Proc. of the International Conference on Electrical, Computer and Energy Technologies (ICECET)
9-10 December 2021, Cape Town-South Africa

ECG Synthesis with Neural ODE and GAN Models

Mansura Habiba
Cloud Solution Lead
IBM in Ireland
Dublin, Ireland
0000-0001-9051-1370

Eoin Borphy
School of Computing & INFANT Research Centre
Dublin City University
Dublin, Ireland
0000-0002-6486-5746

Barak A. Pearlmutter
Department of Computer Science & Hamilton Institute
Maynooth University
Maynooth, Ireland
0000-0003-0521-4553

Tomas Ward
School of Computing & Insight Centre for Data Analytics
Dublin City University
Dublin, Ireland
0000-0002-6173-6607

***Abstract*—This paper uses Neural ODE (NODE) based models to generate continuous medical time series. We also introduce a new technique to design the generative adversarial network (GAN) with Neural ODE (NODE) based Generator and Discriminator. During the Performace evaluation of the proposed model, we find that data generated by a NODE-based generator is more continuous than traditional GAN. Therefore, the proposed GAN model becomes more robust. On the other hand, the traditional GAN model demonstrates unstable training and unsupervised learning, which often make it difficult to determine the quality of the result. For this work, we design NODE based models as a generator and a discriminator to make the GAN model datadriven. The data-driven approach helps us overcome the unstable training of the traditional GAN model and improve the quality of the result. We used different evaluation metrics to quantitatively assess generated synthetic data for real-world applications and data analysis. We also evaluate the existing GAN model and the proposed models to understand the comparative efficiency of medical data synthesis.**

***Index Terms*—Neural Network, Generative Adversarial Network , Neural ODE (NODE), ECG Synthesis, Time-series generation**

I. Introduction

Continuous medical time-series data, such as ECG, is quite complex. ECG signal often exhibits dynamic characteristics, such as irregularity with informative missingness [2], high dimension, multi-variate, high frequency or irregular data sampling rate and dynamic pattern in the dataset. In addition, real data in medical research is often unavailable due to its sensitive nature, privacy concerns, and legal restrictions. As a result, there is a great interest in generating realistic continuous medical time series. Generative adversarial networks (GANs) have shown considerable performance for continuous medical time series generation. Most works on medical data generation, including ECG synthesis [13; 15; 16], are mainly driven by the GAN model and its variants. On the other hand, some recent works on NODEs[4; 10; 11; 14; 18] demonstrate its strength against informative missingness and high dimensional dynamic characteristics of continuous-time series. Therefore, this work is based on the hypothesis that NODE based neural network can improve performance for generating continuous time series as it considers the training data as continuous time data instead of discrete sequence of data.

Continuous medical time series is very susceptible and require additional security. Using real data in medical research is often challenging. We introduced NODE and Generative Adversarial Network (GAN) models to generate continuous medical time series data, e.g. electrocardiogram (ECG). These generated data can be used instead of actual ECG without compromising the quality of analysis. The main contributions of this work are as follows:

- A unique NODE based models for continuous medical time series generation.
- A new technique to leverage Generative Adversarial Neural Network (GAN) along NODE to improve the performance of GAN model in terms of generating continuous medical time series.

II. Background

Different deep learning and GAN [1; 5; 7; 8] models are leading the research for ECG data generation. Improving the performance of GAN model isa well-known research area. For example, [7; 8] explains how domain-specific knowledge such as personalisation or external stimulation can improve the performance of GAN models. However, training personalised data with ECG data is time-consuming and needs multiple environments for training. On the other hand, [1; 5] adopt the hybrid model concept to design the architecture for GAN. [1; 5] models use Recurrent Neural Networks (RNN) as Generator and Convolutional Neural Networks (CNN) as the Discriminator. The fundamental concept of these models is to consider continuous time series as a sequence of discrete-time steps at a fixed rate instead of a continuous function of time (t). Therefore, the performance of the traditional GAN model is not satisfactory for real-time continuous time series generation tasks. In addition, ECG signals often have a high and irregular sampling rate. The dynamical equations of motion for ECG signal can be described as Ordinary Differential Equations (ODE) [17]. ODE can model ECG signals as a function of continuous-time. They can also model the dynamic characteristics of the heart rate, for example, the mean and

