

An Innovative Approach for Long ECG Synthesis with Wasserstein GAN Model

Thi Diem Tran[1(✉)], Thi Thu Khiet Dang[2], and Ngoc Quoc Tran[3]

[1] University of Information Technology, VNU-HCM, Ho Chi Minh City 70000, Vietnam
diemtt@uit.edu.vn

[2] University of International, VNU-HCM, Ho Chi Minh City 70000, Vietnam
BEBEIU17015@student.hcmiu.edu.vn

[3] John Von Neumann Institute, University of Science, VNU-HCM, Ho Chi Minh City 70000, Vietnam
quoc.tran2020@ict.jvn.edu.vn

Abstract. Deep neural networks (DNNs) have set new standards in identifying and classifying irregular patterns in ECG (electrocardiogram) signals, surpassing previous methods. Despite the easy access and affordability of ECG sensors, a critical bottleneck remains the limited availability of reliable data for complex heart rhythms like second and third-degree atrioventricular block, ventricular tachycardia, and supraventricular tachycardia. This shortage has been a significant obstacle to improving DNN algorithms. Recent studies have turned to Generative Adversarial Networks (GANs) to create synthetic ECG data, enhancing the diversity of training datasets. However, much of this research has only managed to produce basic ECG components, missing the intricate details found in real patient data that includes multiple heartbeats. Our research has taken a groundbreaking approach by converting ECG signals into a two-dimensional format, allowing us to utilize advanced GAN models originally developed for image processing. This method has enabled us to generate extended, realistic ECG sequences closely mimicking those from actual patients. We have tested and refined our model using two databases, Physionet and Chapman, and have successfully produced 10-second ECG sequences showcasing a variety of heart rhythms previously unachieved in other studies. Our innovative technique not only surpasses existing methods in generating high-quality, realistic ECG data but also sets a new benchmark in ECG synthesis.

Keywords: ECG generating · Long ECG · Signal to Image · GAN

1 Introduction

An electrocardiogram (ECG) is a time-lapse recording of a re-polarization sequence in the human heart using a piece of standard non-invasive equipment. It is utilized for clinically diagnosing cardiac diseases by understanding abnormalities, for example, second and third-degree atrioventricular block, ventricular

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
O. Gervasi et al. (Eds.): ICCSA 2024, LNCS 14814, pp. 339–351, 2024.
https://doi.org/10.1007/978-3-031-64608-9_22

tachycardia, or ventricular run supraventricular tachycardia, in the signal space [1]. Many applications have embedded DNNs to examine rhythms abnormal in patients, such as Medtronic Linq monitor, Zio monitor, or Apple Watch. Series 4 [2]. However, the performance of DNNs, particularly classification tasks, still needs to improve due to imbalanced data throughout the training model. Augmented data with artificial data generation by creating natural-looking samples that are not available in the original data set is a new approach to increasing the accuracy of DNNs. Furthermore, the ability to create realistic samples with specific attributes might be beneficial in comprehending the original data distribution.

Generative adversarial network, initially described in 2014 [3], is a recently suggested framework for producing generated data that shows remarkable results for producing artificial images. They successfully created and altered data in computer vision and NLP areas, such as high-quality picture production, style transfer, text-to-image synthesis [4,5], etc. GANs have mainly been created and used for picture creation, with just a few research addressing time series. Nonetheless, they have recently shown promising results for the generation of artificial audio [6]. Many current studies have used GANs for ECG signal creation to address the issue of data scarcity. Previous efforts for creating a time-series GAN have mainly relied on Recurrent Neural Network (RNN)-based architectures [7]. To synthesize ECG data, Brophy et al. [8] suggested a GAN architecture that included a Bidirectional LSTM-as Generator and an LSTM-CNN as Discriminator. However, the authors only employed the generator to synthesize ECG data and offered no evaluation of the discriminator network for the arrhythmia classification task. Besides, this GAN-RNN cannot handle lengthy sequential data that suffers from a vanishing gradient issue. Wasserstein GAN [9] claimed the ECG synthesis result was superior to the Denoising Diffusion Probabilistic Models (DDPM), but it was also stopped at one beat ECG generation. Some works have successfully inspired transformer architecture in GAN, intending to improve the quality of synthetic data or create a more efficient training procedure. One of the famous examples of trends using a transformer in 1D signal processing is the TTS-GAN [4]. This research had realistic synthetic time-series data sequences of any length for sinusoidal waves, jumping, and running from the UniMiB dataset. However, it also comes to a halt as it generates an artificial ECG with only one QRS complex (or one beat).

