Augmenting Physiological Time Series Data: A Case Study for Sleep Apnea Detection

Konstantinos Nikolaidis[1]✉, Stein Kristiansen[1], Vera Goebel[1], Thomas Plagemann[1], Knut Liestøl[1], and Mohan Kankanhalli[2]

[1] Department of Informatics, University of Oslo, Gaustadalleen 23B, 0316 Oslo, Norway

[2] Department of Computer Science, National University of Singapore, COM1, 13 Computing Drive, 117417, Singapore

Abstract. Supervised machine learning applications in the health domain often face the problem of insufficient training datasets. The quantity of labelled data is small due to privacy concerns and the cost of data acquisition and labelling by a medical expert. Furthermore, it is quite common that collected data are unbalanced and getting enough data to personalize models for individuals is very expensive or even infeasible. This paper addresses these problems by (1) designing a recurrent Generative Adversarial Network to generate realistic synthetic data and to augment the original dataset, (2) enabling the generation of balanced datasets based on a heavily unbalanced dataset, and (3) to control the data generation in such a way that the generated data resembles data from specific individuals. We apply these solutions for sleep apnea detection and study in the evaluation the performance of four well-known techniques, i.e., K-Nearest Neighbour, Random Forest, Multi-Layer Perceptron, and Support Vector Machine. All classifiers exhibit in the experiments a consistent increase in sensitivity and a kappa statistic increase by between $0.72{\cdot}10^{-2}$ and $18.2{\cdot}10^{-2}$.

Keywords: Augmentation · GAN · Time Series Data.

1 Introduction

The development of deep learning has led in recent years to a wide range of machine learning (ML) applications targeting different aspects of health [24]. Together with the recent development of consumer electronics and physiological sensors this promises low cost solutions for health monitoring and disease detection for a very broad part of the population at any location and any time. The benefits of automatic disease detection and especially early prognosis and life style support to keep healthy are obvious and result in a healthier society and substantial reduction of health expenses. However, there are high demands on the reliability of any kind of health applications and the applied ML methods must be able to learn reliably and operate with high performance. To achieve this with supervised learning, appropriate (labelled) datasets gathered with the physiological sensors that shall be used in a health application are needed for training such that classifiers can learn to sufficiently generalize to new data. However, there are several challenges related to training datasets for health applications including data quantity, class imbalance, and personalization.

In many domains, the quantity of labelled data has increased substantially, like computer vision and natural language processing, but it remains an inherent problem in the health domain [24]. This is due to privacy concerns as well as the costs of data acquisition and data labelling. Medical experts are needed to label data and crowdsourcing is not an option. To enable medical experts to label data, data are typically acquired with two sensor sets. One set with the sensors that should be used in a health application and one sensor set that represents the gold standard for the given task. This problem is magnified by the fact that any new physiological sensor requires new data acquisition and labelling. Furthermore, there is a high probability that the data acquisition results in an unbalanced dataset. Since many health applications aim to detect events that indicate a health issue there should "ideally" be equally many time periods with and without these events. In general, this is unrealistic for a recording from an individual as well as across a larger population that is not selected with prior knowledge of their health issues. For example, in the recent A3 study [30] at the Oslo University Hospital individuals with atrial fibrillation were screened for sleep apnea. In a snapshot from this study with 328 individuals, 62 are classified as normal, 128 with mild apnea, 100 with moderate apnea, and 38 with severe apnea. The severeness of sleep apnea is captured by the Apnea Hypopnea Index (AHI) which measures the average number of apnea events per hour and is classified as follows: AHI$<$15, (normal), 15$\leq$ AHI$<$30, (moderate), AHI$\geq$30, (severe)[3]. It is unrealistic to expect that a sufficiently large dataset for training can be collected from each individual, because it is inconvenient, requires medical experts to label the data, and might be infeasible due to practical reasons for those that develop the application and classifier.

The objectives of this work are to address these problems with insufficient datasets in the health domain: (1) generate synthetic data from a distribution that approximates the true data distribution to enhance the original dataset; (2) use this approximate distribution to generate data in order to rebalance the original dataset; (3) examine the possibility to generate personalized data that correspond to specific individuals; and (4) investigate how these methods can lead to performance improvements for the classification task.

The mentioned problems are relevant for many applications in the health domain. As a proof-of-concept, we focus in our experimental work on the detection of obstructive sleep apnea (OSA). OSA is a condition that is characterized by frequent episodes of upper airway collapse during sleep, and is being recognized as a risk factor for several clinical consequences, including hypertension and cardiovascular disease. The detection and diagnosis is performed via polysomnography (PSG). PSG is a cumbersome, intrusive and expensive procedure with very long waiting times. Traditionally, PSG is performed in a sleep laboratory. It requires

[3] From a ML viewpoint only individuals with severe sleep apnea would produce balanced recordings.

the patient to stay overnight and record various physiological signals during sleep, such as the electrocardiogram, electroencephalogram, oxygen saturation, heart rate, and respiration from the abdomen, chest and nose. These signals are manually evaluated by a sleep technician to give a diagnosis. In our earlier work [18], we could show that ML can be used to classify PSG data with good performance, even if only a subset of the signals is used, and that the quality of collected data with commercial-of-the-shelf respiratory sensors approaches the quality of equipment used for clinical diagnosis [19].