2021 International Conference on Electrical, Computer and Energy Technologies (ICECET) | 978-1-6654-4231-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICECET52533.2021.9698702

standard deviation of the ECG signal and spectral properties such as the LF/HF ratio. Therefore, NODE can provide better performance than GAN. In this paper, we focus on using NODE to reduce the limitation for GAN and other deep learning models for ECG data generation. NODE [3; 12] model provides better results for a continuous-time series generation as it considers the hidden dynamics of training data as a continuous function of time instead of several layers. In addition, an ODE solver can parametrise the hidden state, as shown in Eq. (1).

$$\frac{dh(t)}{dt} = ODESolver(f, h_t, t, \delta_t) \tag{1}$$

Here f is a Neural Network, $h(0)$ is the initial condition of system, the *ODESolver* computes the derivative $\frac{dh(t)}{dt}$ of the output $h(t)$ at time t. This approach of NODE provides faster training time than residual network with constant memory cost instead of linearly increasing memory cost and simpler design for model. In addition, if the *ODESolver* in NODE can leverage the architecture of RNN, it can learn continuous time series in real-time with higher precision [11; 12]. Eq. (2) shows that NODE based on RNN cell [12] can be used as *ODESolver* to compute the output y_t at time t from the initial input y_0 at time t_0.

$$y_t = ODESolver(ODERNNCell, y_0, t, \theta) \tag{2}$$

In this work, we introduced three models for continuous medical time series synthesis. Our goal is to generate high-quality continuous medical data that can reduce medical research limitations for lack of data. In this paper, we mainly focus on ECG synthesis, both Normal Sinus and Arrhythmia. Fig. 1 represents Normal Sinus data and Fig. 2 represents irregular ECG for Arrhythmia. Both ECG signals are continuous in time with a higher data sampling rate.

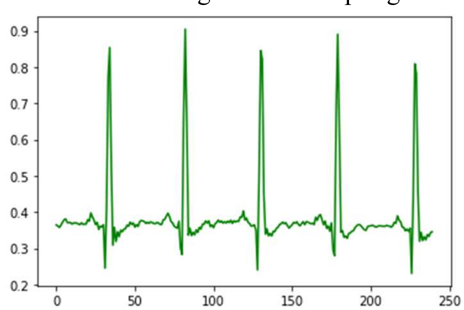

Fig. 1. MIT-BIH Normal Sinus

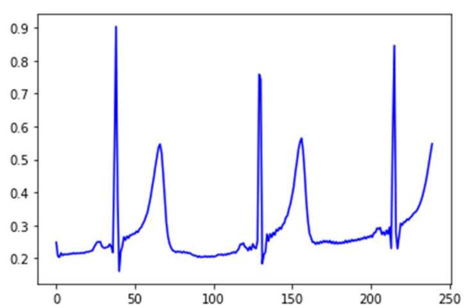

Fig. 2. MIT-BIH Arrhythmia

III. Model Design

In this work, we focused on exploring the strength of NODE for ECG synthesis. Firstly, we designed a NODE based model to produce synthetic continuous-time data, which is described in section III-A. Secondly, we tried to enrich the architecture of the traditional GAN model using NODE. For that, we designed two separate GAN models, where we tried to design the generator and discriminator using NODE.

A. ODEECGGenerator Model

ODEECGGenerator is simply an Neural ODE-RNN [12] NODE model. The building block for this model is a *NeuralODEBlock* that leverage ODE Solver *ODESolver* to train an ODE-RNN (either ODEGRU, or ODELSTM) model. This model learns the dynamics of ECG signal in the form of ODE.

