Experimenting With Generative Adversarial Networks to Expand Sparse Physiological Time-Series Data

Martin BAUMGARTNER[a,1], Alphons EGGERTH[a], Andreas ZIEGL[a], Dieter HAYN[a] and Günter SCHREIER[a]
[a] *AIT Austrian Institute of Technology, Graz / Vienna, Austria*

Abstract. Machine Learning research and its application have gained enormous relevance in recent years. Their usage in medical settings could support patients, increase patient safety and assist health professionals in various tasks. However, medical data is often sparse, which renders big data analytics methods like deep learning ineffective. Data synthesis helps to augment small data sets and potentially improves patient data integrity. The presented work illustrates how Generative Adversarial Networks can be applied specifically to small data sets for enlarging sparse data. Following a state-of-the-art analysis is conducted, experimental methods with such networks are documented, which have been applied to three different data sets. Results from all three sets are presented and take-away messages are summarized. Concluding, the results' quality and limitations of the work are discussed.

Keywords. data analysis, deep learning, neural networks, statistical models

1. Introduction

In recent years, the application of artificial intelligence (AI) has gained significance in various fields of research. Especially machine learning (ML) has proven to be useful for recognizing patterns, predicting events or detecting anomalies. It is unsurprising, that these techniques have been applied to medical research as well.

1.1. Motivation

The usefulness of ML in medical contexts has not only been widely proven in recent academic studies [1–3], but also industry has realized its potential, with big players such as IBM [4], Google [5] and Microsoft [6] being very active in this field. The applications and executions of ML in medicine are diverse, as shown by the examples above. However, ML inherently requires one particular aspect for virtually all use cases: access to large quantities of data. Medical applications are not exempted from this situation. In fact, acquiring or even just accessing medical data constitutes an even more challenging task than in other settings. The European Union's General Data Protection Regulation (GDPR) classifies medical data not only as personal data but also as sensitive data due to the privacy aspects involved. Therefore, health professionals and patients alike are

[1] Corresponding Author: Martin Baumgartner, AIT Austrian Institute of Technology, Giefinggasse 4, 1210 Vienna, Austria, E-Mail: martin.baumgartner@ait.ac.at.

often rightly hesitant and skeptical of providing data for ML researching purposes, especially in the current times' increased focus on data protection and privacy concerns. Synthesizing medical data thus has two inherent advantages: Firstly, the issues of data sparsity and diversity can be addressed by simply creating artificial data. Secondly, the artificial data is highly anonymized and can be used freely without privacy concerns as the data is not from real patients.

1.2. State of the art

The idea of data synthesis is not entirely new, yet research is rather sparse: A very early example by McSharry et al. provides a univariate ECG generation tool using a dynamic mathematical model [7], which was also published on PhysioNet [8] in 2003. Subsequently, this was further developed into a Gaussian Wave-based model by G. Clifford [9], a co-author of the original paper.

Another, different example of data synthesis is Synthea [10]. Their goal is to generate not synthetic data of one specific type but rather entire patient records with realistic medical history. Synthea can model various diseases, focusing on the commonly occurring ones. This approach is helpful when analyzing correlations between diseases and assessing risk factors for future illnesses. They applied logic and rule-based workflows that are supported by huge amounts of real-world data and statistics.

However, more recently, the use of Generative Adversarial Networks (GAN) appears to show promising results. GANs were introduced by Ian Goodfellow et al. in 2014 [11], which fundamentally revolutionized data synthesis research. GANs are constructed with two artificial neural networks, that compete against each other. The first model – called discriminator – is provided with real examples and learns to distinguish between real and fake samples. The second model – called generator – mathematically transforms random noise to create fake data samples that are intended to fool the discriminator. These models learn in cycles and the constant improvement by each model stimulates the learning progress of the other, which stimulates the generator model to produce samples that match the real samples' statistical properties. In theory, once the discriminator accuracy is at 50% - which means, the discriminator cannot distinguish between real and fake – the training process is complete. Since their introduction, GANs have been used for medical data generation. In 2017, a research group developed medGAN, that allows for the generation of multi-label patient records [12]. Another study created synthetic MRI images using GANs [13], which resulted in an improved analysis performance by augmenting a sparse dataset. In 2019, two articles were published, which used similar model architectures to generate artificial ECG signals [14, 15].

