Artificial Intelligence In Medicine 143 (2023) 102583

Contents lists available at ScienceDirect

Artificial Intelligence In Medicine

journal homepage: www.elsevier.com/locate/artmed

Research paper

Domain randomization using synthetic electrocardiograms for training neural networks

Matti Kaisti [a],[*], Juho Laitala [a], David Wong [b], Antti Airola [a]

[a] Department of Computing, Digital Health Lab, University of Turku, Turku 20500, Finland
[b] Department of Computer Science and Centre for Health Informatics, University of Manchester, Manchester, UK

ARTICLE INFO

Keywords:
Deep learning
Electrocardiogram
Domain randomization
Neural network
Synthetic
Explainable AI

ABSTRACT

We present a method for training neural networks with synthetic electrocardiograms that mimic signals produced by a wearable single lead electrocardiogram monitor. We use domain randomization where the synthetic signal properties such as the waveform shape, RR-intervals and noise are varied for every training example. Models trained with synthetic data are compared to their counterparts trained with real data. Detection of r-waves in electrocardiograms recorded during different physical activities and in atrial fibrillation is used to assess the performance. By allowing the randomization of the synthetic signals to increase beyond what is typically observed in the real-world data the performance is on par or superseding the performance of networks trained with real data. Experiments show robust model performance using different seeds and on different unseen test sets that were fully separated from the training phase. The ability of the model to generalize well to hidden test sets without any specific tuning provides a simple and explainable alternative to more complex adversarial domain adaptation methods for model generalization. This method opens up the possibility of extending the use of synthetic data towards domain insensitive cardiac disease classification when disease specific a priori information is used in the electrocardiogram generation. Additionally, the method provides training with free-to-collect data with accurate labels, control of the data distribution eliminating class imbalances that are typically observed in health-related data, and the generated data is inherently private.

1. Introduction

Training neural networks typically requires significant amounts of labelled data that are expensive to collect. This is especially true for healthcare data where expert knowledge is required [1] and open sharing is limited due to privacy concerns [2]. Typically, better predictive performance in deep learning can be achieved by using more data and/or more complex and bigger networks [3].

Even where such large data sets exist for specific clinical data, researchers have consistently found that resulting deep neural network models generalize poorly to external validation data sets [4,5]. One likely reason for this is that noise mechanisms, or other domain-specific but clinically-irrelevant features, are inadvertently learned by the model. One approach for dealing with this is to implicitly train a model to ignore irrelevant features via adversarial learning (adversarial domain adaptation) [6,7]. However, this is only possible in circumstances in which training data are available from multiple domains.

Another approach is to explicitly simulate training data from multiple domains through creating synthetic data, a process sometimes known as domain randomization. This has been shown to be promising in image classification tasks [8–11]. A clear advancement was achieved with domain randomization when photorealism, a requirement of earlier attempts, was abandoned and random perturbations of the environment in non-realistic ways achieved competitive accuracy in testing [9]. The network learned to discriminate between desired and undesired objects by adding randomly different geometric shapes, random textures and random lights into the images. One line of research in health monitoring has shown the benefit of using pre-trained image nets and transfer learning, where the signal is first transformed into an image followed by a fine tuning of the model weights for the final predictive model [12,13]. This removes the need for large application-specific datasets. However, 1D signals are not always well presented as images and either the morphological details or long-term information [14] of the signal is lost. Pre-trained networks are also bounded by the approach chosen during initial training which could be sub-optimal for the task at hand.

Synthetic data is typically generated in one of two ways. The first approach is using generative adversarial network (GAN) [15] where

[*] Corresponding author.
E-mail address: mkaist@utu.fi (M. Kaisti).

https://doi.org/10.1016/j.artmed.2023.102583
Received 27 July 2022; Received in revised form 26 January 2023; Accepted 15 May 2023
Available online 19 May 2023

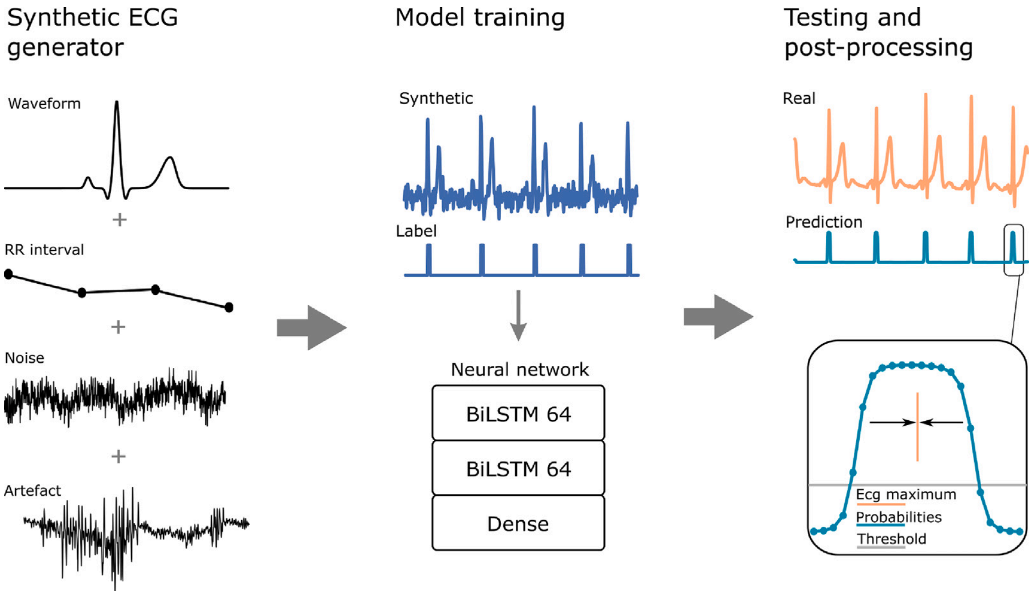

Fig. 1. Principle of synthetic data generation, model training and testing and post-processing step. A mathematical model is used to generate synthetic data where the properties of the waveform shape, RR-intervals and noise can be controlled. Real artefacts are optionally extracted, randomized and added to the synthetic signals. A label array of the same length as the corresponding ECG is used; this array has five ones at the r-wave location and zeros elsewhere. The prediction probabilities are transformed to a single peak location in the post-processing step. The model processes input signals of any length in four second segments and re-constructs a full signal length prediction in post-processing.