Several previous studies have investigated the synthesis of long ECG signals, but their purposes and methodologies differ from our research. For example, the study of Vo et al. (2021) [10] utilized P2E-WGAN for the reconstruction from the realistic Photoplethysmogram (PPG) to ECG signals. They suggested that while collecting high-quality ECG data remains challenging, deriving it from a wearable PPG device would be much more convenient for the patients. In other words, they did not try to generate a fake ECG signal but established a framework to predict ECG data based on the available PPG. As a result, their research still needs to address the lacking-data problem as one of our proposals. The study of Zhang et al. (2021) [7] considered the implementation of a 2D

long short-term memory generative adversarial network (GAN) to produce ECG signals. Their approach generated representative beats and then attached 13 of the same representative beats to each other to form a 10-second ECG signal. The limitation of this approach is that such a framework is only suitable for generating normal rhythms with the defined morphology, not abnormal ones. Heart rate would be the same in all of the generated data because there are always 13 QRS complexes available in a 10-second signal. The research [11,12] had succeeded in long ECG generation with the DDPM model; however, the signal generation velocity in the inference stage of this strategy is a burden. As a result, an architecture that can automatically generate long ECG signals with various morphology and patterns in a short period is required.

Because of the vanishing gradient, instability, and quality of ECG generation, while training GAN directly for 1D long ECG signals, in this paper, we propose a new approach for a generative network that inspired GAN architecture for the image. We convert a 1D signal to a 2D image and optimize the network to achieve high performance on the GAN for images. Instead of a complex network, we toward our approach by using a simple architecture GAN named the DCGAN [13] as our baseline model. This research focuses on generating storage rhythms in ECGs such as second and third-degree atrioventricular block, ventricular tachycardia, or ventricular run supraventricular tachycardia. The Physionet (MIT-BIH Arrhythmia, European ST-T) [14] and Chapman [15] databases are utilized for the training model. We benchmark our proposal with other state-of-the-art architecture for the 1D ECG-GANs model. The result shows outperformance not only for one beat ECG but also for a 10-second ECG signal with various rhythms like acquiring from patients, which has yet to be reported.

2 Methodology

2.1 Premilinaries: Generative Adversarial Networks

Generative Adversarial Networks (GANs) and variants are powerful models for data synthesis. The strategy of GAN is to define a min-max game between two competing networks: a *generator* and a *discriminator*. The *generator* takes the role of generating a sample from random noise, while the *discriminator* is trained to classify a true sample or a generated sample as real or fake. Formally, given a random noise vector z sampled from a prior distribution $p(z)$ (e.g., uniform distribution) and real data sample x drawn from $p(x)$. The *generator* G aims to generate realistic-looking sample $G(z)$ and fool the *discriminator*, and the *discriminator* D tries to distinguish $G(z)$ from x. This game between the *generator* G and the *discriminator* D can be formulated as the following min-max objective function:

$$\min_G \max_D V(G,D) = \mathbb{E}_{x\sim p(x)}[\log(D(x))] + \mathbb{E}_{z\sim p(z)}[\log(1-D(G(z)))] \quad (1)$$

Despite the interesting mechanism and good results in generating data, stabilizing GAN training is a challenge. Many approaches have been proposed to handle this problem, prominent among them are WGAN [16] and WGAN-GP [17].

WGAN stands for Wasserstein GAN, a variant of GAN whose loss function relies on the Wasserstein distance. In WGAN, the *discriminator* is no longer used as a binary classifier but as an evaluation component (called *critic*) used to fit the Wasserstein distance. The objective function of WGAN is formulated:

$$\min_{G} \max_{D \in \mathcal{D}} V(G, D) = \mathbb{E}_{x \sim p(x)}[D(x)] - \mathbb{E}_{z \sim p(z)}[D(G(z))] \tag{2}$$

where $\mathcal{D}$ is the set of 1-Lipschitz functions. WGAN makes the loss function avoid the vanishing gradient problem and stabilizes the GAN training.