All of the examples listed above utilized access to a large-scale dataset. GANs can be understood as a form of deep learning, which traditionally requires high amounts of training data to learn effectively, and thus the demand for extensive data sets is natural. In terms of augmenting sparse datasets, our literature research, revealed only very limited results. A Chinese group enlarged a set of hepatocellular carcinoma samples varying in stages [16]. In total, they had 78 data samples at their disposal. They used an improved version of GANs, the Wasserstein GAN, which was introduced in 2017 by Martin Arjovsky et al. [17]. The difference is an updated loss function (Wasserstein distance), which estimates the models' loss as a measure of "*realness*", which is intended to improve and stabilize the GAN training process. In a non-medical context, another study used 1000 samples – which in a medical scenario would not be considered sparse - in

their investigations into the application of GANs for sparse data sets [18]. Other studies have claimed to work with limited data, yet the amount of training samples used to train the GAN remain undisclosed, to our best knowledge [19, 20].

1.3. This work

This paper documents the experimental work of synthesizing univariate and multivariate data for a medical context using GANs. The emphasis was on applying GANs on small (n < 300) data sets, that are distinctively different from each other. The question to be answered was whether GANs can be used to augment limited data pools to improve machine learning algorithms and whether a "general" GAN can be designed for all types of time-series. Different data pre-processing steps, model architectures and training parameters were used to establish which aspects of GANs are especially important for their application in this context. The focus of this paper was also to highlight important lessons, that might provide valuable information for the research community.

2. Methods

All code was written in Python 3.7. and machine learning model building was done with TensorFlow and Keras.

2.1. Data sources and preprocessing

Three different data sources were used for this work, yet all three of them were of small sample sizes (n < 300). Two univariate and one multivariate time-series datasets were subject of synthetization with GANs. All three data sets are summarized and detailed in Table **1** below. Data set 2 and 3 were taken from a previously conducted clinical study.

1) *ECG data:* The first dataset was comprised of ECG data [21] taken from the publicly available PhysioNet [8] repository "ECG-ID Database". This dataset includes short-term single-lead ECG data from 90 subjects sampled at 500Hz with 12bit resolution. In preprocessing, data was downsampled to 250Hz to reduce computational burden. Further, the samples were separated into sequences of 700 datapoints (2.8s) to normalize the samples' format. The ECG data was not scaled, as the signals naturally range rather reliably between +1 and -1mV.
2) *Timed Up-and-Go test data:* The second dataset consisted of space-time data from a timed up-and-go (TUG) test device, developed at the Austrian Institute of Technology (AIT) [22]. It contained 138 samples of various lengths, which have been time-normalized by downsampling and interpolating to a uniform sequence length of 150, which was the rounded median length of the sequences in the data source. The data samples were scaled individually so that their values range between 0 and 1. Three samples needed to be excluded due to sensor errors, which results in corrupted and unusable data.
3) *Telehealth data:* The third dataset has multivariate telehealth data from real patients, who recorded systolic and diastolic blood pressure, heart rate, bodyweight as well as a patient-assessed comfort score. These measures were recorded once a day normally, however in these samples longer breaks of no data or multiple values for one measure existed, usually due to hospital stays, vacations or equipment failures. These gaps, however, are not to be seen as mistakes or errors in this context, in fact they are important features and the GANs are expected to also simulate such gaps.

The time-domain of this data was treated as a sixth channel of the data to realize this. Recordings of one day, but at different times were consolidated into one value and the data was separated into sequences of 40 data points to achieve a uniform data format.

Table 1. Summary of the three used data sets for experimental data synthesis on small datasets.

	ECG data	TUG data	Telehealth data
Subjects	90	138	105
Data samples	247	135	278
Sequence length	700	150	40
Channels	1	1	6

2.2. Generative Model Architecture

During the experiments, a multitude of different model architectures were investigated. While all models used two artificial neural networks (ANN), their nature varied in different trials. The following three types of networks were used either exclusively or in combination with each other: multilayer perceptrons (MLP), convolutional neural networks (CNN) and recurrent neural networks (RNN). In this section, only the final models will be depicted, further discussion and lessons learned can be found in the *Discussion* chapter.

As all datasets used in this work were of different nature, complexity and dimension, different GANs had to be used in order to produce the wanted outcome. GANs always consist of a generator model and a discriminator model, which in general might use different model architectures. In this work, however, for each of the three datasets CNNs were used, despite the possibly differences in the underlying distributions. The following section will give more detailed information on the model architectures in the final experiments.

All generators followed the same scheme: They were initiated with a fully-connected layer where the amount of nodes matched the sequence length multiplied with the number of channels. The telehealth data generator was comprised of more input nodes to match the increased complexity of multivariate data. Afterwards, a varying number of convolutional layers followed, each partnered with a layer of batch normalization (BN) and a linear activation unit (ReLU). The final output layer was activated with a tanh activation function.