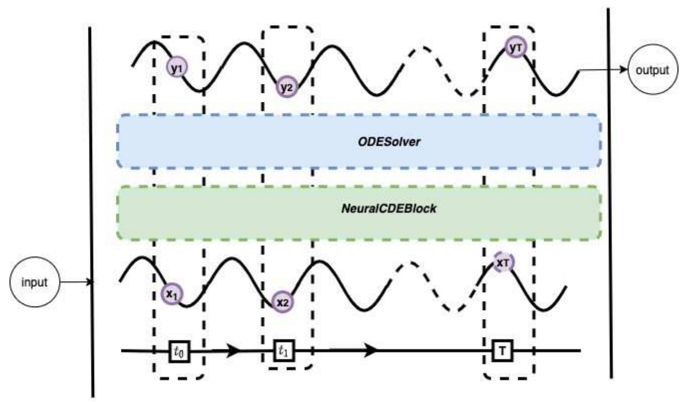

Fig. 3. The architecture for *ODEECGGenerator* model to generate ECG data

Fig. 3 shows that *ODEECGGenerator* receives ECG signal x_t as a continuous function of time (t). Therefore, the input for *ODEECGGenerator* is the state value x_t at a specific time step t The *NeuralODEBlock* in Fig. 3 converts the incoming signal and the hidden state of the system as ODE which is passed to the *ODESolver*. *ODESolver* uses an *ODERNNCell* as shown Eq. (2), which is an ODE-RNN (either *ODEGRU* or *ODELSTM*) model. To solve the ODE within time boundary $[t_0, t]$. *ODERNNCell* is optimized using the parameter γ. The incoming signal contains additional noise (z) to preserve data privacy and keep the model efficient against Adversarial attack. which is also passed as parameter for Eq. (2). Therefore, $\theta = tuple(z, \gamma)$.

$$y_t = ODEECGGenerator(x_t, t) \tag{3a}$$

$$\frac{dy}{dt} = NeuralODEBlock(ODESolver, y_t, t, \theta) \tag{3b}$$

B. ODEGAN Model with ODEGenerator

ODEGAN Model has a NODE generator, called *ODEGenerator*. The discriminator model is similar to the discriminator model used in [5]. Eq. (4b) shows that the *NeuralODEBlock* of *ODEGenerator* uses a *GeneratorFunc* as the neural network (f) for *ODESolver*. *GeneratorFunc* is an recurrent neural network, which can be optimized using parameter θ. Fig. 4 shows the architecture for the *ODEGenerator* generator.

$$y_t = ODEGenerator(y_0, t_0) \tag{4a}$$

$$\frac{dy}{dt} = NeuralODEBlock(GeneratorFunc, y_0, t, \theta) \tag{4b}$$

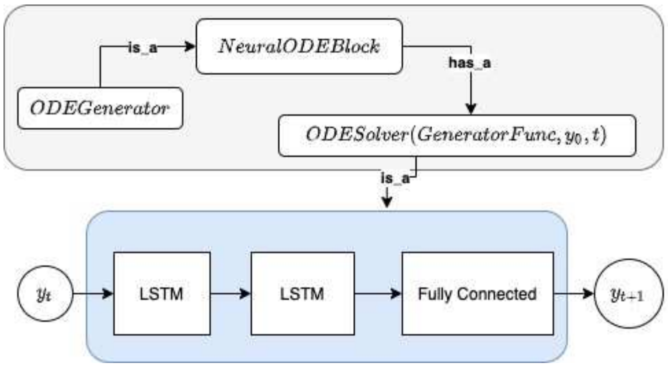

Fig. 4. Block Diagram of Generator Architecture

The Discriminator, as shown in Fig. 5, for this GAN model, is a four-layer 2-dimensional CNN and a Minibatch discrimination layer. The discriminator model also has a noise vector(z) added to the gradient of the optimiser to preserve privacy, as shown in Eq. (4b).

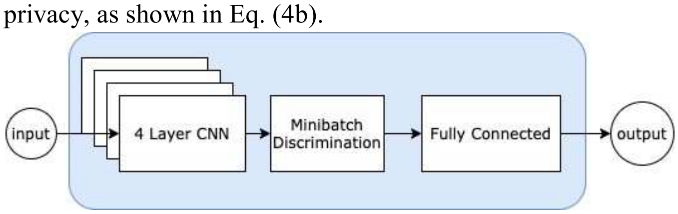

Fig. 5. Block Diagram of Discriminator Architecture

Fig. 6 shows the pipeline for this model. Sample ECG signal y_t is passed to the *ODEGenerator* model. The initial value for the signal is y_0. In order to preserve privacy, additional noise η is added along with parameter (γ) of the *GeneratorFunc*, therefore, $\theta = tuple(z,\gamma)$. NeuralODEBlock converts the

