The Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19)

PGANs: Personalized Generative Adversarial Networks for ECG Synthesis to Improve Patient-Specific Deep ECG Classification

Tomer Golany
Technion-Israel Institute of Technology
Tomer.golany@cs.technion.ac.il

Kira Radinsky
Technion-Israel Institute of Technology
Kirar@cs.technion.ac.il

Abstract

The Electrocardiogram (ECG) is performed routinely by medical personnel to identify structural, functional and electrical cardiac events. Many attempts were made to automate this task using machine learning algorithms including classic supervised learning algorithms and deep neural networks, reaching state-of-the-art performance. The ECG signal conveys the specific electrical cardiac activity of each subject thus extreme variations are observed between patients. These variations are challenging for deep learning algorithms, and impede generalization. In this work, we propose a semi-supervised approach for patient-specific ECG classification. We propose a generative model that learns to synthesize patient-specific ECG signals, which can then be used as additional training data to improve a patient-specific classifier performance. Empirical results prove that the generated signals significantly improve ECG classification in a patient-specific setting.

1 Introduction

Patient-specific modeling (PSM) is the development of computational models of human pathophysiology that are individualized to patient-specific data (Neal and Kerckhoffs 2010). In the last years, PSM has been gaining attention from the research community due to the potential to improve diagnosis. Today, medical diagnosis is based on rough averages derived from clinical trials which might not apply directly to a specific patient condition (Kent and Hayward 2007). PSM can be used to optimize an individual's diagnosis and reach higher accuracies.

In this work, we focus on PSM in the field of cardiology. Electrocardiography (ECG) is a non-invasive tool used for diagnosis and follow-up of cardiac anomalies, functional disorders and cardiac arrhythmias. Any anomaly regarding the heart rhythm or the morphological pattern of the cardiac heart beats as sampled by the ECG, can indicate acute functional emergencies such as acute Myocardial Ischemia or acute rhythm disturbances, meaning an arrhythmia. With the ongoing shortage of trained cardiologists and with the widespread use of home ECGs to monitor patients with cardiac risks, a need to develop fully automated ECG classification mechanisms has arisen. Many studies were conducted in an attempt to reach high performance in ECG classification (Kass and Clancy 2006). Such models were used to reduce interpretation errors by identifying life-threatening arrhythmias or as systems to alert about potential risks to patients.

Most methods today for ECG classification focus on applying classical supervised machine learning methods (Chazal and Reilly 2007; Ye, Kumar, and Coimbra 2012; Escalona-Moran et al. 2014) and manual feature engineering. Lately, deep learning models (Al Rahhal et al. 2016) were successfully applied and reached state-of-the-art results reducing the need for feature engineering. However, in practice, neither of these methods have been able to scale well across different patient's types of ECGs (Kiranyaz, Ince, and Gabbouj 2016). The changing nature of the ECG signal dynamics and morphological characteristics are significantly different across patients, and strongly depend on the patient's physical condition. Given the high variability between patients and internal-variability of heartbeat classification for same patients, building deep ECG models that might be used in practice has been limited.

To scale deep learning to perform personalized ECG classification, subject-specific labeled examples are needed. However, labeling a sufficient amount of ECG samples for each patient is an infeasible task. In this work, we overcome this sparseness of data by learning to synthetically generate personalized ECG signals of different arrhythmias that exhibit similar morphological characteristics to those of the subject of classification. Those are then used to train deep learning models which are better adapted to the subject. To learn patient-specific signals, we devise a novel generative algorithm – Personalized Generative Adversarial Networks (PGANs). PGANs learn to generate personalized arrhythmia-specific ECG signals without any need for subject-specific ECG labeling. This is achieved by learning to generate ECGs that mimic the patient's atrial and ventricular depolarization and repolarization patterns during the specific arrhythmia, which are learned in an unsupervised way from a few minutes of unlabeled ECG signals of the patient.

GANs (Goodfellow et al. 2014) are a class of machine learning algorithms used in an unsupervised machine learning, usually implemented by two deep networks (a generator and a discriminator). The two networks compete with each other in a zero-sum game framework. The generator attempts to learn a latent representation of a distribution, in a way such that the discriminative network, trained to discriminate between instances from the true data distribution and the ones produced by the generator, will have a high loss. As the ECG signals of the subject are not labeled, training a GAN on a specific type of the patient's arrhythmia class is not possible.

In this work, we present PGANs composed of a generator and a discriminator network per an arrhythmia class trained on a large patient population but optimized using a specialized loss function to mimic the morphology of the subject's cardiac signal. We leverage the fact that an ECG signal is represented as a series of waves due to atrial and ventricular depolarization and repolarization: a P wave followed by a QRS set and finally a T wave. We identify those special waves in the patient's unlabeled ECG signal and optimize the adversarial network to generate waves similar to the arrhythmia as exhibited across all patients but with the special morphological cardiac signal of the subject of classification during his personalized arrhythmia. The generated signals are then used as an additional training data for a deep learner. We present empirical results on gold-standard ECG datasets.

