Reconstructing Sparse Wearable Blood Volume Pressure Data using Deep Convolutional Generative Adversarial Networks

Victoria J Armstrong
School of Computing
Queen's University
Ontario, K7L 2N8
victoria.armstrong@queensu.ca

Abstract

This work explores the use of generative adversarial networks in order to reconstruct sparse time series data collected from wearable sensors. Building on successes of GANs for data augmentation and image generation, we have utilized a deep convolutional GAN to generate complete time series data from partially complete time series images. We have evaluated our model under two different lenses: the visual appearance of the reconstructed time series data, and a quantitative measure of how similar the real and corresponding fake time series data are. The model demonstrated promise, but failed to reconstruct time series data. If results were promising, a third testing criteria would have been completed to evaluate the ability of the generated time series data to behave like real data using a state of the art stress classifier. Explanations and future work are discussed to address the issues found in the study.

1 Introduction

Biological data collected from individuals can provide insights into medical conditions, activity levels and affective state [1]. One affective state of particular interest is individual stress levels which can be accurately predicted from physiological data including blood volume pressure (BVP), electrocardiogram (ECG), electroencephalogram (EEG), electromyography (EMG), skin temperature and acceleration [13]. Previously, such data was only able to be collected in clinical settings with stationary, expensive machinery.

With the emergence of wearables, wearable trackers have become prevalent. A number of current wearable technologies have the capability to track a wearer's physiological data to a high degree of accuracy. Using this time series data, machine learning models have been constructed to classify wearer stress to a high degree of accuracy [20]. While wearable data collection is promising, sensor readings are not always accurate and can lead to significant periods of time where data is either not collected or where it is inaccurate and must be discarded as an outlier.

While brief periods of missing data can be accommodated, large portions of missing data can interfere with stress classification. Different methods of data imputation exist to address this issue, but cannot capture the true nature of the time series data, particularly when dealing with sensitive sensor readings [4]. To address this issue, this study proposes harnessing the image generation capacity of generative adversarial networks (GANs), originally proposed by Goodfellow et al. in 2014 [5], to reconstruct large missing portions of data. Given the successes of deep convolutional GANs (DCGANs) [12], this study will utilize a DCGAN architecture to complete partially filled time series data images. The success of this method was evaluated in two ways: image similarity to the naked eye, as well as the

Preprint. Under review.

qualitatively measured difference between the real and generated image using the Frechet Inception Distance (FID) [6].

1.1 Key Contributions

This paper attempted to make the following contribution. Further work needs to be completed to reduce noise and mode collapse in the discriminator in order to produce high quality reconstructed data, as this current work was unable to produce realistic time series data.

- Encode partially filled biological sensor time series data images to a latent vector that is used by the generator to reconstruct the missing data.

2 Related Work

2.1 Generative Adversarial Networks

In 2014, Goodfellow et al proposed a new generative network called the Generative Adversarial Network (GAN) [5]. The general premise of a GAN is the use of a generator and a discriminator. The discriminator is trained to determine if a given sample is from the real distribution, and the generator creates images from a latent vector with the goal of fooling the discriminator [5].

In 2015, Radford et al. proposed the Deep Convolutional GAN, buliding off of Goodfellow et al's work. They focused on three specific CNN architecture changes. First, they use strided convolutions instead of spatial pooling [12]. Next, they addressed the elimination of fully connected layers on top of convolutional layers by feeding the output of the final convolutional layer directly to the discriminators output and the generators input [12]. Lastly, they added batch normalization layers to all but the final layer out the generator and the first layer of the discriminator [12]. It should also be noted that the authors found that while ReLU activation worked well for the generator, save for the final layer which uses Tahn, LeakyReLU should be used in all of the discriminator layers [12].

