Filling Missing Values on Wearable-Sensory Time Series Data

Suwen Lin*† Xian Wu*† Gonzalo Martinez* Nitesh V. Chawla *‡

Abstract

Missing data points is a common problem associated with data collected from wearables. This problem is particularly compounded if different subjects have different aspects of missingness associated with them – that is varying degrees of compliance behavior of individuals (participants) with respect to wearables as well as personal changes in lifestyle and health impacting heart rate. Moreover, despite the varying degree of compliance behavior, the wearable in itself might have glitches that lead to observations being dropped. Thus, any missing value imputation in such data has to not only generalize to the wearable behavior but also to the participant behavior. In this paper, we present a deep learning based approach for imputing missing values in heart rate time series data collected from a participant's wearable. In particular, for each participant, we first leverage his/her historical heart rate records as a reference set to extract the underlying personalized characteristics, and then impute the missing heart rate values by considering both contextual information of the current observations and the user's features learned from previous records. Adversarial training is applied to guide the learning process, which imputed more reasonable heart rate series with the consideration of human health conditions, e.g., heart rate fluctuations. Extensive experiments are conducted on two real-world data to show the superiority of our proposed method over state-of-the-art baselines.

1 Introduction

The availability of such daily activities data, such as heart rate data at a fine granularity, from wearables brings in an unprecedented opportunity to understand and model the patterns of human behavior. There is an especially compelling opportunity to model physiological response, as measured by heart rate, for a wide spectrum of applications, such as human activity identification [11], emotion detection [4], fitness recommendation [23] and user demographics inference [29]. However, there are several challenges that stem from the missingness of data, as the entire stream of heart rate data may not be available for a variety of reasons. This can be due to sensor errors (*e.g.*, wearable malfunction) and inconsistent data collection periods (*e.g.*, wearing behavior and compliance varies by person) [3]. Such incomplete heart rate data not only affects real-time monitoring of a physiologica condition, but also degrades the performance of inference and forecasting tasks that may rely complete heart rate observations.

Time series analysis has drawn progressive attention to impute missing data. In general, this can be achieved by characterizing various temporal patterns of time-ordered sequences [15]. For example, in time series community, averaging [19] or regression [1] strategies are commonly adopted to conduct linear imputations, but might misss the non-linearities in the data. With the advent of deep learning techniques in modeling sequential data, deep neural network models (*e.g.*, LSTM and GRU) have been utilized as an alternative option to fill missing values in time series, which preserve the non-linear temporal correlation structures [6, 18]. Furthermore, a handful of recent extensions leverage hybrid architecture of generative adversarial networks and recurrent neural networks [5] to improve imputation.

Despite their prevalence, we argue that these methods are far from sufficient to capture the complicated sequential transitions and cannot generalize well for heart rate time series imputation. The key reason is that the incomplete heart rate time series data contains many large gaps. Simply filling missing values with semantic information limits the power of most existing methods, especially those relying on recurrent networks. In this work, we ask the question of filling in missing values in heart rate time series, while addressing the challenges of varying reasons of missingness stemming from wearable behavior to human behavior. We posit that a good imputation method is expected to address the following challenges.

- **Capability of coping with time series data with arbitrary missing gaps**. Individuals may wear and take off the heart rate sensors (wearables) at any time. It follows that day-long heart rate records may contain multiple gaps with various lengths. These gaps, if not dealt with properly, may convey misleading information during encoding. For example, when applying convolutional operations to conduct encoding,

*University of Notre Dame, IN, USA. Email: {slin4, xwu9, gmarti11, nchawla}@nd.edu.

†These authors contributed equally to this work.

‡Corresponding author.

if the missing samples are treated equally as other valid inputs, the generated features may be improperly encoded and these improper features can further propagate incorrect information to remaining parts of the network.

- **Imputation with preserving personal characteristics**. Heart rate analysis is very sensitive to small fluctuations in time series data. Cardiovascular disease diagnosis, for example, relies on the monitoring of peaks in heart rate records. Simply semantically inferring missing values with the contextual information does not recover the personal characteristics of the individual and may jeopardize downstream analysis. Thus, an imputation method with the awareness of personal features is required.

To address the above challenges, we propose a heart rate imputation framework, called HeartImp, to fill in missing values in day-long heart rate time series. HeartImp first encodes each day-long time series segments by a convolutional encoder. In this component, HeartImp uses a gating mechanism by re-weighting each feature element after convolution to overcome the challenge of arbitrary locations and lengths of missing gaps. Then, to resolve the second challenge, HeartImp uses imputation with a reference set module, which takes the incomplete day-long time series and a reference set(we will provide the definition later) as the input and generates the imputed values through comprehensive information fusion of inputs. In the training objective, *HeartImp* contains reconstruction loss for imputation correctness and adversarial loss to encourage the coherency between generated and existing heart rate data points.

In summary, the main contributions of this paper, as part of the *HeartImp*framework, are as follows:

- We introduce a gating mechanism to learn a dynamic feature selection strategy after convolution, improving the encoding performance with arbitrary length and location of missing period.
- To fill high-accurate values, we propose using multiple days collected from same person as reference sets to impute. It also captures the personal characteristics and enables discovering the long-distance correlations without increasing convolutional layers.
- We use two real-world datasets of different cohorts to demonstrate that *HeartImp*E significantly outperforms state-of-the-art models, under different scenarios.