WGAN-GP has the same manner as WGAN except the term of *gradient penalty*. Specifically, WGAN-GP uses gradient penalty instead of weight clipping to enforce the 1-Lipschitz constraint for the *critic*, and the right term in the formula (2) becomes:

$$\mathbb{E}_{x \sim p(x)}[D(x)] - \mathbb{E}_{z \sim p(z)}[D(G(z))] + \lambda \mathbb{E}_{\hat{x} \sim p(\hat{x})}[(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2] \tag{3}$$

where λ is the penalty coefficient and $p(\hat{x})$ is the uniform distribution sampled between points in $p(x)$ and $p(z)$. WGAN-GP has been proven to improve GAN training stability, produce high-quality samples, and become much faster convergence power than WGAN.

2.2 Generating Long ECGs Using GAN

Previous studies tackled the ECG synthesis challenge by constructing a direct GAN model or converting the best GAN model for an image to process the signal in the time domain. However, these results cannot provide long ECGs like those obtained from patients. As a result, the output cannot be employed in real-world applications such as rhythm categorization or detecting aberrant heart disorders.

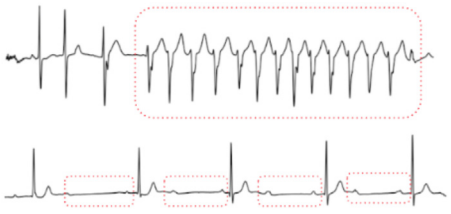

Fig. 1. Top: VT beat with signature V beat group. Bottom: AVB2 rhythm with regular repeats of two P waves.

In this paper, we choose the opposite approach that reuses the GAN model's power on the image and converts the ECG signal to the 2D domain. The intuition of converting the ECG signal into an image and working with it stems very much from the "morphology" character of the ECG signal. Unlike other biological signals (e.g., EEG), an ECG signal contains many morphological features like QRS complexes, P waves, T waves, etc., that professionals need to "see" to diagnose. These characters are more valuable when working with long ECG signals. Figure 1 shows an example of two types of ECG rhythms with different morphological features. To implement our intuition, we propose a new approach consisting of sequential steps: Split (for long ECGs) and draw an image of the ECG signal, train the GAN model with image data, and extract and convert to ECG signal from the generated image. Our proposal is illustrated in Fig. 2.

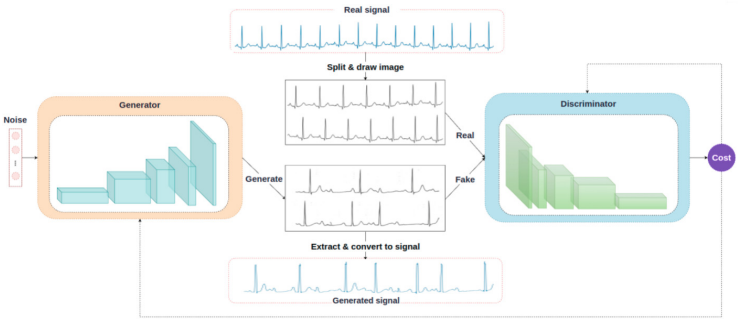

Fig. 2. Flowchart of our approach

In the first step, we plot an ECG signal without the information of magnitude or axes. We believe it is enough to represent the ECG signal on the image without adding extra details (like color). We aim to make our data as simple as possible but still contain the necessary information (like MNIST). The binary image met our expectations. We draw and convert the image to binary mode and then use image processing techniques to make the image look "smoother and softer". Besides, binary images bring convenience for the next steps. A binary image is formatted as a 2-dimensional array (matrix) instead of a 3-dimensional array (tensor) like a color image, making learning easier and reducing storage and computation costs. The extract signal step is done better with binary images, which will be shown in more detail in the following sections.

The next stage is to address the issue of the fit between the available data and the model. Such a problem is simpler to be addressed in one-beat synthesis when it can be fully represented in a symmetric image. Difficulties appear when working with longer ECG signals. Keeping the sample intact for a 10-second ECG signal will result in an extremely asymmetric image when one dimension is much larger than the other. On the other hand, using a symmetric image to represent the ECG signal in the image is shrinking and unnatural, especially when viewed from the perspective of signal amplitude. So we aim for an image of sufficiently large size and "acceptable" asymmetry. To solve this problem, we split the 10 s ECG into two 5 s ECG segments and put them in the same image. We searched through a dozen times and selected the image size that gives the best results as 128×128 for an ECG beat and 240×480 for a 10 s ECG signal. For the GAN model, we choose DCGAN [13] as our baseline model. DCGAN is a lightweight GAN model (compared to current GAN models) that is used a lot and achieves good results when working with image data. The architecture of DCGAN is built on top of convolution and convolution-transpose layers. We made a few adjustments to our problem.