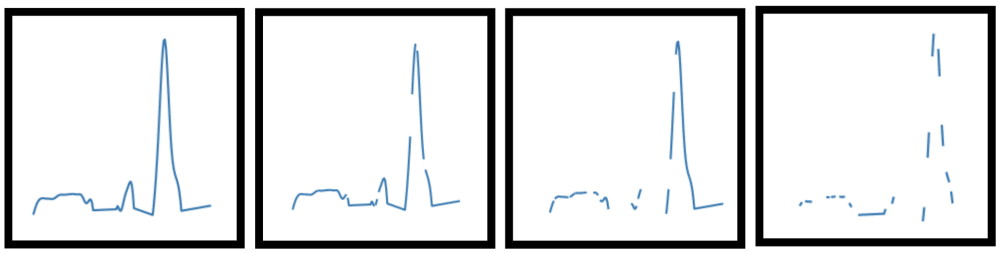
2.2 Time Series Data

Time series data is data that can be ordered chronologically. Time series data is present in a variety of different domains, such as stock market trends, traffic, and healthcare. In the health care field in particular, time series data on biological processes can be collected. As previously mentioned, this data can be used to predict affective state [20]. Various data sets exist that use physiological data with the purpose of detecting stress of affect such as WESAD [16], SWELL [19], ASCERTAIN [9], DREAMER [8] and AMIGOS [14]. A variety of work has been done using such data sets in order to classify periods of stress in users [11, 15, 2]. Of particular interest is the work by Elzeiny et al. which passes time series data images into a convolutional neural network and predicts user stress to a high degree of accuracy, achieving perfect results under some conditions [3].

While emotion recognition data from wearables is prevalent [20], sensor faults can leave chunks of data that with missing values. This can cause challenges in affect prediction if there is no data for a given period of time. Adding missing data back to the set is called data imputation and there are a number of different methods that exist for time series data imputation for deep learning. Naive statistical methods can be used, such as simply filling in the missing data with the mean or mode value [4]. Other more complicated algorithms for data imputation have been studied as well, including GANs with Gated Recurrent Units (GRUs) [4].

2.3 GANs for Time Series Applications

Smith and Smith used two WGANs in tandem to generate realistic time series data [18]. They minimized the Wasserstein distance, which is the distance between two probability distributions [18]. Testing on 70 different data sets they compared the performance of a model trained with real versus fake data and tested on real versus fake data for a total of 4 combinations (train true, test true; train true, test fake; train fake, test true; train fake, test fake) [18]. They demonstrated that time series data can successfully be generated from random noise.



(a) real BVP signal (b) 10% missing data (b) 20% missing data (b) 50% missing data

Figure 1: A sample BVP signal with different percentages of data missing, over a window size of 100

Huang et al. used a DCGAN to do data imputation on traffic data under different rates of missing data [7]. This relates strongly to the work being done in this study. They use a latent vector z sampled from a uniform distribution [7] to generate reconstructed time series data. This research expands on this work to include an encoder block in the generator to encode a sparse time series image. Furthermore, this paper aims to evaluate the generated images by looking at the similarity of the generated time series on more than RME and RMSE metrics.

Koochali et al. trained a conditional GAN to predict the conditional probability distribution of future values [10]. Their model was trained on two artificially generated time series data sets, as well as a real-world internet traffic dataset. The model was evaluated on it's ability to predict one-step ahead and was compared to a G-regression [10]. Their results demonstrated that ForGAN was better able to predict future values in some instances [10]. This relates to the work that will be completed here as it demonstrates the potential for time-series forecasting using generative adversarial networks, however this work will differ by investigating data imputation rather than generation.

3 Method

3.1 Data Set Construction

Data from the WESAD dataset [16] was used to train the network. The dataset contains a variety of biological signals collected from 15 participants under three conditions: stressed, amused and baseline. The data was collected from the Empatica E4, a write worn device, and the RespiBAN, a chest worn device.

Of the available signals, the Blood Volume Pulse (BVP) was used to train the network this project. Collected using photoplethsymography (PPG), where a infrared light is shines through the skin and measures changes in the blood volume present in the individual's arteries. From this BVP signal, heart rate and heart rate variability features can be extracted, which are strong indicators of stressed affect [16].