the aim is to generate new samples with the same statistics as in the training set. The second approach is model driven where a mathematical model with a priori information can be used to generate synthetic data with desired characteristics. This allows users to have full control and transparency of the generated data. It is particularly important because explanations are needed for reliable and responsible use of deep learning models in high stakes medical decision making. [16] One possible way to gain more understanding of model behaviour is to investigate the interaction between the data and model. With controlled changes in the input data it becomes possible to analyse model outputs and study the models inner workings.

In this work, we describe a synthetic signal generator that is able to produce electrocardiograms (ECGs) where characteristics of these signals can be varied in a controlled manner and in part solve and investigate the above mentioned challenges. Here, ECGs are used to train an LSTM network and we demonstrate the learning through r-wave detection using various real-life recordings in testing. The procedure of signal generation is exemplified in Fig. 1 and comprises the generation of (i) signal waveform, (ii) RR-intervals, (iii) noise process and (iv) augmentation of real data artefact. Synthetic signals with varying degree of domain randomization are fed to the network and final estimate of the detected peak indices is achieved through a post-processing step. A flowchart of the computational pipeline is presented in Figure SI 1. We show that neural networks can be effectively trained and such models can achieve better results compared to models trained with real data and that the models trained with synthetic data are robust in different datasets without any input data specific hyperparameter tuning. Furthermore, we show that by having training data with known and controllable properties, insights on network behaviour can be obtained which serves as a step towards explainability.

2. Methods

2.1. Synthetic electrocardiogram generator

The ECG signal generator comprises four main parts: (i) RR-interval generation for controlling the average interval (HR) and its variation (HRV), (ii) waveform model where each of the characteristic waves (p, q, r, s, t) of an ECG can be independently adjusted in terms of amplitude (positive and negative), width and location (We refer the wave locations as fiducial points.), (iii) general noise model allowing a noise realization to be generated from an arbitrary spectrum allowing for example $1/f$ and random walk noises to be generated that roughly resemble motion artefacts and (iv) augmentation of real artefacts where a random segment from recorded artefact signals is added to the generated ECG with random amplitude.

Each synthetic signal is generated using model input parameters. We allow these parameters to change within a predefined range from which we extract randomly and independently each parameter value using a uniform distribution. The limits of this range, and thus the randomization and variance of the input data, are controlled using a scaling coefficient (C) which is controlled independently for RR-intervals, waveform shape, fiducial points and noise. The same C was used for each independent part if not stated otherwise. A scaling coefficient of $C = 1$ mimics the physiological variance of healthy individuals. These randomization limits, that were subsequently scaled, were determined by a combination of values in literature [17] and by fitting the model to real measurements with visual comparison. Some fitting examples are shown in Figure SI 2 which were also used to validate the model. C multiplies the range defined with lower and upper limits l_{low} and l_{high} of each adjustable parameter. The midpoint shifts non-linearly as weighted scaling is used and updated limits are $l_{low} - wl_{low}$ and $l_{high} + wl_{high}$ where $w = (l_{high} - l_{low})/(l_{low} + l_{high}) \times (C - 1)$. This allows parameter limits close to zero to vary more gently. However, this does not exclude inverted waves when the scaling is sufficiently large. Additionally, the t-wave locations were made dependent on heart rate and the distance to r-wave further scales with the square root of the average RR-interval [18]. The lower limit for every noise randomization is zero and this does not change during scaling. This allows some degree of low noise signals to always be in the training set. The starting limits for noise are subjective as the level of noise can vary significantly between devices and situations. The noise limits are adjusted so that the r-waves in most cases are still visually separable from noise when $C = 3$. The starting limits are listed on Table SI I. Overall, the randomization procedure is subjective, but it roughly follows the principle of starting from the signal variation of the healthy in rest and increasing it as high as the model allows when $C = 3$.

2.1.1. RR-intervals

The RR-intervals are modelled as:

$$rr_i = \mu + \beta sin(2\pi f_b t_i) + \gamma \tag{1}$$

where μ is the average RR-interval, the second term is the breathing modulation with coupling coefficient β, breathing frequency f_b and t_i is the sum of previous intervals and γ presents a stochastic component including long-term correlation between RR-intervals [19]. The last term in the presented method is not required as the training is done on short signal segments.

2.1.2. Electrocardiogram waveform

The ECG waveform (p, q, r, s and t waves) are modelled using a Gaussian function as the basis [20] for each wave,

$$\dot{z}_i = \frac{-2\pi ma\phi}{b^2}\exp\left(\frac{-m\phi^2}{2b^2}\right) \tag{2}$$

where ϕ is a linearly increasing phase signal (representing time) with amplitudes $[-\pi, \pi]$ where each phase cycle contains $rr_i * f_s$ samples. A separate phase signal is constructed for each wave. The time difference between the waves is achieved by simply offsetting the beginning of each phase signal by a delay that corresponds to the time difference of a particular event to r-wave. An asymmetry parameter m is used to create a slightly asymmetric t-wave which is typically observed in healthy. Different values of m are given to positive and negative gradient parts to create asymmetrical shape. Parameters a and b for controlling the amplitude and width of a particular event can be adjusted independently. The gradient signals of every wave are summed,

$$e\dot{c}g = \sum_{i\in\{p,q,r,s,t\}} \dot{z}_i \tag{3}$$

The final ECG is obtained with a cumulative numerical integration of $e\dot{c}g$.