We define $C(f, k, s)$ and $C_{ts}(f, k, s)$ to stand for convolution and convolution-transpose layers, f, k, and s are the number of filters, kernel size, and stride value, respectively. Not like the original DCGAN, we use the asymmetric kernel and stride in some convolution and convolution-

Algorithm 1. Convert image to 10-second ECG signal

Input: Input image: **I**, Binary image: **BI**, Width binary image: **W**: 120
Heigth binary image **H**: 480, Resize value **RS**: 2.5, Binary image threshold **BIT**: 250
Key **K**: 1, Threshold **T**: 5, Amplitude **Amp**: 1, Output signal **S**

for k *from* 0 *to* 1 **do**
  $\boldsymbol{BI} = Resize(\boldsymbol{I}, W \leftarrow W * RS, H \leftarrow H * RS)$
  for i *from* 0 *to* H **do**
    for j *from* 0 *to* W **do**
      $BI[i][j] = 1$ **if** $BI[i][j] < BIT$ **else** 0
    end for
  end for
  $\boldsymbol{BI} = MedianFilter(\boldsymbol{BI})$
  Remove pixel columns without information
  Placeholder target array: A
  for i *from* 0 *to* H **do**
    Placeholder index : O
    for j *from* 0 *to* W **do**
      $A = LinearSearchAlgorithm(key \in BI[i][j])$
    end for
    $O = SearchMaxLengthRunAlgorithm(A)$
    $O = (Max(O) - Min(O)) * T\% + Min(O)$
    $S[k][i] = (\frac{W-1-O-\frac{W-1}{2}}{\frac{(W-1)}{2}}) * Amp$
  end for
end for
S = $Concatenate(S[0], S[1])$

transpose layers as our model works with asymmetric images. More specifically, for the *generator*, we build 8 convolution-transpose layers with the configuration $C_{ts}(512, 4, 1) - C_{ts}(256, 4, 2) - C_{ts}(256, 3, 2) - C_{ts}(128, 4, 2) - C_{ts}(128, (3, 4), (1, 2)) - C_{ts}(64, 4, 2) - C_{ts}(64, 4, 2) - C_{ts}(1, 4, 2)$. The input of the *generator* is a vector noise of length 100, and the output is an image of size 240×480x1. For the *discriminator*, we also create 8 convolution layers with the configuration $C(64, 4, 2) - C(64, 4, 2) - -C(128, 4, 2) - C(128, (3, 4), (1, 2)) - C(256, 4, 2) - C(256, 3, 2) - C(512, 4, 2) - C(1, 4, 1)$. The input of the *discriminator* is an image of size 240×480x1, and the output is a scalar. Following each convolution or convolution layer in both *generator* and *discriminator* are batch normalization layers, LeakyRelu as activation function and dropout layer with a ratio of 0.3 to prevent overfitting. To stabilize GAN training and give high-quality results, we also utilize WGAN-GP in our proposal. When using WGAN-GP, we replace the last convolution layer in the *discriminator* with a fully connected layer, the sigmoid activation function after the last layer in vanilla GAN is also removed.

The produced images are then translated into ECG signals in the final stage. We define a workflow, described in Algorithm 1, to enhance this stage. Specifically, the image would be separated into two halves with the shape of 120×480,

one for each of the two five-second signal segments. Image dimensions are amplified 2.5 times to form the new image of 300×1200. This process is conducted to guarantee that the number of pixel columns in the picture is roughly equal to the length of the signals and that the ECG morphological features are not deformed if the two dimensions are not enlarged equally. The image would then be converted to binary using a threshold of 250 and filtered with a median filter to eliminate salt and pepper noise. The image is then evaluated in each pixel column to create the target array A consisting of the positions of pixel value equal to 1 using the Linear Search Algorithm. After that, the Search Max Length Run Algorithm is applied to determine the longest run in array A. After several attempts, the best ECG signals are obtained by setting the threshold T to 5 (choose the position in the longest run at the percentile T). These places would then be translated to the relevant value in the range (-Amplitude, Amplitude) and concatenated to produce an ECG signal. This algorithm shows the flexibility of our proposal when we can control the amplitude of the generated signal.

