Generative models for wearables data

Arinbjörn Kolbeinsson *
University of Virginia
Charlottesville, Virginia
arinbjorn@virginia.edu

Luca Foschini
Sage Bionetworks
Seattle, Washington
luca.foschini@sagebase.org

Abstract

Data scarcity is a common obstacle in medical research due to the high costs associated with data collection and the complexity of gaining access to and utilizing data. Synthesizing health data may provide an efficient and cost-effective solution to this shortage, enabling researchers to explore distributions and populations that are not represented in existing observations or difficult to access due to privacy considerations. To that end, we have developed a multi-task self-attention model that produces realistic wearable activity data. We examine the characteristics of the generated data and quantify its similarity to genuine samples.

1 Introduction

High quality health data is a vital yet scarce resource in modern healthcare. Raw data collection is expensive and time consuming, labelling requires expert knowledge and storage poses privacy concerns. As a result, most health datasets fail to capture the true distribution of the underlying population, particularly in the tails which contain rare conditions and underrepresented attributes [Ganapathi et al., 2022]. Extending these data by generating unseen yet realistic instances can augment the downstream task to allow for novel analyses and hypothesis generation. For downstream tasks to be representative, it is crucial that the generated samples remain realistic and reflective of the data intended for study. However, maintaining realism is a difficult task and must be finely balanced with the requirement to generate new samples instead of simply recreating those seen in the training set. In other fields where data generation is used, the same principle applies. In state-of-the-art image generation [Ramesh et al., 2022, Rombach et al., 2022] this trade-off has been finely balanced.

Methods for time-series generation exist in the literature. [Kang et al., 2020] presented an approach using mixture autoregressive (MAR) models which can be configured to give the time series certain characteristics. One drawback of this approach is that the specific characteristics, such as seasonal strength and stability, need to be quantified and cannot be inferred from the context, such as a medical condition. For healthcare data, Norgaard et al. [2018] presented a Generative Adversarial Network (GAN) for accelerometer and exercise data. [Dash et al., 2020] also used GANs for generation of hospital time-series based on the MIMIC-III dataset. More recently, outside healthcare applications, Srinivasan and Knottenbelt [2022] and Li et al. [2022] have proposed a general architecture based on transformers but train it using the GAN framework.

In this work, we focus on personal health data, specifically multi-modal resting heart rate, sleep and step data, generated by consumer wearable devices (wearables). Applications on the health domain of such data are still emerging, detection of flu and COVID-19 being one example [Shapiro et al., 2021, Merrill and Althoff, 2023]. Our approach features a multi-task self-attention model for wearable activity data synthesis. In summary, our contributions are (1) A synthetic data generator based on self-attention for wearables data, (2) Demonstration that the model can predict future activity through self-supervised learning of over 2 million activity days, and (3) Evaluation of the generative model with quantitative comparisons to genuine real-world data.

*Work done while at Evidation Health Inc, San Mateo, California.

1st Workshop on Deep Generative Models for Health at NeurIPS 2023.

2 Data for training

Dataset. All models were trained and evaluated on the same set of activity data acquired using wearable FitBit trackers. , collected as part of the DiSCover (Digital Signals in Chronic Pain) Project, a 1-year longitudinal study (ClinicalTrials.gov identifier: NCT03421223) [Lee et al., 2021]. The dataset contained day-level data from $10\,000$ individuals who gave permission for use of their data for the purpose of health research. Data were collected over one year, resulting in a total of $2\,737\,500$ person-days of activity data. The data contain three signals: resting heart rate (beats per minute), total sleep (minutes), total steps (step count). The mean age of the participants was 37.3 (SD=10.5, range: 18 to 85) with 72.15% of participants female and primarily Non-Hispanic White (80.5%).

Pre-processing. Day level aggregates were calculated from the minute-level raw data by summing all minutes spent sleeping per day, summing all steps per day and taking the mean resting heart rate per day. Only days with $> 80\%$ coverage were included in the analaysis. Missing data were imputed with the mean feature values per individual. Each feature was then scaled to $[0, 1]$. We then divide the year-long sequences into shorter sequences with a length of 21 days for use as inputs. Although this is much shorter than sequences used with most transformers, we keep this short for the following reason: every source sequence is of length 365, corresponding to each day in the year for an individual. If we use a larger window of, e.g., 100 we could only create three non-overlapping sequences per individual. The shorter sequence length gives us a more diverse set of samples while still capturing a representative time period on the scale of human activity (three weeks).