2.1.3. Noise realizations

Time domain noise realization including white and power-law noise that corresponds a given power spectrum was generated [21,22]. First, a power spectral density (PSD) was defined as

$$PSD = \frac{\rho}{f^\alpha} + \sigma^2 \tag{4}$$

where the components are power-law (first) and white noise (second). The exponent α is used to increase the low frequency (f) noise and when it is e.g., 1 or 2, it reduces to $1/f$ noise and random walk, respectively. The ρ is a constant. This power spectrum was converted to time domain noise realization by first multiplying the amplitude of each frequency bin with an independent zero-mean complex Gaussian random variable of unit variance. Then an inverse-FFT of the randomized spectrum was computed and the real part was kept.

2.1.4. Artefact augmentation of the electrocardiograms

We augmented ECG signal (real and synthetic) with real ECG noise sources [23]. In this approach, baseline wander (BW) and muscle artefact (MA) noises from MIT-BIH Noise Stress Test database [24,25] and a simple generated 60 Hz sine wave representing the powerline interference were added to ECG signals with varying amplitude. Artefact realizations were obtained by. randomly selecting a segment of 1000 samples (same length as the ECG segment) from both BW and MA noise sources. These segments were then multiplied by random numbers from different uniform distributions to alter the strength of these noises. In the case of BW, uniform distribution of [0,10] was used and for MA it was [0,5]. The augmented artefact is one of three different categories; pure BW, pure MA, or a combination of these two. After noise type selection, 60 Hz sine wave representing powerline interference is added to the noise. The magnitude of the unit amplitude sine wave is varied before addition by multiplying it with a random number from a uniform distribution of [0,0.5]. The ranges of used uniform distributions were determined visually and the generated training examples were normalized to [−1,1] range before adding the artefact.

2.2. Datasets

Four different electrocardiogram datasets were used in this work, Glasgow University ECG database (ECG-GUDB) [26], MIT-BIH Normal Sinus Rhythm database [25], MIT-BIH Noise Stress Test database [24,25] and Computing in Cardiology 2017 single atrial fibrillation database (Cinc2017-AF) [27,28]. Both MIT-BIH databases were used only for model training whereas ECG-GUDB and Cinc2017-AF were used solely for testing. This was done in order to test if the trained models can generalize outside their training data. All four databases are publicly available.

ECG recordings of the ECG-GUDB database were obtained from 25 different subjects while performing five different activities (walking, jogging, operating a hand bike, solving a math test and sitting). Each task was recorded with two different setups, loose cables (standard Einthoven leads I-III) and a chest trap. Therefore, ECG-GUDB contains a total number of 250 (25 × 5x2) different ECG recordings. However, only 229 ECG records have annotations available. All ECG recordings were collected with Attys Bluetooth data acquisition board at a sampling frequency of 250 Hz. All r-wave labels were shifted to a maximum within a 16 sample window to ensure an accurate labelling scheme. In this work, we used Einthoven lead II from the loose cables setup and chest strap ECG. We split each of the 229 records into 29 separate non-overlapping four second segments. Thus, in total we use 6641 (229 × 29) four-second long ECG segments for testing. The heart rate distribution is shown in Figure SI 2.

The MIT-BIH Normal Sinus Rhythm database contains 18 long-term ECG recordings with r-wave annotations. Subjects in this database were found not to have significant arrhythmias and it includes 5 men (aged 26 to 45) and 13 women (aged 20 to 50). These recordings were resampled to 250 Hz. Segments during training were selected from a random location of a randomly selected signal. All r-wave labels were shifted to a maximum within a 16 sample window to ensure an accurate labelling scheme.

The MIT-BIH Noise Stress Test database has three different half-hour recordings of ECG noise, which may be consider as some combination of baseline wander, electrode motion artefact and muscle artefact. These recordings represent noise sources typically present in the ambulatory ECG recordings. Segments of noise were collected by placing electrodes such that the ECG signal was not observable. Segments with similar noise were concatenated into a single noise record.

The Cinc2017-AF database contains single lead recordings collected with the AliveCor device. The dataset has 8528 recordings lasting from 9 s to just over 60 s and contains normal sinus rhythm, atrial fibrillation and alternative rhythms. The signals do not have annotated peak labels and the validation was done visually by plotting the detected peak onto the signal under test. From the database, we randomly selected 30 measurements labelled as atrial fibrillation. These signals varied from 15 beats to 120 beats and had in total 1336 beats. The signals are recorded by placing a finger from both hands on the metal plates of the device. Such dry electrode configuration is prone to artefacts. No artefacts were removed and if an r-wave could not be reliably identified, it was not labelled as such.

2.3. Neural network

For all experiments a neural network consisting two bidirectional LSTM layers with 64 units with return sequences set to True followed by a dense layer with sigmoid activation was used with Tensforflow 2.6.0. Each input sample is 1000 samples long, presenting a 4 s ECG (sampling frequency of 250 Hz) and the output of the model is a 1000 samples segment where each sample is the probability of that sample being an r-wave. An r-wave in the training data is constructed as five neighbouring ones centred at the r-wave maximum. The training is done through a generator function that provides either only real or only synthetic samples and with optional real artefact augmentation. The artefact augmentation is independent on the source of ECG samples. Each training is run with a batch size of 32 and step size of 20 for 30 epochs. Binary cross entropy was used as the loss function with the Adam [29] optimizer with an initial learning rate of 0.0003.