In this work, we use different conditional recurrent GAN designs, and four well-known classification techniques, i.e., K-Nearest Neighbor (KNN), Random Forest (RF), Multi-Layer Perceptron (MLP), and Support Vector Machine (SVM) to achieve the aforementioned objectives. Since we want to use datasets that are publicly available and open access, we use the Apnea-ECG and MIT-BIH databases from Physionet [1, 2] for our experiments. The reminder of this paper is organized as follows: Section 2 presents related works, and Section 3 our methods. In Section 4 we evaluate these methods through three experiments. Section 5 concludes this paper.

2 Related Work

Although the GAN framework [12] has recently acquired significant attention for its capability to generate realistic looking images [23, 17], we are interested in time series generation. The GAN is not as widely used for time series generation as for images or videos, however, works which investigate this approach exist [22]. There are also relevant applications for sequential discrete data [31].

Most works are related to Objective 1 [10, 6]. Hyland et al. [10] use a conditional recurrent GAN (based on [21]) to generate realistic looking intensive care unit data, preconditioned on class labels, which have continuous time series form. Among other experiments, they train a classifier to identify a held out set of real data and show the possibility of training exclusively on synthetic data for this task. They also introduce the opposite procedure (train with the real data and test on the synthetic) for distribution evaluation. We use similar methods to synthesize data in the context of OSA, but we expand these techniques by introducing a metric for evaluating the synthetic data quality which is based on their combination. We also investigate methods to give different importance to different recordings. Other works related to medical applications of GANs include [16, 5]. Our work is associated with the use of multiple GANs in combination and uses different design and metrics from the above works (both works use designs based on combinations of an auto-encoder and a GAN). Many approaches that include multiple GANs exist such as [9, 14].

We note that most of the related work with the exception of [5] focuses individually on the synthetic data generation and evaluation, and not how to use these data to augment the original dataset to potentially improve the generalization capability of other classifiers. To the best of our knowledge only few works [8, 25, 20] exist that examine the potential application of GANs to produce realistic synthetic data for class rebalancing of a training dataset. Only one of them uses specifically a recurrent GAN architecture. Finally, we did not find any relevant work that depicts the data distribution as a mixture of different recording distributions, with the end-goal of producing more personalized synthetic data.

3 Method

The goal of data augmentation in this work is to train classifiers to successfully detect in physiological time series data health events of interest. In our use case this means to classify every 30 or 60 second window of a sleep recording as apneic (i.e., an apnea event happened) or non-apneic.

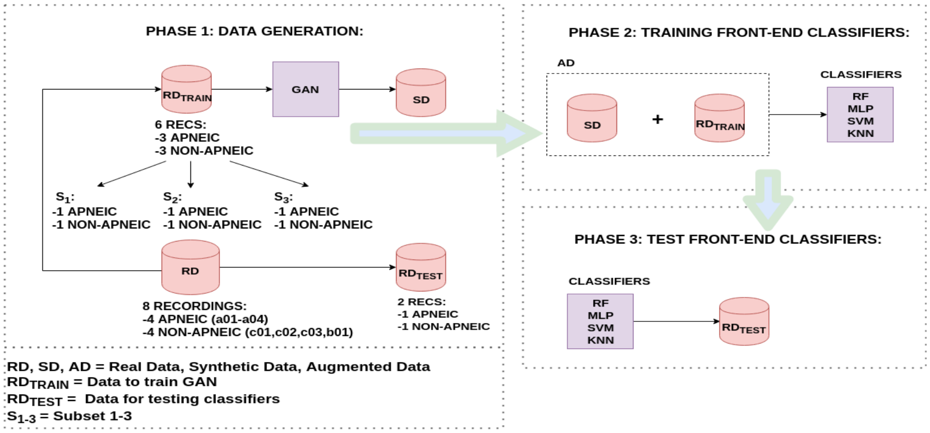

Fig. 1. GAN Augmentation

We use a conditional recurrent GAN to generate a synthetic dataset (SD, see Figure 1) to augment the original training dataset (RD$_{TRAIN}$) (Objective 1) and to rebalance an unbalanced RD$_{TRAIN}$ (Objective 2). Furthermore, we extend the single GAN architecture to a multiple GAN architecture to generate more synthetic data that is potentially closer to the test data to enable personalized training (Objective 3). In this section, we introduce the datasets we use, the two GAN architectures, and the metrics used to evaluate the quality of the generated data.

3.1 Data

We focus on the nasal airflow signal (NAF), because it can adequately be used to train a classifier to recognize apneas and yields the best single signal performance

[18]. Furthermore, NAF is contained in most recordings (in 12 recordings[4]) in the MIT-BIH database. From the Apnea-ECG database we use eight sleep recordings (i.e., a01, a02, a03, a04, c01, c02, c03, b01) that contain the NAF signal with durations 7-10 hours. From MIT-BIH we use the 12 recordings that include the NAF signal. Note that MIT-BIH has low data quality (noisy wave-forms, values out of bounds, etc.), especially when compared to Apnea-ECG.

The sampling frequency is 100Hz for Apnea-ECG and 250Hz for MIT-BIH and all recordings contain labels for every minute window of breathing for Apnea-ECG and for every 30 seconds window for MIT-BIH. These labels classify a window as apneic or non-apneic. For Apnea-ECG, there are four severe OSA, apneic recordings(a01-a04) and four normal, non-apneic recordings (c01-c03,b01). AHIs vary from 0 to 77.4. For MIT-BIH, AHIs vary from 0.7 to 100.8. The only preprocessing we perform is rescaling and downsampling of the data to 1Hz.

