right] - \mathbb{E}_{x\sim p_{\text{g}}(\tilde{x})}\left[D(\tilde{x})\right] \quad \text{(A.3)}$$

Parameter clipping can be used to assure the Lipschitz condition, i.e., keeping the magnitude of the parameters bounded $[-c, c]$, which can easily lead to optimization difficulties (vanishing/exploding gradient) [41].

Gulrajani et al. [41] proposed the WGAN with Gradient Penalty (WGAN-GP) in which, instead of parameter clipping, they penalized (regularize) the loss function with the magnitude of the gradient as a constraint, using a Lagrange multiplier λ:

$$\min_G \max_D \mathbb{E}_{x\sim p_{\text{data}}(x)}\left[D(x)\right] - \mathbb{E}_{x\sim p_{\text{g}}(\tilde{x})}\left[D(\tilde{x})\right] + \lambda\ \mathbb{E}_{\hat{x}\sim p_{\text{data}}(x)}\left[\left(\left\|\nabla_{\hat{x}} D\left(\hat{x}\right)\right\|_2 - 1\right)^2\right] \quad \text{(A.4)}$$

REFERENCES

[1] S. Saadatnejad, M. Oveisi, and M. Hashemi, "LSTM-based ECG classification for continuous monitoring on personal wearable devices," *IEEE J. Biomed. Health Informat.*, vol. 24, no. 2, pp. 515–523, Feb. 2020.

[2] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, Jun. 2002.

[3] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2999–3007.

[4] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, "Generalizing from a few examples: A survey on few-shot learning," *ACM Comput. Surv.*, vol. 53, no. 3, pp. 1–34, May 2021.

[5] I. J. Goodfellow, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Red Hook, NY, USA: Curran Associates, 2014, pp. 2672–2680.

[6] C. Doersch, "Tutorial on variational autoencoders," 2016, *arXiv:1606.05908*.

[7] M. Frid-Adar, E. Klang, M. Amitai, J. Goldberger, and H. Greenspan, "Synthetic data augmentation using GAN for improved liver lesion classification," in *Proc. IEEE 15th Int. Symp. Biomed. Imag. (ISBI)*, Apr. 2018, pp. 289–293.

[8] V. V. Kuznetsov, V. A. Moskalenko, and N. Yu. Zolotykh, "Electrocardiogram generation and feature extraction using a variational autoencoder," 2020, *arXiv:2002.00254*.

[9] T. J. Jun, H. M. Nguyen, D. Kang, D. Kim, D. Kim, and Y.-H. Kim, "ECG arrhythmia classification using a 2-D convolutional neural network," 2018, *arXiv:1804.06812*.

[10] A. Q. Nichol and P. Dhariwal, "Improved denoising diffusion probabilistic models," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 8162–8171.

[11] P. Dhariwal and A. Nichol, "Diffusion models beat GANs on image synthesis," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 8780–8794.

[12] Z. Wang and T. Oates, "Imaging time-series to improve classification and imputation," in *Proc. 24th Int. Joint Conf. Artif. Intell.*, 2015, pp. 3939–3945.

[13] G. B. Moody and R. G. Mark, "The impact of the MIT-BIH arrhythmia database," *IEEE Eng. Med. Biol. Mag.*, vol. 20, no. 3, pp. 45–50, May/Jun. 2001.

[14] A. Goldberger, L. Amaral, L. Glass, J. Hausdorff, P. C. Ivanov, R. Mark, J. Mietus, G. Moody, C. Peng, and H. Stanley, "Components of a new research resource for complex physiologic signals," *PhysioBank, PhysioToolkit, Physionet*, vol. 101, no. 23, pp. e215–e220, 2000.

[15] J. M. L. Alcaraz and N. Strodthoff, "Diffusion-based conditional ECG generation with structured state space models," 2023, *arXiv:2301.08227*.

[16] P. Wang, B. Hou, S. Shao, and R. Yan, "ECG arrhythmias detection using auxiliary classifier generative adversarial network and residual network," *IEEE Access*, vol. 7, pp. 100910–100922, 2019.