The operating principle of the generator function that constructs the training samples can be summarized in five steps (i) Real data: Select randomly 1000 sample segments from the randomly selected ECG recording. Synthetic data: Generate unique random realizations in the generator function directly when needed. (ii) Generate a label vector for every segment based on r-wave indices. The vector has five ones at each r-wave and is zero elsewhere. (iii) If artefact augmentation is used, then normalize the segment to [−1,1] range and add the generated artefact. (iv) Filter the signals with a simple two order Butterworth filter with corner frequencies at 0.5 and 50 Hz. (v) Normalize segment to [−1,1] range. The same filtering and amplitude normalization (steps iv and v) as above were used for test signals.

2.4. Peak detection post-processing

LSTM model predictions are a sequence of probability values that indicate the likelihood of a sample being an r-wave and thus the unambiguous peak location needs to be evaluated from these probability vectors. We followed similar steps as presented in [23]: (i) Split the ECG into segments of 1000 samples with 750 sample overlap. (ii) Use LSTM model to predict sample-wise r-wave probabilities for each segment. (iii) Take the average probabilities from overlapping predictions for each sample. Because ECG segments overlapped, four predictions are produced for each time sample of the ECG signal. Overlapping predictions are averaged to get a single probability value for each sample. (iv) Extract r-wave locations from average probabilities by selecting averaged probability values that are above a predefined threshold of 0.05. These are considered as r-wave candidates. To produce only one peak index for every r-wave, each probability candidate are shifted to index where ECG has the highest amplitude within a ten-sample window. When five or more samples are shifted onto the same index, it is considered as an r-wave. (v) Filter out r-waves that occur unrealistically close. After unique index extraction, there might be some false positives such as pronounced t-waves or noise peaks that were identified as an r-wave. The r-waves that do not have any other r-waves within a threshold distance of 75 samples are considered as valid r-waves and they form the initial set of approved r-waves. All r-waves that occur within the threshold are put into a separate candidate set. Then the candidate set is iterated over by starting from the candidate with highest probability value. In each iteration, the candidate under consideration is compared to the set of approved r-waves. If the candidate is not within the threshold distance of any of the approved r-waves, it is considered a valid r-wave and it is added into the set of approved r-waves.

2.5. Code and data availability

The neural network, artefact augmentation and peak detection post-processing has been described in detail in our earlier work [23]. Minor modifications to training hyperparameters such as learning rate, number of epochs and steps were implemented. No modification based on test set performance on either synthetic nor real data were done. The same training scheme was used throughout the experiments and all models were trained the same amount. Real data used here are all publicly available and the code for synthetic data generation and training is available in [30]. The corresponding training with real data is available at [31].

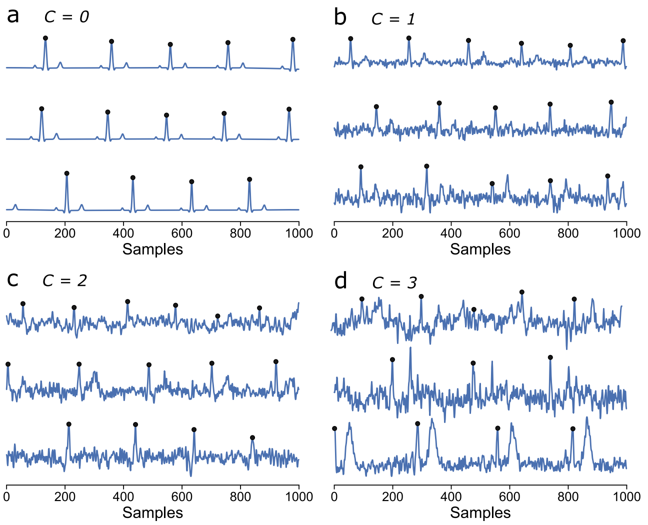

Fig. 2. Examples of random synthetic signals generated using values of $C = 0, 1, 2$ and 3. $C = 1$ produces signals roughly within the physiological range of healthy in rest whereas $C = 0, 2$ and 3 produces minimal, high and extreme variation between the electrocardiograms.

3. Results

The variation of synthetic signals is based on the reported variation of typical ECG [17] on the healthy and the noise properties are judged empirically from typical ECG recordings. This variation range is used as a starting point and scaled to both directions with a scaling coefficient C that signifies how much larger/smaller the range is in a particular synthetic dataset. This is detailed in Section 2.1. Several examples of synthetic ECGs are shown in Fig. 2. The similarity in (a) is evident as the signals are not allowed to have any variance. However, the starting point of every realization is randomized. In (b), the noise clearly distorts the waveforms and the signals resemble typical lowish quality ECGs. Domain randomization in (c) and (d) is high producing unrealistic and noisy ECGs.