2 Problem Formulation

In this section, we introduce preliminary definitions and formalize the problem of heart rate imputation.

Wearable-sensory time series data is generally collected from a set of participants over different time periods. Here, we use $U = \{u_1, ..., u_i, ..., u_I\}$ to denote the participant population in the data collection, where I is the number of participants.

DEFINITION 1. ***Day-long Wearable-Sensory Time Series***. *We define day-long wearable-sensory time series as the measurements collected on one day that started from 12:00 AM to 11:59 PM. Since the wearable sensory data of each participant is collected at different start and end date, the collected data may vary with different duration. We define the collected dates as a vector* $\boldsymbol{t}_i = \{t_i^1, ..., t_i^j, ..., t_i^{J_i}\}$ *in a chronological order, where* t_i^j *denotes the date collected from participant* u_i *on* j*-th day and* J_i *is the total number of collected days. The time series collected on* t_i^j *is presented as* $\boldsymbol{x}_i^j = \{x_i^{(j,1)}, ..., x_i^{(j,\tau)}, ..., x_i^{(j,T)}\}$ *(*$\boldsymbol{x}_i^j \in \mathbb{R}^T$*). Each measurement* $x_i^{(j,k)}$ *is associated with a timestamp information.*

DEFINITION 2. ***Target time series***: *We define target time series as the day-long wearable-sensory time series that we aim to fill the missing values.*

DEFINITION 3. ***Reference time series set***: *We define a reference set as a set of day-long wearable-sensory time series before target time series, which is collected from the same participant/source as the target time series. Formally, it can be presented as a matrix* $\boldsymbol{R}_i^j = \{\boldsymbol{x}_i^1, ..., \boldsymbol{x}_i^{j-1}\}$ *given target time series* $\boldsymbol{x}_i^j$*.*

Problem Statement. Given the target time series and reference time series set of a user, the objective of this work is to learn an imputation function that could automatically impute missing values in the target time series. It can be formally defined as follows:

$$M \cdot \boldsymbol{x}_i^j = f((1 - M) \cdot \boldsymbol{x}_i^j, \boldsymbol{R}_i^j) \tag{2.1}$$

where M is a binary mask corresponding to missing areas with value 1 and observed areas with value 0. f is the function we aim to learn. Note that we do not have any restrictions on the completeness of time series in a reference set.

3 Methodology: HeartImp

In this section, we present the *HeartImp* model that imputes missing parts of time series data based on the contextual information. We first introduce the general framework of *HeartImp* to elaborate the overall flow and then present details in the following subsections. Figure 1 depicts the key components of *HeartImp*.

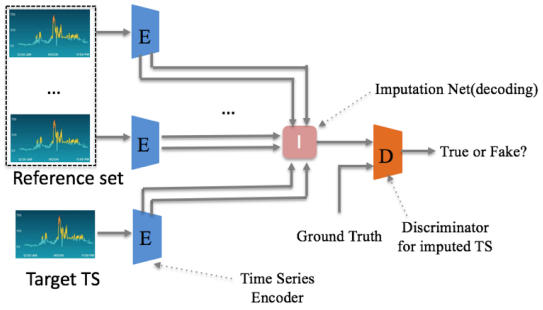


Figure 1: The framework of *HeartImp*.

3.1 General Framework Considering that heart rate data is in variable lengths and human behaviors strongly exhibit daily periodic patterns, we first partition the heart rate data of each individual into a set of day-long time series. For missing observations, we set their corresponding values to zeros by default and these zeros serve as placeholders in time series and differentiate the missing data points from the observed ones with different ranges.

After that, for each day-long time series with missing regions, we leverage an encoder-decoder architecture to impute values in the missing region. Instead of filling values merely based on the observed measurements on the same day, we also introduce a reference set selected from the rest day-long time series of the same collected source/person as a second source to assist the imputation task. Daily heart rates of the same person may exhibit repeat but slightly different patterns across different days. For example, when imputing the time period during sleeping, the effective reference information could be the sleeping duration on other day-long time series. Here, *HeartImp* learns how to incorporate this reference data as a semantic guide with contextual information around the missing period. Consequently, the decoder takes the fused feature representation as an input and generates the imputed target time series.

In the training phase, *HeartImp* utilizes a reconstruction loss to minimize the difference between filled values and ground truth. In addition, we leverage an adversarial loss to make the filled heart rate time series look as real as possible and further encourage the coherence with regards to its contexts.

3.2 Gated Convolutional Module In the encoder-decoder framework of *HeartImp*, we propose a gated convolutional module to handle the incomplete wearable-sensory time series data with missing data at arbitrary positions. This module is a multi-layer convolutional network which aims to extract dependency patterns in the time dimension of daily series in a hierarchical way.