[17] A. M. Delaney, E. Brophy, and T. E. Ward, "Synthesis of realistic ECG using generative adversarial networks," 2019, *arXiv:1909.09150*.

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

[19] E. Adib, F. Afghah, and J. J. Prevost, "Synthetic ECG signal generation using generative neural networks," 2021, *arXiv:2112.03268*.

[20] E. Adib, F. Afghah, and J. J. Prevost, "Arrhythmia classification using CGAN-augmented ECG signals," 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9995088

[21] Z. Ahmad, A. Tabassum, L. Guan, and N. M. Khan, "ECG heartbeat classification using multimodal fusion," *IEEE Access*, vol. 9, pp. 100615–100626, 2021.

[22] H. Cai, L. Xu, J. Xu, Z. Xiong, and C. Zhu, "Electrocardiogram signal classification based on mix time-series imaging," *Electronics*, vol. 11, no. 13, p. 1991, Jun. 2022.

[23] A. Diker, Z. Cömert, E. Avci, M. Togaçar, and B. Ergen, "A novel application based on spectrogram and convolutional neural network for ECG classification," in *Proc. 1st Int. Informat. Softw. Eng. Conf. (UBMYK)*, Nov. 2019, pp. 1–6.

[24] E. Izci, M. A. Ozdemir, M. Degirmenci, and A. Akan, "Cardiac arrhythmia detection from 2D ECG images by using deep learning technique," in *Proc. Med. Technol. Congr. (TIPTEKNO)*, Oct. 2019, pp. 1–4.

[25] C. Hao, S. Wibowo, M. Majmudar, and K. S. Rajput, "Spectro-temporal feature based multi-channel convolutional neural network for ECG beat classification," in *Proc. 41st Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Jul. 2019, pp. 5642–5645.

[26] J. Huang, B. Chen, B. Yao, and W. He, "ECG arrhythmia classification using STFT-based spectrogram and convolutional neural network," *IEEE Access*, vol. 7, pp. 92871–92880, 2019.

[27] A. T. Oliveira and E. G. O. Nobrega, "A novel arrhythmia classification method based on convolutional neural networks interpretation of electrocardiogram images," in *Proc. IEEE Int. Conf. Ind. Technol. (ICIT)*, Feb. 2019, pp. 841–846.

[28] M. Salem, S. Taheri, and J. Yuan, "ECG arrhythmia classification using transfer learning from 2-dimensional deep CNN features," in *Proc. IEEE Biomed. Circuits Syst. Conf. (BioCAS)*, Oct. 2018, pp. 1–4.

[29] B. M. Mathunjwa, Y.-T. Lin, C.-H. Lin, M. F. Abbod, and J.-S. Shieh, "ECG arrhythmia classification by using a recurrence plot and convolutional neural network," *Biomed. Signal Process. Control*, vol. 64, Feb. 2021, Art. no. 102262.

[30] B. M. Mathunjwa, Y.-T. Lin, C.-H. Lin, M. F. Abbod, M. Sadrawi, and J.-S. Shieh, "ECG recurrence plot-based arrhythmia classification using two-dimensional deep residual CNN features," *Sensors*, vol. 22, no. 4, p. 1660, Feb. 2022.

[31] H. Zhang, C. Liu, Z. Zhang, Y. Xing, X. Liu, R. Dong, Y. He, L. Xia, and F. Liu, "Recurrence plot-based approach for cardiac arrhythmia classification using Inception-ResNet-v2," *Frontiers Physiol.*, vol. 12, p. 558, May 2021.

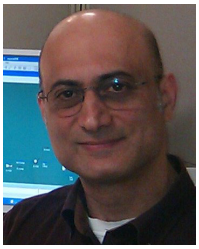
[32] D. P. Kingma and M. Welling, "An introduction to variational autoencoders," 2019, *arXiv:1906.02691*.

[33] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds. Red Hook, NY, USA: Curran Associates, 2020, pp. 6840–6851.