The loss (binary cross-entropy) computed after every epoch during the training process is shown in Fig. 3. In (a) the test loss remains high when randomization is low and the network is unable to learn the relevant signal characteristics. The performance improves systematically as the domain randomization is increased through the scaling coefficient (C) and the best performance is achieved with highest randomization when $C = 3$. The testing ROC-AUC used to rank the model performances in Fig. 4 also improves with increasing C and a clear performance boost is observed when $C > 1$. With low randomization, the loss is getting lower without much improvement in ROC-AUC. This implies that the network is becoming more confident and its correct predictions are more clear, but this does not translate into better ROC-AUC. It is also likely that the testing data has a fair amount of typical and high quality signals that are quite easy to interpret. In particular, negative samples constitute the large majority class, and are not challenging to classify. The harder signals, in turn, are correctly interpreted by the network only with sufficiently high randomization. It is also noteworthy that having a physiologically valid input variation does not result in a high performing model. Instead, the best performing model occurs when randomization is increased clearly beyond what is expected to be in the test data. Comparisons done on models trained with real data surprisingly perform worse than a model trained purely on synthetic data. This is evident in both loss and ROC-AUC where a model trained with real data performs roughly equally well as model trained with a scaling coefficient of 2. However, the test set and training set are different and in part most likely have non overlapping characteristics. Having larger real dataset with more variation in training

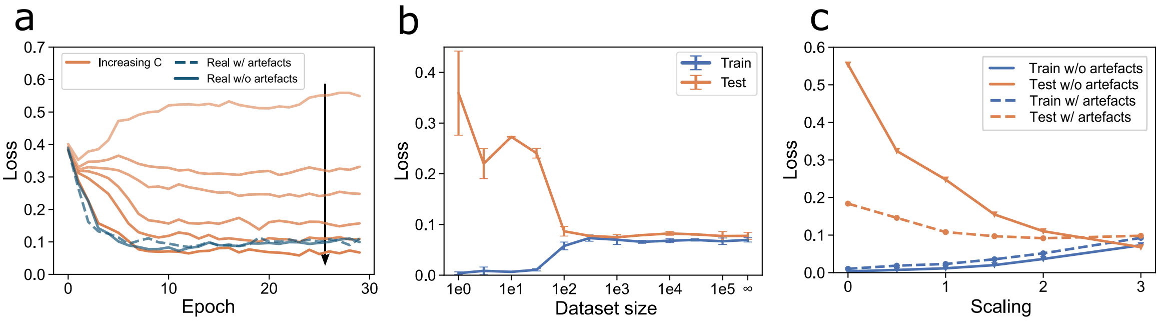

Fig. 3. (a) Test loss during training with increasing randomization scaling coefficient, $C = 0, 0.5, 1.0, 1.5, 2.0$ and 3.0 (orange). Respective curves for real data with (dashed dark blue) and without (dark blue) artefact augmentation. The loss is computed after every epoch on test data. Training with synthetic data is done without artefact augmentation. (b) Learning curves with $C = 3$ showing model performance with different input data sizes. Error bars (mean ±std) are from three independent runs with different seeds. The ∞ means that every synthetic sample is unique and is generated on the fly during training. (c) Training loss compared to test loss with increasing randomization. In (b) and (c) the test loss is an average from last five epochs. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

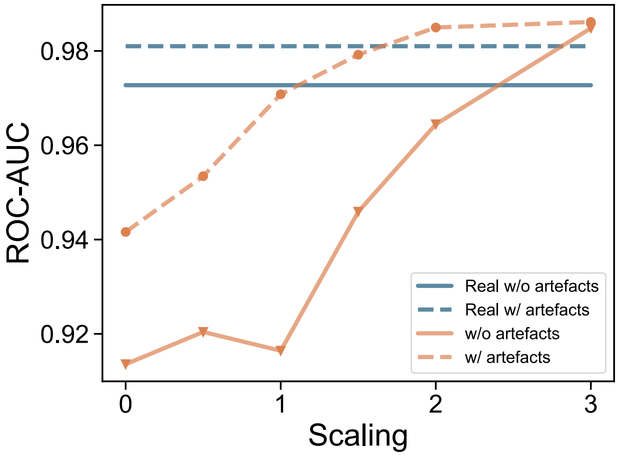

Fig. 4. ROC-AUC on test data (average from last five epochs) with increasing randomization scaling coefficient (C).

is expected to improve the performance and generalization to unseen datasets.

The learning curves are shown in Fig. 3(b). In this experiment, synthetic datasets (C = 3) of varying size are pre-generated and during training the examples are drawn randomly from this set. The infinite size signifies that every training example is unique, which is the used technique in other experiments. Expectedly, the training loss is very low and test loss is high when the number of input samples is minimal. The network simply learns the properties of this data and fails to generalize. As the input data size is increasing, the gap between training and test curves is closing and with sufficient input data size there is a very small variance and bias. The learning curves are run three times with different random signal generations. Low amount of input data produces non-robust training, but as the data size is sufficient the model performances converge.

The training performance with increasing scaling is examined in Fig. 3(c) including synthetic data without augmentation as in previous experiments as well as with the artefact augmentation. We can observe that the augmentation which adds a significant amount of randomization helps the network to learn when C is low. With high randomization, the training and test losses are roughly the same, indicating a high degree of challenge the network has with this training data. The best result is achieved with highest randomization scaling and in such a case not much is achieved with additional artefact augmentation. Augmentation could have a higher impact if testing data was even more challenging or more corrupted by artefacts.

To further validate the performance of training with synthetic data, we compared the models by considering the correctly predicted peaks. Probability vectors were converted into location indices in the post-processing step. The F1-score of each measurement was used for comparison as shown in Fig. 5(a). An F1-score was computed for each measurement. The error plots show the mean of these scores over all measurements and whiskers present the 10% and 90% percentiles. The mean values systematically increase with domain randomization (increasing C) while also reducing the number of signals where the model fails to provide a meaningful result. The results also show that augmenting real artefacts in the signals helps the model to learn which was also observed in testing loss and ROC-AUC evaluation. However, it is noteworthy that artefact augmentation alone does not yield good performance if the synthetic data itself has no to minimal variation. Best results are achieved with highest randomization including artefact augmentation, but improvement is modest compared to only synthetic data with $C = 3$. Synthetic data without augmentation provides better performance than real data with artefact augmentation in these experiments. The overall performance compares favourably to state-of-art [32].