3.2.1 Temporal Dependency Modeling In our constructed convolutional network, we stack the 1-dimensional convolutional layers with increasing the receptive fields for long-range dependence modeling. In particular, for l-th convolution layer in our model, we employ c_i filters $w \in \mathbb{R}^{K \times c_0}$ to perform the 1-dim convolution operation on the $K \times c_i$ matrices, where K and c_o are hyperparameters that indicate kernel size and output channel size, respectively. Formally, the convolution operator is given as follows:

$$\boldsymbol{\mathcal{V}}^l_{i,j} = f\left(w^l_j \cdot \boldsymbol{\mathcal{V}}^{l-1}_{[i-\frac{k^{l-1}}{2}:i+\frac{k^{l-1}}{2},1:c_i]}\right) \tag{3.2}$$

where $\boldsymbol{\mathcal{V}}^l_{i,j}$ denotes the features in the l-th layer, $\cdot$ represents the dot product operation, i and j are the index of time and channel dimension, respectively. $f(\cdot)$ is the activation function. For simplicity, we do not show the bias term in the above equation.

3.2.2 Channel-Wise Recalibration Mechanism However, the generated intermediate features are negatively affected by the zero paddings in missing data points since convolution operations treat padding and observing values equally. To address this issue, we propose a channel-wise gating mechanism to dynamically re-weight the features and to reduce the impact of features generated via padding values or the mixed padding and observing values. Inspired by squeeze and excitement network[16] to recalibrate the information distribution of channel-wise feature responses, we introduce the feature importance vectors among all elements across channels as:

$$z^{cr}_{i,j} = \text{Sigmoid}(w^{cr}_2 \cdot \text{ReLU}(w^{cr}_1 \cdot \boldsymbol{\mathcal{V}}_{i,j})) \tag{3.3}$$

where $z^{cr}_{i,j} \in \mathcal{R}^c_i$ is the feature importance vectors with each element for each channel. w^{cr}_1 and w^{cr}_2 are the learned parameters of two fully-connected neural layers. As such, the final channel-wise feature is given by:

$$\widetilde{\boldsymbol{\mathcal{V}}}_{i,j} = \boldsymbol{\mathcal{V}}_{i,j} \odot z^{cr}_{i,j} \tag{3.4}$$

where $\odot$ is the element-wise product operation.

3.2.3 Encoder-Decoder Configuration In summary, our gated convolutional module consists of five convolutional layers followed by channel-wise recalibration mechanism layers. In the encoder, the five layers are set as the one-dimensional kernel with {5, 3, 3, 3, 3} and filter sizes with {16, 32, 48, 64, 128}, respectively. Accordingly, the structure of the decoder is symmetric to that of the encoder.

3.3 Fusion between Reference Set and Target Time Series To exploit the contextual information around the missing intervals, we further incorporate the knowledge from reference sets in our model. Given the reference set $\boldsymbol{R}_i^j$ for target time series $\boldsymbol{x}_i^j$, we assume that it may involve extra heart rate features to assist the missing part imputation in the target time series. As the reference data could be incomplete, we also apply a gated convolutional encoder to these reference series to alleviate the effect of missing values during feature encoding. To distill effective information from both the target time series and its corresponding reference set, we propose an aggregation mechanism to fuse them. More specifically, we first concatenate the feature responses learned from both reference set and target series, and then we apply dilated convolutions on the concatenated feature responses along both the temporal dimension and the concatenation dimension. In this way, the dilated convolutions yield a conclusive vector, which aggregates the information from both the target and reference sides. The conclusive vector is further fed into an imputation net (decoder). The decoder aims to decode the aggregated information encoded in the conclusive vector and recover the time series information in the target series. In a nutshell, the encoder allows us to utilize knowledge from related time series in the reference set on top of the information distilled from the target series and form aggregated time series representations. The decoder aims to recover the target series based on the aggregated representations. In this way, the target series is reconstructed based on not only its original signals but also reference time series which are highly related.

3.4 Loss Function We train our model by learning the joint loss function of two individual parts. One is the reconstruction loss $\mathcal{L}_{rec}$ from the encode-decoder module. It is to model the temporal structure of the missing area in target time series and to capture the contextual knowledge and internal coherence with the reference series. The other is the adversarial loss $\mathcal{L}_{adv}$, which guides the imputation process to generate as real series as possible. We use the l1-norm distance between the ground truth and the imputed data as the reconstruction loss function as:

$$\mathcal{L}_{rec} = ||M \odot (\mathcal{I}(E(\boldsymbol{x}_i^j \odot M, \boldsymbol{R}_i^j)) - \boldsymbol{x}_i^j)||_1 \tag{3.5}$$

where the binary mask M takes the value 1 for missing points and 0 for the observed points. The $\mathcal{I}(\cdot)$ and $E(\cdot)$ represent the imputation net(decoder) and encoder functions, respectively.