3 Experiments

3.1 Dataset

This study uses Physionet (MIT-BIH Arrhythmia, European ST-T) [14] and Chapman [15] datasets to evaluate performance. First, the MIT-BIH contains ECG signals from 48 participants captured at a sampling rate of 360 Hz. The second Physionet database used in this study is the European ST-T dataset, consisting of 90 annotated ECG recordings sampling at 250 Hz. Besides, the Chapman dataset consists of ECG signals collected from 10,646 patients at a 500 Hz sampling rate. All records were resampled into 250 Hz for convenience during the generation process. We investigate two types of ECG synthesizes, including one beat (one QRS complex) and 10-second signals. First, with case one beat synthesis, we use only the MIT-BIH database (lead II) and divide the dataset into four categories: **N** (Normal beat), **V** (Ventricular beat), **S** (Supraventricular beat), **F** (Fusion of the ventricular and normal beats). The dataset is split into two subsets, including DS1 and DS2 like Hossain et al. research [1]. If DS1 is used for training, DS2 is utilized for inference. In the case using DS1 and DS2 for training, we take N, S, V, and F samples and divide this into 80% and 20% for training and testing, respectively. Second, in the case of the long ECG synthesis, we utilize both Physionet and Chapman datasets to synthesize ECG signals. Due to the lack of rhythms on the atrioventricular block, our study focuses on three rhythms (sinus rhythm, ventricular tachycardia, and supraventricular tachycardia). Based on the notes from the annotation, N, VT, and SVT symbols denote sinus rhythm, ventricular tachycardia, and supraventricular tachycardia, respectively; we spit these rhythms into 10-second ECG signals to make up the data training of our GAN model. Total samples collected from two datasets for N, VT, and SVT include 2782 N, 252 VT, and 635 SVT. As a result, we have a Synthetic Minority Over-sampling Technique (SMOTE) to augment data.

3.2 Evaluation Metrics

To evaluate morphology similarity between the original ECG and ECG synthesis in case one beat, we use four various metrics: Mean Squared Error (MSE), Normalized Mean Squared Error (NRMSE), Structural Similarity Index (SSIM), and Cross-correlation coefficient. It is important to note that the best performance means lower MSE and RMSE and higher SSIM and Cross-correlation coefficient. Besides, we also use the CNN classification model for beat classification tasks with adversarial data made up as the training dataset, and MIT-BIH data is utilized for testing. Accuracy (ACC), Sensitivity(SEN), and Specificity(SPEC) metrics are considered for analysis and comparison with previous studies.

3.3 Setup Hyper-parameters

As described, our GAN model relied on WGAN-GP. We set the penalty coefficient $\lambda = 10$ in Eq. 3. The input vector noise of length 100 is sampled from a standard normal distribution. In this study, the Adam optimizer is used to train the model, employing a learning rate of 0.0002 with $\beta_1 = 0.5$ and $\beta_2 = 0.999$. We used Pytorch 1.11 to train the model in mini-batches with a batch size of 32 in 400 epochs on a GeForce RTX 2070 SUPER GPU.

4 Results and Discussion

4.1 Results on One Beat ECG Synthesis

Table 1. Comparing performance of ECG generator on one beat synthesis with previous research for MIT-BIH.

Dataset	MSE	RMSE	SSI	CCC
DS1 [1]	0.00038	0.0136	0.9982	0.9986
DS1 (ours)	**0.00032**	**0.00107**	**0.9981**	**0.9985**
MIT [1]	0.00241	0.0292	0.9977	0.9425
MIT (ours)	**0.00179**	**0.0237**	**0.9978**	**0.9546**