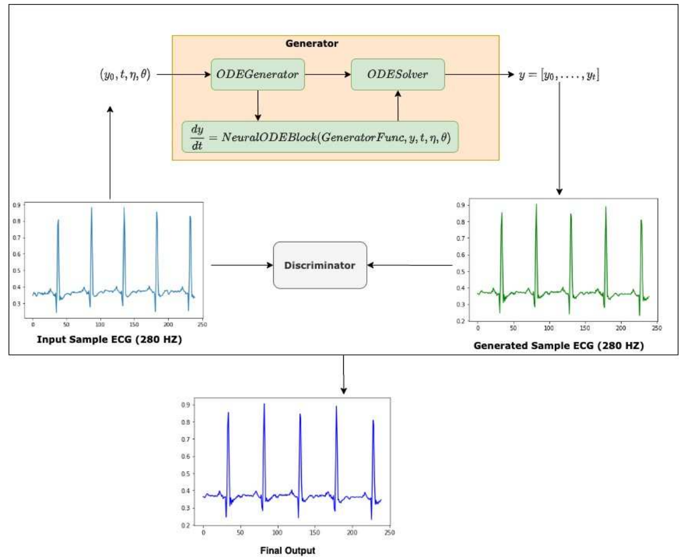

Fig. 6. pipeline for GAN model with ODE generator

signal to ODE, and then *ODESolver* solves the ODE, yielding a generated noisy ECG signal for time T [$t0,\ldots,t$]. The Discriminator then learn to distinguish between the real and the generated signal.

During the training phase, *ODEGenerator* generates continuous time series similar to an actual ECG signal. A log probability (L) matrix evaluates the *ODEGenerator* model by computing L of negative identification of the generated signal by the Discriminator. The optimiser optimises the parameter of the *ODEGenerator* model with a target to minimise L. On the other hand, a cross-entropy loss function evaluates the Discriminator by computing L of correct classification of the natural and generated signal. During the training, the optimiser designated for the Discriminator reduces L to its minimum value.

C. GAN Model with NODE based Generator and Discriminator

For this ODE-GAN-2 model, we designed both Generator and Discriminator using NODE models. The *ODEECGGenerator* model described in section III-A is the Generator for this GAN model. The Discriminator for this model is a NeuralCDE Network [14] which uses a Neural controlled differential equations (CDE) [14] as the *ODESolver* function. The Discriminator leverages the concept of the NeuralCDE network to learn the difference between a real ECG signal, and the generated ECG signalNeuralCDE Network converts the data to a conditional continuous path X using interpolation, and this path X is passed through *ODESolver* to solve the ODE derived from path X to lean the hidden dynamics of the ODE system. Fig. 7 shows that the input time series for this Discriminator has two-channel, e.g. for real ECG signal as well as generated ECG signal.

Fig. 7 shows that the discriminator takes real ECG signal x_t and generated ECG signal y_t as input. Both signals x_t and y_t

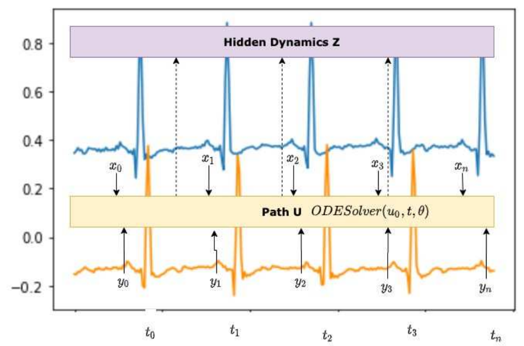

Fig. 7. NeuralCDE based Discriminator

are CDE. Interpolation between x_t and y_t generates an intermediate continuous path *U*. NeuralCDE based Discriminator learns the hidden dynamics (*Z*) of *U* to distinguish real signal and generated signal correctly. Table I describes the parameter used in proposed NeuralCDE based Discriminator.

TABLE I
THE PARAMETERS OF NEURALCDE BASED DISCRIMINATOR

input channels	Length of the Input Sequence hidden dim
	Hidden dimension of the NeuralCDE network output
channels	Output dimension of the NeuralCDE network

IV. PERFORMANCE EVALUATION

The quality and performance of the proposed model are assessed using an experiment with two different tasks, i.e. (i) Generate Normal Sinus ECG and (ii) Generate Arrhythmia ECG. Furthermore, the proposed model is evaluated against state art Neural Network used for ECG Synthesis as described in [5] and NeuralCDE.