The BVP signal was collected from the non-dominant hand of the participant using PPG. The BVP was sampled at 64Hz, giving a total of 13,141 datapoints. Given the amount of data and resources available, only three of the total fifteen participants were used: participant 2, participant 3 and participant 4. The data was preprocessed by removing the negative values at the tail ends of the dataset that corresponded with taking on and the device. In addition, any missing values were removed, although this number made up a very small portion of the collected data.

Following this, the data was broken up into three different data sets of different window sizes of 100 datapoints, 200 datapoints and 500 datapoints over a sliding window with a length equal to half of the total image width (50, 100 and 250 respectively). In order to mimic missing data from a sensor fault, a random percentage of data was removed from each window chunk by randomly generating indices within the given window. Different percentages of data was removed: 10%, 20% and 50%. Figure 1, above, shows a BVP signal with different amounts of missing data.

These images were then saved as 64x64 pixel RGB images to make up the datasets. The images were loaded into pytorch using the Dataloader class. The images were centered, converted to a tensor and

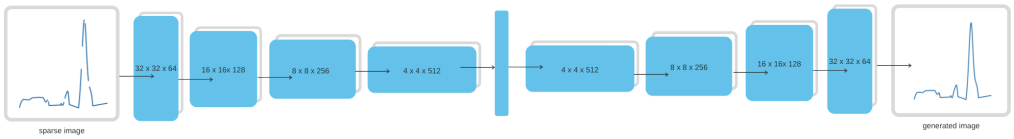


Figure 2: Generator architecture

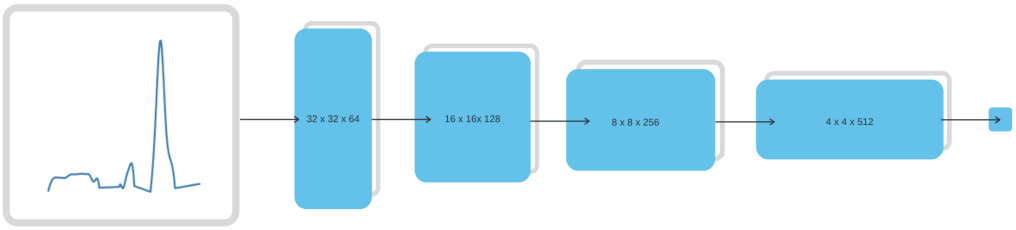


Figure 3: Discriminator architecture

then normalized between [-1, 1]. A 80-20 training to testing split was used to create the final data sets.

3.2 Architecture

This model used in this paper is based off of the Deep Convolutional GAN presented by Radford et al. [12] and (is comprised of a generator or comprises a generator) comprises of a generator and discriminator which work against each other, as was proposed in the initial GAN paper by Goodfellow et al. [5].

The generator is comprised (repitition of word from last sentence, unsure if you care?) of an encoder and decoder block. The encoder block taxes a 64x64 pixel image with 3 RGB channels and compresses it to a latent vector of size 100 by passing it through four convolutional blocks, each with batch normalization and ReLU activation. The decoder takes this latent vector and passes it through five two-dimensional strided convolutional transpose layers to output a 64x64x3 image. Each layer includes batch normalization and ReLU activation, with the final layer using a Tahn activation (following comma is a little awk, doesn't entirely align - I think it's use of "with" after the comma). Figure 2 shows the structure of the generator.

The discriminator uses four strided two-dimensional convolutional layers with batch normalization and leaky-ReLU activation functions in each layer except the last. A true image passed in to the discriminator with dimensions 64x64x3 and the final output of the discriminator uses a sigmoid activation function in order to classify the data as real or fake. Figure 3 shows the architecture of the discriminator.