Further experiments with increasing C either very slightly improved or worsened the results. This is not surprising since the criteria for scaling were that $C = 3$ produces the maximal amount of randomization for all adjustable parts of the signal and further increases lead to noise becoming too dominant where r-waves are completely lost, fiducial points leaking over the designated cardiac cycle and waveforms are breaking down.

The randomization clearly has significant effect on the models ability to learn the desired characteristic. In previous experiments the r-wave was kept at a nominal and modest variance through all experiments and the surroundings were randomized as shown in Table SI I. In Fig. 5(b) we compare the effect of randomization of the desired characteristic, the r-wave, to randomization of everything else than the r-wave. In our case, this can also be viewed as randomizing the samples that associate to label 1 (r-wave) or to label 0 (not r-wave) although this is not strictly true as noise realizations are added to the entire generated signal. The four cases shown in Fig. 5(b) compare the mean F1-scores. The cases for the blue curve are with $C = 0, 1, 2$ and 3 (same as in (a)). For the orange curve, the cases present increasing r-wave randomization. We chose to randomize the r-wave by adjusting the upper limit of amplitude and width parameter to match that of the t-wave and we kept the lower limit intact. This ensures that nominal r-waves are also present in all cases. The surroundings are randomized with $C = 3$ in all cases. The first case is with nominal r-wave randomization (the same case as the last case in previous) and the following cases (2,3,4) have upper limits matching corresponding t-wave upper limits with $C = 1, 2, 3$. As the r-wave shape randomization is increased, the performance drops significantly since the model is now learning to detect various shapes in the electrocardiogram as r-waves.

The synthetic generator produces electrocardiograms where several characteristics (waveform shape, fiducial points, RR-intervals and

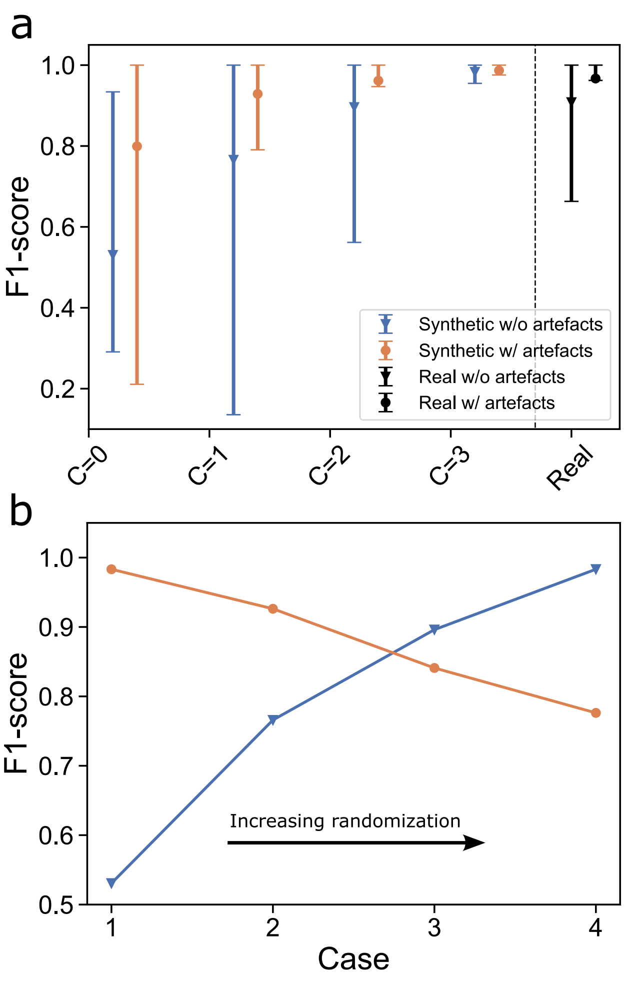

Fig. 5. (a) F1-scores on test data with scaling coefficient $C = 0, 1, 2$ and 3. Error bars are mean with 10% and 90% percentiles for F1-scores over the test data where models are trained without artefact augmentation (blue) and with augmentation (orange). Networks trained with real data are shown on the right (black). (b) Performance comparison when randomization of the feature of interest (r-wave) is increased (orange) as opposed to randomizing the surroundings i.e. everything else, but the r-wave (blue). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

noise) of the signal can be independently changed and randomized. Thus, the effect of each characteristic can be independently tested. As detailed in Table SI II the randomization of fiducial points had the lowest impact, closely followed by RR-intervals. Randomization of the waveform shape results in clearly reduced loss, but not in improved ROC-AUC. This indicates that the model is able to learn easier signals with high confidence, but fails on the more difficult, mostly likely noisy ones. Addition of a significant amount of noise results in a good performance measured by all metrics. This is not unexpected as the model learns to discriminate the characteristic r-wave from rest of the signal which is efficiently randomized with the addition of noise. However, experiments with only r-waves and noise (i.e., the model is modified not to produce any other waves) results in modest performance and inclusion of qrs complex instead of r-wave results in some improvement. Inclusion of t-wave further helps the model learn to discriminate between the two prominent waves. If t-wave is excluded, the model is confused between t and r-wave when presented with real data. The simple combination of the r and t-wave and noise realizations performs surprisingly well and not much else is needed. However, best results are achieved with all randomizations including artefact augmentation.

Table 1
Detection of r-waves in atrial fibrillation ($n = 3$, no. peaks = 1336).