3.2 Single GAN Architecture

In order to solve the problems of too small and unbalanced datasets we generate synthetic data and augment the original dataset. Due to its recent successes in generating realistic looking synthetic data, e.g., images and music, we use the GAN framework, in particular, a conditional recurrent GAN. The conditional aspect allows us to control the class of the generated data (apneic, non-apneic). Thus, data from both classes can be generated and the front-end classifiers are able to learn both apneic and non-apneic event types. The generative network G takes as input random sequence from a distribution $p_z(z)$ and returns a sequence that after training should resemble our real data. The discriminator D takes as input the real data with distribution $p_{Data}(x)$ and the synthetic data from G, and outputs the probability of the input being real data. Using cross-entropy error, we obtain the value function [12]:

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

G has the objective to minimize the probability that D correctly identifies the generated data as synthetic (second term of Eq. 1). D has the objective to maximize the probability to correctly classify data as either real or synthetic.

The objective of the generator is to fool the discriminator such that it classifies generated data as real. Through the training the generator learns to produce realistic looking synthetic data. Consequently, the generated data distribution converges to the real data distribution [12]. Inspired by [10], we use a conditional LSTM [15] as G and D, because we are interested in time series generation of sequentially correlated data. LSTMs are able to store information over extended time intervals and avoid the vanishing and exploding gradient issues [11]. G produces a synthetic sequence of values for the nasal airflow and D classifies each individual sample as real or fake based on the history of the sequence.

[4] slp01, slp02a , slp02b, slp03 , slp04, slp14, slp16, slp32, slp37, slp48, slp59, slp66, slp67x

3.3 Multiple GAN Architecture

The aim for this approach is to ensure that the SD represents in a realistic manner all recordings in RD$_{TRAIN}$. Each person, depending on various environmental and personal factors has different breathing patterns, but individual characterization is possible.

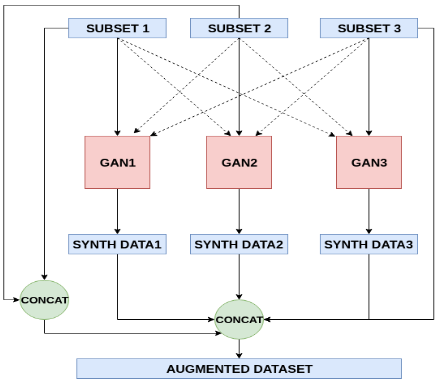

Fig. 2. Three GANs trained separately with a chance to interchange subsets

Such an individualization is often described as bias towards a particular patient [11]. We, in contrast, make the hypothesis that different recording sessions have different data distributions, which together constitute the total apnea/non-apnea distribution of the dataset. In our case different recordings correspond to different individuals. A distinction is made between the recordings and the modes in their distribution since a recording can have more than one mode in its distribution, and different modes in the feature space can be common for different recordings. Since we have insufficient data per recording to successfully perform the experiments of this section, we define disjoint subsets of recordings (hereby called *subsets*), the union of which constitutes the original recording set. Under this hypothesis, the data distribution can be depicted as a mixture of the different recording distributions:

$$p_{Data}(x) = \sum_{i=0}^{k_{rec}} w_{r_i} p_{rec_i}(x) = \sum_{j=0}^{k_{sub}} w_{s_j} p_{sub_j}(x) \tag{2}$$

with:

$$p_{sub_j}(x) = \sum_{l \in sub_j} w_{sb_l j} p_{rec_l}(x) \tag{3}$$

where k_{rec} is the total number of recordings, k_{sub} is the total number of subsets, p_{rec_i} is the data distribution of recording i, and $w_{r_i} = 1/k_{rec}$ assuming equal contribution per recording, p_{sub_j} and w_{s_j} is the distribution and weights of subset j, and $w_{sb_l j}$ the weights of the recording within each subset.

We restate Eq. 1 to explicitly include the distributions of the subsets by dedicating a pair of G and D to each subset. This allows each GAN to prioritize the data from its respective subset, thus making it less probable to exhibit mode collapse for modes contained in the examined recordings. Each subset contains one apneic and one non-apneic recording (see Section 3.1, 4.4).

The goal of this method is to properly represent all recordings in the SD. The potential decrease of collapsing modes due to the use of multiple GANs for different data is an added benefit. There are relevant publications that use similar ensemble techniques to specifically address this issue backed by theoretical or methodological guarantees [29, 14].

Since the amount of data per recording is too low to train GAN with only two recordings, we allow each GAN to train with data from the training subset of another GAN with a controllable probability (see Figure 2). Per iteration, for GANj we perform a weighted dice toss such that $J = (1, 2..., j, ..., k_{sub})$, and $\mathbf{p} = (p_1, p_2, ...p_j, ...p_{k_{sub}})$ where J is a random variable following the multinomial distribution and $\mathbf{p}$ the parameter probability vector of the outcomes. For GANj $p_j = p$, and $p_1 = p_2 = ... = p_i.. = p_{k_{sub}} = \frac{1-p}{k_{sub}-1} \forall i \neq j$ for a chosen value p . Note that the larger the chosen p, the more pronounced the modes of the recording combination that corresponds to GANi will be. It is relatively straightforward to show that:

Proposition 1. *A GAN satisfying the conditions of Proposition 2 of [12] and trained with a dataset produced from the above method will converge to the mixture distribution:* $p_s(\mathbf{x}) = \sum_i^{k_{sub}} w_i p_{sub_i}(\mathbf{x})$ *where* $w_i = P(J = j)$.