The training of the model follows the algorithm outlined in the DCGAN paper [12], as developed by Goodfellow et al. [5]. The loss of the generator is calculated according to (1) and the loss of the discriminator is calculated according to (2).

$$\text{(1) generator\_loss} = log(D(x)) + log(1 - D(G(z)))$$

$$\text{(2) discriminator\_loss} = log(D(G(z)))$$

In order to avoid mode collapse, rather than us the single integer values 1 and 0 as labels for real and fake data respectively, the data was randomly labeled using float values. Values ranging between [0.0, 0.3] were used to label fake data and values between [0.7, 1.2] were used to label real data.

(1)

4 Evaluation

The model was trained for 100 epochs with a learning rate of 0.0002 and a batch size of 64. The Adam Optimizer was used for both the generator and discriminator, and binary cross-entropy was used as the loss function. Experimentation revealed these to be the best hyper parameters, however greater resources would allow for a more exhaustive hyper parameter search. The data used to train the model using a window of 100 data points and a data loss of 20%. If these results were promising, the following evaluation criteria would have been compared across window size and percentage of missing data.

Given the nature of time series data, the reconstructed time series images not only need to be visually passable as time series data, but they must also be as quantitatively similar to the training data as possible. Furthermore, the reconstructed data should behave similarly to the training data under classification tasks. These requirements led to three different evaluation criteria for the reconstructed data.

First, the data was visually compared to the training data to qualitatively compare. An important criteria in this visual comparison is continuity of the graph since the goal of the project is to impute the missing data. The next comparison is the relative similarity between the true image and the reconstructed image. This can be measured using the Frechet Inception Distance (FID) [6]. This metric is calculated using the mean and covariance from the final activation layer formula in (3). A perfect FID score is 0.0, and the greater the number, the less similar two images are to each other. The FID was calculated using code developed by Seitzer [17].

$$\text{(3) FID} = (\mu - \mu_w)^2 + tr(\Sigma + \Sigma_w - 2\sqrt{\Sigma\Sigma_w})$$

Lastly, the model should be evaluated on its performance using existing machine learning methods. This evaluation method was not completed, as the results indicated that the model was not accurately able to represent the data, rendering this particular evaluation method redundant. If the reconstructed data was sufficient, a network similar to that of Elzeiny et al.'s work should be constructed [3]. Afterwards, the network would be trained and tested four separate times in order to determine if the reconstructed data behaves similarly to the true data. First, the network would be trained on the true images and then evaluated on true images, and separately evaluated on reconstructed images. Similarly, the network would then be trained on reconstructed images and then evaluated on true images and reconstructed images. This would give four sets of metrics under the following conditions (training-testing): real-real, real-reconstructed, reconstructed-real, reconstructed-reconstructed. Comparing performance metrics would allow a comparison between the behaviour of the data sets. This method was used by Smith and Smith in their evaluation of WGANs [18].

5 Results

Overall, the results of this experiment showed that this model has potential, but was unable to produce accurate results. The following section summarizes what was determined and the conclusions and future work section addresses how these issues could be addressed. Figure 4 shows the final batch outputs of the network after 100 epochs. As is evident in the right hand side image, the generated time series data does resemble its real counter parts. These images also highlight the volatility of training a generative adversarial network, as a small tweak in a single hyper parameter has lead to large discrepancies in the data. Figure 6 shows the evolution of training from the very first epoch, the fiftieth and lastly the hundredth, demonstrating that while the network is not accurately representing the time series data, it is learning to some extent.

As can be seen in Figure 5, the discriminator loss collapsed to between (0.5 and 1.0) almost immediately, whereas the generator loss hovered fluctuated for the first 30 epochs but then settled into the range of (1.5, 2.5). The FID was calculated using the *pytorch_fid* repository by Seitzer [17]. First, the distance between real and real images was calculated to ensure the method was accurate. Confirming this, the FID between the final generated images from the training dataset was calculated against the real samples. The FID after 50 epochs was 350.850 and after 100 epochs was 342.766. This is decreasing, however not by a significant value. These scores indicate that the images are highly dissimilar from the original data, which is evident looking at the images. Furthermore, the FID for