Model	Precision	Recall	F1
$C = 3$ w/o artefacts	0.985	0.960	0.968
Real w/ artefacts	0.979	0.981	0.979

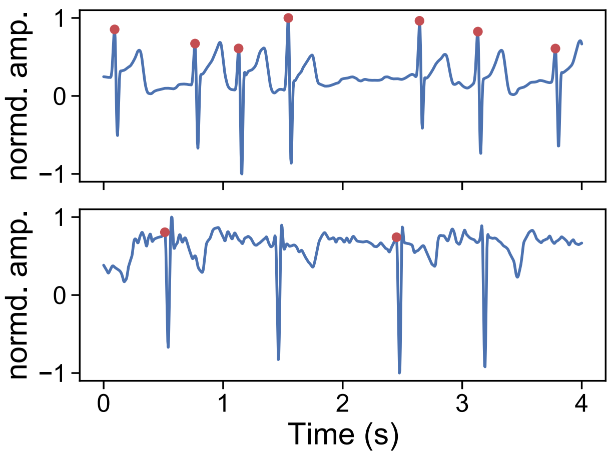

Fig. 6. Examples of peak detection in atrial fibrillation with successful (upper) and unsuccessful (lower) detection. The lower graph shows a qs-wave with prominent downward deflection that the model trained with synthetic data ($C = 3$ w/o artefacts) is unable to detect correctly.

One shortcoming of rule based detection algorithms is the performance during arrhythmia. We tested if the rhythm could be detected in a separate single lead atrial fibrillation test set [27,28] when the model is not specifically trained for it. Results are summarized in Table 1. We used a model trained with synthetic data only (w/o artefacts $C = 3$). Most false detections were due to noise artefacts in the signals or when a prominent downward qs-wave without a clear r-wave was present. The artefacts typically create a single false detection, but the model can completely fail with highly abnormal r-wave as seen in Fig. 6. Several examples from Cinc2017-AF dataset where model fails are shown in Figure SI 5. The model is unreliable in these atrial fibrillation recordings due to a prominent downward qs-wave and a lack of upward deflection. When a sufficient upward deflection is present, the model's predictions are mostly correct even in the presence of downward deflection. As shown in Figure SI 5, the predictions become more unreliable with increased downward deflection and reduced upward deflection.

In such difficult cases, the model trained with real data performed better, most likely due to having some similar abnormal examples during training. Regardless, the model trained with purely synthetic data (with high domain randomization) overall performs well even with abnormal peak shapes, peak inversions and abnormal rhythm, none of which were specifically accounted for during training.

4. Discussion

The model trained with synthetic ECGs was able to produce better predictive performance when tested on real data that included subjects performing various physical activities. We used r-wave detection as an example, which is a prominent feature of the ECG, and simply providing large enough variation provides compelling performance. Having randomizations as large as possible without drowning the r-waves in noise or breaking the model resulted in the best performance. Such randomizations are clearly beyond any physiological domain or what would otherwise be expected to be present in the test signals. Further examinations with synthetic data when different randomization components (waveform, RR-intervals, noise, artefact augmentation) are turned on individually (shown in Table SI II) revealed that simple noise realization is the most effective way to randomize the signal and allow the model to learn. However, if the t-wave was excluded altogether from the generated signals, the performance drops and the model is unable to discriminate between these two prominent waves.

Our experiments also show that ECG waveform location randomizations produce the smallest effect, hinting that the network learns to detect the r-wave from its surroundings with little regard to what specifically happens around it and it is sufficient simply to have high variation on the parts that are not of interest. However, the exclusion of t-wave resulted in poor performance and its presence is required as a counter example for the model. The RR-interval variation produces an interesting result where the loss is high, but ROC-AUC is comparatively high, highlighting the possibility that the network is able to correctly classify the labels, but with low confidence. This results in poor performance in actual r-wave detection when the post-processing step is included for F1-score calculation. The insensitivity to r-wave locations was further tested in atrial fibrillation data and surprisingly the model performed well even in the presence of arrhythmia. However, with highly abnormal r-wave shapes the model occasionally failed and it does not work robustly in such special cases. This could most likely be fixed by introducing other specific r-waves in the training, but this should be done in a specific and controlled manner. Overall, the model performs well against various sources of noise as exemplified in Figure SI 4. In the rare cases that a high and narrow, i.e. spiky artefact is present, the model can falsely detect them as r-waves.

Additionally, the performance with high C is robust and repeated training with different seeds produces closely matching results. Hyperparameter tuning and/or longer training would most likely increase the performance further. Although the synthetic data outperformed the models trained with real data, it is likely that adding significant amounts of highly varying real data from various sources would reduce or flip the performance gap.

5. Conclusion

We presented a method to train neural networks using synthetic data generated from scratch and it achieved good r-wave detection performance on challenging electrocardiogram test sets including recordings during various physiological activities and atrial fibrillation using only one training scheme with high amount of domain randomization.

This approach could be beneficial in training robust networks for various health monitoring applications and it could be extended to cardiac disease detection using vast and available a priori information.

Unlike alternative approaches such as adversarial domain adaptation, the method here does not explicitly require multiple training data sets, as noise mechanisms can be set by hand. Furthermore, the fine control over the synthetic signals may be used to infer reasoning behind complex models and advance the explainability of them and while paving the way of reducing the need for expensive to collect and manually label electrocardiograms.

Declaration of competing interest

The authors state no conflict of interest.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.artmed.2023.102583.

References

[1] Bote-Curiel L, Munoz-Romero S, Gerrero-Curieses A, Rojo-Álvarez JL. Deep learning and big data in healthcare: A double review for critical beginners. Appl Sci 2019;9(11):2331.

[2] Rieke N, Hancox J, Li W, Milletari F, Roth HR, Albarqouni S, et al. The future of digital health with federated learning. NPJ Digit Med 2020;3(1):1–7.

[3] Muralidhar N, Islam MR, Marwah M, Karpatne A, Ramakrishnan N. Incorporating prior domain knowledge into deep neural networks. In: 2018 IEEE international conference on big data. IEEE; 2018, p. 36–45.