MIT: DS1+DS2

After training using the DS1 data part from the MIT-BIH dataset with our GAN model, the extracted signal from the image is compared with the types of beats from the DS2 part. Table 1 indicates the results when evaluating with MSE, NRMSE, SSIM, and Cross-correlation coefficient. Table 1 demonstrates that our adversarial examples generated from DS1 and DS1 + DS2 training datasets outperform previous work [1]. Significantly lower MSE and NRMSE scores, as well as greater SSIM and Cross-correlation coefficients, were found in both test scenarios. In Fig. 3, we provide real and generated ECG signals for each class for comparison and visualization. The graphic expositions that the synthesized signals are highly vibrant and lifelike visually. Comparison with GAN models that work on time domain LSTM-GAN [8], WGAN-GP 1D [9] TTS-GAN [4], Fig. 4 shows that our one-beat ECG synthesis has the same performance as these three studies. Specifically, the amplitudes of all QRS complexes have been normalized in the range [0, 1] for convenience in the comparison. The generated QRS durations in these four models are reasonable, with

values of approximately 0.8 s. However, there are still some minor limitations. While the one of LSTM-GAN reveals significant fluctuations, the one of WGAN-GP 1D illustrates more fluctuation during the post-QRS-complex duration. The TTS-GAN approach seems trivial in the one-beat synthesis, whereas this method needs more verification while working with the longer ECG signal generation.

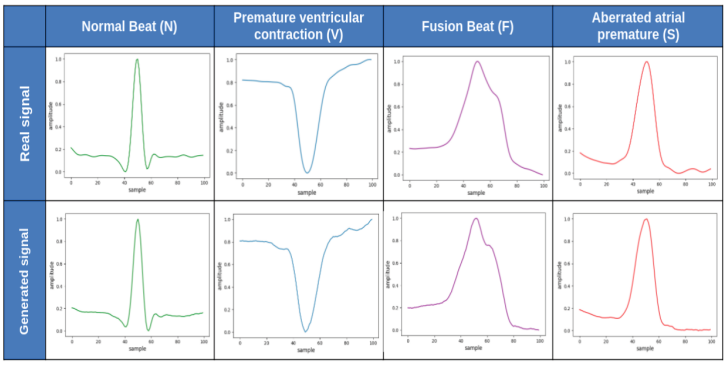

Fig. 3. Visualize a one-beat ECG synthesis on N, S, V, F types.

We also establish a classifier based on the Discriminator architecture of [1] to assess the efficient model and robustness on out-of-distribution and real-world data, which the adversarial scenario mimics for one-beat ECG synthesis. The network takes ECG signals as input and classifies the four categories (N, S, V, F) as output. Specifically, convolution is used four times for downsampling. Except for downsampling convolution, which uses a stride of two, we use a kernel size of three, a stride of one, and padding of one. Following the convolution layer are the Batch-normalization and Leaky-ReLU layers. Following that, we employ three dense layers. Eight convolution layers and two dense layers make up the encoder as described in [1]. Finally, we utilize the Softmax activation function as output for the prediction.

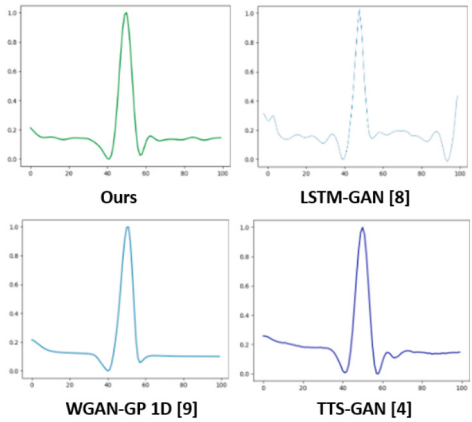

Fig. 4. Compare our generated ECG one beat with other GAN-based models.

We would train and test this model on different datasets to evaluate the influence of generated signals on neural network architecture. Table 2 reveals that our adversarial samples have promoted the overall performance of the architecture. Specifically, while testing the fake dataset with models trained by 80% MIT or

DS1, the evaluated performance illustrates the approximate values, or in other words, slightly higher compared with the one tested with 20% MIT or DS2. This result reveals that our produced signals are similar and can replace the real ones in the training and testing processes. Besides, since all of these adversarial data mimics manually chosen one-beat ECG signals, nearly no noise would be generated, which would be helpful during the classification process. While combining the available training datasets with the fake ones, we noticed that the evaluation metrics of models improved significantly by approximately 0.3% in all of the values. This fact means the generated data has been successful in the augmentation procedure and model performance promotion.