The adversarial loss $\mathcal{L}_{adv}$, on the other hand, is based on Wasserstein Generative Adversarial Network(WGAN), which has shown better performance than existing GAN for various applications [12, 31]. WGAN jointly learns a discriminator and a generator. The learning process is the discriminator trying to distinguish the real sample from the generated one from the generator while the generator learns to predict the more confusing sample for the discriminator. We adopt this method by modeling the generator as the encoder and the imputation net. Then the objective function is given by:

$$\begin{aligned} \mathcal{L}_{adv} = \min_{\mathcal{I},E} \max_{D} \mathbb{E}_{\boldsymbol{x}\in\chi}(\log(D(\boldsymbol{x})) \\ + \log(1 - D(\mathcal{I}(E(\boldsymbol{x} \odot M, \boldsymbol{R}))))) \\ + \gamma\mathbb{E}(||\nabla_{\boldsymbol{x}'} D(\boldsymbol{x}')||_2 - 1)^2 \end{aligned} \tag{3.6}$$

where $D(\cdot)$ is the discriminator function, $\mathbb{E}(\cdot)$ if the expectation function, γ is set as 1 and others are consistent with the above statements. The last term in the function is the gradient penalty term, where $\boldsymbol{x}'$ is sampled from the straight line between points from distribution of generated data and that of real data. Therefore, the final objective function $\mathcal{L}$ is the combination of these two loss function: $\min_{\mathcal{I},E} \max_{D}(\mathcal{L}_{rec} + \lambda\mathcal{L}_{adv})$, where λ is a coefficient to control the importance of regularization term.

4 Experiments

In this section, we demonstrate the effectiveness of the proposed *HeartImp* on two real-world datasets collected from two different wearable sensors for two different population groups. We perform extensive experiments on the gap imputation task with the aim to answer the following research questions:

- **RQ1**: Does *HeartImp* consistently outperform the state-of-the-art temporal data imputation models with respect to different time periods?
- **RQ2**: How does *HeartImp* perform compared to other algorithms across various gap sizes?
- **RQ3**: How do the different key hyper-parameters impact the model performance?
- **RQ4**: How do the different components(*i.e.* the Generative Adversarial Training, the Gated Convolution and the reference infusion) of *HeartImp* contribute to the model performance ?

4.1 Experiment Settings

4.1.1 Data In our evaluation, we conduct experiments on the heart rate time series data from two real-world datasets. These two datasets are collected from two different subject pools and based on two types of wearable sensors. Accordingly, we further notate them

by their sensor types: Garmin data and Fitbit data, respectively.

Garmin data [21]: This dataset spanning from January 2018 to September 2018 is collected from 574 participants (male: 334, female: 240) in a research study of workplace performance, where they record physical activities(*e. g.* heart rate and steps) of employees via Garmin bands in several companies across the United States.

Fitbit data [10]: This dataset contains wearable sensory data via Fitbit Charge bands from 623 students(male:306, female: 317) in a University. The sensor bands were gradually deployed to all the participants from the Fall semester of 2015. In our experiments, we consider the heart rate data from January 2016 to October 2017.

4.1.2 Comparison Baselines To justify the efficacy of *HeartImp*, we compare it with the following baselines from multiple research lines: i) conventional time series imputation methods (*i.e.* linear interpolation, exponentially weighted moving averaging, K-nearest neighbor, Kalman smoothing and Last Observation Carried Forward); ii) deep learning techniques for time series imputation(*i.e.* Denoising Autoencoder and bi-direction RNN); iii)neural network models for image and video inpainting(*i.e.* context encoders and spatial-temporal video completion network).

- Linear Interpolation(LIP) [17]: LIP fits a line polynomials for the given endpoints of the gap and further utilizes the linear formula to compute the missing values.
- Exponentially Weighted Moving Average(EWMA) [14]: EWMA estimates the missing values as a weighted average of the history data points, where the weighting factors decrease exponentially for each timestamp.
- k-Nearest Neighbor(kNNimp) [27]: kNNimp first find the k nearest neighbors and impute the missing points with the weighted average of these neighbor points. The k is a hyperparameter to be tuned in the experiments.
- Kalman Smoothing(Kalman) [25]: This approach is to apply sequentially the Kalman Filter[2] for missing calculation. The underlying model is linear and the distributions are assumed as Gaussian.
- Last Observation Carried Forward(LOCF) [13]: It fills each missing value by the most recent historic data point.
- Denoising Autoencoder(DAE) [8]: DAE is a multilayer encoder-decoder algorithm, which minimizes the reconstruction error of missing values and thus recovers the complete data.
- bi-directional RNN(bRNN) [30]: bRNN is a deep learning architecture based on bi-directional recurrent neural networks. It imputes the missing values from the hidden states of both forward and backward directions.
- Context Encoders(CE) [24]: This method proposes a context encoder to generate the contents of an arbitrary image region conditioned on its surroundings. The loss function is defined by jointly optimizing square loss on reconstruction areas and adversarial loss to discriminate the predictions and ground truth.
- Spatial-Temporal Completion Network for Video Inpainting(ST-Comp) [28]: It combines a 3d convolutional based neural network to explore the temporal consistency and a 2d one to restore the spatial details for each frame of the video.

To apply image-based methods to the time series imputation application, we replace the 2d convolutional operation with 1d convolutional operation and 3d convolutional operation with 2d convolutional operations in their frameworks. All neural network methods(*i.e.* DAE, bRNN, CE, ST-Comp and *HeartImp*) are trained by Adam optimizer. LIP, EWMA, Kalman and LOCF are implemented based on the imputeTS library[22] and kNNimp is built from the fancyimpute package[1]. We also optimize the parameter settings of all baselines in the validation set and report the performance in the testing set.