[4] Alday EAP, Gu A, Shah AJ, Robichaux C, Wong A-KI, Liu C, et al. Classification of 12-lead ECGs: the PhysioNet/Computing in cardiology challenge 2020. Physiol Meas 2021;41(12):124003.

[5] Alice CY, Mohajer B, Eng J. External validation of deep learning algorithms for radiologic diagnosis: A systematic review. Radiol Artif Intell 2022;4(3).

[6] Dinsdale NK, Jenkinson M, Namburete AI. Deep learning-based unlearning of dataset bias for MRI harmonisation and confound removal. NeuroImage 2021;228:117689.

[7] Shang Z, Zhao Z, Fang H, Relton S, Murphy D, Hancox Z, et al. Deep discriminative domain generalization with adversarial feature learning for classifying ecg signals. In: 2021 Computing in cardiology, vol.48. IEEE; 2021, p. 1–4.

[8] Tobin J, Fong R, Ray A, Schneider J, Zaremba W, Abbeel P. Domain randomization for transferring deep neural networks from simulation to the real world. In: 2017 IEEE/RSJ international conference on intelligent robots and systems. IEEE; 2017, p. 23–30.

[9] Tremblay J, Prakash A, Acuna D, Brophy M, Jampani V, Anil C, et al. Training deep networks with synthetic data: Bridging the reality gap by domain randomization. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops. 2018, p. 969–77.

[10] Prakash A, Boochoon S, Brophy M, Acuna D, Cameracci E, State G, et al. Structured domain randomization: Bridging the reality gap by context-aware synthetic data. In: 2019 International conference on robotics and automation. IEEE; 2019, p. 7249–55.

[11] James S, Wohlhart P, Kalakrishnan M, Kalashnikov D, Irpan A, Ibarz J, et al. Sim-to-real via sim-to-sim: Data-efficient robotic grasping via randomized-to-canonical adaptation networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019, p. 12627–37.

[12] Salem M, Taheri S, Yuan J-S. ECG arrhythmia classification using transfer learning from 2-dimensional deep CNN features. In: 2018 IEEE biomedical circuits and systems conference. IEEE; 2018, p. 1–4.

[13] Weimann K, Conrad TO. Transfer learning for ECG classification. Sci Rep 2021;11(1):1–12.

[14] Costa M, Goldberger AL, Peng C-K. Multiscale entropy analysis of complex physiologic time series. Phys Rev Lett 2002;89(6):068102.

[15] Adib E, Afghah F, Prevost JJ. Synthetic ECG signal generation using generative neural networks. 2021, http://dx.doi.org/10.48550/ARXIV.2112.03268, [Online]. Available: https://arxiv.org/abs/2112.03268.

[16] Rudin C. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nat Mach Intell 2019;1(5):206–15.

[17] Thaler M. The only EKG book you'll ever need. Lippincott Williams & Wilkins; 2017.

[18] Vandenberk B, Vandael E, Robyns T, Vandenberghe J, Garweg C, Foulon V, et al. Which QT correction formulae to use for QT monitoring? J Am Heart Assoc 2016;5(6):e003264.

[19] Kantelhardt J, Havlin S, Ivanov PC. Modeling transient correlations in heartbeat dynamics during sleep. Europhys Lett 2003;62(2):147.

[20] McSharry PE, Clifford GD, Tarassenko L, Smith LA. A dynamical model for generating synthetic electrocardiogram signals. IEEE Trans Biomed Eng 2003;50(3):289–94.

[21] Timmer J, Koenig M. On generating power law noise.. Astron Astrophys 1995;300:707.

[22] Kaisti M, Altti M, Poutanen T. Radiometric resolution analysis and a simulation model. Remote Sens 2016;8(2):85.

[23] Laitala J, Jiang M, Syrjälä E, Naeini EK, Airola A, Rahmani AM, et al. Robust ECG R-peak detection using LSTM. In: Proceedings of the 35th annual ACM symposium on applied computing. 2020, p. 1104–11.

[24] Moody GB, Muldrow W, Mark RG. A noise stress test for arrhythmia detectors. Comput Cardiol 1984;11(3):381–4.

[25] Goldberger AL, Amaral LA, Glass L, Hausdorff JM, Ivanov PC, Mark RG, et al. PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals. Circulation 2000;101(23):e215–20.

[26] Howell L, Porr B. High precision ECG database with annotated R peaks, recorded and filmed under realistic conditions. University of Glasgow; 2018, http://dx.doi.org/10.5525/GLA.RESEARCHDATA.716, [Online]. Available: http://researchdata.gla.ac.uk/id/eprint/716.

[27] Clifford GD, Liu C, Moody B, Li-wei HL, Silva I, Li Q, et al. AF classification from a short single lead ECG recording: The PhysioNet/computing in cardiology challenge 2017. In: 2017 Computing in cardiology. IEEE; 2017, p. 1–4.

[28] Goldberger AL, Amaral LAN, Glass L, Hausdorff JM, Ivanov PC, Mark RG, et al. PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. Circulation 2000 (June 13);101(23):e215–20.

[29] Kingma DP, Ba J. Adam: A method for stochastic optimization. 2014, arXiv preprint arXiv:1412.6980.

[30] Kaisti M. Training neural networks with synthetic electrocardiograms. 2021, GitHub Repository, GitHub, https://github.com/mkaist/Training-neural-networks-with-synthetic-electrocardiograms.

[31] Laitala J. Ecg2rr. 2020, GitHub Repository, GitHub, https://github.com/jtlait/ecg2rr.

[32] Peimankar A, Puthusserypady S. DENS-ECG: A deep learning approach for ECG signal delineation. Expert Syst Appl 2021;165:113911.