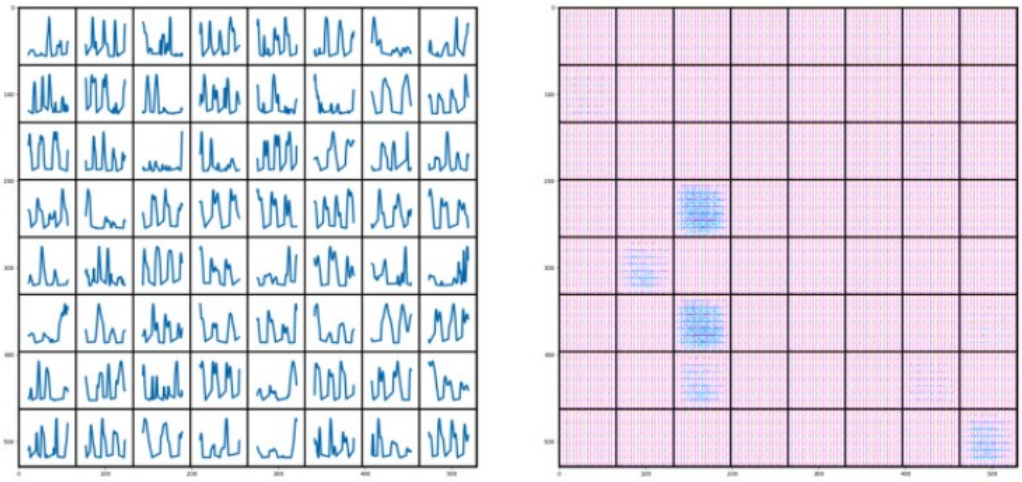


Figure 4: A sample of a batch of 64 real BVP samples (left) and the generator output (right) using the encoded sparse image.

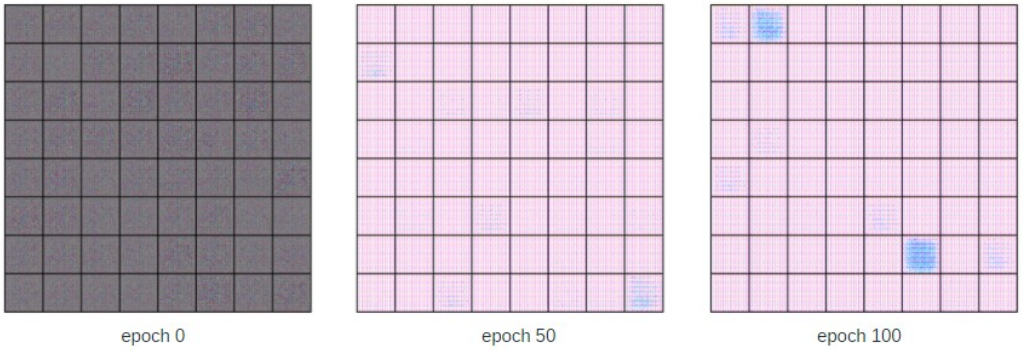


Figure 5: Evolution of training at three checkpoints.

the generated data from the first epoch was 258.284 despite being random noise. This could indicate that FID is not an accurate measure in this case, despite it's previous uses.

As mentioned in the evaluation, the results were poor, so the data was not used to train a simple classification task. This decision was made since the generated data does not resemble time series data so training would be redundant. Lastly, as the image reconstruction was unsuccessful, different window sizes and percentage of data missing was not reported in the results. Only a single window size of 100 data points and 20% missing were used. If the results were promising, all three of the outlined tests would have been completed using this data to see how the model performed with various levels of missing data. However, similarly to above, this evaluation criteria was redundant.

6 Conclusions and Future Work

Overall, this model was unable to reconstruct time series data from partially complete data. While this project allowed exploration of biological signal reconstruction, there are a number of limitations associated with the work. Namely, and most obviously, was the failure of the network to regenerate the data from the partially constructed data. While parameters were tuned to attempt to resolve this, time and resources did not allow a solution to be found. Given more time, exploration into a different number of layers, than what was presented in the initial DCGAN paper would be beneficial. Further exploration into the image prepossessing and encoding of the image could also give insights into how this may have impacted the negative performance of this model.