4.1.3 Evaluation Protocol We conduct experiments on gap imputation on all methods. To generate the data with missing intervals, we first split the wearable-sensory time series every 24 hours. Following the setting of missing interval generation in [9], we separately apply four gap sizes as mean values of a normal distribution generator to randomly select the length of missing intervals in these daily series: two hours, four hours, six hours and eight hours. And the missing starting locations are sampled uniformly from $[0, 24 - \text{gap size}]$. Then, we split the sets of these daily series chronologically. We use 10.5 months, 0.5 months and 1 month for training, validation, and testing. The training and validation time periods are before the start date of testing time period. The testing periods for Garmin data are June, July, August and September 2018, and those for Fitbit data are July, August, September and October 2017. We use the validation set to tune the hyper-parameters and testing set to report the performance. All experiments are evaluated across the consecutive days in the testing set and the average performance is reported.

[1] https://github.com/iskandr/fancyimpute

Table 1: Missing imputation performance across different time periods in terms of RMSE, MAE and MAPE scores, given the gap size of 8.

Data	Garmin Data											
Month	June			July			August			September		
Method	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE
LIP	20.57	15.26	0.1912	20.18	14.99	0.1872	20.20	14.98	0.1881	20.22	15.01	0.1888
EWMA	20.23	14.98	0.1875	19.98	14.8	0.1846	19.97	14.79	0.1856	20.05	14.84	0.1864
Kalman	23.63	15.92	0.1989	21.59	16.43	0.2051	23.36	15.73	0.197	23.85	15.95	0.1999
LOCF	20.57	15.25	0.1912	20.17	14.99	0.1870	20.17	14.96	0.1879	20.19	15.00	0.1885
kNNimp	17.36	12.6	0.1564	17.55	12.69	0.1572	17.94	12.86	0.1594	17.50	12.65	0.1581
bRNN	15.49	11.54	0.1414	15.62	11.65	0.1433	15.71	11.66	0.1417	15.67	11.67	0.1431
DAE	16.86	13.00	0.1642	17.02	13.01	0.1621	17.35	13.35	0.1689	17.09	12.96	0.1587
ST-Comp	15.55	11.57	0.1412	16.00	12.03	0.1488	15.93	11.95	0.1461	15.96	11.85	0.1434
CE	17.11	13.21	0.1663	17.27	13.25	0.1661	17.15	13.32	0.1700	17.16	13.07	0.1603
HeartImp	**14.82**	**11.18**	**0.1399**	**15.25**	**11.16**	**0.1350**	**15.25**	**11.17**	**0.1348**	**15.05**	**11.04**	**0.1339**
Data	Fitbit Data											
Month	July			August			September			October		
Method	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE
LIP	21.46	15.34	0.2021	21.67	15.64	0.2027	22.29	15.95	0.1985	21.80	15.68	0.1971
EWMA	21.44	15.27	0.2011	21.58	15.55	0.2016	22.30	15.91	0.1979	21.75	15.59	0.1959
Kalman	27.06	16.19	0.2130	23.80	16.88	0.2181	23.41	17.31	0.2156	24.67	16.67	0.2094
LOCF	21.43	15.32	0.2020	21.68	15.63	0.2022	22.24	15.93	0.1981	21.78	15.66	0.1969
kNNimp	19.84	14.00	0.1838	20.66	14.42	0.1838	19.54	14.21	0.1752	19.4	14.11	0.1768
bRNN	17.19	12.23	0.1553	16.88	11.79	0.1453	16.69	12.45	0.1540	16.86	12.31	0.1475
DAE	18.13	13.38	0.1719	18.55	13.41	0.1655	19.22	14.24	0.1693	18.66	14.15	0.1725
ST-Comp	17.01	12.21	0.1551	16.48	12.46	0.1682	16.73	12.57	0.1566	17.39	12.78	0.1521
CE	17.29	12.42	0.1573	17.21	12.90	0.1716	16.94	12.44	0.1521	17.17	12.30	0.1442
HeartImp	**16.12**	**11.15**	**0.1388**	**15.42**	**10.83**	**0.1373**	**16.46**	**11.95**	**0.1449**	**16.81**	**12.17**	**0.1441**

We measure the imputation accuracy of all compared methods by using three metrics: Root Mean Square Error (RMSE), Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE), which are widely used in time series imputation tasks. Note that a smaller RMSE, MAE and MAPE value indicates better performance.

4.1.4 Reproducibility We implement the proposed *HeartImp* and its variants using Tensorflow. And they are trained and tested on a server with 4 GPU accelerators. We summarize the key parameter settings of *HeartImp* in Table 2. And the software will be publicly available at the time of publication.

Table 2: Parameter Settings

Parameter	Value
Size of Reference set:	6
Loss regularization λ:	0.001
Learning Rate:	0.0001
Batch Size:	32

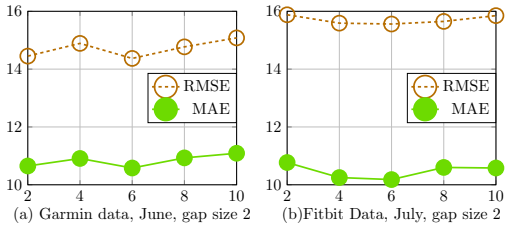
4.2 Performance Comparison (RQ1,RQ2)