In all three use cases, the same discriminator was used. Like the generator model, the discriminator was also utilizing a convolutional neural network architecture. After three one-dimensional convolutional layers, a flattening layer was applied to prepare the data for the final fully-connected layer with only one node for binary classification (real / fake) with sigmoid activation function. **Figure 1** depicts details of the used model architectures. In all three applications the Adam optimizer was used with a learning rate of 0.0002 and decay parameter β_1 of 0.9. Binary crossentropy was used as the loss function in all models. As described earlier, the generators transformed random noise of a certain dimension – sometimes referred to as "*latent dimension*" – which they are provided with. In the TUG and telehealth data case, a length of 100 was chosen as latent dimension and the ECG generator was provided with 1000 random noise values to address the longer sequence length. All models were trained for 10000 epochs, during which both generator and discriminator are updated for different amount of times, establishing a ratio of learning effort between discriminator and generator. The ECG

GAN was trained in a 1:3 (discriminator:generator) ratio, the TUG GAN was trained in a 1:1 ratio and the telehealth GAN in a 2:4 ratio.

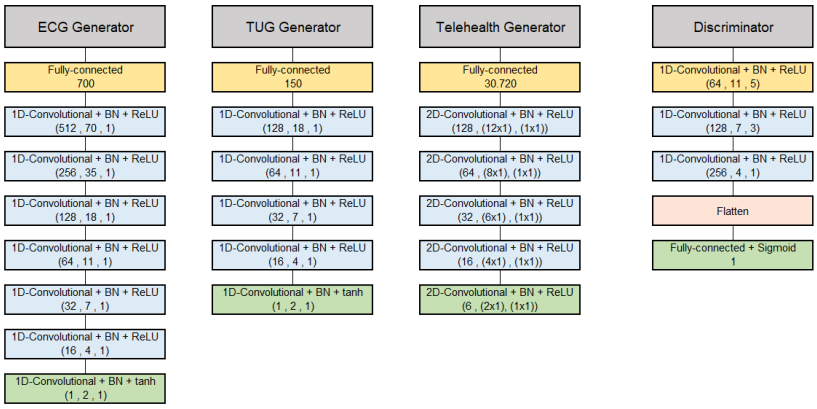

Figure 1. Detailed architectures of the three models. Convolutional layers are described in the format (filters, kernel size, strides).

3. Results

This chapter features examples of synthesized data created from all three data sets. It should be noted, that the following pictures shown are not from the GANs' final training iteration. The pictures in **Figure 2**, **Figure 3** and **Figure 4** depict good examples, showing what the GANs are able to achieve at their top performance.

3.1. Synthesized Data Examples

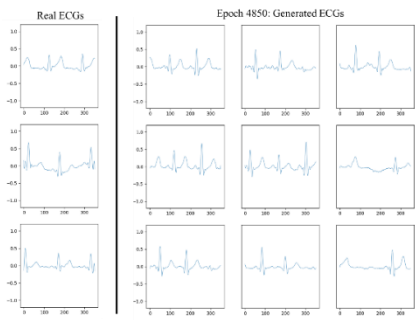

Figure 2. Generated ECG data after 4850 epochs of training. On the left, three randomly chosen real examples are depicted, while the plots on the right show nine generated ECG signals.

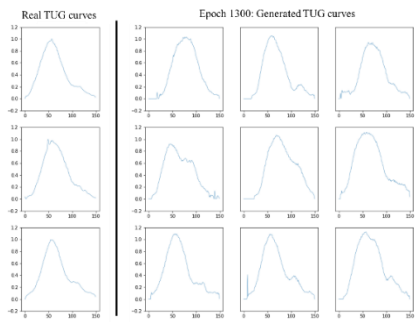

Figure 3. Generated TUG curves after 1300 epochs of training. On the left, three randomly chosen real examples are depicted, while the plots on the right show nine generated TUG signals.

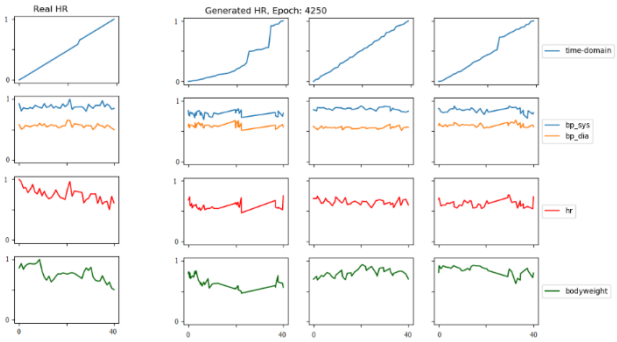

Figure 4. Generated multivariate telehealth data after 4250 GAN training cycles. On the left, one real example is shown, while the on the right three generated examples are depicted.