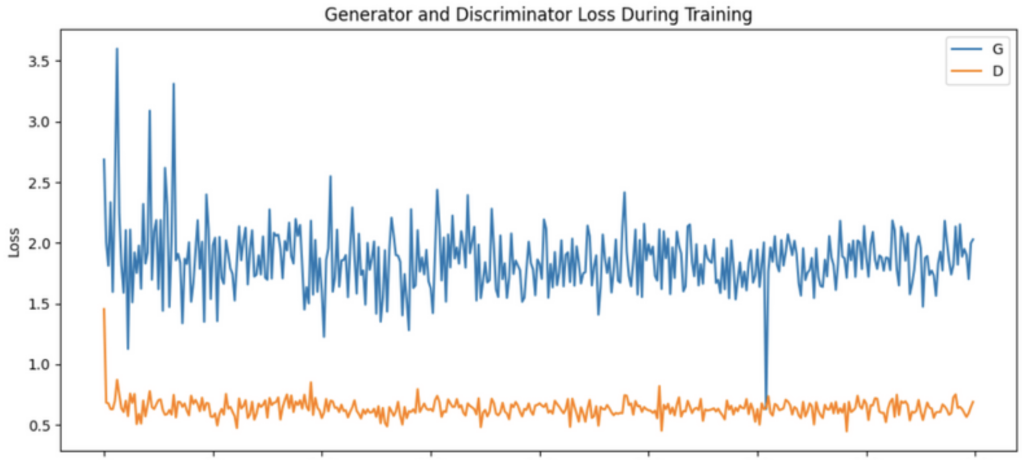


Figure 6: Generator and Discriminator Loss Over 100 epochs

One design decision that I would have changed would be the representation of the data as an array rather than an image. I also would have extracted the heart rate from the BVP signal and used this data instead of the raw BVP signal as I think this would allow the generated images to be used more readily without the need to perform peak detection after the fact. Lastly, if the model were to continue to be trained on images, I think the conversion of the images from RGB to grey-scale could impact the results and should be explored.

Future work would include further training and hyper-parameter tuning in order to achieve better results and perform the evaluation metrics listed above in order to determine the accuracy of the model. If this is eventually successful, future work should explore using these reconstructed images to classify stress. Previous demonstrated that stress classification is possible using CNNs with a high degree of accuracy and automatically filling the data and passing it to the convolutions network could be promising [3]. Lastly, if the model is trained successful using BVP, other biological and non-biological signals should be explored for reconstruction, such as stock market data.

References

[1] CAI, R. Y., RICHDALE, A. L., DISSANAYAKE, C., AND ULJAREVIĆ, M. Resting heart rate variability, emotion regulation, psychological wellbeing and autism symptomatology in adults with and without autism. *International Journal of Psychophysiology 137* (2019), 54–62.

[2] DZIEŻYC, M., GJORESKI, M., KAZIENKO, P., SAGANOWSKI, S., AND GAMS, M. Can we ditch feature engineering? end-to-end deep learning for affect recognition from physiological sensor data. *Sensors 20*, 22 (2020), 6535.

[3] ELZEINY, S., AND QARAQE, M. Stress classification using photoplethysmogram-based spatial and frequency domain images. *Sensors 20*, 18 (2020), 5312.

[4] FANG, C., AND WANG, C. Time series data imputation: A survey on deep learning approaches. *arXiv preprint arXiv:2011.11347* (2020).

[5] GOODFELLOW, I. J., POUGET-ABADIE, J., MIRZA, M., XU, B., WARDE-FARLEY, D., OZAIR, S., COURVILLE, A. C., AND BENGIO, Y. Generative adversarial networks. *CoRR abs/1406.2661* (2014).

[6] HEUSEL, M., RAMSAUER, H., UNTERTHINER, T., NESSLER, B., KLAMBAUER, G., AND HOCHREITER, S. Gans trained by a two time-scale update rule converge to a nash equilibrium.