4.2.1 Overall Performance In our experiments, we evaluate the performance of all compared methods over different settings of testing time periods and gap sizes on the two datasets. Table 1 shows the imputation performances of all approaches across various time periods in terms of RMSE, MAE and MAPE metrics, while the corresponding performances for different gap sizes are summarized in Table 3. From both tables, we have the following key observations: i. The proposed *HeartImp* consistently achieves the best performance over other methods in all cases. For example, the improvements of *HeartImp* over different baselines range from 4.1% to 58.5% for Garmin data in September and those from 6.9% to 54.3% for Fitbit Data in August in term of RMSE scores. We believe the benefits are credited to the joint modeling of temporal dependency and reference sets. ii. Among all compared methods, neural network-based models(*i.e.* *HeartImp*, CE, ST-Comp, DAE and RNN) are significantly better than different types of conventional time series imputation techniques. This further sheds light on the advantages of complex neural structures for missing interval imputation. iii. We also notice that the performance of *HeartImp* is followed by two baselines, RNN and ST-Comp, which both consider and model the temporal consistency of the series. It also indicates the important positive effects of capturing temporal contextual relations in gap imputation.

4.2.2 Performance *v.s.* Time Period We performed experiments of all methods across different training and testing time periods on two datasets while fix-

ing the corresponding gap size. From Table 1, we can observe that our *HeartImp* performs better than other compared methods for different time periods. And we also note that these improvements are relatively stable across time. It further implies the robust efficacy of *HeartImp* in gap imputations.

4.2.3 Performance *v.s.* Gap Size In addition, we investigate the effectiveness of our model for imputation of missing intervals with various lengths(*i.e* gap sizes). Table 3 summarizes the results of all methods for gap sizes of 2,4,6,8 hours, respectively. From the table, *HeartImp* outperforms other methods for all gap sizes. Moreover, another observation is that the performance of our model is more stable than others when gap size increases. This further reflects the robustness of our model for various gap sizes. Similar results were achieved when we evaluated the performance across different combined settings of time periods and gap sizes. This last analysis was not included here due to space limitations.



(a) Garmin data, June, gap size 2 (b)Fitbit Data, July, gap size 2

Figure 2: Hyperparameter Studies in terms of RMSE and MAE.

4.3 Model Analysis(RQ3,RQ4)

4.3.1 Parameter Study(RQ3) We also conducted experiments to study the impact of the key parameter, *i.e.* the size of reference sets. Except for the parameter being tested, all other parameters are set as the default values in Table 2. Figure 2 shows the imputation accuracy of on two datasets as functions of the parameter when others are controlled for. We notice that the *HeartImp* achieves the best results when r is around 6 for two datasets. Additionally, we can see the hyper-parameters have a relatively low influence on the performance of our model across two datasets. It also demonstrates the stable performance of our *HeartImp*.

4.3.2 Model Ablation(RQ4) To better understand *HeartImp*, we evaluate its key components. We consider three variants of the proposed method corresponding to different analytical aspects: (i) Effect of Generative Adversarial Training (*HeartImp*-a): A simplified version of *HeartImp* without generative adversarial training, *i.e.* only consider $\mathcal{L}_{rec}$ in the loss function; (ii) Effect of Gate Convolutional Module (*HeartImp*-c): A simplified version of *HeartImp* which excludes gate convolutional module to capture the temporal dependency; (iii) Effect of Reference Infusion (*HeartImp*-r): A simplified version of *HeartImp* without fusing reference sets.

The results are reported in Figure 3. We can summarize that:(i) *HeartImp* succeeds in gap imputation when compared to the variant *HeartImp*-a, which justifies the positive effects of adopting adversarial training to capture the distribution of time series; (ii) *HeartImp* outperforms *HeartImp*-c in all cases. Thus, the gate convolutional module is efficient to capture the temporal dependency among series; (iii) *HeartImp* achieves better results than the variant *HeartImp*-r. We suspect the reason is the important roles of reference set fusion on modeling the personal characteristics. Therefore, these observations suggest the full version of *HeartImp*jointly provide the promising and effective performance for filling missing intervals in sensory time series.

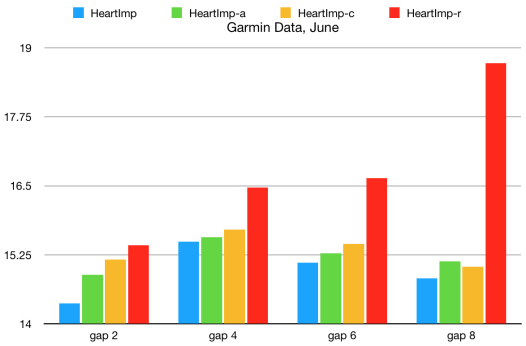


Figure 3: Effects of key components on Garmin data.

5 Related Work

Time Series Imputation There has been an increasing interest and progressive achievements in time series imputation. The conventional methods like the averaging[14] and forecasting from regression model of the time sequence[17], either make linear assumptions on the data or can cause error propagation on the large gap imputations. More recently, many deep learning techniques have been developed to solve this problem[6, 18]. However, they carry their own limitations on modeling individual characteristics, which play an important role on health-related analysis. *HeartImp* is proposed to fill the large gaps while learning and retaining the individual characteristics.

Deep Learning in Healthcare Application With the development of deep learning techniques, introducing deep neural network structure to health-related applications has gained a lot of attention in the past decades[4, 7, 26, 20]. For instance, Cao *et.al.* propose

Table 3: Missing imputation performance across different missing gap size in terms of RMSE, MAE and MAPE scores, given the fixed time period.