[34] A. Razavi, A. Van den Oord, and O. Vinyals, "Generating diverse high-fidelity images with VQ-VAE-2," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 14866–14876.

[35] T. Henighan, J. Kaplan, M. Katz, M. Chen, C. Hesse, J. Jackson, H. Jun, T. B. Brown, P. Dhariwal, S. Gray, C. Hallacy, B. Mann, A. Radford, A. Ramesh, N. Ryder, D. M. Ziegler, J. Schulman, D. Amodei, and S. McCandlish, "Scaling laws for autoregressive generative modeling," 2020, *arXiv:2010.14701*.

[36] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, vol. 2. Cham, Switzerland: Springer, 2009.
[37] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[38] A. Lyashuk. (2021). *ECG Classification*. [Online]. Available: https://github.com/lxdv/ecg-classification/blob/master/README.md
[39] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *J. Mach. Learn. Res.*, vol. 13, pp. 723–773, Mar. 2012.
[40] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," 2017, *arXiv:1701.07875*.
[41] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of Wasserstein GANs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5767–5777.



EDMONMD ADIB (Student Member, IEEE) received the B.S. degree in chemical engineering from the University of Tehran, the M.Sc. degree in chemical engineering specialized in advanced process control and system identification from the Sharif University of Technology, and the master's degree in petroleum engineering from the University of Calgary. He is currently pursuing the Ph.D. degree with the Electrical and Computer Engineering Department, UTSA. He is a Teaching Assistant with UTSA. His research interests include deep generative algorithms and biomedical signal analysis and processing.

AMANDA S. FERNANDEZ (Senior Member, IEEE) received the B.S. degree in computer science from the Siena College, in 2007, and the master's and Ph.D. degrees from the University at Albany State University of New York (SUNY), in 2011 and 2015, respectively. Her research laboratory, the UTSA Vision and Artificial Intelligence Laboratory, focuses on explainable deep learning and optimizations, with applications in computer vision and the physical sciences. She was with Industry, as a Machine Learning Engineer and a Software Developer. She is currently an Assistant Professor with the Department of Computer Science, The University of Texas at San Antonio. She holds 19 issued U.S. patents and the author/co-author to many conference and journal publications in the field of AI.

FATEMEH AFGHAH (Senior Member, IEEE) is currently an Associate Professor with the Electrical and Computer Engineering Department, Clemson University, and the Director of the Intelligent Systems and Wireless Networking (IS-WiN) Laboratory. Prior to joining Clemson University, she was an Associate Professor with the School of Informatics, Computing and Cyber Systems, Northern Arizona University. She is the author/coauthor of more than 120 peer-reviewed publications, six U.S. patents. Her research interests include wireless communication networks, decision-making in multi-agent systems, UAV networks, and artificial intelligence in healthcare. She was a recipient of several awards, including the Air Force Office of Scientific Research Young Investigator Award, in 2019, the NSF CAREER Award, in 2020, the NAU's Most Promising New Scholar Award, in 2020, and the NSF CISE Research Initiation Initiative (CRII) Award, in 2017. She served as an Associate Editor for several journals, including *ACM Transactions on Computing for Healthcare*, *Journal of Network and Computer Applications* (Elsevier), *Ad Hoc Networks*, *Computer Networks*, and *Neural Processing Letters* (Springer).

JOHN J. PREVOST (Senior Member, IEEE) received the B.S. degree in economics from Texas A&M, in 1990, and the B.S. (magna cum laude), M.S., and Ph.D. degrees in electrical engineering from The University of Texas at San Antonio (UTSA), in December 2009, 2012, and December 2013, respectively. In 2015, he Co-Founded and became the Chief Research Officer and the Assistant Director of the Open Cloud Institute. He was the Director of the Product Development, the Director of Information Systems, and the Chief Technical Officer for various technical firms. He is currently an Assistant Professor with the Department of Electrical and Computer Engineering, UTSA. He remains an active consultant in the areas of complex systems and cloud computing and maintains strong ties with industry leaders. His research interests include energy-aware cloud optimization, cloud-controlled robotics, cloud-based communications, and quantum cloud computing.