Although the labels are continuous values, we convert them to a one-hot encoding of 100 evenly-spaced bins. We do this to model the outputs as a softmax distribution. As described by van den Oord et al. [2016], this removes any assumptions about the shape of the distribution and is therefore highly compatible with neural networks and has also been used for audio-generation in Wavenet [Oord et al., 2016].

3 Model and learning

Embeddings. The three input channels (resting-heart-rate, sleep minutes and step count) are embedded in a 64 dimensional space through a learned embedding weights. As the sequences are temporally ordered, it is important to preserve their positional relationships. To do that, they are positionally encoded with learned positional weights that are added to the embedded inputs.

Transformer. The embeddings are passed into a transformer [Vaswani et al., 2017] that consists only of decoder layers. Self-attention is calculated as $attention(Q, K, V) = softmax(QK^T/\sqrt{d_k})V$. Where Q, K and V are the query, key and value matrices, respectively and d_k is the dimensionality of the keys. Decoder-only transformers have been shown to perform well in autoregressive tasks, like next-word predictions [Brown et al., 2020, Rae et al., 2021] and joint learning of multiple tasks [Reed et al., 2022]. Each transformer block begins with layer-normalization to stabilize gradient updates and training. As this is an auto-regressive task, we ensure future information is not used by causal masking, i.e. confining each position to previous positions or the current position. This is implemented by masking the upper-right triangle of the attention weight-matrix.

Finally, each block is completed by a feed-forward network of two dense layers of dimensionality 256 with GeLU activation and dropout probability of 0.1 during training. We stack three of these blocks to form the core of the model, and four attention heads. It is followed with a feed-forward network to an output of three 100-unit vectors, corresponding to the three tasks and 100 bins. A softmax activation is applied to each one to obtain the logits used for loss calculation. This results in a causally-masked multihead multi-task self-attention model that can be trained to model and forecast activity time series.

Loss. As described in detail earlier, we use a softmax distribution of outputs. Then we can minimize the cross-entropy loss between the predicted and true values. We learn the three outputs (resting heart rate, daily steps and sleep minutes) jointly with separate feed-forward network heads. The individual losses are added through shake-shake regularization Gastaldi [2017], a stochastic affine combination.

The combined loss which we minimize is then defined as

$$\mathcal{L}_{combined} = \sum_{i=1}^{N} \alpha_i \mathcal{L}_i$$

where $\boldsymbol{\alpha}$ is a random vector of unit length and $\mathcal{L}_i$ are individual losses. In our case, $N = 3$.

Training. We minimize the loss using Adam [Kingma and Ba, 2015] and an initial learning rate of 10^{-3}, reducing it by a factor of 10 every 5 epochs, with a total of 15 training epochs. The model and training were implemented in PyTorch [Paszke et al., 2019], along with NumPy [Harris et al., 2020] and SciPy [Virtanen et al., 2020], and visualizations in Matplotlib [Hunter, 2007].

We train four different models to compare the effect of increased number of training points on the quality of generated samples. The largest model contains 2 029 230 days, which represent 100% of the available training data. We then train three smaller models with 10%, 1% and 0.5% of the available training data, respectively.

Generating new samples. With the autoregressive model already trained to predict next-day values, synthesizing new sequences is straightforward. We start with a prompt sequence fragment, taken from a held-out set, and input into the trained model. Then, we recursively remove the first day of the sequence and append the next-day predictions to the end. Scaling the temperature of the logits gave more consistent results for resting steps and sleep, we used temperatures of 2, while resting heart rate was kept with a temperature of 1. The three softmax distributions of the output were sampled independently to obtain the next-day value.

4 Results and evaluation

We evaluate the model on two criteria. 1) The prediction accuracy of the model 2) Quantitative evaluation of distance measures and similarity scores between real and generated sequences.