Table 2. Classification results with different training and testing datasets

Training Data	Testing Data	ACC	N		S		V		F	
			SEN	SPEC	SEN	SPEC	SEN	SPEC	SEN	SPEC
80%MIT	**20%MIT**	99.31	99.28	94.36	94.56	99.64	94.93	99.94	95.54	99.54
	FAKE	**99.45**	**99.37**	**95.02**	**94.91**	**99.6**	**95.47**	**99.95**	**95.68**	**99.73**
DS1	**DS2**	97.72	95.87	96	88.19	99.6	93.75	99.64	90.9	96.53
	FAKE	**98.04**	**95.97**	**96.32**	**88.24**	**99.63**	**94.05**	**99.73**	**91.07**	**96.85**
TF1	**20%MIT**	**99.78**	**99.43**	**96.48**	**95.95**	**99.89**	**95.75**	**99.98**	**96.67**	**99.71**
TF2	**DS2**	**98.32**	**96.01**	**96.43**	**89.21**	**99.64**	**94.14**	**99.73**	**91.06**	**96.73**

MIT: DS1+DS2, **TF1**: 80%(DS1+DS2)+FAKE, **TF2**: DS1+FAKE

4.2 Results on Long ECG Synthesis

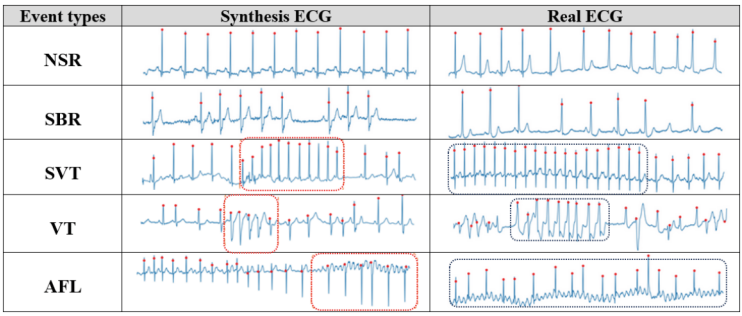

Fig. 5. 10 s ECG synthesis examples for normal sinus (NSR), sinus bradycardia rhythm (SBR) supraventricular tachycardia (SVT), ventricular tachycardia (VT) and atrial flutter (AFL) rhythms

Figure 5 portrays the likenesses and dissimilarities between real and synthesized signals by detecting the R peak position of each with the PanTompkins

algorithm. In summary, Fig. 5 illustrates some comparison of various rhythms, including normal sinus (NSR), sinus bradycardia rhythm (SBR), supraventricular tachycardia (SVT), ventricular tachycardia (VT), and atrial flutter (AFL) rhythms. The morphology shown in Fig. 5, the QRS complexes in the generated data, may be smoothly noticed, just like valid data. Furthermore, each data set accurately emphasized the different properties of event types. For instance, the AFL rhythm characters have a sawtooth appearance. A succession of V beats at higher heart rates causes the VT event. The created SVT data, with shorter RR intervals during the event, is also comparable to the genuine data, with shorter RR intervals.

A well-evaluated long ECG signal should include the acceptable P and T waves. However, after rebuilding and verification on models for 10-second ECG synthesis, a large amount of noise appears when testing with the work [8,9]. Figure 6 also compares the difference in the performance of our proposals with other architectures, including S12-ECG-GAN [7], TTS-GAN [4] and DDPM [11]. In terms of S12-ECG-GAN, the main problem is that the generated 10-second signals are formed from the same 13 representative beats attached. As a result, the generated signals in this framework have no diversity in RR intervals and morphological features. TTS-GAN seems to address such a problem, but the shapes of P and T waves are still unacceptable, with sharp peaks and illogical inclinations. The DDPM result with clear P and T morphology is acceptable like the real one. However, when we conducted this experiment again, the performance to synthesize one sample took approximately 2 min, which required nearly 30 s for our approach with the same heart rate and shape on one long ECG sample. In short, to compare the synthesis rate and output morphology on ECG generation, our suggestion has demonstrated the results outperformance of previous research.