A. Dataset

This experiment uses two different kinds of multi-channel publicly available ECG datasets, e.g. (i) MIT-BIH Arrhythmia dataset on PhysioNet [9] and (ii) MIT-BIH Normal Sinus Rhythm Database [9]. Both of the Databases show unique characteristics. For example, MIT-BIH Normal Sinus Rhythm Database consists of clean ECG recordings. As shown in Fig. 1, this Database shows minimal noise in the ECG continuous series. On the other hand, the ECG recordings for MIT-BIH Arrhythmia on PhysioNet were created by adding calibrated noise to clean the MIT-BIH Normal Sinus Rhythm Database. Therefore, the MIT-BIH Arrhythmia dataset contains a significant amount of noise, as shown in Fig. 2.

B. ODEECGGenerator model Training

Table II describes the settings used for performance evaluation. The dynamic characteristics of proposed models enable us to train them quickly and with fewer parameters. For robust training, the dataset was split at random into 80% for training and 20% for the test.

TABLE II
THE PARAMETERS FOR *ODEECGGenerator* MODEL TRAINING

Parameters	ODEECGGenerator	GAN Model
Batch Size	20	64
dataseize	100	1000
Sequence Length	240	240
Hidden Dimension	50	50
Learning rate	0.0001	0.00005
Number of epoch	100	30 * 1000

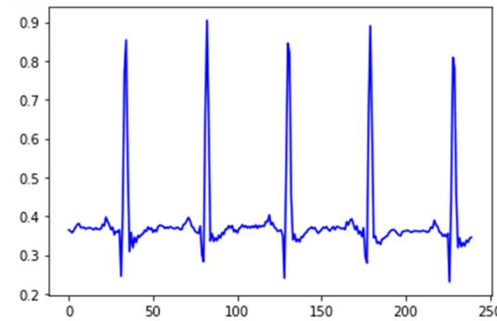

Fig. 8. ODEECGGenerator generated MIT-BIH Normal Sinus Rhythm

Fig. 8 and Fig. 9 show the generated ECG for Normal and Arrhythmia signal by the proposed *ODEECGGenerator* model after training for 100 iterations.

C. GAN model for ECG Synthesis

For comparative analysis, the proposed models are evaluated against the GAN [5] and NeuralCDE [14] models.

Table II describes the parameter used in the implementation of GAN [5] model. Firstly, we compare the GAN model and proposed *ODEECGGenerator* model. GAN model requires a comparatively higher number of parameters than the proposed *ODEECGGenerator* model. The training time for each epoch is also higher in the case of GAN. The GAN model requires a comparatively longer series of data. The training requires three to five hours for 30 epoch in the case of GAN. Fig. 10 shows the generated ECG signal by GAN Model proposed in [5] after training for 1000 iterations.

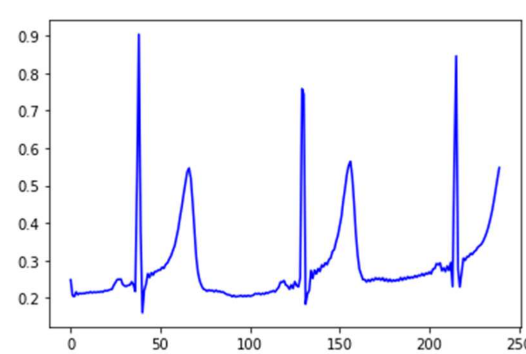

Fig. 9. ODEECGGenerator generated MIT-BIH Arrhythmia Sinus Rhythm

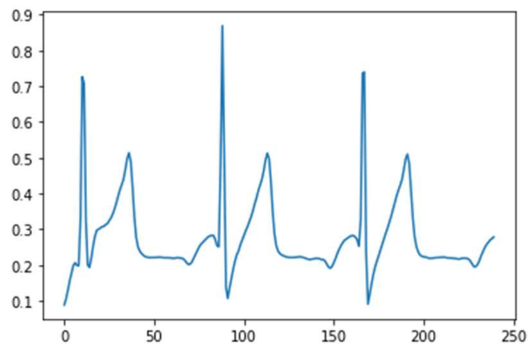

Fig. 10. GAN generated MIT-BIH Arrhythmia signal

The result evaluation between the proposed *ODEECGGenerator* model and the existing GAN models show that NODE enables *ODEECGGenerator* to achieve comparatively higher accuracy with fewer parameters and a concise data series. As a result, the *ODEECGGenerator* model does not need to be trained for longer and is essentially a hidden dimension.