Data	Garmin Data, June											
Gap Size	2			4			6			8		
Method	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE
LIP	16.05	11.74	0.145	17.22	12.6	0.1508	17.72	12.85	0.1513	20.57	15.26	0.1912
EWMA	17.3	12.44	0.1532	18.47	13.4	0.1602	18.85	13.7	0.1618	20.23	14.98	0.1875
Kalman	17.94	12.05	0.1491	19.88	13.12	0.1569	21.42	13.81	0.1622	23.63	15.92	0.1989
LOCF	18.72	13.51	0.1627	20.37	14.9	0.1756	20.5	15.13	0.1806	20.57	15.25	0.1912
kNNimp	16.15	12.15	0.1493	16.9	12.68	0.1529	16.82	12.62	0.1529	17.36	12.6	0.1564
bRNN	15.37	11.29	0.1335	15.55	11.59	0.1369	15.2	11.43	0.1379	15.49	11.54	0.1414
DAE	15.68	11.64	0.1387	16.68	12.59	0.1492	16.71	12.53	0.1481	16.86	13	0.1642
ST-Comp	14.97	11.1	0.132	15.51	11.59	0.1371	15.53	11.4	0.1334	15.55	11.57	0.1412
CE	16.29	12.27	0.1488	17.11	13.11	0.1579	16.71	12.77	0.1547	17.11	13.21	0.1663
HeartImp	**14.37**	**10.58**	**0.1258**	**15.49**	**11.49**	**0.1359**	**15.11**	**11.05**	**0.13**	**14.82**	**11.18**	**0.1399**
Data	Fitbit Data, July											
Gap Size	2			4			6			8		
Method	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE
LIP	16.19	11.09	0.1393	17.71	12.2	0.1515	17.95	12.43	0.1532	21.46	15.34	0.2021
EWMA	17.65	11.79	0.1477	19.19	13.04	0.1618	19.47	13.46	0.1662	21.44	15.27	0.2011
Kalman	18.06	11.72	0.1483	19.9	12.83	0.1593	20.6	13.21	0.1625	27.06	16.19	0.213
LOCF	19.85	13.49	0.167	21.22	14.83	0.1833	21.29	15.13	0.19	21.43	15.32	0.202
kNNimp	17.78	12.83	0.1627	18.35	13.21	0.1661	18.15	13.01	0.1643	19.84	14	0.1838
bRNN	17.29	11.53	0.1378	18.1	12.18	0.1434	17.2	11.82	0.1427	17.19	12.23	0.1553
DAE	16.73	11.53	0.1416	18.4	13.05	0.1579	17.62	12.75	0.158	18.13	13.38	0.1719
ST-Comp	16.38	11.04	0.1329	17.13	**11.64**	0.1404	16.82	11.66	0.1413	17.01	12.21	0.1551
CE	16.35	11.33	0.1396	17.56	12.65	0.1576	17.23	12.22	0.1508	17.29	12.42	0.1573
HeartImp	**15.56**	**10.18**	**0.1215**	**17.03**	**11.64**	**0.1398**	**16.62**	**11.4**	**0.1386**	**16.12**	**11.15**	**0.1388**

a end-to-end deep architecture to model the multi-view sensory data for human mood prediction[4], while Choi *et.al.* explored the feature learning of medical concepts with deep neural models[7]. Our work further investigates this direction to deal with the missing in temporal sensory data, which is also essential to much sensory data analysis work.

6 Conclusion

In this paper, we presented *HeartImp*, a deep learning model for gap imputation in wearable sensory time series. It addresses several limitations of the current state-of-art methods, including dealing with missing intervals of variable sizes and arbitrary locations and modeling intra-sensor/individual variability. *HeartImp* first introduces the historical time series as the reference set to extract personal characteristics. And then the model imputes the missing gaps based on the context information of the missing area and the learned personal features. Adversarial learning was adopted to guide the learning process. Extensive experiments demonstrate that *HeartImp* outperforms the conventional and contemporary approaches.

Acknowledgement

This research is based upon work supported in part by the Office of the Director of National Intelligence(ODNI), Intelligence Advanced Research Projects Activity(IARPA), via IARPA Contract No.2017-17042800007. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

References

[1] I. B. Aydilek and A. Arslan, *A hybrid method for imputation of missing values using optimized fuzzy c-means with support vector regression and a genetic algorithm*, Information Sciences, 233 (2013), pp. 25–35.

[2] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation: theory algorithms and software*, John Wiley & Sons, 2004.

[3] T. Benichou, B. Pereira, M. Mermillod, I. Tauveron, D. Pfabigan, S. Maqdasy, and F. Dutheil, *Heart rate variability in type 2 diabetes mellitus: A systematic review and meta–analysis*, PLoS One, 13 (2018), p. e0195166.

[4] B. Cao, L. Zheng, C. Zhang, P. S. Yu, A. Piscitello, J. Zulueta, O. Ajilore, K. Ryan, and A. D. Leow, *Deepmood: modeling mobile phone*

typing dynamics for mood detection, in KDD, ACM, 2017, pp. 747–755.