4.1 Activity modelling

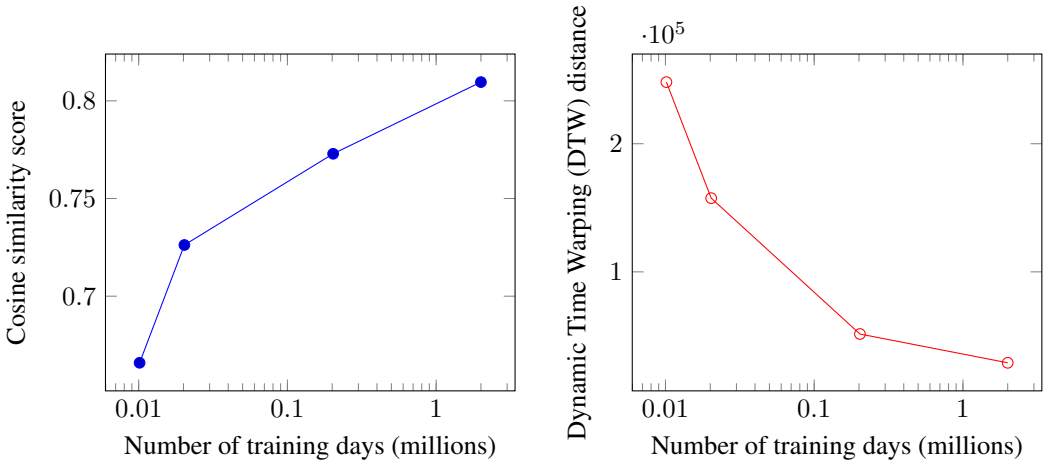
We begin by comparing the accuracy of the next-day predictions with the ground truth real-world data. These results are highlighted in Table 1, where we investigate the effect of training dataset size on test set performance. Increasing the number of training samples has a strong effect, particularly on resting heart rate prediction where the mean absolute error (MAE) is reduced to 1.21 BPM in the case of 2 million training samples. Given only 0.5% of the data, the accuracy is far lower and increasing the number of data always results in a marked increase in accuracy. The effect of increased data has a different effect for both steps and sleep minutes. It appears that going from $\sim$ 20k days to $\sim$ 200k days has a far greater effect than the next order of magnitude, which appears to have no marked difference.

Table 1: Comparison of mean absolute errors (MAE) of next-day resting heart rate (HR), sleep and steps with respect to the size of the training set. MAE reported on the test set. There is a marked difference in terms of accuracy as the number of training samples increases.

Training size (Days)	MAE Resting HR (BPM)	MAE Sleep (Minutes)	MAE Steps (Count)
10 146 (0.5%)	31.9	135.9	4922
20 292 (1%)	18.6	137.2	4444
202 923 (10%)	3.31	58.6	2627
2 029 230 (100%)	1.21	56.2	2830

4.2 Distance and similarity measures

No standard collection of methods exists for scoring differences between time series. However, we make use of two common distance measures: cosine similarity and dynamic time warping (DTW).



(a) Mean pairwise cosine similarity measure of models trained with different training set sizes, compared with real data. Models trained with more data have more similarity with genuine data. The model trained with over 2 million days achieves a score of over 0.810 with the intra-similarity of real data being 0.873.

(b) Mean pairwise dynamic time warping distance of models trained with different training set sizes, compared with real data. The mean distance from the model trained with over 2 million days to the real data is $29\,028$ with the intra-distance of real data being $27\,897$.

Figure 1: The effect of increasing training data on two common distance measures: cosine similarity and dynamic time warping (DTW).

For cosine similarity, we follow the approach of Norgaard et al. [2018] and compare the mean pairwise cosine similarity statistics between real sequences and generated ones. Where the cosine similarity statistic between two sequences X and Y is defined as their normalized dot product $K(X, Y) = \frac{(X \cdot Y)}{\|X\|\|Y\|}$. The mean pairwise cosine similarity score between real sequences in the dataset is 0.873, providing an "target realism" value for this score on the dataset. This captures the intra-dataset variation of the real data distribution.

In further analysis, we calculate the mean pairwise DTW distance [Bundy and Wallen, 1984] using the DtAIdistance library [Wannesm et al., 2022]. The mean pairwise DTW distance in the real dataset is $27\,897$ which provides the "target realism" measure for comparing the distances to the generated data. Figure 1a illustrates the results of this comparison. Increasing the amount of training data has a significant impact on the similarity between generated and real sequences. When only 0.05% of the total available data is used for training ($10\,146$ days), the mean pairwise cosine similarity is 0.666. When 1% of the data is used, the score increases to 0.726, and when 10% is used, it reaches 0.773. The full dataset of over 2 million days yielded the best trained model with a score of 0.810, which is close to the intra-similarity of real data, which is 0.873.

In Figure 1b we see that increasing the size of the training data results in a model that produces sequences much closer to the real data. The increase appears nearly asymptotic to the intra-distance of real data, which is $27\,897$ compared to $29\,028$ for data generated from the model trained on the full dataset. The agreement between the cosine similarity and the DTW distance measures provides further evidence that the model is able to capture the inherent properties of the data and generate similar sequences.

5 Conclusion

This work furthers the exploration of methods for generating synthetic personal health data. It provides researchers with the ability to craft datasets according to their needs while reducing privacy concerns, making study design more efficient and enabling the development of analysis at a faster rate. Moreover, it helps to identify issues before they affect real-world deployment. Our work adds to the existing literature on synthetic data across multiple fields and underscores the potential of generating realistic person-generated health data to enhance and improve health research.