Our contribution in this work is threefold: (1) We present the problem of personalized ECG classification, and present an algorithm requiring no patient-specific *labeled* examples. Specifically, we present PGAN, a personalized adversarial generative algorithm, to generate patient-specific ECG signals by training on arrhythmia present in labeled data over a general population and optimized to mimic the specific patient's morphological cardiac waves. (2) We empirically show that utilizing the synthetically generated personalized ECG instances significantly improves personalized ECG classification using deep learning techniques. To the best of our knowledge, this is the first application, where the instances generated by an adversarial network have been shown to improve supervised classification tasks outside the domain of image synthesis. (3) We share our code online for further research and experimentation: https://bitbucket.org/tomerGolany/ecg_dl

2 Related Work

Extracting interval features and using prior knowledge on the ECG morphology (De Chazal, O'Dwyer, and Reilly 2004) is one of the leading methods for ECG beat-level classification. Each ECG signal is separated to heartbeats using heartbeat detection techniques (Afonso et al. 1999). For each heart beat, features related to the heart beat intervals and ECG morphology are calculated. The combined features are then fed into supervised machine-learning models based on linear discriminants (LDs) (Nasrabadi 2007).

In recent years, the application of deep learning models to ECG classification has become popular. It was applied for numerous tasks, such as Cardiologist-Level Arrhythmia Detection (Rajpurkar et al. 2017) and ECG heartbeat classification (Güler and Übeylı 2005; Prasad and Sahambi 2003). One of the best methods today for ECG heartbeat-level classification is described by (Al Rahhal et al. 2016). They leverage a denoising autoencoder (DAE) to learn features in an unsupervised way from the training data and then use a large number of hidden layers and neurons at each layer to learn a sparse representation of the input. This sparse representation of each signal is fed through a softmax layer in order to classify the signal into one of the five possible beats. We show that superior results are reached by applying a simple LSTM model on ECG gold-standard dataset. Due to its simplicity, superior performance and success in other fields involving temporal sequences (Venugopalan et al. 2015), (Gers, Schraudolph, and Schmidhuber 2002),(Sak, Senior, and Beaufays 2014) we utilize this model as the leading deep model for ECG classification. Patient-specific ECG classification have been addressed in the last few years (Kiranyaz, Ince, and Gabbouj 2016; Jiang and Kong 2007) as means of improving ECG classification as it was shown in the past (Kiranyaz, Ince, and Gabbouj 2016) that the variability between patients is too high for most supervised model to reach satisfactory results on a per-patient basis. The common methods for patient-specific ECG classification require significant labeling of beats of the ECG of the specific patient. However, such labeling still requires a cardiologist making it hard to scale for the growing population of home ECG users and adapt to the changing in-patient morphology. In our work, we present a framework that requires no patient-specific labeling of data. We first learn to synthetically generated patient-specific ECG via a novel GAN-based framework. We first adapt GAN to work produce native ECG signals and then show how to adapt it to a specific patient morphology. Those are then used to train an LSTM model for the specific patient.

3 ECG Background

Electrocardiography (ECG) is a non-invasive tool used for diagnosis and follow up of cardiac anomalies, functional disorders and cardiac arrhythmias. Each beat of the heart can be observed as a series of deflections away from the baseline on the ECG. These deflections reflect the time evolution of electrical activity in the heart which initiates muscle contraction. A full heart beat is also known as cardiac cycle, the period of time that begins with contraction of the atria and ends with ventricular relaxation. A single sinus (normal) cycle of the ECG, corresponding to one heart beat, is traditionally labeled with the letters P,Q,R,S and T on each of its turning points (see figure1. Each such point has unique patterns which define the cardiac cycle, as described by (De Luna, Batchvarov, and Malik 2006). In normal resting hearts, the physiologic rhythm of the heart is normal sinus rhythm (NSR). Normal sinus rhythm produces the prototypical pattern of P wave, QRS complex, and T wave. Generally, deviation from normal sinus rhythm is considered a cardiac arrhythmia. The average length of all type of heartbeats is 600 ms, 200 ms before the R peak and 400 ms after. Cardiac cycles can be partitioned to different classes which can represent either normal heart beats or arrhythmias. The P,Q,R,S,T wave patterns varies between classes of arrhythmias. We will focus on 3 type of heart-beats: Supraventricular ectopic beats, Ventricular ectopic beats and Fusion beats.

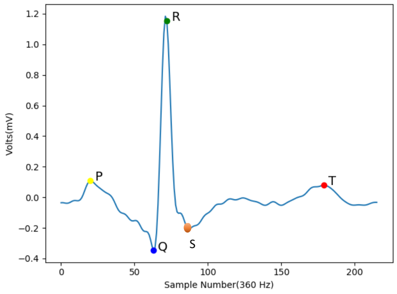

Figure 1: An ECG waves with P,Q,R,S and T waves.