Based on this proposition, this method creates a variation of the original dataset, that gives different predefined importance to the different subsets (see Appendix G for details). The same proposition holds for individual recordings. The value function now for a GAN takes the following form:

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_s(x)}[\log D(x)] + \mathbb{E}_{z \sim p_Z(z)}[1 - \log D(G(z))] \tag{4}$$

3.4 Metrics

Measuring the quality of data produced by a GAN is a difficult task, since the definition of "realistic" data is inherently vague. However, it is necessary, because the performance of the front-end classifiers is not necessarily a direct measurement of how realistic the synthetic data are. In this subsection we introduce the metrics we use to measure the quality of the synthetic data.

T metric: Hyland et al. [10] introduce two empirical evaluation metrics for data quality: TSTR (Train on Synthetic Test on Real) and TRTS (Train on Real Test on Synthetic). Empirical evaluation indicates that these metrics are useful in our case, however each one has disadvantages. To solve some of these issues we combine them via taking their harmonic mean (in the Appendix F we explain problems with these metrics and reasons to use the harmonic mean):

$$T = \frac{2 * TSTR * TRTS}{TSTR + TRTS} \tag{5}$$

MMD: We chose the Maximum Mean Discrepancy (MMD) [13] measurement since other well-established measurements (e.g., log likelihood) are either not well suited for GAN assessment, because plausible samples do not necessarily imply high log likelihood and vice versa [28], or they are focused on images, like the inception score [26]. There is also a wide variety of alternative approaches [3], however we use the MMD since it is simple to calculate, and is generally in line with our visual assessment of the quality of the generated data.

We follow the method from [27] to optimize the applied MMD via maximizing the ratio between the MMD estimator and the square root of the estimator of the asymptotic variance of the MMD estimator (the t-statistic). Inspired by [10], we further separate parts of the real and synthetic datasets to MMD training and MMD test sets (each contains half real and half synthetic data points). To maximize the estimator of the t-statistic for the training data we run gradient descent to the parameters of our kernel (i.e., Radial Basis Function (RBF) with variance σ as parameter). Then we test the MMD measurement on the MMD test set with the parameters that have been optimized with the training set. In the next section we evaluate the data based on these metrics.

4 Evaluation

To analyze how well we can achieve our objectives with the two GAN architectures, we design three experiments. Before we describe these experiments and their results, we analyze in Section 4.1 the quality of the synthetic data with the T-metric, MMD, and visual inspection. In Sections 4.2-4.4 we present and analyze the experiments. Together with accuracy, specificity, and sensitivity, we use the kappa coefficient [7] as performance metric since it better captures the performance of two-class classification in a single metric than accuracy. For all experiments, the pre-processing of the data is minimal (Section 3.1) and we use a wide variety of relatively basic methods as front-end classifiers. This is because we want to focus on investigating the viability of GAN augmentation as a means of performance improvement for a general baseline case. However, the GAN augmentation is applicable to any type of data (e.g., pre-processed apnea data) and is independent of the front-end classifiers. For details about the GAN and the front-end classifiers parameters and design please refer to Appendix A.

4.1 Data Quality Evaluation

To measure the similarity between the synthetic and the real distribution we use the MMD and T-metric (see example in Figure 3). We execute the tests every 10 epochs during training. Both scores improve as the training procedure progresses, until they stabilize (with minor variation). The T-metric is more unstable with epochs with high score in the initial training phase. However,

after epoch 600, the performance of the metric stabilizes around 0.9. Similarly, the majority of MMD variations stop (with few exceptions) around epoch 400.

Another important criterion for recognizing whether the generated data are realistic is the visual inspection of the data. Although not as straightforward as for images, apnea and non-apnea data can be visually distinguished. In Figures 4

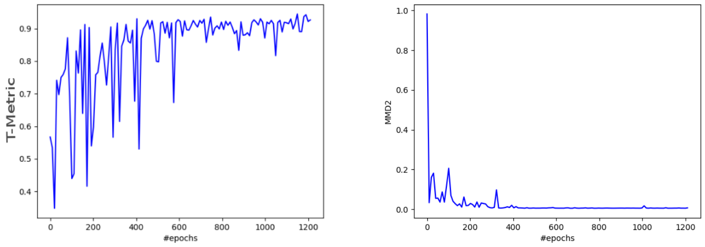

Fig. 3. Mean of T-metric (left) and MMD (right) scores throughout the GAN training

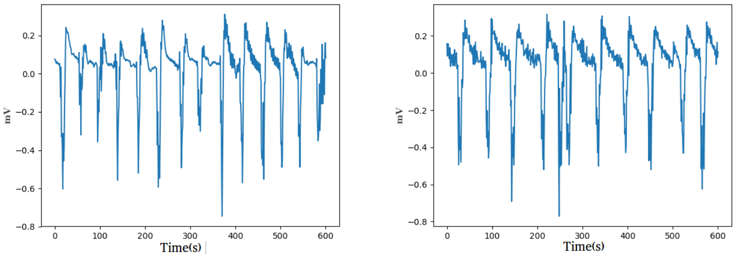

Fig. 4. Real apneic data (left) and good synthetic apneic data (right) for 600sec

and 5 we show examples of real and realistic-looking synthetic data. The generated data are realistic-looking and difficult to distinguish from the real data. For further evaluation of the visual quality and diversity of the real and generated data please refer to the Appendix H.