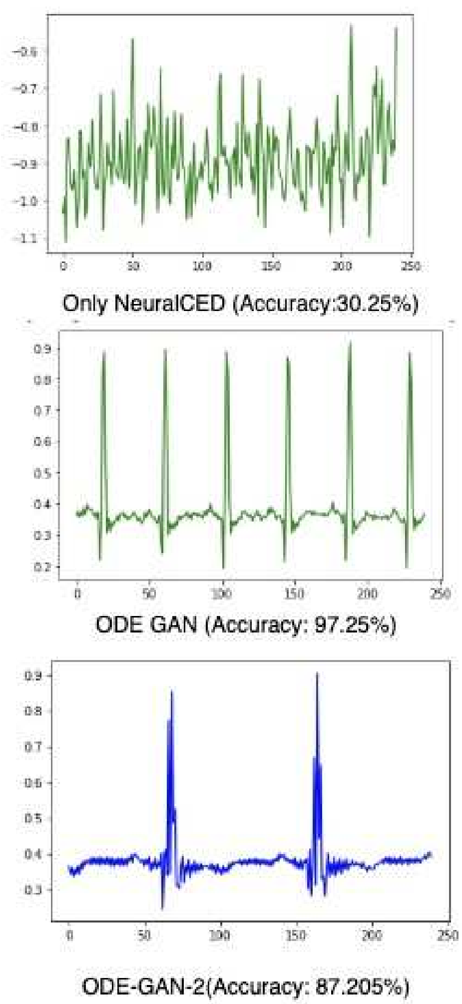

Fig. 11. Generated ECG signals by different models

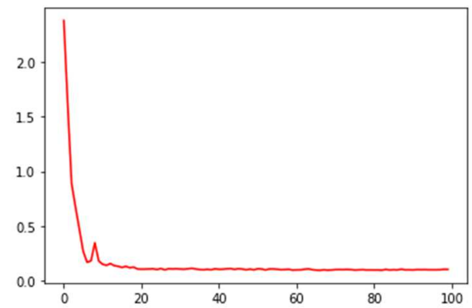

Fig. 12. Training loss for MIT-BIH Normal Sinus Rhythm

Fig. 11 shows the generated ECG by Neural CDE and proposed ODE GAN model with *ODEECGGenerator* as well as ODE GAN model with NODE based generator and discriminator. These generated ECG signals demonstrate that NODE based models can perform significantly better in continuous medical data generation.

D. Comparative Analysis of GAN and proposed models for medical time series generation

Table III shows that proposed models use short sequence length for training data. As a result, the memory for training is constant. On the other hand, the GAN model uses memory in increasing fashion throughout the training. The training also needs less iteration for the proposed model. Therefore, the parameters are fewer than the GAN model.

TABLE III
COMPARATIVE ANALYSIS OF PERFORMANCE

Model	Batch size	Iteration	Seq length	Memory
ODEECGGenerator	50	100	100	constant
ODE GAN	25	100	100	constant
ODE-GAN-2	50	100	100	constant
GAN	32	5000	6647	increasing

The loss shown in Fig. 12 shows that the ODEECGGenerator model can quickly achieve high accuracy for Normal Sinus Rhythm ECG signal. When there is more variant in the time series similar to the Arrhythmia signal, the pattern changes frequently. Arrhythmia signal also takes a comparatively higher time to achieve acceptable accuracy. However, the training time for the ODEECGGenerator model is still significantly lower than the GAN. A similar model, called ECG-ODE-GAN [6] as proposed ODE-Generator GAN model described in III-B tries to learn purely data-driven dynamics from ECG signal. This ECG-ODE-GAN model also shows the impact of physical parameters in morphological descriptors of the ECG signal instances.

For proposed NODE-based models, we observed better performance with batch size > 50. If the batch size is too small, it creates unnecessary noise. The principle of the proposed NODE models is to learn the hidden dynamics of the continuous-time series from the pattern that represents the hidden dynamics most rather than considering each time steps x_t. As proposed NODE based models take the initial state x_0 of any batch as the input, it is redundant to use smaller batch sizes, e.g. 25, 20, 10. For the GAN model, we used a series of lengths 6446, but for the NODE based proposed model, we will use a series of 100 lengths. We observed that length series create instability during the training, and the model's performance dropped significantly. GAN also requires a longer training time. Therefore, we avoid using length time series as training data.