Feature Extraction

There are many techniques to analyze an ECG signal, detect its cardiac cycles and extract the P, Q, R, S, T features from each cardiac cycle. We used a common method implemented in NeuroKit (Makowski 2016). This method is based on filtering and finding local maximals in different areas around each cardiac cycle. We first detect R-peaks as suggested by (Hamilton and Tompkins 1986), and identify the P, Q, R, S, T waves of each cardiac-cycle by finding local maximals and minimals around each R-peak.

Preprocessing

Before extracting cardiac-cycles and its wave values from an ECG signal, preprocessing the whole signal in order to remove noise is necessary. Among all proposals for reducing noise in ECG signals, the simplest and most widely used is the application of recursive digital filters of the finite impulse response (FIR) (Luz et al. 2016).

4 Personalized Generative Adversarial Nets (PGANs) for ECG

In this section, we introduce the framework of ECG GAN optimized for a specific patient ECG signal. We first present a general framework for ECG signal generation using GANs, adapted to the domain of ECG generation. One of the difficulties of creating realistic ECG is sustaining a natural medical cardiac morphology. We therefore devise a novel loss function for the task utilized by the generator. We discuss the details of implementation and optimization of the generator and the discriminator. We call this adapted GAN framework – *ECG GAN*. We then present the Personalized ECG Generative Adversarial Network (PGAN) that extends the ECG GAN with morphological signals derived from patient-specific unlabeled data. The generated ECG signals are then used to train a deep network (Section 5). We empirically show (Section 7) that the additional generated labeled examples significantly improve the ECG classification in a patient-specific setting.

ECG GAN Framework

We formulate the generative adversarial nets for ECG heart beats generation as follows. Let $HB(n) = [hb_1, \ldots, hb_N]$ be an ECG signal taken from a patient in one lead, sliced to heartbeats (cardiac cycles), where $hb(i) = \{v_1, \ldots, v_{216}\}$ represents the voltage values of a single heart-beat. The 216 points represent 200 ms before the R-peak and 400 ms after the R-peak, where the sampling rate is 360 samples per second. All heart-beats are taken from the same type of arrhythmia. For a given heart-beat hb_i, we define $W(hb_i) = [P_i, Q_i, R_i, S_i, T_i]$ to be the wave values of the cardiac cycle. We denote the underlying beats distribution of a given class as a conditional probability $p(hb|Y; W(hb))$, which reflects the distribution of heart-beats given that it is taken from arrhythmia type Y and has feature wave $W(hb)$. Given a set of heart-beats $\{hb_1, \ldots hb_K\}$ taken from P patients, we aim to learn the two following models:

Discriminator As in classic GANs architecture, the discriminator $D(G(z), hb; \theta_D)$, aims to discriminate between a real heart-beat hb and a generated heart-beat $G(z)$ from the generator. $D(hb)$ outputs a single scalar representing the probability that hb came from a real patient rather then p_g.

Generator $G(hb|z; \theta_G)$, which tries to approximate the underlying true ECG heart beat distribution $p(hb|Y, W)$ and generates ECG heart beats with waves similar to W and most likely from class Y. We alter the classic GANs Generator's architecture with a specialized loss function to generate waves similar to natural wave features (P, Q, R, S, and T values) exhibited in the training data.

Generator G and discriminator D act as two opponents: generator G would try to fit $p(hb|Y, W)$ perfectly and generate relevant heart-beats similar to real heart-beats of the desired class to deceive the discriminator, while discriminator D, on the contrary, would try to detect whether these heart-beats are real heart-beats from the data or the ones generated by its counterpart G. Formally, G and D are playing the following two-player $minimax$ game with value function $V(G, D)$:

$$min_G max_D V(D, G) = E_{x \sim p_{data(x)}}[logD(x)] + E_{z \sim p_{data(z)}}[log(1 - D(G(z))] \tag{1}$$

Based on Eq. 1, the optimal parameters of the generator and the discriminator can be learned by alternately maximizing and minimizing the value function $V(G, D)$. In each iteration, discriminator D is trained with positive samples from $p(hb|Y)$ and negative samples from generator $G(hb|Y; \theta_G)$. Each iteration generator G is updated twice with policy gradient under the guidance of D (detailed later in this section) while the discriminator is updated once. Competition between G and D drives both of them to improve their methods until G is indistinguishable from the true connectivity distribution. We discuss the implementation and optimization of D and G as follows.

ECG Discriminator Architecture Given positive ECG heart-beat samples from real patients and negative ECG heart-beat samples from the generator, the objective for the discriminator is to maximize the log-probability of assigning the correct labels to both positive and negative samples. The real heart-beats are taken from 30 minuets ECG records of real patients, sampled at 360 Hz. preprocessing is applied to the ECG signals received as input (as described in Section 3) and they are sliced to single heart-beats.