4.2 Experiment 1: Data Augmentation

In this experiment we investigate whether augmenting RD$_{TRAIN}$ with realistic SD generated from a GAN trained with the same RD$_{TRAIN}$ can have a positive impact on the front-end classifier performance.

Experiment Description: We iterate the experiment 15 times for Apnea-ECG and 10 times for MIT-BIH: We partition RD into RD$_{TRAIN}$ (with 50% of RD data points), RD$_{TEST}$ (25%) and a validation set (25%) via random subsampling. We train the GAN with RD$_{TRAIN}$. The GAN training is very unstable for the data of the two datasets (especially for MIT-BIH), and a good quality based on our metrics and visual inspection does not necessarily correspond to high performance of the front-end classifiers. For this reason, we use the validation dataset to evaluate the front-end classifier performance. We save the trained GAN model periodically throughout training, generate SD, augment RD$_{TRAIN}$, and measure the front-end classifier performance on the validation set. The GAN with the maximum validation set performance, and empirically acceptable MMD and T-metric values is chosen to generate SD. To obtain better performance for the MIT-BIH experiments with MLP and KNN, we concatenated data of many good performing models on the validation set, instead of only using the model with the best validation performance.

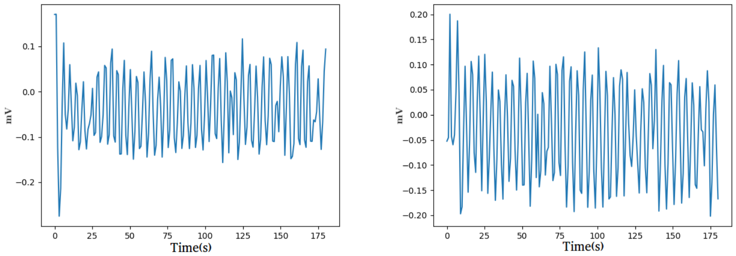

Fig. 5. Real (left) and good synthetic (right) non-apneic data , 175 sec

Results: Due to limited space we present in the main text only the kappa statistic for all front-end classifiers (Table 1), and the accuracy, sensitivity, and specificity for the MLP classifier (Table 2) to indicate the general behaviour we observe for all the classifiers. For accuracy, specificity, and sensitivity for KNN, RF and MLP please refer to Appendix B. We use this presentation convention for all experiments (Appendices C and D for Experiments 2 and 3 respectively).

Table 1. Kappa statistic and standard error for all front-end classifiers for Apnea-ECG and MIT-BIH. All kappa values are multiplied by 100 for legibility.

Kappa statistic ($X\cdot10^{-2}$) for Apnea-ECG (A), and MIT-BIH (M)				
	MLP	RF	KNN	SVM
A: Baseline	85.89±0.36	90.08±0.26	88.12±0.40	74.75±0.40
A: Exp1:Synth	78.29±0.97	83.88±0.56	85.76±0.49	75.04±0.55
A: Exp1:Augm	86.93±0.45	90.88±0.28	90.12±0.37	76.90±0.57
M: Baseline	25.04±0.88	30.95±1.10	27.15±1.01	0.0±0.0
M: Exp1:Synth	18.35±0.86	21.80±0.95	16.84±1.26	11.02±0.96
M: Exp1:Augm	27.01±0.61	33.01±0.87	29.22±1.01	14.93±1.22

Baseline shows the performance of the front-end classifiers trained only with RD_{TRAIN}. For the synthetic case (*Exp1:Synth*) they are trained only with SD, and for the augmented case (*Exp1:Augm*) with RD_{TRAIN} and SD.

For Apnea-ECG, Exp1:Augm exhibits for all front-end classifiers a statistically significant improvement of the mean of the kappa statistic at $p = 0.05$. The p-value for the one-tailed two sample t-test relative to the Baseline is: p= 0.042 (MLP), p=0.035 (RF), p=0.005 (KNN), p=0.002 (SVM). Notice that SD yields a good performance on its own, and even surpasses the performance of the Baseline for the SVM. We assume that this is due to the better balancing of the synthetic data in relation to the real. In SD, 50% of the generated minutes are apneic and 50% non-apneic, whereas in RD_{TRAIN} approximately 62.2% are non-apneic and 37.8% are apneic depending on the random subsampling.

For MIT-BIH, Exp1:Augm shows in most cases a significant improvement of the kappa statistic values relative to the Baseline for all front-end classifiers when we perform the 2-sample one tailed t-test, i.e., p=0.012 (MLP), p=0.062 (RF), p=0.029 (KNN), and $p \simeq 0$ (SVM). The overall performance is very low, due to the very low data quality for this dataset. Since our pre-processing is minimal this is to be expected. Notice that the SVM actually does not learn at all for the Baseline case. In all the iterations we performed, it classifies all minutes as non-apneic. Interestingly, both for Exp1:Synth and Exp1:Augm, there is a big improvement for the SVM, since the algorithm successfully learns to a certain extent in these cases. We assume that this is due to the better class balance (more apneas present in the datasets of Exp1:Synth and Exp1:Augm). Generally, for MIT-BIH the augmentation seems to have a beneficial effect on performance.