References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL `https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html`.

Alan Bundy and Lincoln Wallen. Dynamic time warping: Alias: dynamic programming in speech recognition. *Catalogue of Artificial Intelligence Tools*, pages 32–33, 1984.

Saloni Dash, Andrew Yale, Isabelle Guyon, and Kristin P Bennett. Medical time-series data generation using generative adversarial networks. In *Artificial Intelligence in Medicine: 18th International Conference on Artificial Intelligence in Medicine, AIME 2020, Minneapolis, MN, USA, August 25–28, 2020, Proceedings 18*, pages 382–391. Springer, 2020.

Shaswath Ganapathi, Jo Palmer, Joseph Alderman, Melanie Calvert, Cyrus Espinoza, Jacqui Gath, Marzyeh Ghassemi, Katherine Heller, Francis Mckay, Alan Karthikesalingam, et al. Tackling bias in ai datasets through the standing together initiative. *Nature Medicine*, 2022.

Xavier Gastaldi. Shake-shake regularization. *ArXiv preprint*, abs/1705.07485, 2017. URL `https://arxiv.org/abs/1705.07485`.

Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, 2020. doi: 10.1038/s41586-020-2649-2. URL `https://doi.org/10.1038/s41586-020-2649-2`.

J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3): 90–95, 2007. doi: 10.1109/MCSE.2007.55.

Yanfei Kang, Rob J Hyndman, and Feng Li. Gratis: Generating time series with diverse and controllable characteristics. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 13(4):354–376, 2020.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL `http://arxiv.org/abs/1412.6980`.

Jennifer L Lee, Christian J Cerrada, Mai Ka Ying Vang, Kelly Scherer, Caroline Tai, Jennifer LA Tran, Jessie L Juusola, and Christine N Sang. The discover project: protocol and baseline characteristics of a decentralized digital study assessing chronic pain outcomes and behavioral data. *medRxiv*, pages 2021–07, 2021.

Xiaomin Li, Vangelis Metsis, Huangyingrui Wang, and Anne Hee Hiong Ngu. Tts-gan: A transformer-based time-series generative adversarial network. In *Artificial Intelligence in Medicine: 20th International Conference on Artificial Intelligence in Medicine, AIME 2022, Halifax, NS, Canada, June 14–17, 2022, Proceedings*, pages 133–143. Springer, 2022.

Mika A Merrill and Tim Althoff. Self-supervised pretraining and transfer learning enable\titlebreak flu and covid-19 predictions in small mobile sensing datasets. In *Conference on Health, Inference, and Learning*, pages 191–206. PMLR, 2023.

Skyler Norgaard, Ramyar Saeedi, Keyvan Sasani, and Assefaw H Gebremedhin. Synthetic sensor data generation for health applications: A supervised deep learning approach. In *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 1164–1167. IEEE, 2018.

Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *ArXiv preprint*, abs/1609.03499, 2016. URL https://arxiv.org/abs/1609.03499.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 8024–8035, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/bdbca288fee7f92f2bfa9f7012727740-Abstract.html.

Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. Scaling language models: Methods, analysis & insights from training gopher. *ArXiv preprint*, abs/2112.11446, 2021. URL https://arxiv.org/abs/2112.11446.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *ArXiv preprint*, abs/2204.06125, 2022. URL https://arxiv.org/abs/2204.06125.

Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, et al. A generalist agent. *ArXiv preprint*, abs/2205.06175, 2022. URL https://arxiv.org/abs/2205.06175.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022.

Allison Shapiro, Nicole Marinsek, Ieuan Clay, Benjamin Bradshaw, Ernesto Ramirez, Jae Min, Andrew Trister, Yuedong Wang, Tim Althoff, and Luca Foschini. Characterizing covid-19 and influenza illnesses in the real world via person-generated health data. *Patterns*, 2(1):100188, 2021.

Padmanaba Srinivasan and William J Knottenbelt. Time-series transformer generative adversarial networks. *ArXiv preprint*, abs/2205.11164, 2022. URL https://arxiv.org/abs/2205.11164.

Aäron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In Maria-Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 1747–1756. JMLR.org, 2016. URL http://proceedings.mlr.press/v48/oord16.html.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris,

Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2.

Khendrickx Wannesm, Aras Yurtman, Pieter Robberechts, Dany Vohl, Eric Ma, Gust Verbruggen, Marco Rossi, Mazhar Shaikh, Muhammad Yasirroni, ZW Todd, et al. Wannesm/dtaidistance: v2. 3.5. *Zenodo: Genève, Switzerland*, 2022.