ECG Discriminator Optimization The ECG heart-beats are then fed through four conventional layers. All weights were initialized from a zero-centered Normal distribution with a standard deviation of 0.02. Between each layer we perform batch normalization and apply a LeakyReLU activation function, where the slope of the leak is set 0.2. The final layer is a sigmoid layer which classifies whether the heart-beat is from the real data or from the generator (fake). Formally, the discriminator loss function is a sum of two cross-entropy functions H:

$$\begin{aligned} &Loss(D) = H(hb_{real}, 1) + H(hb_g, 0) \\ &= [-1 * logD(hb_{real}) - (1-1)log(1 - D(hb_{real}))] + \\ &[-0 * logD(hb_g) - (1-0)log(1 - D(hb_g))] \\ &= -logD(hb_{real}) - log(1 - D(hb_g)) \end{aligned} \tag{2}$$

Where $hb_g \sim p_{data(hb)}$, i.e., hb_{real} is an ECG heart beat taken from the real training data, and hb_g is an ECG heart-beat generated from the generator network, $hb_g = G(z)$, where $z \sim N(\mu, \sigma)$ is a Gaussian distribution with $\mu = 0$ and $\sigma = 1$.

ECG Generator Architecture The ECG GAN Generator architecture differs from that of the classic GAN by attempting to keep natural morphological wave features (P, Q, R, S, and T values), Figure 3 describes the new architecture. The generator input layer draws 100 random numbers from a Gaussian distribution $z \sim N(\mu, \sigma)$ with $\mu = 0$ and $\sigma = 1$, which are then fed to a dense layer. The output of the dense layer is funneled through three de-convolution layers. Between each layer batch normalization (Ioffe and Szegedy 2015) is performed followed by a ReLU activation function. Instead of generating one cardiac cycle, the final layer of the generator is a dense layer with 216*3 neurons which produces an output vector of size 216*3 (step (1)). The output dimensions of the generator corresponds to the same dimensions of three cardiac cycle, i.e., the generator generates three cardiac cycles. We consider only the middle cardiac-cycle as the generated ECG signal (step (2)), and use the two other cycles to calculate the wave features P, Q, R, S, T (Section 3), which are then used in the loss function described below.

ECG Generator Optimization The Generator loss function is defined by a combination of the classical cross-entropy loss, which tries to generate cardiac-cycles that will fool the discriminator, and a Mean Square Error (MSE) function which tries to generate fake heart-beats which are more morphologically similar to real heart-beats. The MSE loss function constrains the generated heart-beat wave values, P, Q, R, S, T to be as close as possible to the wave values extracted from real heart-beats in the training data.

Cross-Entropy Loss: in contrast to the discriminator, the generator aims to minimize the log-probability that the discriminator correctly assigns negative labels to the samples generated by G. Specifically, we optimize the generator network by feeding its output into the discriminator network and optimizing the generator's weights in a way that the discriminator network will predict that the generated ECG heart beat is real. Therefore, we feed the examples from the generator into the discriminator and label them as real (y = 1). We then perform Adam optimization to correct only the weights of the generator. Formally, the cross-entropy loss function of the generator is as following:

$$\begin{aligned} &Loss_H(G) = H((hb_g, 1)) = H((G(z), 1)) \\ &= [-1 * logD(G(z)) - (1-1)log(1 - D(G(z)))] \\ &= -logD(G(z)) \end{aligned} \tag{3}$$

Where $hb_g = G(z)$ is an ECG heartbeat generated from the generator network, and $z \sim N(\mu, \sigma)$ is a Gaussian distribution with $\mu = 0$ and $\sigma = 1$. We set the hyperparameters as suggested by (Radford, Metz, and Chintala 2015).

MSE loss: In order to improve the generator's ability to generate fake heart-beats which are more morphologically similar to real heart-beats, we constrain the generated heart-beat wave values, P, Q, R, S, T to be as close as possible to the wave values extracted from real heart-beats in the training data. The P, Q, R, S, T values are calculated by the algorithm described in section 3. We add a penalty to the loss function for how far the waves from the output of the generator are from the wave values extracted from the training data (Figure 4). We define the MSE generator loss as:

$$\begin{aligned} &Loss_{mse}(G) = MSE(P_{data}, P_g) + \\ &MSE(Q_{data}, Q_g) + MSE(R_{data}, R_g) \\ &+ MSE(S_{data}, S_g) + MSE(T_{data}, T_g) \end{aligned} \tag{4}$$

where, : $MSE(Y, \hat{Y}) = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$, and Y is the vector of observed values of the variable being predicted, and $P_{data}, Q_{data}, R_{data}, S_{data}, T_{data}$ are calculated from the training data.

The total loss function of the generator is the sum of the two above loss functions:

$$Loss(G) = Loss_{mse}(G) + Loss_H(G) \tag{5}$$

The optimization is performed using Adam optimizer (Kingma and Ba 2014) and a learning rate of 0.0002 as suggested for training GAN models in (Radford, Metz, and Chintala 2015). The generator weights are trained twice each iteration while the discriminator is trained once as suggested in (Radford, Metz, and Chintala 2015).