Table 2. Accuracy specificity and sensitivity for the MLP classifier

MLP Classifier Apnea-ECG (A), and MIT-BIH (M)			
	Acc.	Spec.	Sens.
A: Baseline	93.19±0.17	94.78±0.19	90.83±0.39
A: Exp1:Synth	89.26±0.49	85.48±1.14	95.02±0.94
A: Exp1:Augm	93.66±0.20	94.62±0.24	92.28±0.46
M: Baseline	64.6±0.37	75.95±1.16	48.41±1.26
M: Exp1:Synth	59.76±0.5	61.6±2.58	57.17±3.16
M: Exp1:Augm	64.7±0.25	69.92±0.78	57.08±1.22

From Table 2 we notice that for Exp1:Augm, the MLP (both for MIT-BIH and Apnea-ECG) exhibits a clear improvement in sensitivity and a small drop in specificity. This pattern is present for all front-end classifiers. For Exp1:Augm there is always a clear improvement in sensitivity, and either a small increase or decrease in specificity. This is an important advantage in a healthcare context since sensitivity reflects the ability of a classifier to recognize pathological events. This observation serves as a motivation for Experiment 2.

Implications for OSA Detection: The goal of this experiment is to reflect a real application scenario in which we have relatively equal amount of data from different patients to train with, and we perform classification for these patients. An example could be mobile OSA detection for patients after monitoring. It serves as an indication that augmentation with synthetic data can yield performance improvements for classifiers that are trained with the goal of OSA detection.

4.3 Experiment 2: Rebalancing Skewed Datasets

To analyze how well the single GAN architecture can be used to rebalance a skewed dataset, we need to create a skewed dataset because Apnea-ECG is nearly balanced with a ratio of 62.2% non-apneic and 37.8% apneic.

Experiment Description: We separate RD into RD$_{TRAIN}$ and RD$_{TEST}$ on a per-recording basis instead of a per event-basis as in the previous experiment. We randomly choose one apneic and one non-apneic recording as RD$_{TEST}$ (i.e., a01 and b01 respectively), and as RD$_{TRAIN}$ we use the remaining six recordings. We choose to evaluate this scenario using Apnea-ECG since it is the dataset for which our front-end classifiers exhibit the better performance.

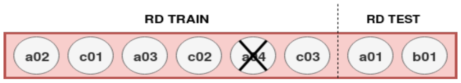

Fig. 6. Training and Test sets for Experiment 2

To create an unbalanced dataset, one apneic recording (i.e., a04 chosen randomly) is removed from the training dataset RD$_{TRAIN}$ (Figure 6) resulting in 72.2% non-apneic and 27.8% apneic training data. The augmentation in this experiment rebalances the classes to 50% apneic and 50% non-apneic. This means that we only generate apneic data with the GAN (i.e., SD contains only apneic minutes) and combine them with the original dataset to form AD.

Table 3. Kappa statistic and standard error for all front-end classifiers.

Exp2: Kappa statistic (X·10^{-2}) a01b01-unbalanced				
	MLP	RF	KNN	SVM
Baseline	88.44±0.54	91.92±0.26	93.16±0.16	74.6±0.2
Exp2:Augm	93.40± 0.63	94.56±0.16	94.76±0.45	92.88±0.64

Note that a04 is removed from the training set both for the baseline/augmented training of the front-end classifiers and also for the training of the GAN, i.e., the apneic minute generation relies only on the other two apneic recordings. A validation set is extracted from a01 and b01. Throughout the training of the GAN the validation set is periodically evaluated by the front-end classifiers which are trained each time with AD. We choose the model that generates the SD with which the front-end classifiers perform the best on the validation set. For this experiment we perform five iterations.

Results: The results are shown in Tables 3 and 4. For Exp2:Augm we train the front-end classifiers with AD (i.e., apneic SD and RD$_{TRAIN}$ without a04), and for the Baseline we train with RD$_{TRAIN}$ without a04. In both cases we evaluate on RD$_{TEST}$. Compared to Baseline, a performance improvement occurs

Table 4. Accuracy, specificity and sensitivity for MLP

Exp2: MLP a01b01-unbalanced Acc.,Spec.,Sens.			
	Acc.	Spec.	Sens.
Baseline	94.22±0.27	99.44±0.09	89.12±0.44
Exp2:Augm	96.70±0.31	98.82±0.24	94.62±0.51

for Exp2:Augm. This can be noticed in terms of accuracy for the MLP (Table 4, first column) and in terms of kappa for all front-end classifiers (all columns of Table 3). The SVM seems to benefit the most from the rebalancing process. Again, in terms of specificity and sensitivity we notice a similar behaviour as in the previous experiment, i.e., increased sensitivity and stable specifity.

To further evaluate the potential of the proposed technique we compared the results with results when training with the Synthetic Minority Over-sampling Technique (SMOTE) [4]. For all classifiers the proposed method is marginally to significantly superior (i.e.,MLP: $88.7\pm0.25\cdot10^{-2}$, SVM: $90.9\pm0.41\cdot10^{-2}$, KNN: $94.54\pm0.36\cdot10^{-2}$, RF: $93.42\pm0.27\cdot10^{-2}$).

Implications for OSA Detection: OSA data are generally very unbalanced towards non-apneic events. This experiment implies that GAN augmentation with synthetic data can be used to efficiently rebalance OSA data. This has a positive effect on the detection of apneic events and on the overall classification performance for OSA detection, based on the classifiers we experimented with.

4.4 Experiment 3: Personalization with Multiple GANs

In this experiment, we analyze whether we can improve performance by indirect personalization during GAN training. By *Personalization* we mean that we aim to make the learned distribution of the GAN to approach the specific distribution of the RD_{TEST} for a given proximity metric (MMD). Since we do not use a01 and b01 for the training of the GAN the method we apply is indirect. We use a01 and b01 from Apnea-ECG as RD_{TEST}.