One should keep in mind, that the different amounts of training cycles are not directly comparable, as the internal ratio of training the networks lead to different numbers of training total training iterations. For example, after 100 iterations, the ECG generator was trained 300 times, the TUG generator 100 times and the telehealth generator 400 times.

3.2. Lessons Learned

As model architectures are concerned, with our data convolutional networks showed the best results with our data. The simpler multilayer perceptrons did not produce satisfactory signals, regardless of the number of nodes or the model depth. Perhaps, they simply lack the computational power to find the necessary relations in the data. Despite them being developed specifically for time-series and also being applied successfully as GAN generators [14, 15], RNNs did also not generate promising results. They have been applied as discriminators successfully, but they proved to be more computationally expensive without being superior in result. As generators both simple recurrent layers as well as long short-term-memories (LSTM) have not produced the wanted outcome. The main issues were extreme training instability, frequent mode collapse and noisy results.

Although, in other studies the application of Wasserstein GANs has been reported to improve training stability and results [16, 17], this was not reproducible with the data from this study. Regardless of model architecture or model complexity, Wasserstein GANs delivered worse results at higher computational cost. It can be speculated that their application is best for image generation.

As the layer design is concerned, the application of convolutional layers with a large kernel size (> 10% of sequence length) have drastically improved the quality of outcome. It appears, that these layers are responsible for the overall structure of the generated samples, designing where exactly the extracted features are located in time.

The selection of the latent dimensions' size was also found to impact the outcome. A high latent dimension constitutes more degrees of freedom for the generator, which results in a higher necessary training effort but also in more possibilities of generation. A balance needs to be found that gives the generator enough freedom to transform signals without providing an overwhelming amount of degrees of freedom.

The internal training ratio also has proved to be a critical adaptation. It is a necessary measure to establish a balance between generator and discriminator. As the generator is generally provided with a more challenging task, increasing its training cycles helped to prevent discriminator overpowering.

Despite all efforts, training instability could not be resolved entirely. In many cases, the GAN showed steady and quality progress to achieve a satisfactory result, just to completely deteriorate in the next epoch, sometimes without ever recovering from it. Currently, it is unknown why this phenomenon occurred so frequently. Perhaps, this could be attributed to one of the networks overfitting and thus completely throwing the model off balance. Another theory is that in small data sets, the selection of provided samples is rather small and an unlucky batch sampling leads to the model learning irrational features.

Contrary to the statement made in the *Introduction* chapter, the discriminator accuracy proved to be inadequate for tracking the model's progression. In the final, satisfactory result, the accuracy often was not 50% as expected but far above it, reaching 100% in rare cases.

4. Discussion

4.1. Interpretation of Results

The synthetic ECGs in **Figure 2** show good qualities such as clear ECG features (P-waves, QRS-complexes and T-waves) in a meaningful and regular frequency in 7 of the 9 depicted examples. The commonly appearing issue of mode collapse – meaning the repeated generation of the exact same sample – has not been observed at this training level as the signals vary in amplitude (real-valued), frequency and phase shift. The two unsatisfactory signals appear to show the expected features, yet not in a regular fashion. This phenomenon was encountered during the trials with different model architectures, perfectly illustrated with the following example in **Figure 5** below.

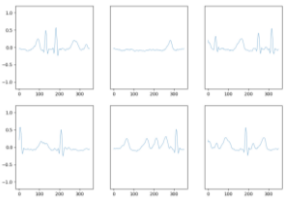

Figure 5. ECG data with alternative architectures. Signals show ECG features but in misplaced position.

This effect is attributed to a lack of temporal resolution – meaning that filter kernel sizes were big enough to correctly learn features but too small to place them into an overall correct time-series context.

The TUG data constitutes a rather simple signal, which also led to satisfactory results, when assessed through visual comparison by the authors. Its usefulness must be determined in a follow-up study by testing them on machine learning tasks.

In the telehealth data generation, the self-assessed comfort score was omitted during the synthetization due to its categorical nature ('good', 'okay, 'bad'), which proved to be too challenging in this context. This data set is more challenging to assess. While some obviously correct features are recognizable (e.g. correlation between systolic and diastolic blood pressure), the overall grade of realism is hard to determine. The GAN was able to reproduce the wanted jumps in the time-domain, simulating periods without data. Any other interpretations of the signals are speculation and thus the quality of this synthetic data will be assessed when it is used in further context.