Personalized ECG GAN

To address the variance between different patients which causes lower ECG classification accuracies, we propose an alternative method of training the ECG GAN in a way that it will learn to generate heart-beats which are closer to a specific patient ECG morphology. The two underlying principles of PGANs are as follows:

1. Learning a GAN model *per patient*. Intuitively, we would like to create an ECG GAN that generates waves similar to the wave features (P, Q, R, S, and T values) of the patient. Specifically, we try to extract the wave features as exhibited during the arrhythmia the generator attempts to generate. We suggest a model which leverages unclassified beats of the subject of classification. Those beats are

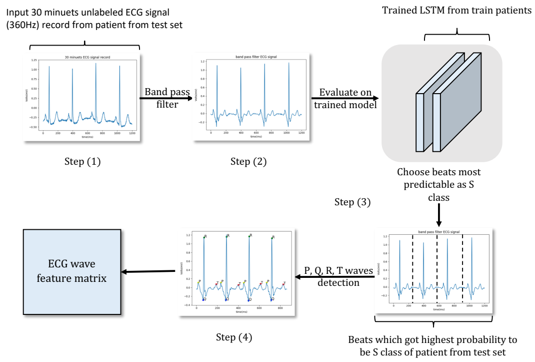

Figure 2: Feature extraction pipeline. Extract wave features of heart beats from specific patients in the test set.

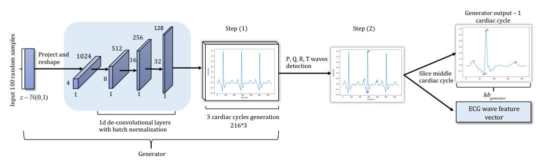

Figure 3: Generator model pipeline. The generator creates three cardiac cycles in order to extract wave features from the middle cardiac cycle

extracted from the beginning of the ECG signal recorded from the patient. We feed the patient's beats to a pre-trained LSTM classifier (see Section 5 for details of the model) tuned for predicting arrhythmia classes over a general patient population. That is, the classifier is trained beforehand using training data over all patients (excluding the specific patient of classification). We then feed to the trained LSTM the heart-beats from the specific patient (Figure 2). We select 50 heart-beats from the specific-patient which were predicted by the LSTM classifier with the highest probability to belong to the class we would like to generate with the generator (step (3)). From those 50 heart-beats we extract the P, Q, R, S, T wave values (step (4)). The values are used in the generator MSE loss.

2. Constraining the generator to produce ECG signals with properties of the specific patient. In order to improve the generator's ability to generate heart-beats which are more close to real heart-beats, and specifically close to the patient's heart beats, we try to constrain the generated heart-beat wave values, P, Q, R, S, T to be close as possible to the wave values extracted from the patient, while still close enough to the real training data, so the discriminator will be fooled. This time the MSE loss of the generator adds penalty to the loss function for how far they are from the wave feature values extracted from the *specific test patient* rather from the training data. By that we approximate the generated cardiac-cycle morphology to the specific-patient morphology (see Figure 4). The generator loss is now:

$$Loss(G) = H(D(G),1) + MSE(P_{subj}, P_g) + MSE(Q_{subj}, Q_g) + MSE(R_{subj}, R_g) + MSE(S_{subj}, S_g) + MSE(T_{subj}, T_g)$$

Where $H(D(G),1)$ is the cross-entropy loss function from before and the second part is mean-square error between the wave values generated from the generator and wave values from *subject's* heart-beats.

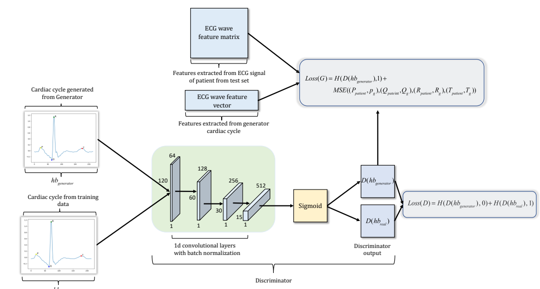

Figure 4: Description of the ECG patient-specific Discriminator and the two parts of its loss function. The discriminator takes as input real heart-beat from the training data and a fake heart-beat from the output of the generator. Each heart-beat is fed through 4 convolution layers. The last layer is a feed-forward layer with sigmoid activation which outputs the probability that the input heart-beat was taken from the real data or from the generator output. The new loss function for the generator is as follows: the first part is the original GAN loss function which is the cross-entropy loss function. The second part is an MSE function which penalizes for how far the wave features of the generated heart beat are from the wave features extracted from those of the subject's.