Experiment Description: Based on the discussion in Section 3.3, we separate our training recordings into three subsets (Figure 7). Then we create three GANs (GAN 1, GAN 2, and GAN 3) and we use each subset to train the respective GAN, with a non-zero probability of choosing another subset for the gradient update based on a weighted dice toss (see Section 3.3). We set $p = 0.4$ (see Figure 2), i.e., for one gradient update of GAN 1, the mini-batch is selected with probability 0.4 from Subset1, and probability 0.3 from Subset 2 and 3. We do the same for GAN 2 and 3. The choice of p is made via experimental evaluation.

Proposition 1 implies that through this training, a GAN converges to a mixture of distributions with weights for each subset distribution j equal to $P(J = j)$ (see Eq. 4). By controlling $P(J = j)$ we control the weights of the mixture, and thus the degree to which each subset of recordings is represented in SD.

We use the validation set from a01 and b01 (obtained as in Experiment 2) for two purposes: (1) to evaluate the SD from the three GANs (SD 1, SD 2 and

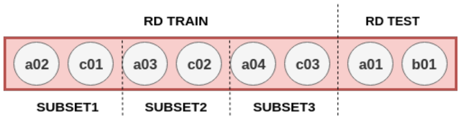

Fig. 7. Training and Test sets for Experiment 3

Table 5. Kappa statistic for front-end classifiers

Exp3: Kappa statistic (X·10^{-2}), a01b01 as RD_{TEST}				
	MLP	RF	KNN	SVM
Baseline	92.36±0.37	92.88±0.38	93.12±0.21	88.20±0.37
Exp3:Augm	93.08±0.59	93.6±0.62	94.50±0.39	91.72±0.94
Exp3:AugmP	93.36±0.40	94.36±0.31	94.58±0.17	93.92±0.23

SD 3) and (2) to calculate the MMD between SD 1-3 and this validation set. We examine two cases: In Exp3:Augm, SD 1, SD 2, and SD 3 are combined with RD_{TRAIN} to form AD. SD 1, SD 2, and SD 3 combined have the same size as RD_{TRAIN}. In Exp3:AugmP, we identify the SD that has the lowest MMD in relation to the validation set, and use the corresponding GANi to generate more data until SDi has the size of RD_{TRAIN}. AD is formed by combining RD_{TRAIN} and SDi. In Exp3:AugmP, we perform indirect personalization, since the SDi selected originates from the GAN that best matches the distribution of RD_{TEST} based on the MMD metric. This occurs since the validation set is also extracted from a01 and b01. The experiment is repeated 5 times.

Results: The results are found in Tables 5 and 6. We see that the general behavior is similar to the previous experiments. Again there are improvements for the augmented cases in relation to the Baseline. There are improvements in sensitivity and a small drop in specificity for the MLP cases, which is the case also for the other classifiers (with the exception of RF).

Generally, Exp3:AugmP, exhibits slightly better performance both in terms of kappa and accuracy. SVM and RF seem to gain the most benefits from this approach. Interestingly, in Exp3:AugmP SVM surpasses MLP in terms of kappa.

Table 6. Accuracy, specificity and sensitivity for MLP

Exp3: MLP a01b01 Acc.,Spec.,Sens.			
	Acc.	Spec.	Sens.
Baseline	96.18±0.18	98.92±0.07	93.54±0.25
Exp3:Augm	96.54±0.29	98.4±0.19	94.74±0.51
Exp3:AugmP	96.68±0.20	98.64±0.18	95.2±0.25

To further investigate the viability of Exp3:AugmP method we examine in Appendix E different recording combinations as RD_{TEST} (i.e., a02c01, a04b01 and a03b01) and perform Baseline and Exp3:AugmP evaluations for the front-end classifiers. For all cases, for all front-end classifiers we notice improvements for the kappa statistic, that vary from (RF, a02c01):$0.28 \cdot 10^{-2}$ to (MLP, a03b01): $27.12 \cdot 10^{-2}$, especially for low performing cases, e.g., for the (MLP, a03b01) case Baseline kappa is $57.4 \cdot 10^{-2}$ and Exp3:AugmP kappa is $84.5 \cdot 10^{-2}$.

Implications for OSA Detection: This experiment implies that personalization can indeed have a positive impact on classification performance for the detection of OSA. Even the simple indirect approach of Exp3:AugmP exhibits performance advantages for all front-end classifiers in relation to when it is not applied in Exp3:Augm.

5 Conclusion

In this work we examined how dataset augmentation via the use of the GAN framework can improve the classification performance in three scenarios for OSA detection. We notice that for all the cases the augmentation clearly helps the classifiers to generalize better. Even for the simpler classifiers like KNN, we see that augmentation has a beneficial effect on performance. The largest performance improvement is achieved for the SVM for Experiment 2, and in all the cases the metric that increases the most is sensitivity. This leads us to believe that the class balancing that GAN can provide with synthetic data can be useful in situations for which one class is much less represented than others. This is even more pronounced in cases like OSA detection where the vast majority of the data belongs to one of two classes.

As a next step we plan to investigate the viability of creating synthetic datasets that are differentially private. As health data are in many cases withheld from public access, we want to investigate the performance of front-end classifiers when using synthetic datasets that have privacy guarantees and examine how this impacts the performance of the classifiers. Additionally, for the NAF signal, the Apnea-ECG dataset contains only severe apneic or non-apneic patients, and MIT-BIH is generally too noisy. Thus, we want as a next step to investigate different datasets that contain more patients with average AHIs so that the GAN can also map transitional OSA states that are realistic. This could potentially help a classifier to further capture apneic characteristics from a wider range.