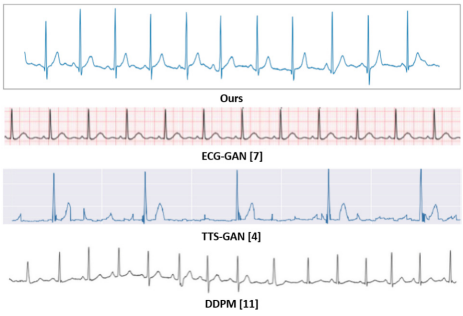

Fig. 6. Comparison 10 s ECG synthesis between our proposal with previous works

5 Conclusion

This paper proposes a new approach that works on the image domain rather than the time domain for the long ECG synthesis. To generate 10-second ECG signals, we only utilize a baseline simple GAN (DCGAN). This novel idea has been handled in synthesizing particular features inside ECG, such as P and T waves. Our models cannot validate performance for atrioventricular block rhythms since none exist in the dataset. However, we have succeeded in 10-second ECG synthesis for sinus, ventricular tachycardia, or supraventricular tachycardia rhythms. In terms of creating realistic ECGs, our technique consistently outperforms state-of-the-art benchmarks. In the future, we may look at merging the differential

privacy framework into our method to develop adaptable ECGs that can govern multiple frequencies and amplitudes.

Acknowledgments. This research is funded by Vietnam National University Ho Chi Minh City (VNU-HCM) under grant number DS2024-26-05.

References

1. Hossain, K.F., et al.: ECG-Adv-GAN: Detecting ECG adversarial examples with conditional generative adversarial networks. In: 2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA), pp. 50–56, IEEE (2021)
2. Rajpurkar, P., Hannun, A.Y., Haghpanahi, M., Bourn, C., Ng, A.Y.: Cardiologist-level arrhythmia detection with convolutional neural networks. arXiv preprint arXiv:1707.01836 (2017)
3. Creswell, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B., Bharath, A.A.: Generative adversarial networks: an overview. IEEE Signal Process. Mag. **35**(1), 53–65 (2018)
4. Li, X., Metsis, V., Wang, H., Ngu, A.H.H.: TTS-GAN: a transformer-based time-series generative adversarial network. arXiv preprint arXiv:2202.02691 (2022)
5. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of StyleGAN. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8110–8119 (2020)
6. Su, J., Jin, Z., Finkelstein, A.: HIFI-GAN-2: studio-quality speech enhancement via generative adversarial networks conditioned on acoustic features. In: 2021 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA), pp. 166–170, IEEE (2021)
7. Zhang, Y.-H., Babaeizadeh, S.: Synthesis of standard 12-lead electrocardiograms using two-dimensional generative adversarial networks. J. Electrocardiol. **69**, 6–14 (2021)
8. Brophy, E.: Synthesis of dependent multichannel ECG using generative adversarial networks. In: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, pp. 3229–3232 (2020)
9. Adib, E., Fernandez, A.S., Afghah, F., Prevost, J.J.: Synthetic ECG signal generation using probabilistic diffusion models. IEEE Access **11**, 75818–75828 (2023)
10. Vo, K., et al.: P2E-WGAN: ECG waveform synthesis from PPG with conditional wasserstein generative adversarial networks. In: Proceedings of the 36th Annual ACM Symposium on Applied Computing, pp. 1030–1036 (2021)
11. Tran, D.T., Tran, Q.N., Dang, T.T.K., Tran, D.H.: A novel approach for long ECG synthesis utilize diffusion probabilistic model. In: Proceedings of the 2023 8th International Conference on Intelligent Information Technology, pp. 251–258 (2023)
12. Alcaraz, J.M.L., Strodthoff, N.: Diffusion-based conditional ECG generation with structured state space models. Comput. Biol. Med. **163**, 107115 (2023)
13. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434 (2015)
14. Moody, G.B., Mark, R.G.: The impact of the MIT-BIH arrhythmia database. IEEE Eng. Med. Biol. Mag. **20**(3), 45–50 (2001)

15. Zheng, J., Zhang, J., Danioko, S., Yao, H., Guo, H., Rakovski, C.: A 12-lead electrocardiogram database for arrhythmia research covering more than 10,000 patients. Sci. Data **7**(1), 1–8 (2020)
16. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein GAN, vol. 30, no. 4,arXiv preprint arXiv:1701.07875 (2017)
17. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.C.: Improved training of wasserstein GANs. In: Advances in Neural Information Processing Systems, vol. 30 (2017)