5 Patient-Specific Deep ECG Classification

In Section 4, we described PGAN – a framework for generating ECG signals that exhibit both a certain arrhythmia and are adapted to a specific patient. In this section, we discuss how those patient-specific ECGs are used to train a deep learner ECG classifier showed in Section 7 to empirically reach better results in patient-specific ECG classification. Figure 5 presents the architecture of the LSTM model, which was found to have superior results on ECG gold-standard dataset. The system is trained over a large corpus of 66000 heart-beats (Section 7). We filter each signal as described in Section 3 (step (2)). The filtering is necessary in order to detect correctly the R-peaks and P, Q, S, T of each cardiac cycle(Described in section 3). and slice the signal to cardiac-cycles around each R-peak (step (3)). The sliced heart-beats are fed to an LSTM classifier (step(4)) with 2 layers of 512 neurons each. The last layer is a softmax layer which classifies the heart-beats to 5 different arrhythmia classes (step(5)). Once the model is trained, we continue its training with the additional generated ECG signals of the patient of classification. We provide several empirical results showing how the classifier performance improves

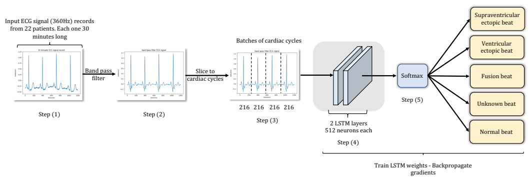

Figure 5: Proposed LSTM model to classify heart-beats to 5 type of arrhythmias.

as a function of the number of the synthesized ECG signals added (Section 7).

6 Experimental Evaluation

ECG Dataset

Our framework consists of ECG recordings taken from the MIT-BIH database (Moody, Mark, and Goldberger 2001). MIT-BIH arrhythmia database is the most popular public dataset in discovering and clustering arrhythmias. It is considered the gold-standard evaluation data for ECG classification tasks (Moody, Mark, and Goldberger 2001). The database contains 48 half-hour ECG records, obtained from patients studied by the BIH Arrhythmia Laboratory between 1975 and 1979. Each record contains two 30-minutes ECG lead signals digitized at 360 samples per second. The database contains annotations for both heartbeat class information and timing information verified by independent expert. A total of 109,492 labeled heartbeats. In this work, we consider three classes of arrhythmia, and neglect two classes – the normal heart beats and the unclassified beats. We excluded those classes from our work as they have low medical value and small amount of training data (below 100).

Experimental Methodology

We follow the dataset partitioning as described by (De Chazal, O'Dwyer, and Reilly 2004) and (Al Rahhal et al. 2016). Both follow the AAMI recommendations for the ECG classification task. This partition makes sure that patients data is not mixed between the training and the testing sets. In agreement with the AAMI recommended practice, the four recordings containing paced beats were removed from the dataset. We train an LSTM model in a leave-one-out fashion, each time extracting a different specific patient for a test. We report the area under the curve (AUC) the ROC using the beats annotations ground truth reported from experts and the scores of each classifier for each class. For the test, we only consider patients with more than 800 heartbeats from the desired class.

Models Evaluated

We present results for the following models classifying one class of arrhythmia:

No ECG Generation LSTM model (Section 5) trained on all patients with no additional synthesized ECG examples.

Non-Personalized ECG Generation

1. **Vanilla GAN** – LSTM model trained on all patients with additional synthesized ECG examples from a classical GAN model.
2. **DCGAN** – LSTM model trained on all patients with additional synthesized ECG examples from a DCGAN (Radford, Metz, and Chintala 2015) model. As DCGAN have shown superior performance on several tasks (Radford, Metz, and Chintala 2015; Yeh et al. 2017) we present results on generated examples from this class of generative adversarial networks.
3. **ECG GAN** – LSTM model trained on all patients with additional synthesized ECG examples from our proposed ECG GAN model (Section 4) but with no adaptation to the specific test patient.

Personalized ECG Generation

1. **Personalized LSTM** – We devised a personalized LSTM model following a transfer learning schema (Donahue et al. 2013). We train the LSTM on all patients and add predicted ECG samples from the specific patient (similarly to the methodology presented in Section 4). On average 800-1000 patient-specific beats are added.
2. **Personalized GAN** – To understand the contribution of learning ECG features morphology during personalized ECG synthesis, we devise the Personalized GAN approach. An LSTM model is trained on all patients with additional synthesized ECG from a personalized GAN model. The GAN model is a vanilla GAN without learning ECG morphology, a done by the ECG GAN. The personalized GAN discriminator is trained on all training patients and in addition predicted ECG samples from the specific patient are added (similarly to the methodology presented in Section 4) (i.e., with additional 800-1000 patient-specific beats).
3. **Personalized DCGAN** – We wish to study the contribution of the specific generator architecture to the classification. DCGAN is considered state of the art for image generation. We therefore devise a model which is similar to Personalized GAN but with a DCGAN architecture. That is, the LSTM is trained with synthesized examples from a personalized DCGAN.
4. **PGAN** – LSTM model trained on all patients with additional synthesized ECG examples from our proposed PGAN model (Section 4).

The number of synthetically generated beats by the generative networks, which were added to the training set, is a parameter of the model. We experimented with the following values: $0, 500, 1000, 5000, 8000, 10000, 15000, 20000$.

7 Results

Figures 6–8, present our results for the 3 arrhythmia classes.