V. Conclusion

We have presented three different models for continuous medical data synthesis. Firstly, we introduce a new technique to learn the hidden dynamics of ECG signals by ODE-RNN only. Secondly, we introduced a generative adversarial network with a NODE Generator and a standard CNN based Discriminator. For the third model, we experimented with a NODE as Generator and NeuralCDE as the Discriminator. Finally, we demonstrated that Neural ODE models could improve the accuracy of standard generative adversarial networks. As NODE models are data-driven generative models, they perform better as the Generator in generative adversarial networks. These hybrid NODE based GAN networks can generate any system that can be described as ODE. The performance evaluation also shows that these models perform better when the batch size for training data is small (30-50) and has fewer parameters. Furthermore, if the pattern in the signal is regular, the train time is lower than the irregular input signal. Proposed models are also suitable for generating systems described by PDEs, as they are data-driven and continuous in time.

References

[1] E. Brophy. Synthesis of dependent multichannel ECG using generative adversarial networks. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 3229–3232, 2020.

[2] Z. Che, S. Purushotham, K. Cho, D. Sontag, and Y. Liu. Recurrent neural networks for multivariate time series with missing values. *Scientific reports*, 8(1):6085, 2018.

[3] Z. Che, S. Purushotham, K. Cho, D. Sontag, and Y. Liu. Recurrent neural networks for multivariate time series with missing values. *Scientific reports*, 8(1):1–12, 2018.

[4] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud. Neural ordinary differential equations. *Advances in Neural Information Processing Systems*, 31:6571–6583, 2018.

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

[6] T. Golany, D. Freedman, and K. Radinsky. ECG ODEGAN: Learning ordinary differential equations of ECG dynamics via generative adversarial learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, number 1, pages 134–141, 2021.

[7] T. Golany and K. Radinsky. PGANs: Personalized generative adversarial networks for ECG synthesis to improve patient-specific deep ECG classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, number 01, pages 557–564, 2019.

[8] T. Golany, K. Radinsky, and D. Freedman. SimGANs: Simulator-based generative adversarial networks for ECG synthesis to improve deep ECG classification. In *International Conference on Machine Learning*, pages 3597– 3606. PMLR, 2020.

[9] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley. PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. *Circulation*, 101(23):e215–e220, 2000.

[10] W. Grathwohl, R. T. Q. Chen, J. Bettencourt, I. Sutskever, and D. Duvenaud. Ffjord: Free-form continuous dynamics for scalable reversible generative models. *arXiv preprint arXiv:1810.01367*, 2018.

[11] M. Habiba and B. A. Pearlmutter. Neural ODEs for informative missingness in multivariate time series. *arXiv preprint arXiv:2005.10693*, 2020.

[12] M. Habiba and B. A. Pearlmutter. Neural ordinary differential equation based recurrent neural network model. *arXiv preprint arXiv:2005.09807*, 2020.

[13] K. G. Hartmann, R. T. Schirrmeister, and T. Ball. EEG-GAN: Generative adversarial networks for electroencephalograhic (EEG) brain signals. *arXiv preprint arXiv:1806.01875*, 2018.

[14] P. Kidger, J. Morrill, J. Foster, and T. Lyons. Neural controlled differential equations for irregular time series. *arXiv preprint arXiv:2005.08926*, 2020.

[15] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunning-´ ham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4681–4690, 2017.

[16] D. Li, D. Chen, B. Jin, L. Shi, J. Goh, and S.-K. Ng. Mad-gan: Multivariate anomaly detection for time series data with generative adversarial networks. In *International Conference on Artificial Neural Networks*, pages 703–716. Springer, 2019.

[17] P. E. McSharry, G. D. Clifford, L. Tarassenko, and L. A. Smith. A dynamical model for generating synthetic electrocardiogram signals. *IEEE transactions on biomedical engineering*, 50(3):289–294, 2003.

[18] Y. Rubanova, R. T. Q. Chen, and D. Duvenaud. Latent ODEs for irregularly-sampled time series. *arXiv preprint arXiv:1907.03907*, 2019.