[7] HUANG, T., CHAKRABORTY, P., AND SHARMA, A. Deep convolutional generative adversarial networks for traffic data imputation encoding time series as images. *CoRR abs/2005.04188* (2020).

[8] KATSIGIANNIS, S., AND RAMZAN, N. DREAMER: A database for emotion recognition through EEG and ECG signals from wireless low-cost off-the-shelf devices. *IEEE J. Biomed. Health Informatics 22*, 1 (2018), 98–107.

[9] KOLDIJK, S., SAPPELLI, M., VERBERNE, S., NEERINCX, M. A., AND KRAAIJ, W. The SWELL knowledge work dataset for stress and user modeling research. In *Proceedings of the 16th International Conference on Multimodal Interaction, ICMI 2014, Istanbul, Turkey, November 12-16, 2014* (2014), A. A. Salah, J. F. Cohn, B. W. Schuller, O. Aran, L. Morency, and P. R. Cohen, Eds., ACM, pp. 291–298.

[10] KOOCHALI, A., SCHICHTEL, P., DENGEL, A., AND AHMED, S. Probabilistic forecasting of sensory data with generative adversarial networks–forgan. *IEEE Access 7* (2019), 63868–63880.

[11] KULCHYK, J., AND ETEMAD, A. Activity recognition with wearable accelerometers using deep convolutional neural network and the effect of sensor placement. In *2019 IEEE SENSORS* (2019), IEEE, pp. 1–4.

[12] RADFORD, A., METZ, L., AND CHINTALA, S. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434* (2015).

[13] SAGANOWSKI, S., KAZIENKO, P., DZIEŻYC, M., JAKIMÓW, P., KOMOSZYŃSKA, J., MICHALSKA, W., DUTKOWIAK, A., POLAK, A., DZIADEK, A., AND UJMA, M. Consumer wearables and affective computing for wellbeing support. *arXiv e-prints* (2020), arXiv–2005.

[14] SANTAMARIA-GRANADOS, L., ORGANERO, M. M., GONZÁLEZ, G. A. R., ABDULHAY, E. W., AND N., A. Using deep convolutional neural network for emotion detection on a physiological signals dataset (AMIGOS). *IEEE Access 7* (2019), 57–67.

[15] SARKAR, P., AND ETEMAD, A. Self-supervised learning for ecg-based emotion recognition. In *2020 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2020, Barcelona, Spain, May 4-8, 2020* (2020), IEEE, pp. 3217–3221.

[16] SCHMIDT, P., REISS, A., DÜRICHEN, R., MARBERGER, C., AND LAERHOVEN, K. V. Introducing wesad, a multimodal dataset for wearable stress and affect detection. In *Proceedings of the 2018 on International Conference on Multimodal Interaction, ICMI 2018, Boulder, CO, USA, October 16-20, 2018* (2018), S. K. D'Mello, P. G. Georgiou, S. Scherer, E. M. Provost, M. Soleymani, and M. Worsley, Eds., ACM, pp. 400–408.

[17] SEITZER, M. pytorch-fid: FID Score for PyTorch. `https://github.com/mseitzer/pytorch-fid`, August 2020. Version 0.1.1.

[18] SMITH, K. E., AND SMITH, A. O. Conditional gan for timeseries generation. *arXiv preprint arXiv:2006.16477* (2020).

[19] SUBRAMANIAN, R., WACHE, J., ABADI, M. K., VIERIU, R. L., WINKLER, S., AND SEBE, N. ASCERTAIN: emotion and personality recognition using commercial sensors. *IEEE Trans. Affect. Comput. 9*, 2 (2018), 147–160.

[20] THIEME, A., BELGRAVE, D., AND DOHERTY, G. Machine learning in mental health: A systematic review of the HCI literature to support the development of effective and implementable ML systems. *ACM Trans. Comput. Hum. Interact. 27*, 5 (2020), 34:1–34:53.