Personalized vs Non-Personalized LSTM

Across all classes the results for the Personalized LSTM reached AUC values significantly lower of that of the Non-

Personalized LSTM. The heartbeat class which had the highest gap of AUC between the non-personalized to the personalized LSTM had a gap of 0.15 and the heartbeat class which had the lowest gap of AUC between the Non-personalized to the personalized LSTM had a gap of 0.1. For all classes the AUC of the non-personalized LSTM was better then the Personalized LSTM. We conclude that adding a small amount of predicted examples from the specific-patient ECG adds too much noise to the LSTM training. Achieving more samples requires longer ECG monitoring which reduces the medical value of such a system. The ability of the generative models to add significant amounts of training examples carries higher promise as shown in next experiments.

The Effect of ECG Synthesis

We study the effect of adding synthesized ECG signals on the classifier ability to distinguish between one heartbeat class versus the rest of the classes. We observe that the LSTM model trained with added synthetic examples from any GAN model significantly outperforms the LSTM model trained without synthetic examples added for the S(supraventricular) and F(Fusion) heartbeats. For the V(ventricular ectopic) heart-beats, we see that the LSTM classifier performed very well without adding synthetic heart-beats to the training set, achieving AUC of 0.99. The LSTM kept the high score when adding synthetic heart-beats from all type of GANs. We observe there is no monotonicity in performance as a function of number of synthesized examples added. There were some events where the synthetic heart-beats achieved lower AUC. For most models we see that the performance first drops and then fluctuates till reaches better performance than the LSTM trained with no synthesized examples. This is due to the randomness of the generator creation of heart-beats. We observe the improvement is more significant in arrhythmia classes, which are composed of several medical conditions. We conjecture that those classes are harder to learn given the same amount of training generated by the PGAN. We conclude that the practice of adding synthesized examples to the training of the LSTM model significantly improves its performance for heartbeat classification when tuning the number of synthesized examples added.

The Effect of Adapting to ECG Morphology

The ECG GAN (Section 4) presents a novel loss functions whose goal is to adapt its output to ECG morphology with natural P, Q, R, S, T waves. We observe that for the F and V heart-beat class, the classification with ECG-GAN synthesized ECG significantly outperforms the classic GAN and DCGAN, but does not perform as well for S beats. In the S beats, the variation between the train and test patients is the highest compared to other beats classes. We conclude that learning to adapt to natural ECG morphology is important for better classification, but the need to adapt to a specific-patient morphology is crucial. PGAN solves this issue by leverages our proposed ECG-GAN model and adapts it to the specific-patient morphology with patient-specific P, Q, R, S, T waves.

The Effect of Personalization

We wish to study the effect of adding personalized versus non personalized ECG signals. The personalized GAN and DCGAN don't outperform the non personalized models across all classes (we only observe a gain on F beats). We observe that PGAN outperforms all non-personalized models across all classes, reaching the following state-of-the-art performance: F beats: 0.95 AUC; S beats: 0.85 AUC; V beats: 0.99 AUC. Additionally, it reaches superior performance compared to all personalized models. We note that the variance of the classification performance is low and goes between 0.01 - 0.05 in the different heart-beat classes. We conclude that adding personalized ECG signals improves ECG classification. However, the method of generating the personalized ECG is of high importance.

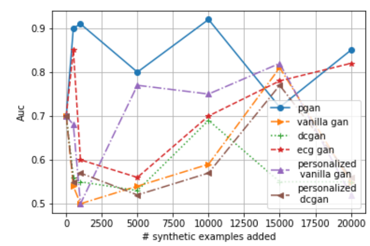

Figure 6: Average AUC comparison on LSTM which classifies heart-beat of type F - Fusion beats

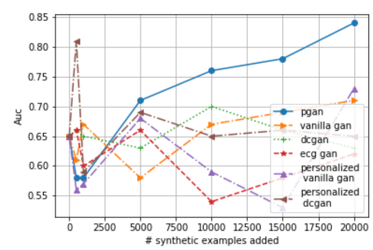

Figure 7: Average AUC comparison on LSTM which classifies heart-beat of type S - supraventricular ectopic beats

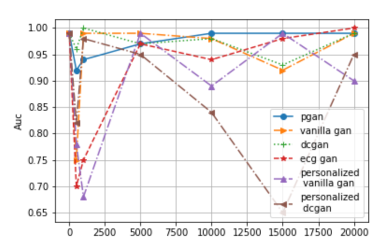

Figure 8: Average AUC comparison on LSTM which classifies heart-beat of type V - ventricular ectopic beats

8 Conclusions

In this work, we study the problem of personalized ECG classification, which is of high importance due to high variability across patients. We present a general framework for generating natural ECG signals by constraining the generative model to produce natural P, Q, R, S, T waves. We evaluate the performance on LSTM classifier for ECG classification and show that it reaches high results compared to non-constrained generative models. We then present PGAN, a personalized adversarial generative algorithm, to generate patient-specific ECG signals by training on arrhythmia present in labeled data over a general population and optimized to mimic the specific patient's morphological cardiac waves. We empirically show that utilizing the synthetically generated personalized ECG instances significantly improves personalized ECG classification using deep learning techniques.

References

Afonso, V. X.; Tompkins, W. J.; Nguyen, T. Q.; and Luo, S. 1999. Ecg beat detection using filter banks. *IEEE transactions on biomedical engineering* 46(2):192–202.