Acknowledgement: This work was performed as part of the CESAR project (nr. 250239/O70) funded by The Research Council of Norway.

References

1. https://physionet.org/physiobank/database/apnea-ecg/ (1999), [Online; accessed 26-3-2019]
2. https://physionet.org/physiobank/database/slpdb/ (1999), [Online; accessed 26-3-2019]
3. Borji, A.: Pros and cons of gan evaluation measures. Computer Vision and Image Understanding **179**, 41–65 (2019)

4. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: synthetic minority over-sampling technique. Journal of artificial intelligence research **16**, 321–357 (2002)
5. Che, Z., Cheng, Y., Zhai, S., Sun, Z., Liu, Y.: Boosting deep learning risk prediction with generative adversarial networks for electronic health records. In: 2017 IEEE International Conference on Data Mining (ICDM). pp. 787–792. IEEE (2017)
6. Choi, E., Biswal, S., Malin, B., Duke, J., Stewart, W.F., Sun, J.: Generating multi-label discrete patient records using generative adversarial networks. arXiv preprint arXiv:1703.06490 (2017)
7. Cohen, J.: A coefficient of agreement for nominal scales. Educational and psychological measurement **20**(1), 37–46 (1960)
8. Douzas, G., Bacao, F.: Effective data generation for imbalanced learning using conditional generative adversarial networks. Expert Systems with applications **91**, 464–471 (2018)
9. Durugkar, I., Gemp, I., Mahadevan, S.: Generative multi-adversarial networks. arXiv preprint arXiv:1611.01673 (2016)
10. Esteban, C., Hyland, S.L., Rätsch, G.: Real-valued (medical) time series generation with recurrent conditional gans. arXiv preprint arXiv:1706.02633 (2017)
11. Goodfellow, I., Bengio, Y., Courville, A.: Deep learning. MIT press (2016)
12. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in neural information processing systems. pp. 2672–2680 (2014)
13. Gretton, A., Borgwardt, K., Rasch, M., Schölkopf, B., Smola, A.J.: A kernel method for the two-sample-problem. In: Advances in neural information processing systems. pp. 513–520 (2007)
14. Hoang, Q., Nguyen, T.D., Le, T., Phung, D.: Multi-generator generative adversarial nets. arXiv preprint arXiv:1708.02556 (2017)
15. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
16. Hwang, U., Choi, S., Yoon, S.: Disease prediction from electronic health records using generative adversarial networks. arXiv preprint arXiv:1711.04126 (2017)
17. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1125–1134 (2017)
18. Kristiansen, S., Hugaas, M.S., Goebel, V., Plagemann, T., Nikolaidis, K., Liestøl, K.: Data mining for patient friendly apnea detection. IEEE Access **6**, 74598–74615 (2018)
19. Løberg, F., Goebel, V., Plagemann, T.: Quantifying the signal quality of low-cost respiratory effort sensors for sleep apnea monitoring. In: Proceedings of the 3rd International Workshop on Multimedia for Personal Health and Health Care. pp. 3–11. ACM (2018)
20. Mariani, G., Scheidegger, F., Istrate, R., Bekas, C., Malossi, C.: Bagan: Data augmentation with balancing gan. arXiv preprint arXiv:1803.09655 (2018)
21. Mirza, M., Osindero, S.: Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784 (2014)
22. Mogren, O.: C-rnn-gan: Continuous recurrent neural networks with adversarial training. arXiv preprint arXiv:1611.09904 (2016)
23. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434 (2015)

24. Ravì, D., Wong, C., Deligianni, F., Berthelot, M., Andreu-Perez, J., Lo, B., Yang, G.Z.: Deep learning for health informatics. IEEE journal of biomedical and health informatics **21**(1), 4–21 (2017)
25. Rezaei, M., Yang, H., Meinel, C.: Recurrent generative adversarial network for learning imbalanced medical image semantic segmentation. Multimedia Tools and Applications pp. 1–20 (2019)
26. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved techniques for training gans. In: Advances in neural information processing systems. pp. 2234–2242 (2016)
27. Sutherland, D.J., Tung, H.Y., Strathmann, H., De, S., Ramdas, A., Smola, A., Gretton, A.: Generative models and model criticism via optimized maximum mean discrepancy. arXiv preprint arXiv:1611.04488 (2016)
28. Theis, L., Oord, A.v.d., Bethge, M.: A note on the evaluation of generative models. arXiv preprint arXiv:1511.01844 (2015)
29. Tolstikhin, I.O., Gelly, S., Bousquet, O., Simon-Gabriel, C.J., Schölkopf, B.: Adagan: Boosting generative models. In: Advances in Neural Information Processing Systems. pp. 5424–5433 (2017)
30. Traaen, G.M., Aakerøy, L., et al.: Treatment of sleep apnea in patients with paroxysmal atrial fibrillation: Design and rationale of a randomized controlled trial. Scandinavian Cardiovascular Journal (52:6,pp. 372-377), 1–20 (January 2019)
31. Yu, L., Zhang, W., Wang, J., Yu, Y.: Seqgan: Sequence generative adversarial nets with policy gradient. In: Thirty-First AAAI Conference on Artificial Intelligence (2017)