4.2. Limitations and Outlook

One important take-away from our experiments is, that currently developing a "general" GAN suitable for different time-series is simply unrealistic. Our experiments showed, that GANs should be adapted to fit the individual use case.

The ultimate test for this synthetic data is whether it is beneficial to real machine learning tasks. Standard statistical evaluation is not applicable. Different similarity measures have been tried during this experimental process, yet no method was found that could generally be applied to such diverse data sets. The importance of developing a proper scoring system must not be underestimated and will be topic of a follow-up study. Two of three data sets are non-public, which reduces reproducibility of the results.

References

[1] T. J. Bright *et al.*, "Effect of Clinical Decision-Support Systems: A Systematic Review", *Annals of Internal Medicine* **157**(1) (2012) 29–43.

[2] A. Esteva *et al.*, "Dermatologist-level classification of skin cancer with deep neural networks", *Nature* **542**(7639) (2017) 115–118.

[3] P. Rajpurkar *et al.*, "CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning", *arXiv.org* (2017).

[4] M. N. Ahmed *et al.*, "Cognitive Computing and the Future of Healthcare", *IEEE Pulse* **8**(3) (2017) 4–9.

[5] J. Powles *et al.*, "Google DeepMind and healthcare in an age of algorithms", *Health and Technology* **7**(4) (2017) 351–367.

[6] Microsoft, "Project Hanover", https://www.microsoft.com/en-us/research/project/project-hanover/, last access: Jan-08-2020.

[7] P. E. McSharry *et al.*, "A dynamical model for generating synthetic electrocardiogram signals", *IEEE Transactions on Biomedical Engineering* **50**(3) (2003) 289–294.

[8] A. L. Goldberger *et al.*, "PhysioBank, PhysioToolkit, and PhysioNet", *Circulation* **101**(23) (2000) 215–220.

[9] O. Sayadi *et al.*, "Synthetic ECG generation and Bayesian filtering using a Gaussian wave-based dynamical model", *Physiological measurement* **31**(10) (2010) 1309–1329.

[10] J. Walonoski *et al.*, "Synthea: An approach, method, and software mechanism for generating synthetic patients and the synthetic electronic health care record", *Journal of the American Medical Informatics Association* **25**(3) (2017) 230–238.

[11] I. Goodfellow *et al.*, "Generative Adversarial Nets", in *Advances in Neural Information Processing Systems 27*, 2014, 2672–2680.

[12] E. Choi *et al.*, "Generating Multi-label Discrete Patient Records using Generative Adversarial Networks", *Journal of Machine Learning Research* **68** (2017) 1–20.

[13] H.-C. Shin *et al.*, "Medical Image Synthesis for Data Augmentation and Anonymization Using Generative Adversarial Networks", in *2018 Workshop on Simulation and Synthesis in Medical Imaging*, 2018, 1–11.

[14] F. Zhu *et al.*, "Electrocardiogram generation with a bidirectional LSTM-CNN generative adversarial network", *Nature Scientific Reports* **9**(1) (2019) 6734.

[15] A. M. Delaney *et al.*, "Synthesis of Realistic ECG using Generative Adversarial Networks", *arXiv.org* (2019) 1–16.

[16] Y. Liu *et al.*, "Wasserstein GAN-Based Small-Sample Augmentation for New-Generation Artificial Intelligence", *Engineering* **5**(1) (2019) 156–163.

[17] M. Arjovsky *et al.*, "Wasserstein GAN", *arXiv.org* (2017).

[18] Y. Wang *et al.*, "Transferring GANs: Generating images from limited data", *Lecture Notes in Computer Science* (2018) 220–236.

[19] S. Gurumurthy *et al.*, "DeLiGAN : Generative adversarial networks for diverse and limited data", in *30th IEEE Conference on Computer Vision and Pattern Recognition*, 2017, 4941–4949.

[20] M. Hu *et al.*, "Exploring Bias in GAN-based Data Augmentation for Small Samples", *arXiv.org* (2019).

[21] T. S. Lugovaya, "Biometric human identification based on electrocardiogram", University "LETI" Saint-Petersburg, 2005.

[22] A. Ziegl *et al.*, "Timed Up-and-Go Device for Unsupervised Functional Assessment of Elderly Patients", in *eHealth2017 - Health Informatics meets eHealth*, 2017.