Al Rahhal, M. M.; Bazi, Y.; AlHichri, H.; Alajlan, N.; Melgani, F.; and Yager, R. R. 2016. Deep learning approach for active classification of electrocardiogram signals. *Information Sciences* 345:340–354.

Chazal, P., and Reilly, R. 2007. A patient-adapting heartbeat classifier using ecg morphology and heartbeat interval features. 53:2535–43.

De Chazal, P.; O'Dwyer, M.; and Reilly, R. B. 2004. Automatic classification of heartbeats using ecg morphology and heartbeat interval features. *IEEE transactions on biomedical engineering* 51(7):1196–1206.

De Luna, A. B.; Batchvarov, V. N.; and Malik, M. 2006. The morphology of the electrocardiogram. *The ESC Textbook of Cardiovascular Medicine Blackwell Publishing* 1–35.

Donahue, J.; Jia, Y.; Vinyals, O.; Hoffman, J.; Zhang, N.; Tzeng, E.; and Darrell, T. 2013. Decaf: A deep convolutional activation feature for generic visual recognition. *arXiv preprint arXiv:1310.1531*.

Escalona-Moran, M.; Soriano, M.; Fischer, I.; and R Mirasso, C. 2014. Electrocardiogram classification using reservoir computing with logistic regression. 19.

Gers, F. A.; Schraudolph, N. N.; and Schmidhuber, J. 2002. Learning precise timing with lstm recurrent networks. *Journal of machine learning research* 3(Aug):115–143.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *Advances in neural information processing systems*, 2672–2680.

Güler, İ., and Übeylı, E. D. 2005. Ecg beat classifier designed by combined neural network model. *Pattern recognition* 38(2):199–208.

Hamilton, P. S., and Tompkins, W. J. 1986. Quantitative investigation of qrs detection rules using the mit/bih arrhythmia database. *IEEE transactions on biomedical engineering* (12):1157–1165.

Ioffe, S., and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.

Jiang, W., and Kong, S. G. 2007. Block-based neural networks for personalized ecg signal classification. *IEEE Transactions on Neural Networks* 18(6):1750–1761.

Kass, R., and Clancy, C. 2006. *Basis and Treatment of Cardiac Arrhythmias*. Handbook of Experimental Pharmacology. Springer Berlin Heidelberg.

Kent, D., and Hayward, R. 2007. Limitations of applying summary results of clinical trials to individual patients: the need for risk stratification. *JAMA* 298:1209–1212.

Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kiranyaz, S.; Ince, T.; and Gabbouj, M. 2016. Real-time patient-specific ecg classification by 1-d convolutional neural networks. *IEEE Transactions on Biomedical Engineering* 63(3):664–675.

Luz, E. J. d. S.; Schwartz, W. R.; Cámara-Chávez, G.; and Menotti, D. 2016. Ecg-based heartbeat classification for arrhythmia detection: A survey. *Computer methods and programs in biomedicine* 127:144–164.

Makowski, D. 2016. Neurokit: A python toolbox for statistics and neurophysiological signal processing (eeg, eda, ecg, emg...). memory and cognition lab' day, 01 november, paris, france. https://github.com/neuropsychology/NeuroKit.py.

Moody, G. B.; Mark, R. G.; and Goldberger, A. L. 2001. Physionet: a web-based resource for the study of physiologic signals. *IEEE Engineering in Medicine and Biology Magazine* 20(3):70–75.

Nasrabadi, N. M. 2007. Pattern recognition and machine learning. *Journal of electronic imaging* 16(4):049901.

Neal, M., and Kerckhoffs, R. 2010. Current progress in patient-specific modeling. *Briefings in Bioinformatics* 11(1):111–126.

Prasad, G. K., and Sahambi, J. 2003. Classification of ecg arrhythmias using multi-resolution analysis and neural networks. In *TENCON 2003. Conference on Convergent Technologies for the Asia-Pacific Region*, volume 1, 227–231. IEEE.

Radford, A.; Metz, L.; and Chintala, S. 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*.

Rajpurkar, P.; Hannun, A. Y.; Haghpanahi, M.; Bourn, C.; and Ng, A. Y. 2017. Cardiologist-level arrhythmia detection with convolutional neural networks. *CoRR*.

Sak, H.; Senior, A.; and Beaufays, F. 2014. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In *Fifteenth annual conference of the international speech communication association*.

Venugopalan, S.; Rohrbach, M.; Donahue, J.; Mooney, R.; Darrell, T.; and Saenko, K. 2015. Sequence to sequence-video to text. In *Proceedings of the IEEE international conference on computer vision*, 4534–4542.

Ye, W.; Kumar, B.; and Coimbra, M. 2012. Heartbeat classification using morphological and dynamic features of ecg signals. 59:2930–41.

Yeh, R. A.; Chen, C.; Lim, T.-Y.; Schwing, A. G.; Hasegawa-Johnson, M.; and Do, M. N. 2017. Semantic image inpainting with deep generative models. In *CVPR*, volume 2, 4.