[5] W. Cao, D. Wang, J. Li, H. Zhou, L. Li, and Y. Li, *Brits: Bidirectional recurrent imputation for time series*, in NIPS, 2018, pp. 6775–6785.

[6] Z. Che, S. Purushotham, K. Cho, D. Sontag, and Y. Liu, *Recurrent neural networks for multivariate time series with missing values*, Scientific reports, 8 (2018), p. 6085.

[7] E. Choi, M. T. Bahadori, L. Song, W. F. Stewart, and J. Sun, *Gram: graph-based attention model for healthcare representation learning*, in Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2017, pp. 787–795.

[8] A. F. Costa, M. S. Santos, J. P. Soares, and P. H. Abreu, *Missing data imputation via denoising autoencoders: the untold story*, in International Symposium on Intelligent Data Analysis, Springer, 2018, pp. 87–98.

[9] M. Elshenawy, M. El-Darieby, and B. Abdulhai, *Automatic imputation of missing highway traffic volume data*, in 2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), IEEE, 2018, pp. 373–378.

[10] L. Faust, R. Purta, D. Hachen, A. Striegel, C. Poellabauer, O. Lizardo, and N. V. Chawla, *Exploring compliance: observations from a large scale fitbit study*, in Proceedings of the 2nd International Workshop on Social Sensing, ACM, 2017, pp. 55–60.

[11] J. Feng, Y. Li, C. Zhang, F. Sun, F. Meng, A. Guo, and D. Jin, *Deepmove: Predicting human mobility with attentional recurrent networks*, in WWW, ACM, 2018, pp. 1459–1468.

[12] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, *Improved training of wasserstein gans*, in Advances in neural information processing systems, 2017, pp. 5767–5777.

[13] D. Hedeker, R. J. Mermelstein, and H. Demirtas, *Analysis of binary outcomes with missing data: missing= smoking, last observation carried forward, and a little multiple imputation*, Addiction, 102 (2007), pp. 1564–1573.

[14] C. C. Holt, *Forecasting seasonals and trends by exponentially weighted moving averages*, International journal of forecasting, 20 (2004), pp. 5–10.

[15] J. Honaker and G. King, *What to do about missing values in time-series cross-section data*, American Journal of Political Science, 54 (2010), pp. 561–581.

[16] J. Hu, L. Shen, and G. Sun, *Squeeze-and-excitation networks*, in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 7132–7141.

[17] H. Junninen, H. Niska, K. Tuppurainen, J. Ruuskanen, and M. Kolehmainen, *Methods for imputation of missing values in air quality data sets*, Atmospheric Environment, 38 (2004), pp. 2895–2907.

[18] Y.-J. Kim and M. Chi, *Temporal belief memory: Imputing missing data during rnn training.*, in IJCAI, 2018.

[19] Z. Liao, X. Lu, T. Yang, and H. Wang, *Missing data imputation: a fuzzy k-means clustering algorithm over sliding window*, in 2009 Sixth International Conference on Fuzzy Systems and Knowledge Discovery, vol. 3, IEEE, 2009, pp. 133–137.

[20] F. Ma, J. Gao, Q. Suo, Q. You, J. Zhou, and A. Zhang, *Risk prediction on electronic health records with prior medical knowledge*, in KDD, ACM, 2018, pp. 1910–1919.

[21] S. M. Mattingly, J. M. Gregg, P. Audia, A. E. Bayraktaroglu, A. T. Campbell, N. V. Chawla, V. Das Swain, M. De Choudhury, S. K. D'Mello, A. K. Dey, et al., *The tesserae project: Large-scale, longitudinal, in situ, multimodal sensing of information workers*, in Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems, ACM, 2019, p. CS11.

[22] S. Moritz and T. Bartz-Beielstein, *imputets: time series missing value imputation in r*, The R Journal, 9 (2017), pp. 207–218.

[23] J. Ni, L. Muhlstein, and J. McAuley, *Modeling heart rate and activity data for personalized fitness recommendation*, (2019).

[24] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, *Context encoders: Feature learning by inpainting*, in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 2536–2544.

[25] S. Sarkka, A. Vehtari, and J. Lampinen, *Time series prediction by kalman smoother with cross-validated noise density*, in 2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No. 04CH37541), vol. 2, IEEE, 2004, pp. 1653–1657.

[26] Y. Suhara, Y. Xu, and A. Pentland, *Deepmood: Forecasting depressed mood based on self-reported histories via recurrent neural networks*, in WWW, ACM, 2017, pp. 715–724.

[27] O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, and R. B. Altman, *Missing value estimation methods for dna microarrays*, Bioinformatics, 17 (2001), pp. 520–525.

[28] C. Wang, H. Huang, X. Han, and J. Wang, *Video inpainting by jointly learning temporal structure and spatial details*, arXiv preprint arXiv:1806.08482, (2018).

[29] P. Wang, J. Guo, Y. Lan, and J. Xu, *Your cart tells you: Inferring demographic attributes from purchase data*, in WSDM, ACM, 2016, pp. 173–182.

[30] J. Yoon, W. R. Zame, and M. van der Schaar, *Multi-directional recurrent neural networks: a novel method for estimating missing data*, (2017).

[31] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, *Generative image inpainting with contextual attention*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 5505–5514.